SANDIA REPORT

SAND2000-0443
Unlimited Release
Printed March 2000

Quantitative Tomographic Measurements of Opaque Multiphase Flows

Darin L. George, John R. Torczynski, Kim A. Shollenberger, Timothy J. O'Hern and Steven L. Ceccio

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (703) 605-6000
Web site: http://www.ntis.gov/ordering.htm

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

SAND2000-0443
Unlimited Release
Printed March 2000

Quantitative Tomographic Measurements of Opaque Multiphase Flows

Darin L. George, John R. Torczynski, Kim A. Shollenberger, Timothy J. O'Hern
Engineering Sciences Center
Sandia National Laboratories
P. O. Box 5800
Albuquerque, New Mexico 87185-0834

Prof. Steven L. Ceccio
Department of Mechanical Engineering and Applied Mechanics
The University of Michigan
Ann Arbor, Michigan 48109-2121

Abstract

An electrical-impedance tomography (EIT) system has been developed for quantitative measurements of radial phase distribution profiles in two-phase and three-phase vertical column flows. The EIT system is described along with the computer algorithm used for reconstructing phase volume fraction profiles. EIT measurements were validated by comparison with a gamma-densitometry tomography (GDT) system. The EIT system was used to accurately measure average solid volume fractions up to 0.05 in solid-liquid flows, and radial gas volume fraction profiles in gas-liquid flows with gas volume fractions up to 0.15. In both flows, average phase volume fractions and radial volume fraction profiles from GDT and EIT were in good agreement. A minor modification to the formula used to relate conductivity data to phase volume fractions was found to improve agreement between the methods. GDT and EIT were then applied together to simultaneously measure the solid, liquid, and gas radial distributions within several vertical three-phase flows. For average solid volume fractions up to 0.30, the gas distribution for each gas flow rate was approximately independent of the amount of solids in the column. Measurements made with this EIT system demonstrate that EIT may be used successfully for noninvasive, quantitative measurements of dispersed multiphase flows.

Acknowledgments

The authors wish to thank the following individuals for their assistance and interactions. Dr. Bernard A. Toseland of Air Products and Chemicals, Inc. provided useful guidance and information about industrial-scale bubble columns. A portion of this work was funded by the Federal Energy Technology Center under Field Work Proposal FEW 8616. Thanks go to Dr. Ann Tassin Leger, formerly of the University of Michigan, for her assembly and testing of the original version of the Sandia/Michigan EIT system and Dr. Douglas R. Adkins for his early work with the GDT system. Prof. Milorad P. Dudukovic of Washington University in St. Louis is acknowledged for his advice on the development of validation testbeds for multiphase flows. The authors also wish to thank the technical staff at Sandia whose expertise kept the EIT validation studies moving forward: C. “Buddy” Lafferty, Tom Grasser, Jaime Castañeda, John O’Hare, “Rocky” Erven, and W. Craig Ginn. Dr. Arthur C. Ratzel of Sandia National Laboratories is thanked for his management and support of this research.

Contents

Figures

Tables

Nomenclature

Quantities listed without units are dimensionless.

Roman Symbols

a	phase interfacial area per unit volume of flow (m^2/m^3 or $1/m$)
A_{col}	cross-sectional area of bubble column (m^2)
C, C_n	arbitrary constant or parameter
C_D	drag coefficient
D_b	bubble diameter (m)
D_{col}	diameter of bubble column (m)
D_{cyl}	diameter of cylindrical test object (cm)
D_L	mass diffusion coefficient into liquid phase (m^2/s)
D_{Sauter}	Sauter mean bubble diameter (m)
$\mathbf{D}$	tomographic measurement domain
E	Maxwell-Hewitt conductivity function
f	frequency (Hz)
f_b	bubble size distribution function
f_{large}	fraction of gas volume in bimodal bubble size distribution attributed to large bubbles
f_{max}	value of f_{large} at which monodisperse and bimodal Maxwell-Hewitt relations differ most
f_μ	gamma attenuation distribution function
F_n	weight function for fundamental voltage at electrode n
g	gravitational constant, 9.807 m/s^2
H_{col}	height of multiphase mixture in bubble column (m)
I	injected current (A)
j_G	gas volumetric flux density ($m^3/s/m^2$ or m/s)
$\bar{j}_G$	alternative notation for superficial gas velocity ($m^3/s/m^2$ or m/s)
j_{Gd}	gas volumetric drift flux ($m^3/s/m^2$ or m/s)
j_L	liquid volumetric flux density ($m^3/s/m^2$ or m/s)
j_{Ld}	liquid volumetric drift flux ($m^3/s/m^2$ or m/s)
j_m	mixture volumetric flux density ($m^3/s/m^2$ or m/s)

J	injected current per unit length of electrode (A/m)
k	fitted coefficient
k_L	mass transfer coefficient into liquid phase (m/s)
K	fixed parameter
l_P	Prandtl mixing length scale (m)
L	distance above floor of bubble column (cm)
M	number of possible independent measurements at the tomographic domain boundary, also number of impedance elements in the computed domain reconstruction
M_G	gas phase momentum per unit mass (kg·m/s/kg or m/s)
$\mathbf{n}$	unit normal vector outward from the domain boundary
N	number of electrodes on the domain boundary
p_{col}	fluid pressure in two-phase bubble column flow (Pa or kPa)
p_v	vapor pressure of liquid at ambient temperature (Pa or kPa)
q	charge flux on the domain boundary (A/m^2)
Q_G	gas volumetric flow rate (m^3/s)
r	radial coordinate (m)
$\tilde{r}$	normalized radial coordinate
R_b	radius of bubble, drop, or particle (m)
R_b^*	nondimensionalized radius of bubble, drop, or particle
R_{col}	radius of bubble column or EIT testbed (m)
Sh	Sherwood number, $k_L D_b / D_L$
T_L	liquid temperature (ºC or K)
T_G	gas temperature (ºC or K)
u_b	bubble rise velocity (m/s)
$u_{b\infty}$	terminal velocity of a single bubble, drop, or particle (m/s)
u_{Dd}	dispersed phase drift velocity (m/s)
u_G	gas phase velocity (m/s)
$\bar{u}_G$	column-averaged gas phase velocity (m/s)
u_{Gd}	gas phase drift velocity (m/s)
u_L	liquid phase velocity (m/s)
u_{Ld}	liquid phase drift velocity (m/s)
u_r	relative velocity or slip velocity (m/s)
$\langle u_r \rangle$	relative velocity averaged over the flow cross section (m/s)
U_G	superficial gas velocity, equal to gas volumetric flux density averaged over the flow cross section (m^3/s/m^2 or m/s)

$U_{G,trans}$	superficial gas velocity at regime transition ($m^3/s/m^2$ or m/s)
U_m	superficial mixture velocity, equal to mixture volumetric flux density averaged over the flow cross section ($m^3/s/m^2$ or m/s)
V	voltage or electric potential (V)
$\overline{V}$	voltage averaged over the measurement period (V)
V_{col}	volume of fluid in bubble column (m^3)
V_n	fundamental voltage solution at electrode *n*
$V_n^{(ij)}$	fundamental voltage solution at electrode *n* for injection at electrode *i* and withdrawal at electrode *j*
$\tilde{V}_m^{(ij)}$	measured voltage at electrode *m* for injection electrode *i* and withdrawal electrode *j* (V)
$\hat{V}_m^{(ij)}$	weighted voltage "measurement" for injection electrode *i*, withdrawal electrode *j*, and measurement electrode *m*
w	distance from center of insulating test object (m)
$w_m^{(ij)}$	fundamental voltage weight for injection electrode *i*, withdrawal electrode *j*, and measurement electrode *m*
x	Cartesian coordinate in bubble column or domain (m)
y	Cartesian coordinate in bubble column or domain (m)
z	Cartesian or cylindrical axial coordinate (m)

Greek Symbols

α	Maxwell-Hewitt coefficient
β	momentum correction factor
γ	complex electrical conductivity (S/m or μS/cm)
δ_{qr}	Kronecker delta function
$\bar{\varepsilon}_{cyl}$	volume fraction of testbed occupied by test object
ε_D	volume fraction of dispersed phase
$\varepsilon_{D,MAX}$	volume fraction of dispersed phase at maximum packing
ε_{df}	"dense phase" volume fraction of small bubbles in a liquid-small bubble mixture, excluding volume of large bubbles
$\bar{\varepsilon}_{df}$	dense phase holdup, column-averaged volume fraction of small bubbles in a liquid-small bubble mixture excluding volume of large bubbles
ε_G	gas volume fraction
$\langle\varepsilon_G\rangle$	gas volume fraction averaged across the column cross section
$\bar{\varepsilon}_G$	gas holdup or volume-averaged gas volume fraction
$\varepsilon_{G,small}$	volume fraction of small bubbles in two-phase flow

$\varepsilon_{G,large}$	volume fraction of large bubbles in two-phase flow
$\bar{\varepsilon}_{G,small}$	column-averaged volume fraction of small bubbles
$\bar{\varepsilon}_{G,large}$	column-averaged volume fraction of large bubbles
$\bar{\varepsilon}_{G,trans}$	gas holdup at regime transition
ε_I	insulating phase volume fraction
ε_L	liquid volume fraction
$\bar{\varepsilon}_L$	volume-averaged liquid volume fraction
ε_S	solid volume fraction
$\langle \varepsilon_S \rangle$	solid volume fraction averaged across the column cross section
$\bar{\varepsilon}_S$	volume-averaged solid volume fraction
$\bar{\varepsilon}_S^{NOM}$	nominal column-averaged solid volume fraction
η_C	dynamic viscosity of continuous phase (kg/m/s)
η_D	dynamic viscosity of dispersed phase (kg/m/s)
η_{eff}	effective dynamic viscosity of non-Newtonian fluid (kg/m/s)
η_L	liquid dynamic viscosity (kg/m/s)
η_m	mixture dynamic viscosity (kg/m/s)
θ	azimuthal coordinate (degrees)
μ	gamma ray attenuation coefficient (1/cm)
$\hat{\mu}$	gamma ray attenuation coefficient averaged over the beam path (1/cm)
$\langle \mu \rangle$	gamma ray attenuation coefficient averaged over the measurement cross section (1/cm)
μ_G	gas phase gamma ray attenuation coefficient (1/cm)
μ_L	liquid phase gamma ray attenuation coefficient (1/cm)
μ_S	solid phase gamma ray attenuation coefficient (1/cm)
ν_L	liquid kinematic viscosity (m^2/s)
ν_{mol}	molecular kinematic viscosity (m^2/s)
ν_t	turbulent eddy viscosity (m^2/s)
ξ	variable in the Maxwell-Hewitt conductivity function
ρ	density (kg/m^3)
ρ_C	density of continuous phase (kg/m^3)
ρ_D	density of dispersed phase (kg/m^3)
ρ_G	gas density (kg/m^3)
ρ_L	liquid density (kg/m^3)

ρ_S	solid density (kg/m^3)
σ	real component of complex electrical conductivity (S/m or µS/cm)
$\overline{\sigma}$	effective or domain-averaged real conductivity (S/m or µS/cm)
σ_B	effective conductivity of mixture of liquid and bimodal (small and large) bubbles (S/m or µS/cm)
σ_{df}	effective conductivity of "dense phase" mixture of liquid and small bubbles, excluding large bubbles (S/m or µS/cm)
σ_L	real electrical conductivity of liquid phase (S/m or µS/cm)
σ_s	interfacial surface tension (N/m)
τ_{rz}	shear stress in the radial direction on a plane perpendicular to the vertical axis (kg/m/s^2)
ω	permittivity or dielectric constant (F/m or µF/cm)
$\tilde{\omega}$	permittivity of medium relative to vacuum
$\tilde{\omega}_L$	relative permittivity of liquid
ω_0	permittivity of vacuum, 8.854×10^{-12} F/m

Quantitative Tomographic Measurements of Opaque Multiphase Flows

1. Introduction

The spatial distribution of materials in dispersed multiphase flows is important to many chemical and energy industries. For example, bubble-column reactors are used to carry out chemical synthesis and other processes in gas-liquid or gas-solid-liquid flows; in indirect coal liquefaction, a reactive gas is bubbled through a catalyst-laden slurry. When scaling slurry bubble-column reactor vessels to industrial sizes, it is important to ensure that the distribution of the different materials within the vessel is acceptable. A spatially nonuniform gas distribution within the reactor can reduce process efficiency by inducing large-scale, buoyancy-driven recirculating flows (Jackson *et al.,* 1996). Thus, it is desirable to be able to accurately measure or predict material distributions in three-phase flows.

Because direct numerical simulation of multiphase flows based on first principles is presently impractical, computational models for industrial use must rely on a combination of theory and experimental correlations to properly simulate such flows. Techniques that measure the distribution of each phase in multiphase flows have the potential to improve the control of such processes and can also be useful for validating computational models (Plaskowski *et al.,* 1995; Torczynski *et al.,* 1997). As the requirements for accuracy of computational models become more demanding, the techniques used to measure relevant flow properties must deliver more detailed and accurate information.

A dilemma encountered in multiphase flow measurements is that probes or instruments should be placed outside the flow domain so as not to disturb the flow itself, but phase distributions cannot easily be measured from the boundary. Tomography offers a possible solution to this dilemma, since tomographic methods have the potential to determine spatial phase distributions without disturbing the flow. Many of the tomographic instruments available to study multiphase flows have been derived from devices for medical applications. Examples of these methods include X-ray tomography; nuclear magnetic resonance (NMR, also known as magnetic resonance imaging or MRI); positron emission tomography (PET); and acoustic tomography. While the spatial resolution of these methods can be substantial (acoustic tomography can have a resolution of less than 1 mm, MRI a resolution of 10^{-5} m), these methods are often limited to the study of slowly changing systems because of the long acquisition times required for accurate quantitative data (ranging from minutes for MRI to hours for PET and other radiation-based methods). The systems listed above can also be difficult to use because of safety and cost considerations. In recent years the medical community has pioneered the use of electrical-impedance tomography (EIT), which uses measurements of the electrical impedance at the boundary of a test domain to reconstruct the impedance within the domain. EIT has the potential for much faster quantitative measurements of phase distributions than other tomographic methods. Researchers are now attempting to implement EIT systems for the nonintrusive study of multiphase flows. Dickin *et al.* (1993) and Ceccio and George (1996) report on the development of EIT systems specifically for multiphase flows, while a review of current developments in many of the other fields of tomography can be found in George *et al.* (1998a).

At Sandia National Laboratories, initial research into multiphase tomographic methods was performed as part of the Laboratory-Directed Research and Development (LDRD) project "Advanced Tomographic Flow Diagnostics for Opaque Multiphase Fluids." The final report for this LDRD (Torczynski *et al.*, 1997) examined several techniques for measuring phase distributions in vertical bubble-column flows, including instruments that provided spatially averaged or local quantities and other methods designed to yield full-field, spatially resolved measurements. Two tomographic techniques, gamma-densitometry tomography (GDT) and electrical-impedance tomography (EIT), were applied to two-phase flows and compared to established measurement methods and to one another. The EIT system was developed as a joint effort between Sandia National Laboratories and the University of Michigan (O'Hern *et al.*, 1995; Torczynski *et al.*, 1996a).

GDT and other radiation-based tomographic methods are relatively mature and are known to measure spatial phase distributions accurately (Hewitt, 1978). Shollenberger *et al.* (1997a) applied the GDT technique at Sandia to vertical gas-liquid flows at industrially relevant conditions and reported that GDT accurately measured the phase distributions (see also Torczynski *et al.*, 1997). The main disadvantage of radiation-based methods, as suggested above, is that the data collection times required are usually longer than the time scales of time-dependent multiphase flows. EIT can acquire information much more quickly than GDT, but several issues with EIT are currently under investigation, including its usefulness for multiphase flows, its spatial resolution, and its ability to make accurate quantitative measurements (Webster, 1990; Jones *et al.*, 1993; Plaskowski *et al.*, 1995; Ceccio and George, 1996). While Torczynski *et al.* (1997) were able to report successful EIT measurements only of a dilute solid-liquid suspension, subsequent improvements to data acquisition and reduction yielded the first validated, quantitative phase distribution measurements with EIT (George *et al.*, 1998a).

This report documents the work conducted since the publication of the LDRD report, specifically the application of the EIT and GDT techniques to the task of providing spatially resolved information on dispersed multiphase flows. Torczynski *et al.* (1997) briefly discussed the complexity of multiphase fluid mechanics as a motivation for this work; Chapter 2 of this report expands upon this topic by summarizing recent investigations into the physics of vertical bubble-column flows. Subsequent chapters discuss the application of GDT and EIT at Sandia to two-phase and three-phase vertical column flows. Chapter 3 includes a brief introduction to EIT theory and a description of the Sandia/Michigan EIT system. Recent improvements to the EIT hardware are described, and numerical and experimental validation tests of the EIT reconstruction algorithm are presented. The EIT system is then evaluated in Chapter 4 for its ability to measure realistic solid-liquid and gas-liquid flows. Since the Sandia GDT system had already been successfully applied to multiphase flows (Adkins *et al.*, 1996; Torczynski *et al.*, 1996b; Shollenberger *et al.*, 1997a), it was used to assess the accuracy of the EIT system in two-phase flow measurements. Finally, a three-phase tomographic method combining GDT and EIT is presented in Chapter 5 and applied to several vertical solid-gas-liquid flows to provide information useful for the design and scaleup of bubble-column reactors. Typically, EIT systems have been validated by imaging static conductivity distributions or (in a few cases) by quantifying small-scale two-phase flows. This study marks a significant application of EIT to quantitative measurements of multiphase flows on a scale approaching that of bubble-column reactors and other industrial flows.

2. A Review of Vertical Multiphase Flow Studies

This report discusses the application of tomographic techniques to vertical multiphase flows, which are flows of considerable importance to the chemical and nuclear industries. For example, bubble-column reactors are commonly used in the chemical industry to carry out fermentation, extraction, or coal liquefaction. These reactors are essentially vertical columns in which gas enters from the bottom and bubbles through a continuous liquid phase or solid-liquid slurry. The ability to predict the phase distributions, rate of mass transfer, and chemical reaction rates in these columns is a key to improving performance, but scaling the hydrodynamic and transport behavior from the laboratory to industrial conditions remains a formidable task. In the nuclear power industry, boiling is used as a highly efficient heat transfer mechanism in a class of reactors known as boiling water reactors (BWRs). Here the working fluid undergoes nucleate boiling and exhibits a range of flow regimes along the vertical coolant channels in the core. The local vapor phase distribution influences fuel heat transfer characteristics and core reactivity. Improved understanding of the vertical two-phase flows in BWRs under normal and abnormal conditions is currently leading to increases in reactor efficiency, simplified designs, and improved safety.

This chapter reviews the literature on the vertical flows found in two-phase and three-phase chemical reactors, particularly bubble-column reactors. It is in this area that electrical-impedance tomography (EIT) is currently being applied at Sandia to try to obtain information for bubble-column reactor design. This review begins with a discussion of the variables relevant to describing and predicting vertical multiphase flows and flows in bubble-column reactors. Experimental data and correlations from the literature are presented, then models of vertical flows are reviewed. The chapter ends with a survey of current methods of scaling the relevant parameters from the laboratory setting to industrial conditions. Successful scaling is the key to much-needed improvements in the design of bubble-column reactors and similar facilities.

2.1. Relevant Quantities in Vertical Multiphase Flows

The first task in modeling a process for predictive purposes is to determine the variables that must be measured and correctly scaled between laboratory and prototype conditions. Beyond the obvious choices of fluid properties and variables of state — density, temperature, pressure, surface tension, and so on — hydrodynamic variables and phase spatial distributions must be examined to characterize multiphase flows correctly. One of the first journal articles to present a method for analyzing and interpreting experimental data from two-phase flow systems was that of Zuber and Findlay (1965). Many physical quantities were defined or cited in that reference that have become standard for describing multiphase flows. The definitions, introduced below, assume a gas-liquid flow with motion along only one axis, although the equations are equally applicable to other two-phase flows with motion in one dimension.

The first useful quantity is the volumetric flux density, j_ϕ, defined as the volume flow rate of phase ϕ per unit area of the column or duct perpendicular to the direction of flow. The volumetric flux density of each phase is defined in terms of the local phase volume fraction, ε_ϕ, and the local phase velocity, u_ϕ.

$$j_G = \varepsilon_G u_G \tag{2.1}$$

$$j_L = (1-\varepsilon_G)u_L = \varepsilon_L u_L \tag{2.2}$$

The volumetric flux density of the mixture is defined as the sum of the local values for each phase, so that in gas-liquid flow

$$j_m = j_L + j_G . \tag{2.3}$$

Another useful quantity is the drift velocity, $u_{\phi d}$. This is defined as the phase velocity with respect to the mixture volumetric flux density, and may also be thought of as the phase velocity relative to a hypothetical surface moving with the average velocity of the mixture at the location of interest.

$$u_{Gd} = u_G - j_m \tag{2.4}$$

$$u_{Ld} = u_L - j_m \tag{2.5}$$

Similarly, the volumetric drift flux is defined as the volumetric flux density of the phase relative to the hypothetical surface (Wallis, 1969; Shah and Deckwer, 1983).

$$j_{Gd} = \varepsilon_G u_{Gd} = (1-\varepsilon_G) j_G - \varepsilon_G j_L \tag{2.6}$$

$$j_{Ld} = \varepsilon_L u_{Ld} = (1-\varepsilon_L) j_L - \varepsilon_L j_G = -j_{Gd} \tag{2.7}$$

Finally, the difference in u_G and the local velocity of the liquid phase, u_L, is referred to as the relative velocity or "slip velocity" (Shah and Deckwer, 1983):

$$u_r = u_G - u_L = \frac{j_G}{\varepsilon_G} - \frac{j_L}{1-\varepsilon_G} . \tag{2.8}$$

By convention the relative velocity is the velocity of the gas relative to the liquid; u_G and u_L have the same sign if the one-dimensional motions of the two phases are in the same direction. If there is no flow of the liquid phase in or out of the bubble column, the reactor is said to be operating in "batch mode," and consequently, $j_L = u_L = 0$ and $u_r = u_G$. Many investigators have found the relative velocity to be a strong influence on gas-liquid flows.

In many cases, local quantities such as volumetric flux density and drift flux cannot be measured directly, but the experimenter can easily obtain values averaged over the area of the column or flow duct. For example, the area-averaged gas volumetric flux density can be computed directly from the gas volumetric flow rate, Q_G, and the area of the flow:

$$U_G = \bar{j}_G \equiv \frac{1}{A_{col}} \int_A j_G dA_{col} = \frac{Q_G}{A_{col}}. \tag{2.9}$$

This particular average is commonly called the "superficial gas velocity." (By convention, superficial velocities of phase ϕ are denoted by U_ϕ rather than $\bar{j}_\phi$.) Similarly, the column-averaged gas volume fraction, $\bar{\varepsilon}_G$, has historically been called "gas holdup" because it can be determined from the rise in the free surface of the fluid mixture with gas flow in the column:

$$\bar{\varepsilon}_G \equiv \frac{1}{V_{col}} \int_V \varepsilon_G dV_{col} = \frac{\Delta H_{col}}{H_{col} + \Delta H_{col}}. \tag{2.10}$$

(In this chapter, $\bar{\varepsilon}_G$ will be referred to as "gas holdup" because of the prevalence of the term in the literature; however, "average gas volume fraction" is the more accurate description.) Early works almost exclusively report correlations and measurements of average quantities, such as U_ϕ and $\bar{\varepsilon}_\phi$, whereas local values – *e.g.*, radial phase distributions – have been reported in the literature only recently with the advent of particle imaging velocimetry and tomographic techniques.

2.2. Experimental Data and Correlations

This section discusses investigations of flow patterns, gas volume fractions, and mass transfer parameters in vertical bubble columns. Particular attention is paid to operating conditions and phase properties that influence the flows, as this is important in the scaleup of laboratory results to industrial conditions. Most of the experimental data taken in vertical column flows involves gas-liquid flows; although slurry bubble-column reactors and fluidized bed reactors involve a solid phase, little data has been reported for such systems.

The flow regimes of vertical gas-liquid flows are well known and have been correlated for a wide range of flow parameters (Shah and Deckwer, 1983; Carey, 1992). Examples of the flow regimes relevant to bubble-column hydrodynamics are sketched in Figure 2.1. At low superficial gas velocities, the bubbles are roughly uniform in diameter and are much smaller than the column itself. The bubbles are dispersed rather evenly in the liquid phase, rise at about the same velocity, and rarely interact with one another. This flow regime is known as "homogeneous bubbly flow" or simply "bubbly flow." At higher superficial gas velocities, larger bubbles form, the flow becomes heterogeneous, and the behavior of the phases depends upon the size of the column. In larger columns of the size found in industrial applications, small and large bubbles coexist, and the larger bubbles rise at higher velocities, generating turbulent

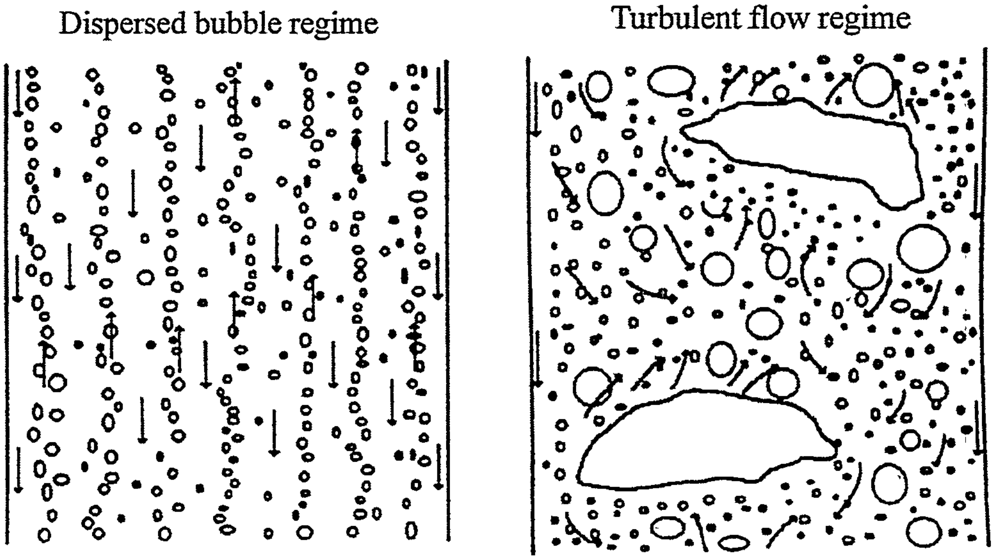


Figure 2.1. Vertical gas-liquid flow regimes of interest in bubble-column hydrodynamics. (Left) homogeneous bubbly flow, (right) churn-turbulent flow. All arrows denote the direction of liquid motion. Adapted from Chen *et al.* (1994).

motion in the liquid phase. This flow is known as "churn-turbulent" flow. In small-diameter vertical columns – typically columns with internal diameters less than 15 cm – the walls act to prevent churn-turbulent flow and cause the gas phase to coalesce into slugs with diameters comparable to the column itself. This may occur in columns with diameters larger than 15 cm if the liquid phase is sufficiently viscous, but "slug flow" is not typically found in bubble-column reactors and is therefore not of concern here.

The flow regime map for air-water bubble columns (Figure 2.2), which indicates the regime as a function of column diameter and superficial gas velocity, is typical of general gas-liquid flows. Many of the investigations reviewed in this chapter are concerned with the transition between gas-liquid flow regimes or with the lower limit on column diameter for which churn-turbulent flows occur.

2.2.1. Gas Volume Fraction Correlations and Observations

Most of the experimental data taken in vertical column flows is of gas volume fractions in gas-liquid flows. This is due not only to the wide variety of methods available for measuring ε_G, but also because of the influence of this quantity on chemical reaction rates and liquid recirculation in bubble-column reactors.

One of the first comprehensive studies of gas volume fraction in bubble columns was that of Akita and Yoshida (1973), in which the rise in liquid level during gas flow was measured and translated into gas holdup for 24 different combinations of liquids, gases, liquid temperature, and column diameter. The paper presented a correlation based on dimensional analysis that fit the

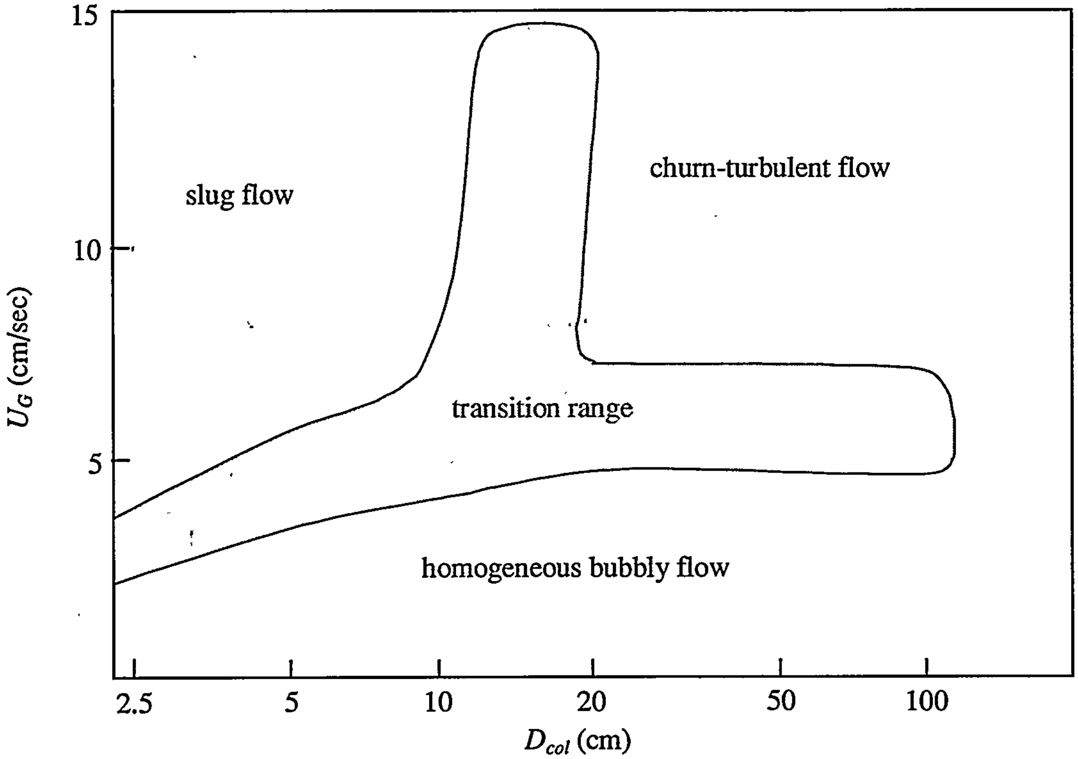


Figure 2.2. Flow regime map for vertical air-water bubble columns (adapted from Shah and Deckwer, 1983).

gas holdup in bubbly flow well. The dimensional analysis included surface tension, σ_s, the superficial gas velocity, U_G, and the column diameter, D_{col}, as independent variables and yielded the correlation

$$\frac{\bar{\varepsilon}_G}{(1-\bar{\varepsilon}_G)^4} = C\left(\frac{gD_{col}^2\rho_L}{\sigma_s}\right)^{1/8}\left(\frac{gD_{col}^3}{\nu_L^2}\right)^{1/12}\frac{U_G}{\sqrt{gD_{col}}}, \tag{2.11}$$

where $C = 0.20$ for pure liquids and nonelectrolyte solutions and 0.25 for electrolyte solutions. According to the authors, earlier experiments had shown the effect of the column diameter on gas holdup to be negligible for columns with diameters of 15 cm or larger. Only columns meeting this criterion were used in their experiments, and their final correlation was chosen to reflect the independence of $\bar{\varepsilon}_G$ on column diameter (note that D_{col} cancels on the right-hand side of Eq. 2.11). Deckwer and Schumpe (1993) advocate a similar correlation for gas flows in viscous liquids and non-Newtonian fluids,

$$\bar{\varepsilon}_G = 0.20\left(\frac{gD_{col}^2\rho_L}{\sigma_s}\right)^{-0.13}\left(\frac{gD_{col}^3\rho_L^2}{\eta_{eff}^2}\right)^{0.11}\left(\frac{U_G}{\sqrt{gD_{col}}}\right)^{0.54}, \tag{2.12}$$

where the liquid kinematic viscosity in Eq. 2.11 has been replaced by the ratio of the effective dynamic viscosity to the liquid density, η_{eff}/ρ_L, to account for non-Newtonian behavior. Note also that this correlation does show a dependence on column diameter ($\bar{\varepsilon}_G \propto D_{col}^{-0.20}$).

The independence of $\bar{\varepsilon}_G$ from D_{col} observed by Akita and Yoshida has been confirmed by later researchers (Shah *et al.*, 1982; Wilkinson *et al.*, 1992; Kumar *et al.*, 1994, 1997) and is the result of the absence of slug flow in industrially relevant liquids when $D_{col} \geq 15$ cm. This value is generally accepted as the lower limit for scaleup of experimental results to industrial scales, provided the following conditions are met: (1) the experimental flow is in the churn-turbulent regime, and (2) the experimental liquid column has a height-to-diameter ratio such that $(H_{col}/D_{col}) \geq 5$ (Joshi *et al.*, 1998). As the first condition suggests, flows in highly viscous or non-Newtonian liquids will have a different lower scaleup limit on D_{col}.

Influences on gas volume fraction with flow rate have been studied extensively, possibly because gas volumetric flow rate is the easiest variable to modify in bubble-column experiments. The relationship between gas holdup and superficial gas velocity is known to vary with flow regime. It is generally accepted as a characteristic of the bubbly flow regime that $\bar{\varepsilon}_G$ increases linearly with U_G (Wilkinson *et al.*, 1992; Reilly *et al.*, 1994). In the churn-turbulent regime the relationship is generally believed to take the form $\bar{\varepsilon}_G \propto U_G^n$, where $n < 1$. Reported values of n range from 0.4 to 0.8 (Wilkinson *et al.*, 1992; Shah *et al.*, 1982). In churn-turbulent flows observed by Reilly *et al.* (1994), the relationship between $\bar{\varepsilon}_G$ and U_G was linear in a few gas-liquid systems, but the slope was consistently less steep than in the bubbly flow regime.

Wilkinson *et al.* (1992) experimentally investigated the effects of column dimensions on $\bar{\varepsilon}_G$ and other quantities to assist in the scaleup of laboratory results to industrial conditions. Measurements were made of nitrogen holdup by an overflow method in four liquids, with surface tensions differing by a factor of 3, viscosities varying by a factor of 50, and fluid pressures ranging from atmospheric to 2.0 MPa. The group confirmed earlier findings that gas holdup decreases with increasing column diameter until $D_{col} > 15$ cm, beyond which the diameter has no effect. For columns with liquid height-to-diameter ratios such that $3 < (H_{col}/D_{col}) < 5$, increasing the column height was found to induce stronger liquid circulation, thereby reducing gas volume fraction gradients between the central column region, the sparger or entrance region, and any foamy region present at the free surface. The net result of an increase in H_{col} over this range was to decrease $\bar{\varepsilon}_G$ within the column. For columns with absolute height $H_{col} > 3$ m or columns with $(H_{col}/D_{col}) > 5$, the influence of circulation and end effects became negligible. Ueyama and Miyauchi (1979) found D_{col} to have little influence on the functional relationship between $\bar{\varepsilon}_G$ and U_G, both in their gas holdup model and in experimental data with which the model was compared; notably, some of the data came from experiments with $D_{col} = 13.8$ cm, less than the commonly accepted limit of 15 cm below which slug flow occurs.

Wilkinson (1991) investigated the influence of sparger hole size on gas holdup. In tests with different gases and liquids, sparger holes had no effect on $\bar{\varepsilon}_G$, provided that the hole diameter was larger than 2 mm. Joshi *et al.* (1998) reported a similar critical hole diameter for several experiments in churn-turbulent air-water flows and flows involving an aqueous solution of carboxymethyl cellulose. Below their critical hole diameter of 3 mm, however, $\bar{\varepsilon}_G$ increased as hole diameter decreased.

The properties of the liquid and gas phases are also relevant quantities that can be easily varied experimentally, and whose effects have yet to be understood fully for correct scaleup of column behavior. A review by Wilkinson (1991) found that with all other conditions being equal, column-averaged gas volume fractions increase with gas density (see also Deckwer and Schumpe, 1993; Krishna and Ellenberger, 1996). Increases in ρ_G increase bubble instability and breakup, which reduces the typical bubble diameter, D_b; decreases the terminal velocity of the bubbles, $u_{b\infty}$; and causes an increase in gas holdup through continuity (see Eq. 2.1). Reilly *et al.* (1994) measured the effects of gas density on gas holdup using a differential pressure method, using gases ranging from helium (ρ_G = 0.164 kg/m^3) to CO$_2$ (ρ_G = 1.84 kg/m^3) and liquids ranging from trichloroethylene (η_L = 5.75×10^{-4} kg/m/s) to a viscous petroleum hydrocarbon (η_L = 2.43×10^{-3} kg/m/s). Increases in the overall column pressure from atmospheric to p_{col} = 1.1 MPa were also used in experiments to increase gas density. The gas volume fraction in both bubbly and churn-turbulent flow regimes was deduced to be a function of the gas phase momentum, M_G, which incorporates the influences of both gas density and gas velocity. The following power law relationship was determined for both bubbly and churn-turbulent flows:

$$\bar{\varepsilon}_G = C_0\beta^n\left[\frac{\rho_G U_G}{(1-\bar{\varepsilon}_G)\rho_L}\right]^n = C_1 M_G^n. \tag{2.13}$$

The exponent n and the constants C_0 and C_1 ($\equiv C_0\beta^n$) all depend upon flow regime, and the units of the constants are defined so as to render the correlation nondimensional. The parameter β is a correction factor that accounts for differences between the column-averaged axial gas phase velocity, $\bar{u}_G$, and the local values of u_G, which can include radial or azimuthal components.

Log-log plots of $\bar{\varepsilon}_G$ versus M_G justified this power-law correlation under experimental conditions and indicated appropriate values of n. In the bubbly flow regime, n = 1, and the constant C_1 is determined by the gas density. Data from churn-turbulent flows of all gas-liquid systems investigated fell on a line described by n = 1/3. One assumption in the derivation of Eq. 2.13 was found to fail under high-pressure "pilot plant" conditions: at high gas holdups ($\bar{\varepsilon}_G$ > 0.5), the total mass of the gas phase relative to that of the liquid phase is no longer negligible, invalidating the approximation for M_G. Reilly and co-workers discuss how the gas momentum term must be evaluated based on the masses of both phases to extend the applicability of the power law to higher gas pressures.

Concerning liquid properties, the only properties determined by Reilly *et al.* (1994) to significantly influence gas holdup in ambient two-phase flows were ρ_L (included in Eq. 2.13) and surface tension, σ_s. From a variety of gases bubbled through trichloroethylene, water, and industrial hydrocarbons, Reilly and co-workers found the effect to be described best by the relationship $\bar{\varepsilon}_G(1-\bar{\varepsilon}_G) \propto \sigma_s^{-0.12}$ or $C_1 \propto \sigma_s^{-0.12}$ in bubbly flows. The effect on churn-turbulent flows was smaller and was not quantified. Wilkinson *et al.* (1992) reviewed the work of earlier authors and found that results often depend on the method used to vary liquid properties. They did point out, however, that factors that lower the bubble coalescence rate will increase gas holdup through reduction of bubble size.

Of particular importance to the scaleup of laboratory results to operating conditions is the effect of temperature and pressure on phase properties. Zou *et al.* (1988) reported measurements of gas holdup in air-liquid co-current column flows at elevated operating temperatures representative of industrial conditions. Data were taken in air-alcohol, air-saline, and air-water systems, and a single correlation for $\bar{\varepsilon}_G$ in the three systems was obtained using dimensional analysis and the experimental data. The dimensional analysis is noteworthy for the inclusion of the vapor pressure of the liquid phase, p_v, which was significantly different at the higher liquid temperatures than at ambient conditions.

The data indicated a strong dependence of gas volume fraction upon liquid temperature; with increasing T_L, the air bubbles became smaller, and $\bar{\varepsilon}_G$ increased in a nearly second-order fashion. While no explanation for decreasing bubble size was advanced by the authors, decreases in η_L and σ_s with increasing T_L are likely candidates, as these encourage smaller bubbles and bubble breakup. Air-alcohol flows were found to have higher gas holdups for the same volumetric flow rates and conditions; because of the higher vapor pressure of alcohol relative to water, higher amounts of vapor were collected by the air bubbles during their rise toward the free surface. The addition of NaCl to the water phase also produced a higher value of $\bar{\varepsilon}_G$, and the change could be accounted for by a temperature-dependent correction. The increase in gas holdup with the addition of electrolyte is attributed to a lower coalescence rate of the gas phase. Though the exact mechanism involved is not discussed, the decrease of surface tension with the added electrolyte is the most probable cause. Finally, it was noted that earlier correlations of $\bar{\varepsilon}_G$, including that of Hughmark (1967) for the bubbly flow regime, failed at elevated temperatures because the correlations assumed constant values of fluid properties.

Lin *et al.* (1998) published an informative work on scaleup of bubble-column phenomena in which bubble-scale and column-scale phenomena were studied for dependence upon both operating temperatures and pressures. Single-bubble rise velocities, u_b; bubble coalescence and breakup; and gas holdup, $\bar{\varepsilon}_G$, were observed for nitrogen gas in a column of Paratherm NF (a heat transfer fluid) at system pressures up to 20 MPa and liquid temperatures from T_L = 27°C to 78°C. Bubble rise velocities were distinctly affected by changes in physical properties of both phases with pressure and temperature. From the Paratherm-N_2 data, u_b was found to decrease with increases in fluid pressure, the prevalent cause being the resultant decrease in the density difference $\Delta\rho = \rho_L - \rho_G$ and its effect on the buoyancy force. An extensive discussion of the effects of pressure and temperature on bubble formation, coalescence, and breakup can also be found in the paper. In summary, the distribution of the bubble diameter, D_b, shifted toward smaller diameters with an increase in fluid pressure (as the resulting increases in ρ_G and gas momentum encouraged the formation of smaller bubbles), while changes in temperature of the system resulted in competing effects on D_b. Both u_b and D_b, of course, influence the gas holdup.

Similar observations of pressure and bubble velocity were made by Adkins *et al.* (1996) in air-water flows at elevated pressure. A review of experimental data in the literature (Joshi *et al.*, 1998) finds that, depending upon flow conditions and bubble shape, drag forces on an individual bubble are proportional to D_b^m, where *m* ranges from 1 (for spherical bubbles in low Reynolds number flows) to 3.25 (for ellipsoidal bubbles at high Reynolds numbers in contaminated liquids). As argued by Adkins *et al.* (1996), since buoyancy forces scale as D_b^3, a force balance would determine that bubble velocity is proportional to D_b^{3-m} and almost never inversely proportional to bubble diameter. The decrease in both D_b and u_b with increasing pressure is therefore to be expected.

In general, $\bar{\varepsilon}_G$ was found by Lin *et al.* (1998) to increase with rising system temperature in agreement with Zou *et al.* (1988). The precise influence of T_L and T_G on $\bar{\varepsilon}_G$ is complex, depending on the interaction between η_L, ρ_G, and σ_s, which can be unique to each multiphase system. The gas holdup was also found to increase with rising fluid pressure, which may be rationalized by the corresponding decreases in u_b and D_b described above. These pressure effects were more pronounced at higher gas volumetric fluxes, all else being equal. The end result of variations in both pressure and temperature on $\bar{\varepsilon}_G$ may be thought of as an imbalance between viscosity on one hand and gas momentum and surface tension on the other.

The last topic of this section is the behavior of local values of gas volume fraction, as opposed to average gas holdup. While the investigations described to this point consider only the column-averaged gas volume fraction, tomographic methods are available to measure time-averaged spatial profiles of ε_G. Kumar *et al.* (1997) used gamma-ray computed tomography to measure gas volume fractions in two-dimensional cross sections of air-water vertical column flows. The distributions were nearly axisymmetric and were circumferentially averaged over θ to produce distributions in *r* alone. In general, their results were described well by the customary power law (see, for example, Ueyama and Miyauchi, 1979),

$$\varepsilon_G(r) = C_\varepsilon \frac{n+2}{n}\left[1 - C_w\left(\frac{r}{R_{col}}\right)^n\right], \tag{2.14}$$

where C_ε is a parameter related to $\bar{\varepsilon}_G$, C_w accounts for the nonzero gas volume fraction at the wall, and the exponent *n* quantifies the "steepness" of the profile. For the larger columns used in tests, a value of *n* between 2 and 2.5 was in best agreement with the data. Gas volume fractions averaged over the measurement cross section (area averages) were also computed from the data and compared to gas holdup values (volume averages) from differential pressure measurements and bed expansion measurements to validate the tomographic approach.

Effects of several operating parameters on $\varepsilon_G(r)$ were reported. In both bubbly and churn-turbulent flow, the gas distributions were influenced by D_{col}, with profiles becoming more peaked with increasing column diameter over the range 10 cm $\leq D_{col} \leq$ 30 cm. However, the area-averaged gas volume fraction, $\langle \varepsilon_G \rangle$, is in line with the accepted trend of the volume-averaged gas volume fraction, $\bar{\varepsilon}_G$, of remaining unchanged with diameter, provided that $D_{col} >$ 15 cm. In general, an increase in U_G led to an increase in $\varepsilon_G(r)$ everywhere except near the wall, making the parabolic profile steeper; this effect was more pronounced in bubbly flows. Additional experiments with a water-isopropanol mixture as the liquid phase revealed that the increase in $\varepsilon_G(r)$ with decreasing surface tension extended all the way out to the column walls because the smaller gas bubbles that result are able to migrate closer to the periphery of the flow.

Adkins *et al.* (1996) also used gamma-ray tomography to determine the radial distribution of air in an industrial-scale bubble column and to investigate the effects of pressure on the gas phase distribution. The bubble column was a stainless steel pressure vessel with D_{col} = 48 cm, considerably larger than the lower scaleup limit of 15 cm. Air-water vertical flows at fluid pressures from 177 kPa to 432 kPa were analyzed for trends in $\bar{\varepsilon}_G$ and $\varepsilon_G(r)$. The profiles were similar to those of Kumar *et al.* (1997): highest at the column axis, parabolic in shape, and increasing more at the axis than at the walls as U_G increased. For a given superficial velocity, however, the gas volume fraction profile was found to increase uniformly across the column as fluid pressure was increased. Later studies in the same bubble column with air and Drakeol 10 (a light mineral oil) exhibited the same trends in $\varepsilon_G(r)$ with pressure (Torczynski *et al.*, 1997).

2.2.2. Mass Transfer Correlations

Bubble-column reactors exist to synthesize an end product, usually through chemical reactions. Mass transfer rates between phases have also been measured under laboratory conditions in the hope of translating results to full-scale conditions. The rate of mass transfer to phase ϕ from another phase is expressed as the product of a mass transfer coefficient, k_ϕ, and the interfacial area between phases per unit volume, a. The product $k_L a$ is usually referred to as a volumetric coefficient for mass transfer to the liquid phase, and has units of reciprocal seconds. For correlations, k_ϕ is often combined with the mass diffusion coefficient, D_ϕ, and a relevant length scale to form a Sherwood number, Sh. For example, mass transfer from a gas bubble of diameter D_b to a surrounding continuous liquid phase can be described by the quantity

$$Sh \equiv \frac{k_L D_b}{D_L}. \tag{2.15}$$

The dimensionless group $(k_L a D_b^2 / D_L)$ is also commonly used.

An early paper by Hughmark (1967) proposed a semi-theoretical correlation for the liquid phase mass transfer coefficient in bubble columns, based on published data for air and a variety of liquids in the bubbly flow regime. The correlation agreed with the published data to within about 20%. Akita and Yoshida (1973) postulated another correlation for *Sh* based on dimensional analysis and experiments with oxygen bubbled through various liquids. To determine the mass transfer coefficient empirically, liquid samples were analyzed chemically for oxygen content after each test. While the correlation was in fair agreement with the data, the authors recommended "due caution" in its use with other systems. An improvement by Öztürk *et al.* (1987) on this correlation determined from a variety of gas-liquid systems is in better agreement with experimental data (average error 13%) and accounts for changes in gas density:

$$\left(\frac{k_L a D_b^2}{D_L}\right) = 0.62\left(\frac{\nu_L}{D_L}\right)^{0.5}\cdot\left(\frac{g\rho_L D_b^2}{\sigma_s}\right)^{0.33}\left(\frac{gD_b^3}{\nu_L^2}\right)^{0.29}\left(\frac{U_G}{\sqrt{gD_b}}\right)^{0.68}\left(\frac{\rho_G}{\rho_L}\right)^{0.04}. \tag{2.16}$$

Another study of mass transfer was made by Wilkinson *et al.* (1994). In this set of experiments, air was bubbled through a sodium sulfite solution, and the oxidation rate of the sodium sulfite was measured. The goal was to find the influence of fluid pressure on bubble sizes and mass transfer coefficients. Both $\bar{\varepsilon}_G$ and the volumetric mass transfer coefficient, $k_L a$, were found in experiments to increase significantly with increasing pressure of the phases, especially at high superficial gas velocities. Given the observation in the previous section that the average bubble size decreases as the pressure in the column rises, the trend in mass transfer with pressure is to be expected, since the interfacial area density, *a*, and the gas holdup are related by

$$a = \frac{6\bar{\varepsilon}_G}{D_{Sauter}}. \tag{2.17}$$

The Sauter mean bubble diameter, D_{Sauter}, is defined as the diameter of a hypothetical bubble having the same volume per unit surface area as the average of all bubbles in the flow. If the bubble size distribution in the flow of interest is $f_b(D_b)$, then the Sauter mean bubble diameter may be expressed as (Fan and Zhu, 1998)

$$D_{Sauter} \equiv \frac{\int_0^\infty D_b^3 f_b(D_b)dD_b}{\int_0^\infty D_b^2 f_b(D_b)dD_b}. \tag{2.18}$$

The trend in $k_L a$ with pressure observed by Wilkinson and co-workers was reversed at fluid pressures approaching 1.5 MPa when a foamy structure formed in the sodium sulfite solution. This foam decreased the local liquid volume fraction, limited the amount of sulfite in solution available to be oxidized, and decreased the mass transfer rate. Other variables found to improve the mass transfer coefficient included superficial gas velocity (higher U_G led to turbulence and to bubble breakup) and the addition of electrolytes such as sodium sulfite (which reduced the coalescence rate).

Based on a limited set of data from this study, Wilkinson *et al.* (1994) proposed an empirical correlation for D_{Sauter} that accounted for fluid pressure and its effect on gas density.

$$\frac{g\rho_L D_{Sauter}^2}{\sigma_s} = 8.8\left(\frac{U_G\eta_L}{\sigma_s}\right)^{-0.04}\left(\frac{\sigma_s^3\rho_L}{g\eta_L^4}\right)^{-0.12}\left(\frac{\rho_L}{\rho_G}\right)^{0.22} \tag{2.19}$$

For the mass transfer coefficient itself, they tentatively suggested the following relationship between two cases in which the same gas is introduced at different conditions.

$$\frac{[(k_L a)\rho_G]_2}{[(k_L a)\rho_G]_1} = \left\{\frac{[(\bar{\varepsilon}_G)\rho_G]_2}{[(\bar{\varepsilon}_G)\rho_G]_1}\right\}^n \tag{2.20}$$

Here n is estimated to range from 1.0 to 1.2. It was noted that validation of the general relationship requires data from other liquids and further research of the influences of ρ_G and p_{col} on $\bar{\varepsilon}_G$ and $k_L a$.

2.2.3. Flow Regime Transitions

Because of the increased mixing in turbulent flows, a useful parameter in bubble-column design is the transition point from bubbly flow to churn-turbulence in gas-liquid vertical flows, or the transition point to an analogous regime in gas-solid or three-phase flows. Wilkinson *et al.* (1992) discuss influences on the transition from homogeneous to churn-turbulent flow in gas-liquid vertical flows. Through empirical correlations they determined that the superficial gas velocity at transition $U_{G,trans}$ and the gas holdup at transition $\bar{\varepsilon}_{G,trans}$ increase under the following circumstances: (1) with the addition of electrolytes to the liquid phase (which reduces bubble coalescence), (2) with increases in gas density (because of the increase in bubble breakup), and (3) with decreases in η_L and σ_s (which reduce initial bubble size). From regression analysis, they obtained the following empirical equation for $\bar{\varepsilon}_{G,trans}$:

$$\bar{\varepsilon}_{G,trans} = 0.5\cdot\exp(-193\rho_G^{-0.61}\eta_L^{0.5}\sigma_s^{0.11})\,. \tag{2.21}$$

Valid ranges of the phase properties are: $\sigma_s = 0.02 - 0.073$ N/m, $\eta_L = 0.0004 - 0.055$ kg/m/s, $\rho_G = 0.09 - 3.8$ kg/m^3. The typical error of this correlation was just under 10% of the experimental value of $\bar{\varepsilon}_{G,trans}$. By comparison, Reilly *et al.* (1994) used their power-law relationship (Eq. 2.13) to derive the transition formulas

$$\bar{\varepsilon}_{G,trans} = 0.59C_1^{1.5}\left(\frac{\rho_G^{0.96}\sigma_s^{0.12}}{\rho_L}\right)^{0.5}, \qquad (2.22)$$

where the coefficient C_1 assumes the value for churn-turbulent flow, and

$$U_{G,trans} = \frac{1}{2.84}\frac{\sigma_s^{0.12}}{\rho_G^{0.04}}\bar{\varepsilon}_{G,trans}(1-\bar{\varepsilon}_{G,trans}). \qquad (2.23)$$

These formulas were derived from data over the following ranges: $\sigma_s = 0.024 - 0.073$ N/m, $\rho_L = 685 - 1460$ kg/m^3, $\rho_G = 0.16 - 13.0$ kg/m^3. The holdup correlation was compared to high-pressure pilot plant data from other investigators as well as the data used in its own derivation. For experimental holdup values less than 0.5, the relative error was less than 10%, slightly better than the Wilkinson correlation. The error increased rapidly at higher holdup values, however, requiring the adjustments described earlier in conjunction with Eq. 2.13.

Reilly *et al.* (1994) and Krishna *et al.* (1994) also found in experiments that increasing ρ_G extended the stability of the bubbly flow regime to higher $\bar{\varepsilon}_{G,trans}$. Krishna and Ellenberger (1996) found Eqs. 2.22 and 2.23 to be more accurate than Eq. 2.21 for a variety of gas-liquid systems. They also found that a theoretical criterion by Biesheuvel and Gorissen (1990) for the instability of homogeneous bubbly flows overpredicted $\bar{\varepsilon}_{G,trans}$ and should be considered to be an upper limit on the gas volume fraction at transition. In actual bubbly flows, destabilizing influences (such as an imperfect gas distribution) will lead to an earlier transition to churn-turbulence than the criterion predicts. They recommended further study of the influence of the liquid phase properties on $\bar{\varepsilon}_{G,trans}$ and $U_{G,trans}$.

2.3. Mathematical Multiphase Flow Models

Until recently, multiphase measurement methods have only been able to produce time-averaged and volume-averaged quantities that could not reveal instantaneous flow structures. Tomographic methods, such as the electrical impedance method discussed in the next three chapters, have the potential to provide the instantaneous three-dimensional phase distributions and flow fields needed to understand dynamic flows. The alternative to experimentation, multiphase mathematical modeling, suffers from the current inability to determine three-dimensional phase and velocity distributions. In practice, experimental observations of $\varepsilon_G(r,\theta,z)$ must be used as a starting point, or simplified one-dimensional predictive models must be used. A comprehensive review of hydrodynamic bubble-column models is given by Shah and Deckwer (1983), including information on bubbly flows, flow regimes, averaged liquid flow patterns, bubble size populations, and bubble coalescence and breakup. Useful tools for bubble-column reactor design can also be found in Deckwer and Schumpe (1993).

Two models of gas volume fraction distribution and three models for phase velocity are discussed in this section; one of the latter can be applied to a wide variety of two-phase systems. These have been derived from first principles and substantiated by empirical data and correlations. A starting point for many models of vertical multiphase flows is the two-fluid model. This model treats the multiphase fluid as a continuum in which the equations of conservation of mass and momentum and the first and second laws of thermodynamics may be written for each phase in the flow. The continuum model implies that the phases can overlap one another in space, and that phase changes and chemical reactions can alter the amount of mass in each phase (but not the total mass in the flow). By averaging terms in the equations over time, space, or other appropriate variables, the two-fluid model may be reduced to more manageable forms, such as the one-dimensional two-phase bubble-column model used to derive Eqs. 2.1 through 2.10. The one-dimensional two-phase bubble-column model assumes steady-state, axisymmetric flow that is fully developed and involves no end effects. The largest source of uncertainty in the two-fluid model (even in its simplified one-dimensional two-phase form) is the stress tensor, which has been simplified in different ways with varying degrees of success (Kumar *et al.*, 1994).

2.3.1. One-Dimensional Gas Volume Fraction Models

In the early paper by Zuber and Findlay (1965), a general expression was derived for the area-averaged gas volume fraction, $\langle \varepsilon_G \rangle$, in one-dimensional, dispersed two-phase flow. This expression accounted for nonuniform flows and concentration distributions and included the effects of local relative velocities between the phases. Their iterative formulation applied to any two-phase flow regime with no heat or mass transfer:

$$\langle \varepsilon_G \rangle = \frac{U_G / U_m}{C_0 + C_1} , \tag{2.24}$$

where U_m is the superficial velocity of the mixture. The effects of the local relative velocities between phases are accounted for by the second term in the denominator, defined as

$$C_1 = \frac{\frac{1}{A_{col}} \int_A \varepsilon_G u_{Gd} dA_{col}}{\langle \varepsilon_G \rangle U_m} . \tag{2.25}$$

The effects of nonuniform flow and phase concentrations are included in the distribution parameter C_0, which can take on several mathematical forms. While both terms in the denominator are defined analytically, simpler expressions can be derived for fully developed flow profiles in the churn-turbulent bubbly flow and slug flow regimes. The analytical model compared well to churn-turbulent and slug flow data, inasmuch as reasonable assumptions were made where values of the key variables were unavailable in the literature. Shah and Deckwer (1983) discuss analytical expressions for u_{Gd} that may be derived from the same model assuming no bubble-bubble interaction.

A gas-liquid model advocated by Wilkinson and co-workers (Wilkinson, 1991; Wilkinson *et al.*, 1992) treats the transition to churn-turbulent flow as the addition of large bubbles to a constant number of small, homogeneous bubbles. This model assumes that the small bubble population (and thus the volume fraction $\bar{\varepsilon}_{G,small}$) increases linearly with U_G in the bubbly flow regime. At transition it is assumed that the small bubble volume fraction remains constant, and additional gas flow produces only large bubbles. Deckwer and Schumpe (1993) found that the model agreed with experimental data to within 10%, but their data invalidated the assumption of a constant small bubble population and indicated a need for improvement.

Krishna and Ellenberger (1996) modified this model by using an analogy to gas-solid fluidized beds. Their modifications were justified by empirical gas holdup data of air in water; air in liquid hydrocarbons; and helium, argon, and SF_6 in tetradecane. Vertical columns used in the tests had diameters ranging from D_{col} = 10 cm to 63 cm, beyond the lower scaleup limit of 15 cm. Using the dynamic gas disengagement method, they determined the volume fractions $\bar{\varepsilon}_{G,large}$ and $\bar{\varepsilon}_{G,small}$ of "large" and "small" bubbles in the column. They then grouped the small bubbles and liquid together into a "dense" phase and identified the large bubbles as the "dilute" phase. The key quantity in their revised model is the dense phase volume fraction, $\bar{\varepsilon}_{df}$, defined by analogy to fluidized beds as the volume fraction of small bubbles in the liquid alone, excluding the volume of large bubbles:

$$\bar{\varepsilon}_{df} \equiv \frac{\bar{\varepsilon}_G - \bar{\varepsilon}_{G,large}}{1 - \bar{\varepsilon}_{G,large}} = \frac{\bar{\varepsilon}_{G,small}}{\bar{\varepsilon}_{G,small} + \bar{\varepsilon}_L} . \tag{2.26}$$

Krishna and Ellenberger found that $\bar{\varepsilon}_{df}$, not $\bar{\varepsilon}_{G,small}$, was independent of superficial gas velocity beyond the transition to turbulence. This indicates that the volume fractions of small bubbles and liquid with respect to the flow volume decreased by the same proportion with increases in U_G and $\bar{\varepsilon}_{G,large}$. Studies with a variety of frits and a gas sparger demonstrated that $\bar{\varepsilon}_{df}$ and $\bar{\varepsilon}_G$, but not $\bar{\varepsilon}_{G,large}$, were dependent on the gas distribution, with more uniform distributions at the entrance region yielding higher gas holdups. They also reported that $\bar{\varepsilon}_{df}$ increased with gas density.

Using their modified two-phase model, the bubble coalescence model of Darton *et al.* (1977), their dynamic gas disengagement data, and the empirical formulas for transition of Reilly *et al.* (1994) (Eqs. 2.22 and 2.23), Krishna and Ellenberger produced the following correlation for the large bubble volume fraction for $U_G > 0.1$ m/s:

$$\bar{\varepsilon}_{G,large} = 0.268 \frac{(U_G - U_{G,trans})^{0.58}}{D_{col}^{0.18}} \tag{2.27}$$

The correlation reflects their observation that $\bar{\varepsilon}_{G,large}$ is not affected by liquid properties, gas density, or the gas phase distribution, but is significantly affected by the column diameter. The error of Eq. 2.27 in predicting large bubble holdups was 24% on average, but much less than the error of the original correlations by Wilkinson. Further research into analytical or semi-empirical models of gas holdup is indicated.

2.3.2. Mass Transfer Models

A model commonly used to simulate liquid mixing phenomena in a vertical bubble column is the axial dispersion model, which lumps mixing phenomena within the liquid phase into a single dispersion coefficient. The model is less than effective, however, and Degaleesan *et al.* (1997) cite the wide variety of correlations for the dispersion coefficient in the literature as evidence. Deckwer and Schumpe (1993) suggest the simulation of circulation cells within the column as an improvement over axial dispersion and similar lumped models.

Degaleesan *et al.* (1997) developed and evaluated such a bubble-column model for mass transfer that compartmentalizes the column into upflow, downflow, and end regions. This approach simulates, on a coarse scale, the liquid mixing in bubble-column reactors that is driven by radially varying gas volume fractions and turbulent eddies. Each finite volume is governed by its own set of convection and diffusion equations for the concentration of chemical species, and separate equations quantify the transfer of the species between volumes. The concentration is expressed as a function of position and three parameters: liquid velocity, gas volume fraction, and turbulent eddy diffusivity. Using a radioactive particle tracking method, these parameters were measured in churn-turbulent flow in an air-water bubble column of diameter D_{col} = 19 cm. The measured quantities were then put into the model to successfully predict the motion of dye injected at the bottom of the column. To test the usefulness of the model in scaleup situations, the values of $u_L(r)$ and eddy diffusivity from the air-water system were extrapolated to churn-turbulent conditions in a three-phase slurry bubble-column reactor (D_{col} = 46 cm) used for methanol synthesis. Nuclear densitometry measurements of the radial gas volume fraction profile in the reactor supplied the remaining input to the model. The convective diffusion model did a "fair to good job" of predicting the concentration of radioactive tracers injected into the column, and determined the recirculation rate of the liquid phase well. By comparison, axial dispersion models consistently underpredicted or overpredicted the radioactive tracer concentration. Work is continuing to substantiate the correlations used in the scaleup from laboratory to industrial conditions.

2.3.3. Phase Velocity Models

Beginning with the steady-state equations of motion for a gas-liquid flow of infinite height, Ueyama and Miyauchi (1979) solved for the liquid velocity radial profile, $u_L(r)$, and cross-sectionally-averaged relative velocity, $\langle u_r \rangle$, in a cylindrical bubble column, for both "batch mode" (no net liquid flow) and a recirculating flow. For the shear stress term, the derivation assumes that the turbulent kinematic viscosity, ν_t, is much larger than the molecular kinematic viscosity, ν_{mol}, everywhere except near the column wall. The law of the wall is used in the boundary conditions, and a gas volume fraction profile of the following form is assumed:

$$\frac{\varepsilon_G(r)}{\bar{\varepsilon}_G} = \frac{n+2}{n}\left[1-\left(\frac{r}{R_{col}}\right)^n\right]. \qquad (2.28)$$

The end results include a complex expression for $u_L(r)$ and an example result for $\langle u_r \rangle$ for the case of $n = 2$. Specific solutions from this model require empirical values for v_t and an assumed value of the exponent n in the gas volume fraction profile. Experimental values of $\langle u_r \rangle$ and $u_L(r)$ from literature on recirculating bubble-column flows were in agreement with the model, but values of n were chosen based on the gas volume fraction data itself. Shah and Deckwer (1983) reported an extension of the model to a more general profile of $\varepsilon_G(r)$, and presented a previously unpublished solution for $u_L(r)$ by H. P. Riquarts based upon the same assumptions made by Miyauchi and co-workers.

Ishii and Zuber (1979) successfully developed versatile expressions for drag coefficients and drift velocities of bubbles, drops, and solid particles in a dispersed two-phase flow. They began with the two-fluid model, then applied drag similarity criteria and incorporated a mixture viscosity model. The drag similarity approach involves a relationship between the drag coefficients of a single "particle" of the dispersed phase in two extreme cases: (1) a continuous liquid medium with no other particles, and (2) a multiparticle system in which the continuous phase contains many other particles. The reference drag coefficient for the single particle-infinite medium case may be taken from any appropriate correlation for the flow of interest, and the authors have used such correlations to derive drift velocities for a matrix of dispersed phases and flow regimes. The mixture viscosity model considers not only the properties of the dispersed and continuous phases, but also the resistance to particle motion caused by the presence of other nearby particles in the fluid, and is given as

$$\frac{\eta_m}{\eta_C} = \left(1-\frac{\varepsilon_D}{\varepsilon_{D,MAX}}\right)^{-2.5\varepsilon_{D,MAX}\frac{\eta_D+0.4\eta_C}{\eta_D+\eta_C}}. \qquad (2.29)$$

$\varepsilon_{D,MAX}$ is the volume fraction of the dispersed phase at the maximum possible packing. Ishii and Zuber suggest $\varepsilon_{D,MAX} = 0.62$ for monodisperse solid particles and 1.0 for fluid particles; although the theoretical value of $\varepsilon_{D,MAX}$ for solid spheres is 0.75, their value is based upon the maximum packing observed in practical situations. In Eq. 2.29 and the summary of drift velocity formulas which follows, the subscript *C* refers to the continuous phase, whereas *D* marks the dispersed phase of drops, bubbles or particles.

For solid or fluid "particles" in viscous flows, the recommended formula for the drift velocity of the dispersed phase is

$$u_{Dd} \cong u_{b\infty}(1-\varepsilon_D)^{2.0}\frac{\eta_C}{\eta_m}\frac{1+\psi(R_b^*)}{1+\psi(R_b^*)\left[\sqrt{1-\varepsilon_D}\frac{\eta_C}{\eta_m}\right]^{6/7}}, \qquad (2.30)$$

where

$$\psi(R_b^*) = 0.55\left[(1+0.08R_b^{*3})^{4/7}-1\right]^{0.75}, \tag{2.31}$$

the nondimensionalized radius of the "particle" is given by

$$R_b^* = R_b\left(\frac{\rho_C g|\rho_C-\rho_D|}{\eta_C^2}\right)^{1/3}, \tag{2.32}$$

the terminal velocity, $u_{b\infty}$, is described by

$$u_{b\infty}|u_{b\infty}| = \frac{8}{3}\frac{R_b}{C_D\rho_C}(\rho_C-\rho_D)g\,, \tag{2.33}$$

and the drag coefficient, C_D, is computed from an appropriate drag correlation for a single particle in an infinite medium at the corresponding Reynolds number. For solid particles at higher Reynolds numbers, which occur at $R_b^* \geq 34.65$, Eq. 2.30 simplifies to

$$u_{Dd} = u_{b\infty}(1-\varepsilon_D)^{2.0}\frac{\eta_C}{\eta_m}\frac{18.67}{1+17.67\left[\sqrt{1-\varepsilon_D}\,\frac{\eta_C}{\eta_m}\right]^{6/7}}\,. \tag{2.34}$$

For liquid drops or gas bubbles at higher Reynolds numbers, two formulas are derived. The first is for flow regimes in which the "particle" distorts and turbulent eddies influence its motion. Since the drag coefficient on the particle no longer depends upon viscosity, the formulas for terminal velocity and drift velocity simplify to produce

$$u_{Dd} \cong u_{b\infty}(1-\varepsilon_D)^n, n = \begin{cases} 1.75, \eta_C >> \eta_D \\ 2, \eta_c \approx \eta_D \\ 2.25, \eta_C << \eta_D \end{cases}, \tag{2.35}$$

where the magnitude of the terminal velocity is given by

$$|u_{b\infty}| = \sqrt{2}\left(\frac{g\sigma_s|\rho_C-\rho_D|}{\rho_C^2}\right)^{1/4}, \tag{2.36}$$

and the sign of $u_{b\infty}$ depends upon the sign of the density difference $\rho_C-\rho_D$ (*i.e.*, whether the particle rises or falls). Equation 2.35 is applicable above the following viscosity number:

$$\frac{\eta_C}{\left(\rho_C\sigma_s\sqrt{\sigma_s/g|\rho_C-\rho_D|}\right)^{0.5}} \geq 0.11\frac{1+\psi(R_b^*)}{\left[\psi(R_b^*)\right]^{8/3}}\,. \tag{2.37}$$

The second formula for a dispersed fluid phase applies to fully churn-turbulent flow, where "particles" can influence one another directly. In this instance the effective drag coefficient for droplets or cap bubbles simplifies further, and a force balance yields

$$|u_{b\infty}| = C\left(\frac{g\sigma_s|\rho_C-\rho_D|}{\rho_C^2}\right)^{1/4}\left(\frac{\rho_C-\rho_D}{|\rho_C-\rho_D|}\right)(1-\varepsilon_D)^{1/4}, \tag{2.38}$$

with $C = \sqrt{2}$ for bubbles and $\pi/2$ for droplets.

These velocity formulas compared satisfactorily to data and empirical correlations for flows with many different dispersed phases, including solid particles, liquid drops, bubbles, foams, and liquid-liquid dispersions. Agreement with measured data was particularly good for low dispersed-phase volume fractions. The authors considered these results to support both the drag similarity method and the mixture viscosity model.

Finally, Kumar *et al.* (1994) addressed the biggest difficulty with the two-fluid model: the Reynolds shear stress term has not yet been accurately modeled or predicted in either laboratory or industrial flow scales, making the scaleup of laboratory results difficult. Many versions of the one-dimensional bubble-column equations have been developed with different models for the shear stress term, τ_{rz}, but no model has successfully recreated all velocity or gas volume fraction data. Usually an expression is obtained for τ_{rz} in terms of ν_t, the turbulent kinematic viscosity, or l_P, Prandtl's mixing length scale (see also White, 1991):

$$\tau_{rz} = -\rho_L \nu_t\left(\frac{du_L}{dr}\right) = -\rho_L l_P^2\left(\frac{du_L}{dr}\right)\left|\frac{du_L}{dr}\right|. \tag{2.39}$$

In both expressions, du_L/dr is the radial gradient of the liquid axial velocity. The quality of fits to experimental data on u_L is sensitive to the model chosen for these quantities, particularly the model for l_P. Kumar and co-authors began with the expression for shear stress involving l_P, based on the fact that shear stress and velocity gradients can be derived from experimental measurements. Their goal was to use experimental data to determine the behavior of the mixing length scale and thus predict the effects of scaleup on bubble-column hydrodynamics.

Experimental measurements were made of the local gas volume fraction distribution, $\varepsilon_G(r,\theta,z)$, and local axial liquid velocity profile, $u_L(r,\theta,z)$, in vertical columns of five different diameters using radioactive particle tracking and tomographic techniques. Radial profiles of the Prandtl length scales of the flow were computed for one of the columns (D_{col} = 19 cm), and these $l_P(r)$ profiles were used with the one-dimensional model and the $\varepsilon_G(r)$ profiles to predict $u_L(r)$ in all five columns. The predicted and measured liquid velocity profiles were in close agreement for all columns greater than 15 cm in diameter. As expected, the model results for smaller columns deviated from the measured data because of the presence of slug flow instead of churn-turbulent flow in the column. The use of data from the 19-cm column to create valid correlations for larger columns was evidence that scaleup of velocity profiles is possible.

Because disparate values of U_G and D_{col} were used, no trends in l_P were evident, and a universal correlation for the mixing length was not reported. It was suggested that such a correlation or parametric model for conditions ranging from experimental to prototype columns would improve the accuracy of the one-dimensional, two-fluid model.

2.4. Reactor Design and Scaleup Issues

While the experiments and the analytical and semi-empirical models described in this chapter were intended to advance the body of knowledge on bubble-column reactors and multiphase flows in general, more must be learned to improve reactor design and scaleup from test conditions to operating conditions. On the experimental side, scaleup is made difficult by the lack of data taken under industrial conditions. Relatively few experiments have been conducted at elevated temperatures and pressures, although the works of Zou *et al.* (1988) and Lin *et al.* (1998) have addressed this issue. On the analytical side, challenges remain in developing scaleup models based on physics instead of experimental correlations and "know-how." For instance, Kumar *et al.* (1994) mention in passing that a two-dimensional model derived from the full two-fluid model exists, but that analytical and numerical solutions are not yet practical. The difficulty of modeling shear stress in the two-fluid model, even in its simplification to the one-dimensional two-phase model, is another example. Mathematical modeling of the three-dimensional flow also suffers from the fact that gas volume fraction profiles, which strongly influence circulation and chemical reaction rates, have yet to be determined analytically and so must be provided by experimental observation.

In the review by Deckwer and Schumpe (1993) of bubble-column design tools, a conceptual list is presented of phenomena known to influence bubble-column operation (see Figure 2.3). In particular, hydrodynamics, heat transfer, and mass transfer are identified as the scale-dependent properties that pose problems in the application of experimental data to design work. As solutions, the gas holdup correlation of Akita and Yoshida (1973), Eq. 2.11, and the mass transfer correlation of Öztürk *et al.* (1987), Eq. 2.16, are recommended for conservative design use; note, however, that the column diameter used in the latter study was less than 15 cm. Reaction kinetics and the number of reactive species (not the same as the number of phases) are mentioned as factors that can easily be included in mathematical models but have rarely been studied experimentally.

On the other hand, dimensional effects have been extensively studied. As discussed earlier, the common observation that gas holdup is insensitive to column dimensions when D_{col} > 15 cm and $(H_{col} / D_{col}) > 5$ has provided a lower limit for scaleup studies of the effects of gas flow on other quantities. The same insensitivity does not hold, however, for radial gas volume fraction profiles (Kumar *et al.*, 1997) or the bubble-size distribution (Krishna and Ellenberger, 1996). Column size is also believed to influence the liquid mixing scales, which in turn may influence reaction rates. From their model for $u_L(r)$, Ueyama and Miyauchi (1979) deduced that the turbulent kinematic viscosity, which is related to the mixing scale by Eq. 2.39, would increase as $D_{col}^{1.5}$ or a higher power, but the dependence had not yet been confirmed for large

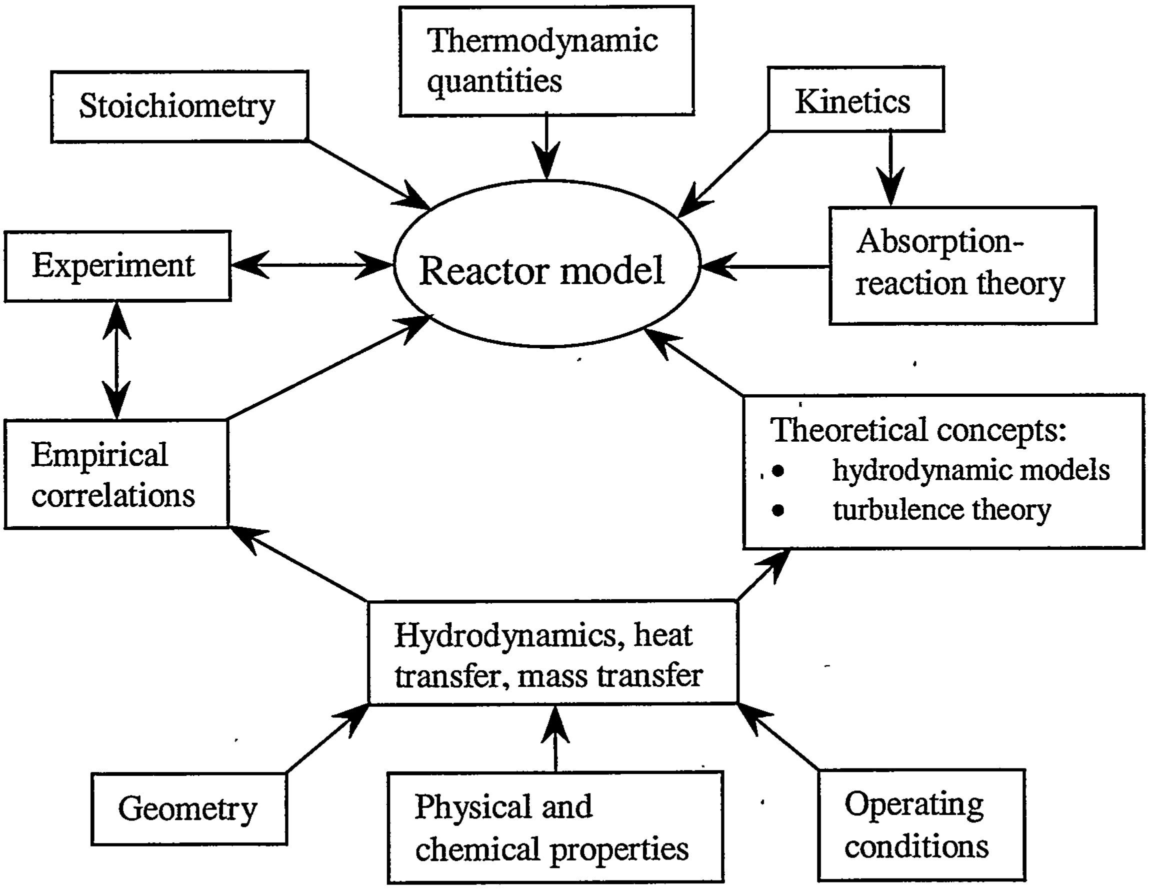


Figure 2.3. Conceptual diagram of the modeling process for bubble-column reactors. Adapted from Deckwer and Schumpe (1993).

columns. The work of Kumar *et al.* (1994) indicates that mixing effects can be scaled from small to large columns, once their dependence on other conditions is determined.

On the subject of mixing, Deckwer and Schumpe (1993) state that liquid mixing has little effect on reaction rates or reactor performance and that gas mixing, which does influence performance, is hard to quantify and therefore to predict. They recommend further studies of gas phase structures, bubble rise velocities, and related gas phase hydrodynamics. This points to the most serious deficiency in current experimental studies of bubble-column flows: measurements are typically made of time-averaged, volume-averaged quantities under the assumption that the flow field is steady-state and one-dimensional in the axial direction. Bubble-column reactors and fluidized beds are known to have three-dimensional, dynamic flow structures, but because most instrumentation is limited to measurements of average quantities, these multidimensional instantaneous flow structures are lost. Methods must be developed that can provide the instantaneous flow field information needed to characterize dynamic structures and gas mixing.

The last issue that must be addressed for successful bubble-column design is the lack of information for solid phases in multiphase flows. Fluidized beds involve a "dense phase" of solid particles, and many chemical processes involve a powdered catalyst suspended in a liquid hydrocarbon; this makes knowledge of the effects of solid properties on mass transfer and phase distributions desirable. Unfortunately, practical models of multiphase flows involving a solid phase are scarce, although the model by Ishii and Zuber (1979) includes drift velocities for solid particles in viscous and inviscid flows. A review by Wilkinson *et al.* (1992) of experimental work in the literature concluded that the addition of solids to a gas-liquid flow increases bubble size and decreases gas holdup slightly, all else being equal.

Daly (1990) conducted experiments in two slurry bubble-column reactors of diameters 5 and 21 cm using iron oxide and silica powders, molten wax, and nitrogen gas as the solid, liquid, and gas phases, respectively. Solid volume fraction axial profiles were measured for column-averaged solid volume fractions of up to 0.30; the axial profiles were found to be uniform for particles 5 μm in diameter or less, but particles 20 μm in diameter or larger tended to settle and produce nonuniform axial distributions. This was counteracted by the introduction of a co-current flow in the solid/liquid slurry at a velocity larger than the terminal settling velocity of the solid phase. Several tests were conducted with a dual-source gamma densitometer with the intent to measure radial and axial volume fraction profiles of all three phases, but similarities in the gamma attenuation coefficients of the phases limited the resolution of measurements.

To address the unresolved issues of phase spatial variations and effects of solids in multiphase flows and to add to the knowledge base in these areas, the remaining chapters of this report present an investigation of radial volume fraction profiles in solid-liquid, gas-liquid, and solid-gas-liquid vertical column flows. While the experiments are conducted at ambient conditions, the column diameter (D_{col} = 19 cm) and fluid height ($H_{col} / D_{col} = 7.6$) in gas-liquid and three-phase flows are acceptable from the standpoint of scaleup. The experiments are also notable in that they mark the first application of electrical-impedance tomography to the quantitative measurement of volume fraction profiles in industrially relevant vertical multiphase flows.

3. The Sandia/Michigan Electrical-Impedance Tomography (EIT) System

The review in Chapter 2 of the current knowledge of vertical multiphase flows concluded with a discussion of the information that is lacking in experimental and analytical models. On the experimental side, instantaneous (rather than time-averaged) phase information and the influences of a solid phase on other phases in the flow are some of the data needed to improve the understanding of bubble-column reactors. On the analytical side, spatial variations in phase distributions are needed for three-dimensional flow modeling, but only one-dimensional radial phase distributions regularly appear in the literature; multidimensional measurements are extremely rare.

Electrical-impedance tomography (EIT) has been investigated in the literature for its ability to determine phase distributions in one and two dimensions, most often in a qualitative manner through phase "images." An electrical-impedance tomography system has been developed collaboratively by Sandia National Laboratories and the University of Michigan. This system is being extensively evaluated for its ability to quantitatively measure, rather than qualitatively image, phase distributions in solid-liquid, gas-liquid, and solid-gas-liquid vertical column flows. The motivation for this study is that quantitative EIT reconstructions are more useful in addressing the issues of analytical flow modeling and the effects of a solid phase. Also, EIT systems have the potential to obtain data over time scales of milliseconds, a feature that may allow EIT to obtain information on "instantaneous" flow structures that can, in turn, improve predictions of mixing and chemical reaction rates.

This chapter presents the details of the Sandia reconstruction algorithms and the Michigan EIT hardware. The development of both the hardware and software has been documented during their evolution (O'Hern *et al.*, 1995; Torczynski *et al.*, 1996a, 1997; George *et al.*, 1998a, 1998b, 1999c), but this chapter provides greater detail on the electronics and documents the reconstruction codes themselves. The chapter begins with an introduction to EIT theory, followed by a description of the EIT hardware and electronics. Next, the reconstruction algorithms are described along with the numerical and experimental tests used to validate them. The chapter concludes with the choice of electrode geometry for multiphase flow measurements, a process involving comparisons of domain reconstructions with numerical models. While the EIT system described in this chapter is not fast enough to obtain "instantaneous" data from multiphase flows, subsequent chapters will evaluate its ability to obtain accurate, quantitative phase measurements on time scales of tens of seconds to minutes, more quickly than other more common tomographic methods.

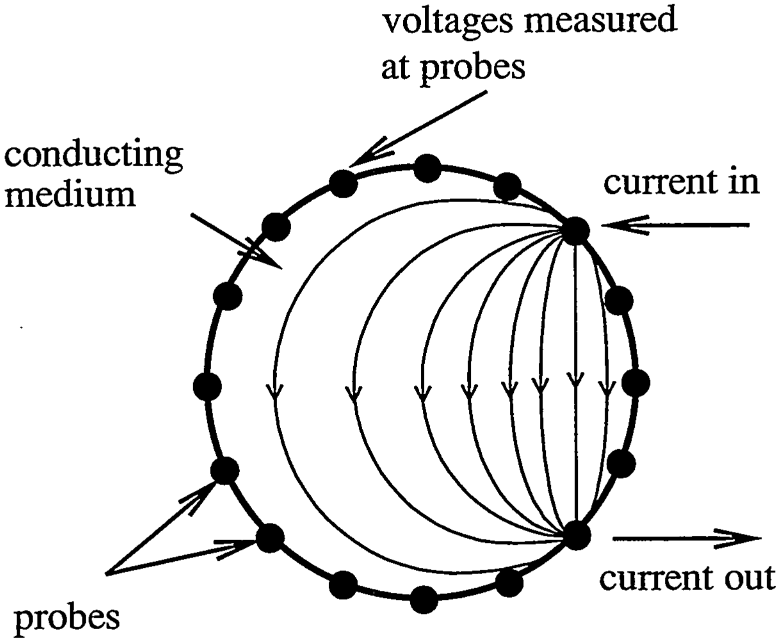


Figure 3.1. Conceptual diagram of an EIT system applied to a circular domain.

3.1. EIT Theory

In electrical-impedance tomography, a number of electrodes are mounted to the surface of a domain of interest, **D**. As a prescribed current is injected into the domain at one electrode and withdrawn at another electrode, all electrode voltages are measured relative to a reference voltage (Figure 3.1). These measurements at the domain boundary are then used to reconstruct the impedance distribution within the domain and infer the phase distribution.

Alternating current (AC) is used in EIT applications to avoid polarization effects on the electrodes. For AC electrical conduction with field frequencies on the order of tens of megahertz or lower, the electric potential within the domain, V, is related to the complex electrical conductivity of the domain, γ, by (Webster, 1990)

$$\nabla \cdot \gamma \nabla V = 0. \tag{3.1}$$

Equation 3.1 assumes that no charge sources or sinks are present in **D** and is identical in form to the steady-state heat-conduction equation. The boundary conditions on **D**, which enforce conservation of charge, are given by

$$\mathbf{n} \cdot \gamma \nabla V + q = 0, \tag{3.2}$$

where **n** is the unit normal vector outward from the domain boundary, and q is the charge flux on the boundary. Multiple measurements of q and V at the boundary of **D** are used to reconstruct the conductivity distribution within **D**. An iterative reconstruction process is required to arrive at a conductivity distribution that will yield the measured boundary conditions. The phase distribution is then inferred from the reconstructed conductivity or impedance distribution.

In practice, current injections and voltage measurements are performed at a finite number of locations on the boundary — *i.e.*, at the electrodes — and these measurements may be averaged over a portion of the boundary surface. Consequently, the limited resolution of the reconstructed conductivity field will be strongly related to the number and locations of the electrodes used to probe the domain. If N electrodes are used, the domain can be modeled as an N-port impedance network. A prescribed current is injected into the domain at one port, known as the current source or injection electrode, and withdrawn at another port, the current sink or withdrawal electrode. The resulting voltage distribution around the "network" can be measured at $(N - 1)$ ports, with one port (often the current sink) referenced to ground. The total number of linearly independent voltage measurements at the domain boundary, M, is then given by

$$M = \frac{N(N-1)}{2} . \tag{3.3}$$

M is also the number of independent impedance elements that can be determined from these voltage "projections" and used to model the domain.

To reconstruct the impedance distribution in the domain, a "candidate" impedance distribution is first constructed. A set of voltage projections is then computed from the candidate distribution. The candidate projections are compared to voltages measured on the boundary during current injection, and their difference is characterized by some error criterion. Finally, the candidate distribution is modified based on the error, and a new set of candidate projections is computed. This iterative process continues until some minimum error criterion is satisfied. Different reconstruction methods vary in the manner in which the impedance distribution is modified; the reader is referred to George (1999a) for an extensive discussion of reconstruction methods and their accuracy.

EIT reconstruction is an ill-conditioned problem. In the case of the forward problem, where a known conductivity distribution is used with Eqs. 3.1 and 3.2 to solve for voltages on the domain boundary, a single voltage solution exists for each distribution in the domain. In the inverse problem, where voltages on the boundary are used to solve for the interior conductivity distribution, such a one-to-one relationship does not generally exist. As a result, the reconstructed impedance solution may not represent a phase distribution that is physically possible or may not satisfy requirements of continuous dependence on the problem data. A more important consequence of ill conditioning is that the solution is sensitive to noise in the projections. As the number of electrodes and the spatial resolution of the solution increase, the ill-conditioning and the sensitivity to noise increase as well. When available, *a priori* information about features of the impedance distribution may be used to aid in reconstruction, as is the case for the vertical two-phase flows examined in this report.

Several simplifications are generally employed in EIT that make it possible to reconstruct the impedance distribution in the domain using discrete finite-element methods; the validity of these assumptions should be noted. First, current is commonly assumed to travel in a two-dimensional domain. In practice, applied fields are three-dimensional, especially for small, unguarded electrodes; unless the test domain is carefully chosen to be uniform in the third dimension, the 2-D image which results will reflect the 3-D nature of the excitation field. (While three-dimensional current flow has been used for some elementary cases, including the vertical bubble-column flows in this work, full three-dimensional EIT reconstruction would require impractical amounts of computational effort and resources.) Another common simplification involves representing the conductivity by a piecewise constant function – *e.g.*, a constant within each element of a finite-element mesh. This approach is not followed here; instead, the conductivity is represented as a global function of position and of several conductivity parameters. By using a global conductivity function, regularization is implicitly included in the solution procedure. It is also assumed in EIT that the impedance distribution within the domain does not change significantly over time while voltage projections are acquired. To satisfy this assumption, EIT systems must acquire projections rapidly, on time scales shorter than the characteristic time scales of multiphase flows. In the vertical multiphase flows examined in this report, this cannot be achieved, so the uncertainty introduced by temporal averaging of voltages prior to reconstruction must be quantified. This consideration is discussed in more detail later in this report.

More information on EIT theory and the development of EIT systems for the study of multiphase flows may be found in Dickin *et al.* (1993); Jones *et al.* (1992, 1993, 1994); and Ceccio and George (1996).

3.2. System Description

3.2.1. EIT Hardware and Electronics

A block diagram of the Sandia/Michigan EIT system (Torczynski *et al.*, 1997; George *et al.*, 1998b) is shown in Figure 3.2. The system hardware consists of an electrode array; a signal generator; a voltage-controlled current source; multiplexers connecting the electrode array to the current source, current ground, and measurement electronics; an instrumentation amplifier and phase-sensitive demodulators; a low-pass filter to eliminate high-frequency components from the demodulated signals; and a Data Translation® model DT2839 data acquisition card. The data acquisition card contains a 12-bit analog-to-digital converter that measures the demodulated signal components and a digital controller that can be used to select electrodes for current injection, current ground, and voltage measurement. The card also acts as an interface to a Pentium PC that operates the entire system. A photograph of the system is shown in Figure 3.3; a complete circuit diagram of the EIT package appears in Appendix A.

A variety of schemes have been employed in EIT systems to acquire voltage projection sets. In principle, voltages need only be measured across the two electrodes used to inject and withdraw current, and different combinations of injection and withdrawal electrodes will produce the complete set of voltages needed in reconstruction. Systems that employ this approach are referred to as "two-port" systems. In practice, however, the contact impedances at the interfaces between the two current-bearing electrodes and the domain will cause significant voltage drops

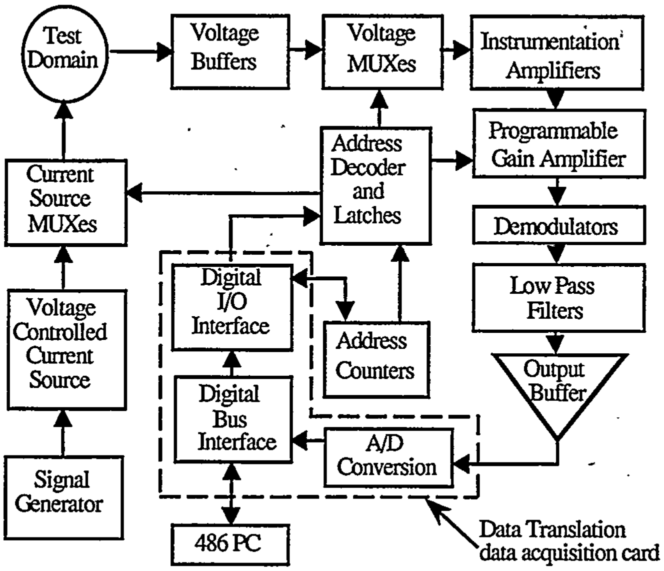


Figure 3.2. Block diagram of the Sandia/Michigan EIT system.

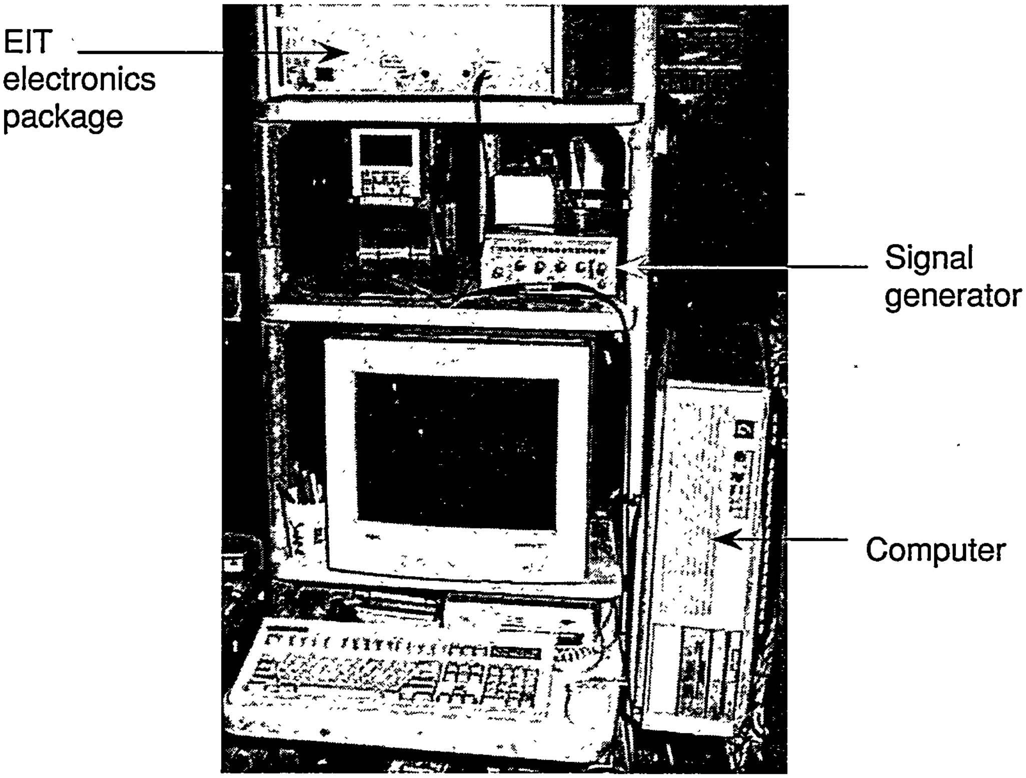


Figure 3.3. Photograph of the Sandia/Michigan EIT system.

across the interfaces and introduce errors in the reconstruction. Furthermore, to acquire a set of projections with a good signal-to-noise ratio, significant current should be induced within the entire domain to produce measurable voltage changes at the boundary, but two-port systems can concentrate current near the boundary and leave the interior of the domain unprobed.

Consequently, "four-port" systems have been devised that use two electrodes to inject and withdraw current and two other electrodes to measure differences in boundary voltages.

The Sandia/Michigan EIT system is a four-port system that injects a controlled current through one electrode, grounds a second electrode to withdraw the current, and measures voltages at all electrodes in sequence relative to the ground electrode. For completeness, a "projection set" consists of measurements at all *N* electrodes for each injection and ground combination, with a total of $N^2(N-1)/2$ voltage measurements. Note that this is a factor of *N* greater than the number of independent pieces of information, *M* (Eq. 3.3); the extra information is used to reduce the impact of noise on the reconstructions. The measurement at the injection electrode is not used in reconstructions in the Sandia/Michigan system because of the possible error from contact impedance discussed above (Torczynski *et al.*, 1996a, 1997).

The domain is excited with a 50-kHz AC electric field. At this frequency, the resistive component of impedance dominates for the air-water and polystyrene-water flows considered here (Ceccio and George, 1996), a fact which will be demonstrated in Section 3.2.2. At the head of the excitation circuitry is a Hewlett-Packard Model 3312A function generator that produces a stable, low-distortion 50-kHz sine wave of peak-to-peak magnitude 1 V. This magnitude was chosen based on the resistance and transient behavior of the multiplexers used to select electrodes for current injection and voltage measurement. The signal is sent through a buffer (a single operational amplifier, referred to hereafter as an "op-amp") to a voltage-controlled current source (VCCS, see Appendix A) that creates an injection current proportional to the voltage of the reference sine wave. In the present configuration, an input of 1 V p/p yields a current output of approximately 125 μA rms.

To keep the current amplitude stable over a wide range of domain impedances, the VCCS employs two op-amps in a positive feedback design: an OP42 that acts as the heart of the current source, and a National Semiconductor model LH0032, which buffers the current and provides positive feedback to the OP42. The output of the VCCS is AC coupled to prevent the transmission of a polarizing DC current to the electrodes. The conductivity of the domain must be checked and adjusted as needed before experiments, since high conductivity can cause saturation of the VCCS. A Burr-Brown model INA110 instrumentation amplifier connected to the sensing resistor of the VCCS provides a voltage proportional to the current output; this voltage may be measured by the data acquisition board to monitor the current output for stability or signs of saturation.

The reference sine wave is sent not only to the VCCS but also to phase-sensitive demodulators for use in domain voltage measurements. An all-pass filter also uses the reference sine wave to generate a quadrature reference for measurement purposes. The all-pass filter may be tuned with two potentiometers to ensure that the quadrature reference is 90º out of phase with, and has the same magnitude as, the original carrier reference.

The electrode array is connected to the VCCS and current ground via two Analog Devices model MUX16 analog multiplexers ("muxes" for short). Multiplexers are also used to connect the electrode array to the differential amplifier used to measure electrode voltages. Latch commands issued to the muxes determine which electrodes are selected for current injection, grounding, and voltage measurement, and the speed of these commands is the largest factor influencing the speed of the EIT system. Originally the controlling software issued these latch commands through the data acquisition card's digital controller and then triggered voltage measurements by the card. Later, integrated circuit (IC) counters were added to the EIT hardware to select injection, ground, and measurement electrodes in sequence and to trigger voltage measurements once the multiplexers were latched. In early tests with a 486 PC, the acquisition time for each 16-electrode voltage projection set was reduced from 2.7 seconds with the software command option to less than 0.75 seconds with IC counters. For this reason the software-controlled option is called "slow mode," whereas the IC-controlled option is referred to as "fast mode." Validation tests conducted using both fast and slow acquisition modes (presented later in this chapter) indicate no significant loss of accuracy with the increased acquisition speed. Presently, a Pentium PC is used to operate the system; each projection set requires 1.5 seconds in "slow mode" and less than one-half second in "fast mode."

Coaxial cable is used to carry current to and from the electrodes. To minimize current loss through the cables, voltage followers (OP42 op-amps) bring the shielding of each current-bearing cable to the same voltage as the electrode. A separate set of cables connects the electrodes to the muxes that control voltage measurements, eliminating any voltage drops from the measurements that would occur across the current-bearing cables. OP42 op-amps are also used as buffers between the electrodes and the voltage muxes.

As the measurement electrode and the ground electrode are selected by the voltage muxes, their voltage signals are passed to a Burr-Brown model PGA202 differential amplifier that measures the difference between the two voltages. The voltage on the grounded electrode is always used as the reference so that the PGA202 output represents the voltage on the measurement electrode relative to ground. The advantage of such a measurement method is that common mode noise that may be present on all the electrodes and coaxial cables is subtracted out of each measurement. The PGA202 was chosen for its good common mode rejection ratio at 50 kHz and for its programmable gain, which can allow the measurement of voltage differences over a wide dynamic range. For the experiments reported here, the gain was always set to unity, since adjustments to liquid conductivity were found to be more practical in obtaining a useful voltage range.

The signal from the differential amplifier is passed to a pair of phase-sensitive demodulators (Analog Devices model AD630). These separate the signal into two components: one in phase with the EIT carrier reference signal and the other out of phase with the carrier by 90°. One demodulator multiplies the measured electrode voltage by the carrier reference signal to obtain the carrier component, while the other uses the quadrature reference signal to obtain the out-of-phase component. By the rules of trigonometric multiplication, both demodulators produce a DC voltage superimposed on an AC voltage of 100 kHz, twice the reference frequency. These outputs are filtered using four-pole, low-pass Butterworth filters with a cutoff frequency of 25 kHz to yield only the DC components, which are proportional to the carrier and quadrature components of the measured voltages. (The recovery of both in-phase and out-of-phase signals permits the determination of the real and imaginary parts of the domain impedance, a feature that may be used in the future to distinguish between more than two phases in the domain.) The DC voltages are finally passed to the DT2839 data acquisition board, which measures the two voltages on separate analog input lines with a minimum conversion time of 2.4 μs. Two other pairs of demodulators and Butterworth filters process the voltage signal from the

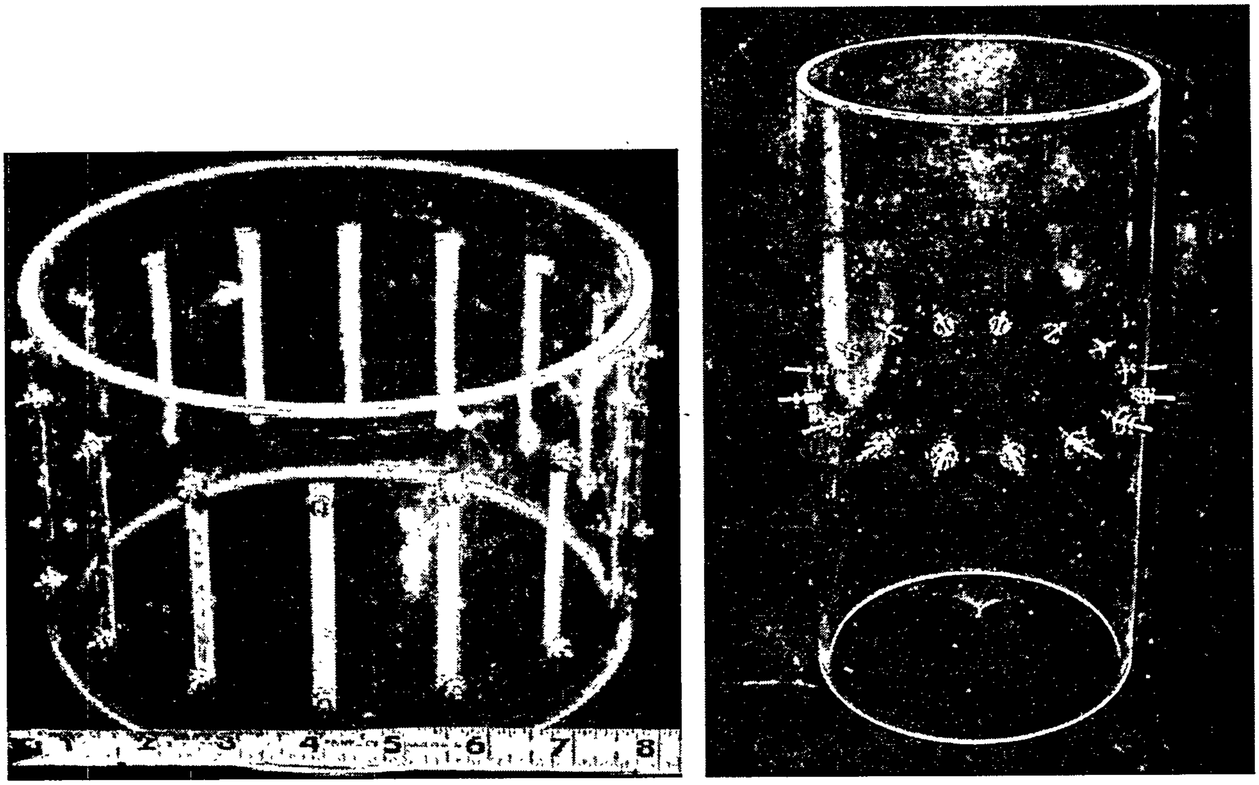


Figure 3.4. EIT electrode arrays. (Left) strip electrode array, (right) disk electrode array. Both cylinders are of the same diameter; the scale in the left-hand photo is in inches.

INA110 amplifier that monitors injection current, so that the DT2839 may measure injection current on two other input lines if necessary.

The electrode arrays used with the Sandia/Michigan system were fabricated for use as stand-alone testbeds in validation tests as well as for use in a transparent bubble column described in the next chapter. Two of these arrays are shown in Figure 3.4. Sixteen strip electrodes, each with an aspect ratio of 12, were fashioned from stainless steel strips 0.64 cm wide, 7.62 cm high, and 76 μm thick. These were mounted at equal azimuthal intervals in a Lucite cylinder with an inner diameter, D_{col}, of 19.05 cm, a wall thickness of 0.64 cm, and a height of 12.7 cm. Other electrode arrays were used during early experiments in order to evaluate the merits of different electrode geometries. Disk electrodes 1.27 cm in diameter and square electrodes 1.27 cm on a side, all 0.16 cm thick, were also fabricated from stainless steel. These were mounted at equal intervals in Lucite cylinders with the same inner diameter and wall thickness as the strip electrode cylinder but with a height of 38.1 cm, twice the inner diameter.

Each electrode geometry is known to have advantages and disadvantages. Strip electrodes can, in theory, produce relatively two-dimensional electric fields that are easier to model, provided that their length is two or more times the domain diameter. The strip electrodes do not meet this criterion, however, as is obvious from Figure 3.4. The point-like electrodes are able to produce current within a smaller domain in the axial direction than strip electrodes and are structurally more robust. However, more potential is required to inject a given amount of current through the smaller point-like electrodes, and voids in a multiphase system may momentarily cover a disk or square injection electrode completely and severely reduce current flow into the domain. The preliminary experiments performed to evaluate these different electrode geometries are described in Section 3.4.

Two data acquisition programs on the PC are used to control the measurement electronics and data acquisition card. These codes, `FASTEIT.BAS` and `SLOWEIT.BAS`, are listed in Appendices B and C, with subroutines common to both codes listed in Appendix D. Both programs are written in Microsoft® QuickBasic™ version 4.5 (Microsoft Corporation, 1991) and call QuickBasic subroutines supplied with the SP0131 Software Toolkit (Data Translation, Inc., 1994) to drive the DT2839 and manage PC extended memory. Both codes initialize the data acquisition board and EIT electronics, initiate current injection, manipulate the collected voltage data, analyze the data for irregularities, and save the data for use by reconstruction codes. As implied by their names, `FASTEIT` arms the IC counters to quickly control the multiplexers and rotate through electrode combinations, whereas `SLOWEIT` issues the commands to the multiplexers through the digital controller on the DT2839. `FASTEIT` also allocates enough dynamic buffers in PC extended memory to collect all voltage projection sets before data manipulation, while `SLOWEIT` manipulates the data after each projection set is collected.

As implied by Eq. 3.3, the number of independent impedance parameters in the domain model increases approximately as N^2, N being the number of electrodes, but the acquisition time of the Sandia/Michigan system increases approximately as N^3. An option was installed in the software and hardware to obtain voltage projection sets using either 8 or 16 electrodes. Although a limited number of tests were performed with rings of eight electrodes, the tests were not extensive enough to justify their use in evaluating the system. This option is still available, should further study of the tradeoff between resolution and acquisition time be warranted.

3.2.2. Finite-Element Algorithm for Reconstruction of Conductivity Distributions

This subsection summarizes the reconstruction algorithm used with the Sandia/Michigan EIT system. The algorithm is based on the YWT method described by Yorkey *et al.* (1987) (see also George, 1999a). The original Sandia algorithm has been documented by Torczynski *et al.* (1997) and is shown in flowchart form in Figure 3.5. The reader is referred to that report for complete details of the original algorithm, which computed a quadratic radial conductivity profile to represent bubble-column flows. In this subsection, the recent extension of the three-dimensional algorithm to quartic profiles is discussed.

In the algorithm, the medium is surrounded by an insulating boundary through which current is injected or withdrawn at discrete electrodes. A finite-element method (FEM) representation is generated for the voltage equation governing the domain, Eq. 3.1. Unlike most finite-element methods, the conductivity is not represented by a piecewise constant function that is discontinuous at element boundaries. Instead, the electrical conductivity is represented as a global function of position and of one or more conductivity parameters. To implement the

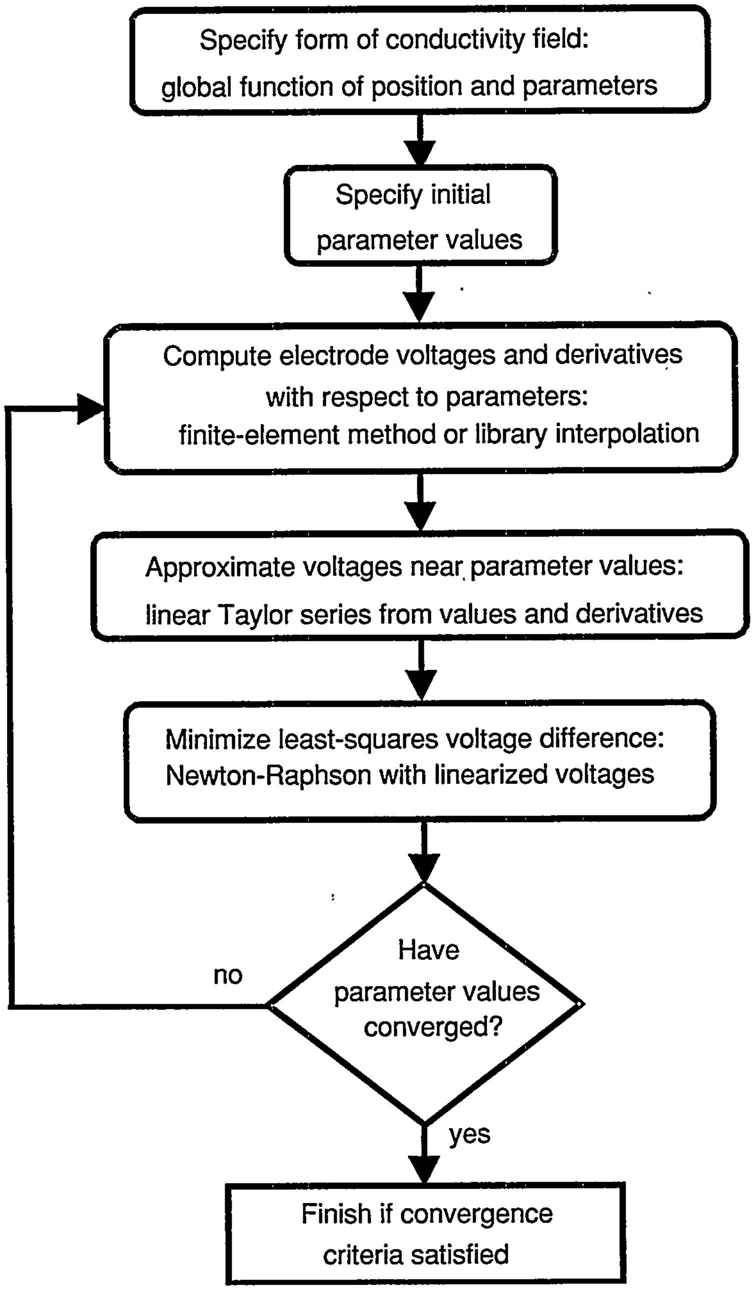

Specify form of conductivity field:
global function of position and parameters
Specify initial
parameter values
Compute electrode voltages and derivatives
with respect to parameters:
finite-element method or library interpolation
Approximate voltages near parameter values:
linear Taylor series from values and derivatives
Minimize least-squares voltage difference:
Newton-Raphson with linearized voltages
Have
parameter values
converged?
no
yes
Finish if convergence
criteria satisfied

Figure 3.5. Flow chart of EIT reconstruction algorithm.

Neumann boundary conditions (Eq. 3.2), current flow is specified at the electrodes such that the net current into and out of the domain is zero.

The finite-element equations generated from Eqs. 3.1 and 3.2 are solved to find (1) the predicted voltages at the electrodes as functions of the conductivity parameters and (2) the derivatives of the electrode voltages with respect to the conductivity parameters. Subsequently, these parameters are adjusted by a Newton-Raphson algorithm to minimize the least-squares difference between the computed and experimental electrode voltages at those electrodes that do not carry current. Since no voltages are prescribed in the computational boundary conditions, the computed voltage solution of Eq. 3.1 is unique only to within an arbitrary additive constant. The value of the additive constant is determined during the least-squares minimization process.

The algorithm is simplified by assuming that resistive effects dominate over capacitive effects in the medium. An appropriate criterion for this assumption can be obtained by separating the complex conductivity in Eq. 3.1 into its real and imaginary components (see, for example, Somersalo *et al.*, 1992):

$$\gamma = \sigma + i2\pi f\tilde{\omega}\omega_0 . \tag{3.4}$$

Here σ is the DC (or real) conductivity, $\tilde{\omega}$ is the normalized permittivity (or dielectric constant) of the medium, ω_0 is the permittivity of a vacuum (8.85×10^{-12} F/m), and f is the AC frequency. For resistance to dominate in the multiphase flows considered here, the following constraint on the liquid conductivity must be satisfied:

$$\sigma_L >> 2\pi f\tilde{\omega}_L\omega_0 . \tag{3.5}$$

For $f = 50$ kHz applied to water, for which $\tilde{\omega}_L \approx 80$, this constraint is satisfied if $\sigma_L >> 2$ µS/cm. This condition is often met by tap water and can also be attained by adding electrolytes to deionized water. In this work, controlled amounts of sodium chloride or sodium nitrate were added to deionized water to ensure that liquid conductivities were at least one hundred times this lower limit. In the remainder of this report, resistive effects are assumed to dominate over capacitive effects in all experiments and calculations, so that $\gamma \to \sigma$ in Eqs. 3.1 and 3.2.

Separate computer codes have been written to implement this algorithm in two and three dimensions. The two-dimensional code `FEMEIT`, listed in Appendix E, models arbitrary domains and represents electrodes by mesh nodes, essentially mathematical points of zero width. `FEMEIT` generates and solves the FEM equations using global conductivity functions from a subroutine library and applies the Newton-Raphson algorithm to determine the final conductivity parameters. Conductivity functions in the library include a constant conductivity, a circular insulating region at an arbitrary position, and various analytical conductivity distributions. As an example, the following formula is used for the spatial variation of conductivity, $\sigma(x,y)$, in the case of a circular insulator (Torczynski *et al.*, 1996a):

$$\sigma(x,y) = C_1\left\{1+\frac{K_1}{2}\left[\tanh\left(\frac{w-C_2}{K_2}\right)-\tanh\left(\frac{w+C_2}{K_2}\right)\right]\right\}, \tag{3.6}$$

where

$$w^2 = (x-C_3)^2+(y-C_4)^2 . \tag{3.7}$$

This function represents a cylindrical region of normalized radius C_2, centered at $(x,y)=(C_3,C_4)$, with a boundary thickness proportional to $2K_2$. Well inside the region, the conductivity is approximately $C_1(1-K_1)$, and well outside, the conductivity is approximately

C_1. When K_1 is slightly less than unity and K_2 is slightly less than the element size, this function represents an insulating cylinder of varying position and radius. This function was used to recreate "phantom" cylinders in validation tests described later in this chapter.

The three-dimensional code EITAXI (Appendix F) models cylindrical domains within which the conductivity varies only radially but the voltage fields vary three-dimensionally due to the finite axial extent of the electrodes. As applied to multiphase flows, this code reconstructs axisymmetric and vertically uniform phase distributions without the error inherent in the unrealistic assumption that the electric field lines are similarly limited to two-dimensionality. The electrodes around the cylindrical domain are taken to be identical and positioned at equal azimuthal intervals. Unlike FEMEIT, EITAXI represents each electrode with its actual shape rather than as a mathematical point. The conductivity field is represented as a global function of radial position and of conductivity parameters. Rather than solving the three-dimensional finite-element representation of the voltage field directly, as FEMEIT does in the two-dimensional case, EITAXI uses library files that contain the fundamental electrode voltages at discrete values of the conductivity parameters (Torczynski *et al.*, 1996a, 1997). These library files are computed in advance using the commercial finite-element code FIDAP (Fluid Dynamics International, 1996), as demonstrated later in this subsection. A cubic-spline interpolation is used to determine the fundamental electrode voltages for intermediate values of the conductivity parameters. To find the parameter values that best describe the experimental conductivity distribution, the least-squares difference between the experimental voltages and appropriate linear combinations of the computed voltages is determined. A Newton-Raphson algorithm is then used to determine the values of the conductivity parameters that minimize this least-squares difference.

As part of the axisymmetric reconstruction process, FIDAP generates a set of nondimensional, fundamental voltage solutions, $\{V_0 \ldots V_{N/2}\}$, for the parameter libraries used by EITAXI in evaluating experimental data (Torczynski *et al.*, 1997). The supplementary computer code EITFUN (described and listed in Appendix G) was written to compute fundamental voltage solutions from axisymmetric experimental datasets, the reverse of the usual process. Comparisons of the computational and experimental fundamental solutions for no-flow conditions allowed different electrode geometries to be evaluated, as described at the end of this chapter. Another code listed in Appendix H, EITFUL, uses fundamental voltage solutions to build axisymmetric datasets that can be used to validate the reconstruction algorithms.

The simplification of the conductivity field in EITAXI to allow only radial variations is justified by the fact that all experiments are conducted in cylindrical domains, and azimuthally invariant phase distributions are reasonably expected for the conditions studied in this work. Various conductivity distributions employed to date in EITAXI include a uniform conductivity profile; symmetric second-order (quadratic) and fourth-order (quartic) polynomials in the radius, r; and a cylindrical insulating inclusion centered in the domain. Originally a quadratic radial conductivity profile was used to represent bubble-column flows in all EIT reconstructions. However, results from the combination of EIT and gamma-densitometry tomography (GDT) in early three-phase experiments (George *et al.*, 1999a; see also Chapters 4 and 5) suggested that a quartic profile would be necessary. The lack of a quartic term in the EIT algorithm and the presence of one in the GDT algorithm were proposed as an explanation for slight on-axis maxima in reconstructed solid volume fraction profiles. To evaluate this proposal and to make

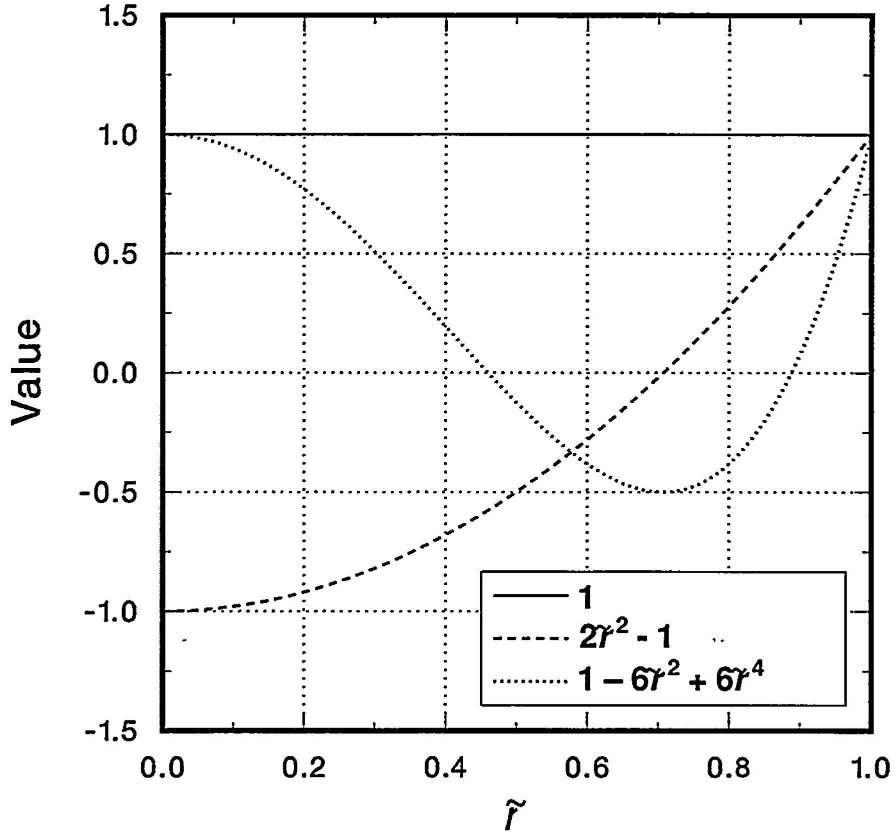


Figure 3.6. Basis functions used in EIT quartic algorithm.

EIT and GDT equally general, the EIT algorithm described by Torczynski *et al.* (1997) was extended to include a quartic term. The remainder of this subsection describes the quartic model and its development.

The following representation of a quartic radial conductivity profile was chosen:

$$\frac{\sigma(r)}{\sigma_L} = \frac{1 + C_1\left(2\tilde{r}^2 - 1\right) + C_2\left(1 - 6\tilde{r}^2 + 6\tilde{r}^4\right)}{C_0}, \tag{3.8}$$

where C_1 and C_2 are the conductivity shape parameters, C_0 is a scale parameter, $\tilde{r} = r / R_{col}$, r is the radial coordinate, and R_{col} is the radius of the circular domain – *e.g.*, the testbed or bubble column. The basis functions (the functions of $\tilde{r}$ in parentheses) were chosen so that the average of the product of any two different basis functions over the cross-sectional area of the domain vanishes,

$$\int_0^1 1\left(2\tilde{r}^2 - 1\right) 2\tilde{r}\,d\tilde{r} = \int_0^1 1\left(1 - 6\tilde{r}^2 + 6\tilde{r}^4\right) 2\tilde{r}\,d\tilde{r} = \int_0^1 \left(2\tilde{r}^2 - 1\right)\left(1 - 6\tilde{r}^2 + 6\tilde{r}^4\right) 2\tilde{r}\,d\tilde{r} = 0, \tag{3.9}$$

and so that the values of the basis functions at $\tilde{r} = 1$ are 1. Figure 3.6 shows a plot of these three basis functions.

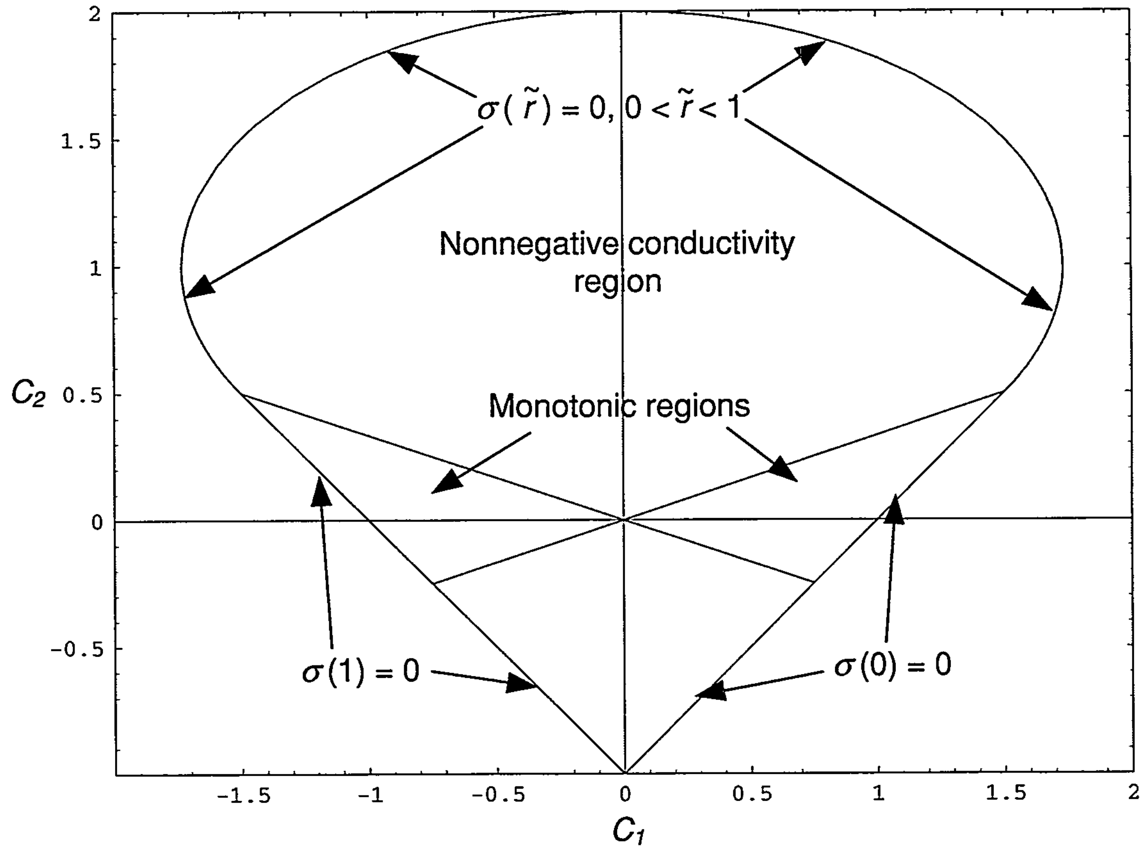


Figure 3.7. Region of parameter space in which conductivity is nonnegative at all radial locations.

The coefficients C_1 and C_2 that multiply the quadratic and quartic basis functions describe a two-dimensional parameter space. The condition that the conductivity must remain nonnegative for all radial locations constrains the coefficients to lie only within a portion of this parameter space. The boundary of the physically allowable portion of this two-dimensional space, within which the conductivity is nonnegative at all radial locations, is governed by the following equations, as shown in Figure 3.7:

$$|C_1| \le C_2 + 1 \text{ when } -1 \le C_2 \le \frac{1}{2}, \tag{3.10a}$$

$$|C_1| \le \left\{3\left[1-(1-C_2)^2\right]\right\}^{1/2} \text{ when } \frac{1}{2} \le C_2 \le 2. \tag{3.10b}$$

Monotonic conductivity profiles obey the following additional constraint, also shown in Figure 3.7.

$$|C_2| \le |C_1|/3 \tag{3.11}$$

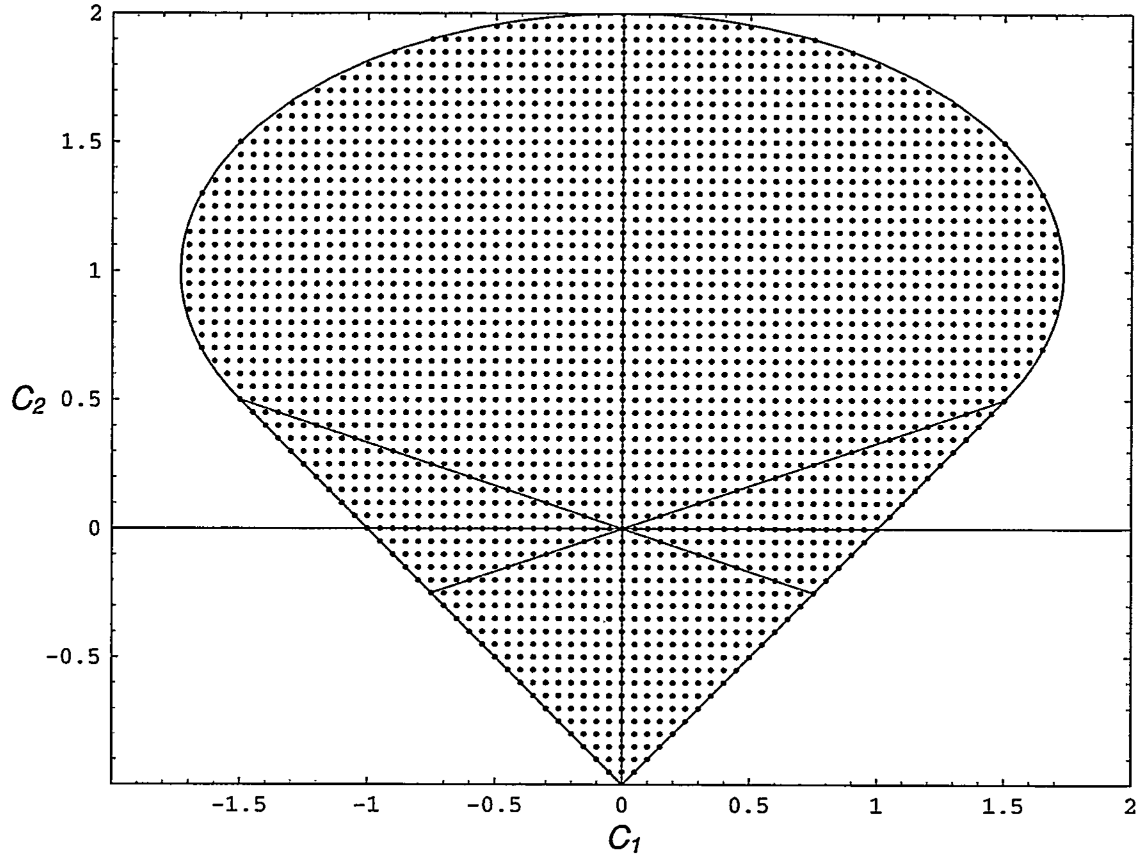


Figure 3.8. Discrete points in parameter space at which FIDAP computations were performed (2675 total).

The fundamental voltage solutions were computed for values of C_1 and C_2 within these constraints. Figure 3.8 shows the discrete points in the two-dimensional parameter space at which the fundamental voltage solutions were computed. A uniform spacing of 0.05 in both coordinate directions was employed, yielding 2675 points within the physically allowable region of parameter space. FIDAP was again used to produce the fundamental voltage solutions at the prescribed points. The FIDAP command language (Fluid Dynamics International, 1996) was employed to execute FIDAP computations in sequence and automatically loop over all the prescribed points in parameter space; approximately 80 hours on a Sun workstation was required to complete these calculations. The EIT electrode geometry used by FIDAP and `EITAXI` corresponds to the strip electrode ring shown in Figure 3.4. The mesh used in the computations, shown in Figure 3.9, has two symmetry planes and terminates at a height of two diameters. This mesh was previously found to produce acceptable accuracy (Torczynski *et al.*, 1998), although the accuracy degrades somewhat as the boundary of parameter space is approached – *i.e.*, when the conductivity vanishes at some radial coordinate within the computational domain.

With the fundamental voltage solutions computed for all points within the parameter space, the final step was to create smooth functions representing the solutions at all points within the physically meaningful region of parameter space, rather than just at the discrete points at which computations were made. Smooth functions are needed because the Newton-Raphson

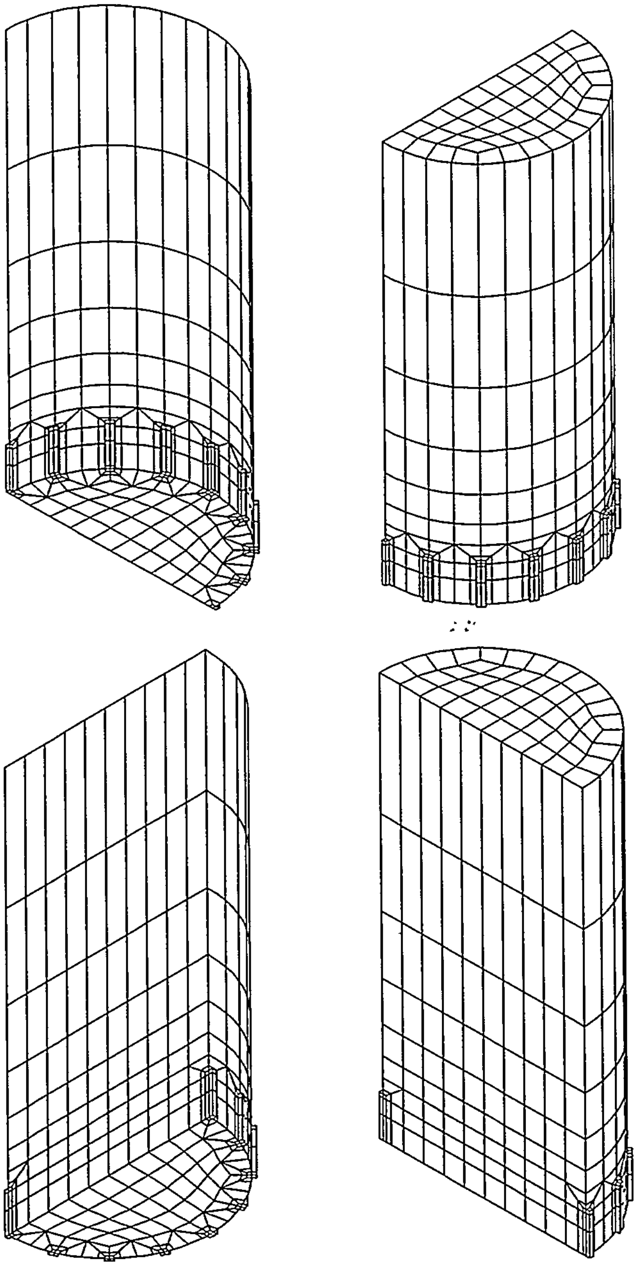


Figure 3.9. FIDAP mesh on which fundamental voltage solutions were computed.

method described earlier requires continuous, well-behaved first derivatives of the fundamental voltage solutions with respect to the conductivity parameters. Because the conductivity vanishes at some location within the physical domain for parameter values lying on the parameter-space boundary, the fundamental voltage solutions experience rapid changes as this boundary is approached. The point $(C_1, C_2) = (-1.5, 0.5)$ is a particularly strong singularity, since both the conductivity and its radial derivative $\partial\sigma/\partial r$ vanish at $\tilde{r} = 1$ for these particular values. The large voltage values encountered near the boundary and the rapid variation in voltage values as this boundary is approached render polynomial fits inaccurate; however, polynomial fits to the reciprocals of the values of the fundamental voltage solutions are well behaved as the boundary is approached.

The symbol manipulation code Mathematica (Wolfram, 1996) was used to fit polynomials of degree 24 in both C_1 and C_2 (containing a total of 625 fitting coefficients *k*) to the reciprocal nontrivial voltages of the fundamental solutions, with derivatives determined by the chain rule:

$$\frac{1}{V_n} \to \sum_{i=0}^{24} \sum_{j=0}^{24} k_{nij} C_1^i C_2^j, \; n = 1\mathrm{K}\ N/2. \tag{3.12}$$

Note that by definition the ground electrode voltage $V_0 = 0$ and is not included in the fits. Of the 2675 points in (C_1, C_2) parameter space, 2670 are used in the fits: the other five points are too close to the parameter-space boundary to be computed accurately on the mesh shown in Figure 3.9 and corrupt the fit if included. For comparison, the computed values and the polynomial fit for voltage V_7 along the curves $C_1 = 0$ and $C_2 = 0$ are shown in Figure 3.10, and are seen to agree closely.

3.2.3. Determination of Volume Fraction Distributions from Conductivity Distributions

Once the converged parameters from `FEMEIT` or `EITAXI` have been used to construct a conductivity distribution, a constitutive model is required to relate the electrical conductivity to the conducting and insulating phase volume fractions. When the continuous liquid phase is the only conducting phase, the Maxwell-Hewitt relation (Maxwell, 1881; Hewitt, 1978) can be used to relate $\sigma(r)$, the local mixture conductivity, and σ_L, the known liquid conductivity, to the local liquid volume fraction $\varepsilon_L(r)$:

$$1-\varepsilon_L(r) = \frac{1-[\sigma(r)/\sigma_L]}{1+\frac{1}{2}[\sigma(r)/\sigma_L]} \text{ (three-dimensional inclusions).} \tag{3.13}$$

The expression above assumes three-dimensional inclusions of an insulating phase dispersed in the conducting phase, such as gas bubbles or insulating solid particles dispersed in salt water, and is applied locally in all tomographic reconstructions in this report. An analogous relation can be derived for the somewhat artificial case of two-dimensional insulating inclusions dispersed in a conducting phase (*e.g.*, parallel insulating posts in salt water):

$$1-\varepsilon_L(r) = \frac{1-[\sigma(r)/\sigma_L]}{1+[\sigma(r)/\sigma_L]} \text{ (two-dimensional inclusions).} \tag{3.14}$$

Equation 3.14 is not used in any of the tomographic reconstructions in this study, but is used to assess the amount of uncertainty introduced into the reconstructions by the Maxwell-Hewitt constitutive model.

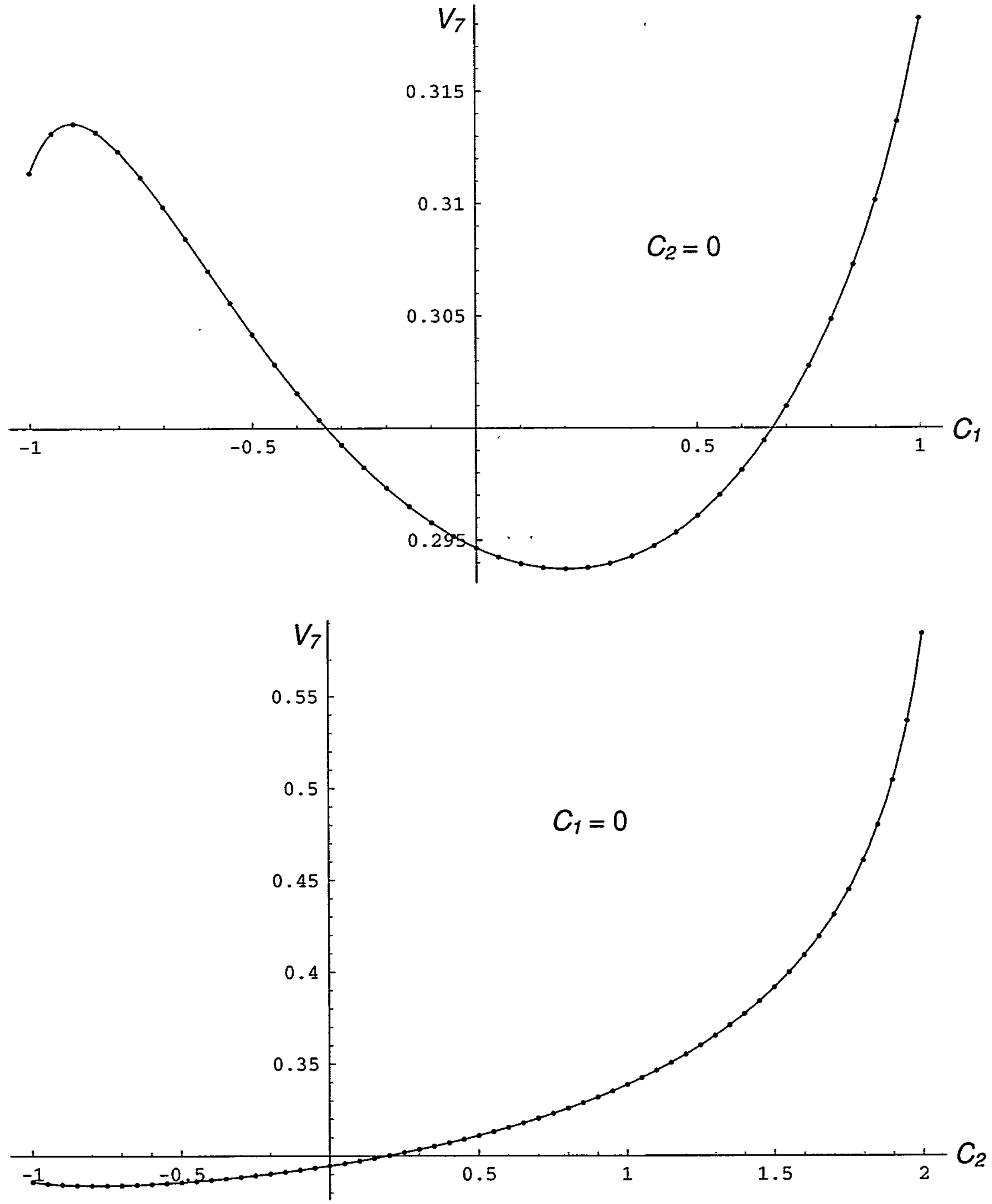


Figure 3.10. Plots of fundamental voltage V_7 along selected curves. Points are computed with FIDAP, curves are best fits.

By continuity, the quantity $1-\varepsilon_L(r)$ equals the local insulating phase volume fraction in either geometry. To determine the domain-averaged phase volume fractions, the profiles can be analytically integrated over the domain. The values of σ_L used to obtain phase volume fractions in all experiments were determined with the EIT system itself, using measurements of the domain with only liquid present. A good measurement of liquid conductivity is crucial for obtaining quantitative data and requires careful control of test conditions, since conductivity can vary significantly with temperature and the presence of contaminants in the liquid.

3.3. Validation of Reconstruction Algorithms

The algorithms described above for reconstructing conductivity profiles have been validated using both analytically and physically derived projection sets. The physical data were obtained during tests of the complete EIT system with the detection of a series of insulating inclusions in the test domain. The constitutive relationship used to translate conductivity profiles into phase volume fraction profiles was also investigated for accuracy, and the sensitivity of reconstructions to noise in the measured voltages was determined. This section describes these tests and presents the results.

3.3.1. Numerical Validation of Conductivity Reconstruction Algorithm

Several calculations were performed to validate the conductivity reconstruction algorithm. The two tests described in this section were first documented in Torczynski *et al.* (1997) but are repeated here for completeness. The first test involved an analytical result for the two-dimensional voltage distribution $V(x, y)$ in a circular domain of constant conductivity σ. The boundary conditions were defined by a current per unit electrode length, J, injected at the boundary point (x_o, y_o) and withdrawn at the boundary point $(x_o, -y_o)$; this yielded the analytical result

$$V(x,y) = (J/2\pi\sigma)\ln\left[\frac{(y_o+y)^2+(x_o-x)^2}{(y_o-y)^2+(x_o-x)^2}\right]. \tag{3.15}$$

The uniform domain was bounded by 16 electrodes located at equal intervals of 22.5º around the perimeter. The radius of the circular domain, the current per unit length and the constant conductivity were all set to unity.

By rotating the coordinates of the analytical results derived from selected injection and withdrawal combinations, the boundary voltages for all possible injection and withdrawal pairs were determined from the analytical solutions. These analytical voltages were input to FEMEIT, along with mesh information, to find the effect of mesh size on reconstruction accuracy. The conductivity function was chosen to be a single unknown constant throughout the domain. Table 3.1 shows the dependence of the constant conductivity determined by FEMEIT for five meshes of appreciably different nodal density. An ideal result would be a calculated conductivity of unity. In all cases the calculated result is close to unity, and is converging to unity with increasing nodal density (Torczynski *et al.*, 1996a).

Table 3.1. Effect of FEMEIT mesh refinement on reconstruction accuracy of a uniform domain. The correct analytical solution is $\sigma = 1$.

Mesh	Number of nodes	Number of elements	Computed σ
A	25	32	1.0443
B	81	128	1.0122
C	169	288	1.0047
D	289	512	1.0023
E	441	800	1.0013

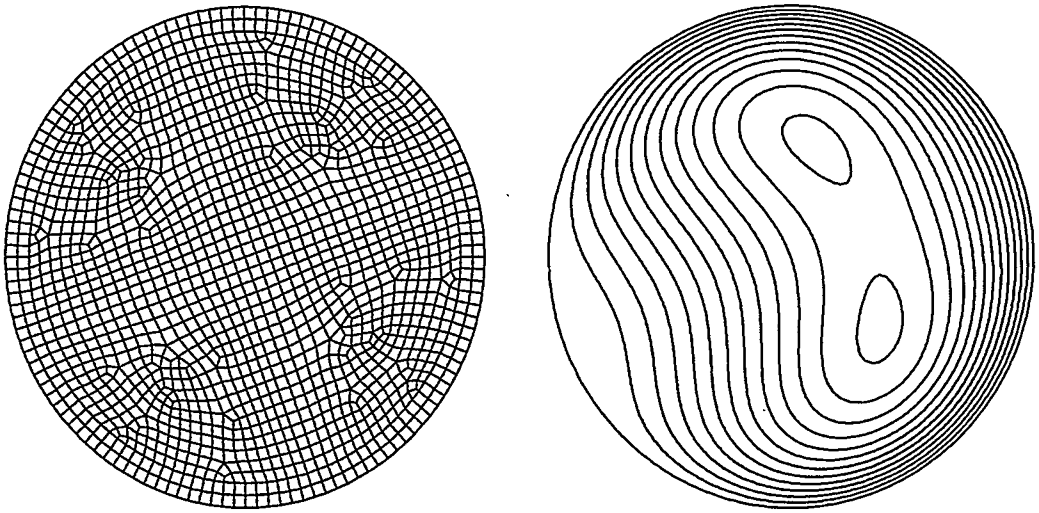


Figure 3.11. Finite-element mesh used by FIDAP in validation tests (left) and synthetic benchmark conductivity field (right). Adapted from Torczynski *et al.* (1996a).

In the second set of validation calculations, the finite-element code FIDAP was used to compute the boundary voltages corresponding to the synthetic spatial conductivity distribution shown in Figure 3.11. This distribution mimics a bubble-column flow with an excess of bubbles in the upper right quadrant. The electrode geometry was identical to that used in the first problem, but a highly refined mesh was used to guarantee that the FIDAP solution would be mesh-independent. The computed voltages from FIDAP were input to FEMEIT to test its ability to reproduce the original conductivity distribution, and reconstructions were performed using the same five meshes used in the first calculation set (Table 3.1).

Figure 3.12 shows the most refined mesh used for this simulation (mesh E of Table 3.1) and the reconstructed conductivity field produced by the EIT algorithm for this mesh. Agreement is seen to be good. To quantify the effect of mesh refinement on reconstruction accuracy, average and rms conductivity values were computed for the FEMEIT results on each mesh and are listed in Table 3.2. As the mesh is refined, the average conductivity approaches the analytical average value of 2/3, and the rms value tends to zero.

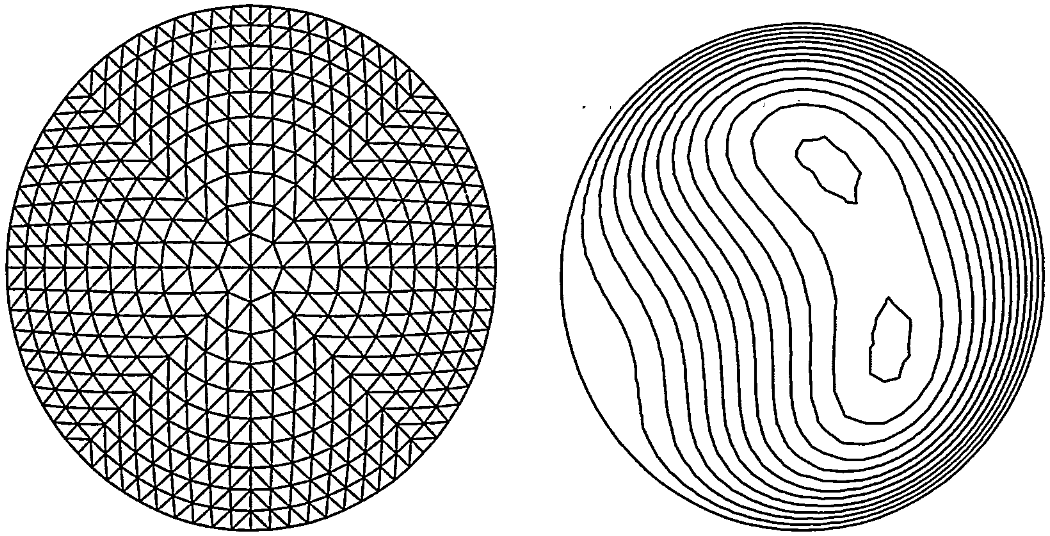


Figure 3.12. Finite-element mesh used by FEMEIT in validation tests (left) and reconstructed conductivity field (right). Adapted from Torczynski *et al.* (1996a).

Table 3.2. Effect of `FEMEIT` mesh refinement on reconstruction accuracy of a synthetic conductivity distribution. The analytical average of the synthetic distribution is $\sigma = 2/3$.

Mesh	Computed σ	rms deviation in σ
A	0.679	0.170
B	0.676	0.027
C	0.669	0.007
D	0.666	0.003
E	0.666	0.003

3.3.2. Experimental Validation of Reconstructed Conductivity Profiles

The reconstruction algorithm was also validated using experimental data from both the strip and point-like electrode arrays. Validation experiments with the strip electrodes were performed by capping the bottom of the cylinder containing the electrode array (Figure 3.4) to form a static testbed, filling the testbed with a saltwater solution of known conductivity, and placing smaller Lucite cylinders of known diameter in the testbed to act as insulating inclusions. EIT voltage measurements were then taken of the domain and used to reconstruct the location and position of the insulating "phantom." This type of test has been studied extensively by other investigators of EIT (*e.g.*, Duraiswami *et al.*, 1997), and was performed with the original configuration of the Sandia/Michigan EIT system as well (Torczynski *et al.*, 1997).

To limit the test domain in the vertical direction, the free surface of the liquid and the bottom Lexan end cap were aligned with the top and bottom ends of the strip electrodes, respectively. This eliminated axial electric field variations and enforced two-dimensionality.

Table 3.3. Comparison of actual and reconstructed geometries of insulating inclusions in two-dimensional validation tests with strip electrodes.

Actual conditions			PC electrode selection (SLOWEIT)			IC electrode selection (FASTEIT)		
$D_{cyl}/2R_{col}$	r/R_{col}	$\bar{\varepsilon}_{cyl}$	$D_{cyl}/2R_{col}$	r/R_{col}	$\bar{\varepsilon}_{cyl}$	$D_{cyl}/2R_{col}$	r/R_{col}	$\bar{\varepsilon}_{cyl}$
0	–	0	0.062	0.000	0.004	0.059	0.000	0.004
0.266	0.596	0.071	0.271	0.597	0.074	0.271	0.597	0.074
0.300	0.000	0.090	0.309	0.020	0.095	0.308	0.020	0.095
0.534	0.189	0.285	0.530	0.184	0.281	0.530	0.184	0.280

Cylinders of different diameters were placed both eccentrically and concentrically with the vertical axis of the testbed to mimic different distributions of gas in a bubble column. Data were acquired using both the PC software (SLOWEIT) and IC counters (FASTEIT) to select electrode combinations.

The conducting medium was deionized water with sodium chloride solution added to yield a conductivity of $\sigma_L = 420 \pm 5$ µS/cm for all strip electrode validation cases. Experience with the EIT system has shown that the conductivity of the continuous phase must be chosen carefully for each geometry and multiphase system to obtain good resolution in the voltage measurements. Because the EIT system employs a constant injection current, a conductivity that is too high will lower the voltages across the domain to the point of poor resolution, while a conductivity that is too low will produce voltages beyond the input range of the data acquisition board. The conductivities used in these and all other tests were selected to provide a large dynamic range of measured voltages without exceeding the input limit of the DT2839.

Equations 3.6 and 3.7 were employed as the spatial conductivity function in the FEMEIT reconstructions. Values of $K_1 = 0.99$ and $K_2 = 0.03$ were used, and the resulting reconstructions were found to be insensitive to modest changes from these values. Table 3.3 presents, for each test case, the actual diameter D_{cyl} of the test object and the radial offset distance r of the object center from the testbed central axis, both normalized by the domain radius, R_{col}. The table also presents the normalized computed diameter, $D_{cyl}/2R_{col} = C_2$, and the offset distance, $r/R_{col} = \sqrt{C_3^2 + C_4^2}$, determined from the EIT data. A domain-averaged inclusion volume fraction, $\bar{\varepsilon}_{cyl} = (D_{cyl}/2R_{col})^2$, was also computed from the actual and reconstructed object dimensions. Figure 3.13 shows the reconstructed conductivity distributions for two of the four test cases; the light areas indicate the object boundaries, numerical transition regions from high conductivity outside the object to zero conductivity within the object.

The reconstructions are in good agreement with actual conditions, reproducing the volume fraction to within 0.005 in $\bar{\varepsilon}_{cyl}$ and the distance r of the inclusion from the central axis to within 3%. For the smaller inclusions the diameter of the object (and hence the average volume fraction) was always slightly overpredicted; in the case of the largest test object, the volume fraction was underpredicted. The source of the disagreement is believed to be differences

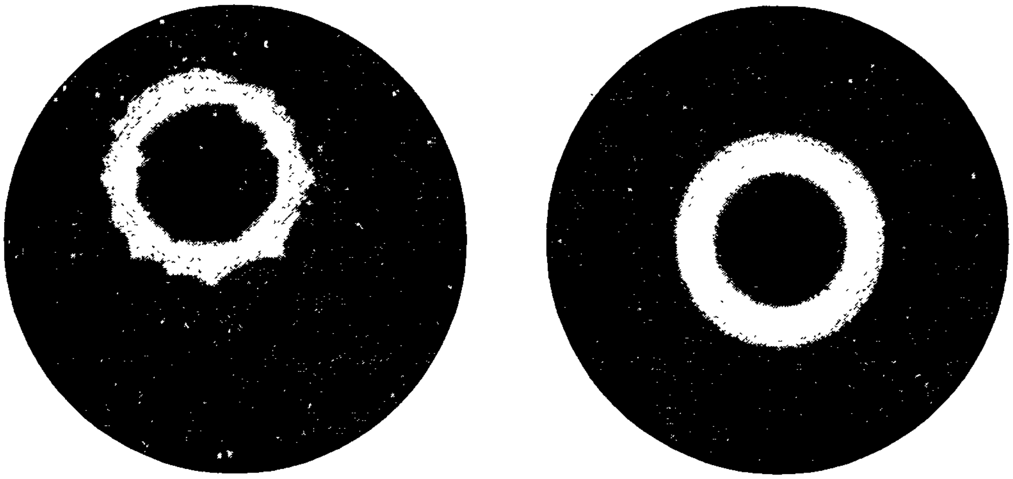


Figure 3.13. Conductivity reconstructions for two-dimensional validation tests with inclusions. (Left) D_{cyl} = 5.07 cm, r = 5.68 cm; (right) D_{cyl} = 5.72 cm, r = 0.

Table 3.4. Fundamental voltage solutions for 16 strip electrodes in the transparent bubble column.

Fundamental voltage	Computational value	Experimental value
V_0 (reference)	0	0
V_1	0.002275	0.002295
V_2	0.009430	0.009481
V_3	0.02260	0.02252
V_4	0.04432	0.04392
V_5	0.08030	0.07963
V_6	0.1454	0.1443
V_7	0.2947	0.2930
V_8 (current-bearing)	1.327	1.232

between the measured and reconstructed voltages along the domain boundary. An example of these differences can be found by comparing the computed and experimental fundamental voltages $\{ V_1 \ldots V_7 \}$ in Table 3.4. While the computed voltages match the corrected experimental data to within 1% at each non-current-bearing electrode, changes in the sign and magnitude of the difference indicate slightly different "shapes" of the computed and experimental voltage profiles along the boundary. This would produce slight differences between the actual and reconstructed conductivity profiles.

Notably, the reconstruction algorithm produced a small object when none was present. This indicates that objects of diameters $D_{cyl}/2R_{col} \leq 0.06$ cannot be reconstructed and that the uncertainty in $\bar{\varepsilon}_{cyl}$ is on the order of 0.004. This uncertainty is attributed to slight differences

Table 3.5. Comparison of actual and reconstructed geometries of insulating inclusions in three-dimensional validation tests with disk electrodes.

Actual conditions		PC electrode selection (SLOWEIT)		IC electrode selection (FASTEIT)	
$D_{cyl}/2R_{col}$	$\bar{\varepsilon}_{cyl}$	$D_{cyl}/2R_{col}$	$\bar{\varepsilon}_{cyl}$	$D_{cyl}/2R_{col}$	$\bar{\varepsilon}_{cyl}$
0	0	0.072	0.005	0.028	0.001
0.266	0.071	0.277	0.077	0.270	0.073
0.530	0.281	0.530	0.281	0.530	0.281

between the computational and experimental boundary voltages for a uniform conductivity field, similar to those for nonuniform fields discussed above.

Validation tests were also conducted with the disk and square electrodes (George *et al.*, 1998b). In these tests, a cylindrical testbed was created by placing the electrode cylinder between two more Lucite sections of the same diameter and wall thickness but 19.05 cm high. This created a test vessel with a height-to-diameter ratio of 4, tall enough to encompass the significant electric field lines. Test cases were created with tall Lucite cylindrical "phantoms" which extended along the entire height of the testbed. Because the three-dimensional reconstruction code EITAXI does not contain a library conductivity function for an eccentrically placed inclusion, all "phantoms" were centered on the vertical axis of the testbed. The conductivity in these tests was σ_L = 604 ± 2 μS/cm, higher than the value in the strip electrode tests because of the smaller electrode area available for current injection.

Data were again taken using both the PC software and IC counters to select electrode combinations. Table 3.5 presents, for each test, the normalized diameter of the test object, the normalized diameter determined from the voltage data, and the domain-averaged volume fraction computed from the reconstructed diameter. The reconstructions in the table are again in good agreement with actual conditions, reproducing the volume fraction to within 0.006 in $\bar{\varepsilon}_{cyl}$. The same overprediction of smaller inclusions is seen here as was observed with the strip electrodes, but the larger inclusion is reproduced correctly.

3.3.3. Validation of Maxwell-Hewitt Relations

To assess the accuracy of the Maxwell-Hewitt relations (Eqs. 3.13 and 3.14), the computational fluid dynamics code FIDAP was used to perform simulations of the voltage field in various multiply periodic geometries. Two-dimensional square lattices of insulating squares and circles and three-dimensional simple-cubic lattices of cubes and spheres were examined; the region outside the insulating inclusions was filled with a uniformly conducting medium. Because of the symmetry and periodicity of these geometries, the computational domain consisted of only one quadrant of a two-dimensional unit cell or one octant of a three-dimensional unit cell. In each case the computational domain had a side length of unity and a conductivity of unity outside the insulating inclusion. Voltages of zero and unity were applied on two opposing sides of the domain, and insulating conditions were applied on the remaining

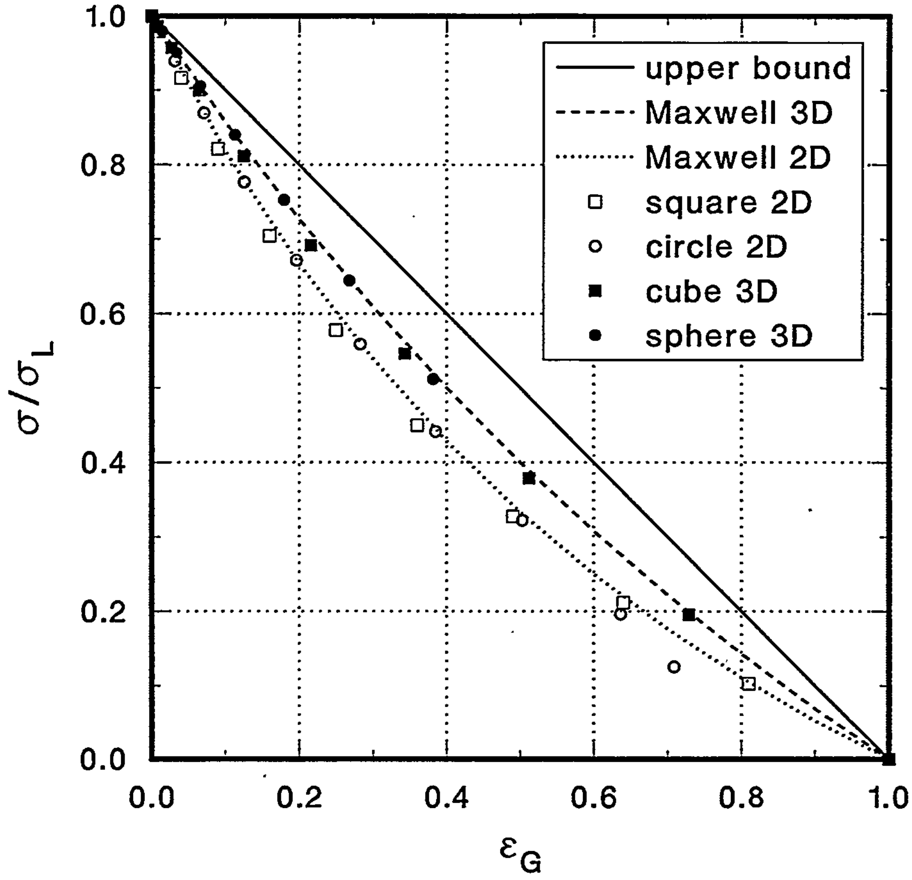


Figure 3.14. Comparison of effective conductivities computed by FIDAP with predictions from the Maxwell-Hewitt relations.

sides by virtue of symmetry. In the absence of an insulating inclusion – *i.e.*, when the domain is simply a conducting square or cube – these conditions lead to a current of unity passing through the domain along one of its principal directions. When an inclusion is present, the current is reduced from this value by the ratio of the effective conductivity to the actual conductivity, $\bar{\sigma}/\sigma_L$ (per Ohm's Law).

Figure 3.14 compares these effective conductivity ratios with the values predicted by the Maxwell-Hewitt relations for two-dimensional and three-dimensional insulating inclusions. The symbols represent FIDAP results, which are the exact solutions for converting conductivity measurements to insulating volume fractions. The dashed lines are values predicted by Eqs. 3.13 and 3.14. The upper bound, determined by the limiting case where current travels through a single layer of liquid and a single layer of gas arranged in "parallel," is plotted as the solid line. It can be seen that using the Maxwell-Hewitt relations for insulating objects of these shapes will cause the insulating phase volume fraction to be only slightly overestimated. The results for circles and spheres closely follow the Maxwell-Hewitt relations until the inclusions touch. It is concluded that these relations are of uniformly good accuracy so long as inclusions are not highly distorted and can "fill space without overlapping."

The Maxwell-Hewitt relation has been shown to be reasonably accurate for monodisperse bubble-size distributions. However, in churn-turbulent bubble-column flow there may be a bimodal distribution of bubble sizes, with many small bubbles and some large bubbles. It is important to quantify the errors that result from applying the monodisperse Maxwell-Hewitt relation to this situation. To this end, a bimodal distribution of bubbles is examined for which the ratio of bubble diameters is large compared to unity. The Maxwell-Hewitt relation is used recursively in this analysis to determine the effective conductivity of this medium.

Consider a mixture of a liquid of conductivity σ_L and an insulating gas. The gas has a total volume fraction of ε_G, and is distributed in the form of "large" bubbles that have a volume fraction of $f_{\text{large}}\varepsilon_G$ and "small" bubbles that have a volume fraction of $(1-f_{\text{large}})\varepsilon_G$. The large-bubble fraction, f_{large}, obeys the constraint $0 \le f_{\text{large}} \le 1$. The Maxwell-Hewitt relation can be applied recursively to determine the effective conductivity of this medium. In a model similar to that of Krishna and Ellenberger (1996) in Section 2.3.1, the liquid and the small bubbles form a "dense phase" mixture with an effective conductivity σ_{df} given by

$$\frac{\sigma_{df}}{\sigma_L} = \frac{1-\varepsilon_{df}}{1+\frac{1}{2}\varepsilon_{df}}, \tag{3.16}$$

where (compare to Eq. 2.26)

$$\varepsilon_{df} = \frac{(1-f_{\text{large}})\varepsilon_G}{1-f_{\text{large}}\varepsilon_G .} = \frac{\varepsilon_{G,small}}{1-\varepsilon_{G,large}}. \tag{3.17}$$

The large bubbles and the "dense phase" form another mixture, denoted *B* (bimodal), with an effective conductivity σ_B given by

$$\frac{\sigma_B}{\sigma_{df}} = \frac{1-\varepsilon_{G,large}}{1+\frac{1}{2}\varepsilon_{G,large}}, \tag{3.18}$$

where

$$\varepsilon_{G,large} = f_{\text{large}}\varepsilon_G . \tag{3.19}$$

These relations can be combined to determine the effective conductivity of the bimodal bubble mixture in terms of the total gas volume fraction and the fraction f_{large} of the gas in large bubbles, the bimodal Maxwell-Hewitt relation:

$$\frac{\sigma_B}{\sigma_L} = \left[\frac{1-f_{\text{large}}\varepsilon_G}{1+\frac{1}{2}f_{\text{large}}\varepsilon_G}\right]\left[\frac{1-\varepsilon_G}{1-\frac{3}{2}f_{\text{large}}\varepsilon_G+\frac{1}{2}\varepsilon_G}\right]. \tag{3.20}$$

When $f_{\text{large}} = 0$ or $f_{\text{large}} = 1$, this expression reduces to the monodisperse Maxwell-Hewitt relation, essentially Eq. 3.13:

$$\frac{\sigma}{\sigma_L} = \frac{1-\varepsilon_G}{1+\frac{1}{2}\varepsilon_G}. \tag{3.21}$$

For any value of f_{large} between 0 and 1, the bimodal curve lies slightly beneath the monodisperse curve. Thus, the assumption of a monodisperse bubble-size distribution in the presence of a bimodal bubble-size distribution would cause the gas volume fraction to be slightly overpredicted. The large-bubble fraction at which the maximum difference between the monodisperse and bimodal Maxwell-Hewitt relations occurs for a given total gas volume fraction is then

$$f_{\max} = \frac{1-\sqrt{1-\varepsilon_G}}{\varepsilon_G}. \tag{3.22}$$

Evaluating the bimodal Maxwell-Hewitt relation at $f_{\text{large}} = f_{\max}$ and comparing the resulting relation with the monodisperse Maxwell-Hewitt relation enables determination of the maximum possible error in gas volume fraction, $(\Delta\varepsilon_G)_{\max}$, as a function of true gas volume fraction, ε_G. The error $(\Delta\varepsilon_G)_{\max}$ never exceeds 0.028, the value at $\varepsilon_G = 0.723$, and is less than 0.01 for $\varepsilon_G \leq 0.3$. Since this degree of uncertainty is comparable to the uncertainty of the monodisperse Maxwell-Hewitt relation itself, it appears that bubble-size distribution does not significantly affect the accuracy of the Maxwell-Hewitt relation.

3.3.4. Other Uncertainty Considerations

It is important to assess the degree to which uncertainties in experimental voltage measurements translate into uncertainties in the material distribution profiles reconstructed by EIT. Table 3.6 shows the fundamental voltage solution from the library file used by EITAXI that corresponds to conductivity parameter values of $C_0 = 1$, $C_1 = 0$, and $C_2 = 0$. Also shown are fundamental voltage solutions into which small errors have been deliberately introduced. Specifically, an error of 0.0001 (smaller than uncertainties seen from experimental projection sets with 25 measurements averaged per set) is added to the indicated voltage. These erroneous fundamental voltage solutions are used to create pseudo-experimental voltage projection sets that are subsequently analyzed by EITAXI.

Table 3.7 shows the values of the conductivity parameters that result from introducing these errors. Several observations can be made about these results. First, in the "no errors" case, EITAXI returns values of the conductivity parameters that are slightly different from the values used to create the projection set. The differences are acceptably small, below 10^{-3}, and are caused principally by the precision to which the voltages are written in the experimental voltage data file. Increasing this "experimental" precision reduces the differences substantially, although

Table 3.6. Fundamental voltage solutions used in uncertainty analysis. Deliberate errors are underlined.

Voltage	No errors	Case 1	Case 2	Case 3
V_0	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00
V_1	0.227477E-02	0.227477E-02	0.227477E-02	0.227477E-02
V_2	0.943045E-02	0.943045E-02	0.943045E-02	0.9$\underline{5}$3045E-02
V_3	0.225976E-01	0.225976E-01	0.225976E-01	0.225976E-01
V_4	0.443181E-01	0.443181E-01	0.44$\underline{4}$181E-01	0.443181E-01
V_5	0.802967E-01	0.802967E-01	0.802967E-01	0.802967E-01
V_6	0.145450E+00	0.145$\underline{5}$50E+00	0.145450E+00	0.145450E+00
V_7	0.294655E+00	0.294655E+00	0.294655E+00	0.294655E+00
V_8	0.132742E+01	0.132742E+01	0.132742E+01	0.132742E+01

Table 3.7. Conductivity parameter values resulting from voltages in Table 3.6.

Coefficient	No errors	Case 1	Case 2	Case 3
C_0	0.10001E+01	0.99885E+00	0.10011E+01	0.99976E+00
C_1	-.26065E-03	0.70206E-02	-.38374E-02	-.32062E-03
C_2	-.79597E-03	0.16049E-01	-.14456E-01	0.15128E-02

some differences attributed to finite numerical precision still remain. Second, the errors in the conductivity parameter values in all cases are smallest for C_0, intermediate for C_1, and largest for C_2. Third, the errors decrease as the index of the voltage in which the error is introduced is decreased. These last two observations are in harmony with the concepts that the largest of the fundamental voltages are the most important in the reconstruction, and that the coefficients of the higher-degree polynomials require more of the larger voltages to be measured accurately. Fourth, the errors in case 3 are acceptably small, but the errors in cases 1 and 2 (particularly in C_2) are not. This suggests that experimental datasets should average more than 25 measurements for each projection set, the number used in early tests and experiments with the EIT system.

3.4. Selection of Electrode Geometry for Multiphase Flow Measurements

The final evaluation of strip and point electrode geometries used measurements of baseline liquid conductivities taken with each electrode array in the vertical bubble column described in the next chapter. These measurements were taken during preliminary tests of vertical gas-liquid flows in the column, in periods of no gas flow. Fundamental voltage solutions were derived from these no-flow data with EITFUN and compared to calculated solutions from FIDAP for the appropriate electrode geometry with a uniform conductivity field in three dimensions. The comparisons gave a clear indication of the best choice for multiphase flow measurements.

The fundamental voltages from the square and circular electrode data were in poor agreement with calculated values; the experimental voltages were similar at all non-current-bearing electrodes. Besides implying a potential for significant disagreement between actual and calculated conductivity profiles, this similarity indicated that under multiphase flow conditions, the electrodes would have poor sensitivity to variations in the conductivity profile. In contrast, the fundamental voltage solutions from strip electrode data were in excellent agreement with the computational values at the non-current-bearing electrodes 0 through 7 (see Table 3.4). Although the voltages at the current-bearing electrode (V_8) differed somewhat from the fundamental solution, probably because of contact impedance, the reconstruction algorithm does not use this quantity (Torczynski *et al.*, 1996a and 1997). The voltages at non-current-bearing strip electrodes also varied significantly from one electrode to the next, implying a higher sensitivity to changes in the domain conductivity profile. Several possible explanations for the higher performance of the strip electrodes have been advanced, including their larger area compared to the square and circular electrodes (4.84 cm^2 versus 1.61 cm^2 and 1.27 cm^2, respectively); the fact that the strip electrodes more closely resemble "two-dimensional" electrodes; and the smaller fraction of the column circumference subtended by the strips compared to the squares and disks.

As seen in Table 3.4, multiplying the strip electrode experimental values by 1.007 brings them into agreement with computed voltages to almost three significant figures. A systematic difference of 0.7% between computational and experimental voltages is reasonable, since experimental uncertainty in measurements of injected current and liquid conductivity is about ±0.5% and computational accuracy is similar. Based on these results, strip electrodes were used exclusively to reconstruct the radial profiles of the multiphase flows discussed in the remainder of this report.

3.5. Summary

In this chapter the Sandia/Michigan EIT hardware and reconstruction algorithms have been presented and evaluated. The electronics used to inject current into the domain and measure voltages at the boundary have been described in detail, including improvements to reduce acquisition time. An algorithm has been implemented in two and three dimensions to reconstruct distributions of insulating phases within a continuous conducting phase, such as gas bubbles or insulating solid particles in conducting liquid. Both the two-dimensional and three-dimensional codes are capable of reconstructing uniform and radially varying conductivity distributions that are useful in the analysis of vertical bubble-column flows. The three-dimensional algorithm has recently been expanded to allow quartic radial gas volume fraction profiles, so that EIT may be used with the same generality as the gamma-densitometry tomography method employed at Sandia.

As a validation test of the reconstruction algorithm alone, boundary voltages were computed from analytical conductivity distributions and input to the algorithm; the original conductivity distributions were accurately reproduced by the reconstruction code. Next, using a method common to many EIT studies, known geometries of insulating inclusions in a conducting domain were measured with the Sandia/Michigan EIT system and reproduced with the reconstruction codes and measured voltage data. The combination of hardware and software reproduced the volume fraction of each inclusion to within 0.006 and its position to within 3%, demonstrating the ability of the system to quantitatively measure static insulating phase distributions. An assessment of the uncertainty in the quartic reconstruction algorithm indicated that the use of higher-order coefficients requires: (1) high accuracy in the largest experimental voltages near the injection electrode, and (2) datasets that average more than the 25 measurements typically used in early EIT tests. Finally, a commercial finite-element code was used to investigate the accuracy of the Maxwell-Hewitt equations used to convert conductivity distributions to phase volume fraction distributions. It was concluded that these equations will accurately determine insulating phase volume fractions from measured conductivity data, with a small but acceptable tendency to overpredict the amount of insulating phase. With these successful validation tests, the EIT system and the Maxwell-Hewitt equations can now be applied to the measurement of insulating volume fraction distributions in vertical multiphase bubble-column flows.

4. Evaluation of the Sandia/Michigan EIT System in Two-Phase Flows

In this chapter, a series of studies is presented in which the EIT system described in Chapter 3 was applied to solid-liquid and gas-liquid two-phase flows. The Sandia/Michigan EIT system and an established gamma-densitometry tomography (GDT) system were simultaneously applied to the same flows, and the results were compared to judge the accuracy of the EIT system for two-phase flow measurements (Shollenberger *et al.*, 1997b; George *et al.*, 1998a, 1999c). The GDT system developed at Sandia has already been successfully applied to multiphase flows (Adkins *et al.*, 1996; Torczynski *et al.*, 1996b; Shollenberger *et al.*, 1995, 1997a), making it a useful validation tool.

The EIT system was applied to the task of *in situ* measurement of dispersed solid-liquid and gas-liquid flows to determine its ability to measure volume-averaged phase fractions and spatial phase distributions. In the solid-liquid tests, the size of the dispersed solid phase was much smaller than the size of the test vessel. In the gas-liquid bubbly flows, the scale of the dispersed phase (air bubbles) was larger than in the solid-liquid mixture but still small compared to the scale of the apparatus. This increase in phase scale represents a progression toward industrial-scale multiphase flows, where EIT may ultimately be applied to resolve questions about phase distributions and effects of solid phases in three-phase flows.

Liquid conductivity was controlled during the two-phase experiments by adding sodium nitrate to deionized water, ensuring that resistive effects would dominate over capacitive effects in EIT measurements as explained in Section 3.2.2. This marks a change from the validation tests in Chapter 3, where sodium chloride was used; sodium nitrate was chosen for all subsequent tests to minimize the possibility of electrode corrosion.

The governing equation for the computational domain, Eq. 3.1, assumes that no charge sources or sinks are present in the domain. For multiphase experiments, this assumption requires that no grounded conductors (except for the EIT withdrawal electrode) may be in contact with the conducting medium anywhere within 10 column diameters of the electrode ring. To enforce this requirement, only electrically isolated instruments (cartridge heaters and thermocouples) were allowed at the flow boundaries, and mechanical devices within the flows were electrically isolated or given nonconducting coatings.

The remainder of this chapter begins with a brief description of the Sandia GDT system, then continues with discussions of the experimental conditions of each two-phase flow test, the GDT and EIT results, and an assessment of EIT accuracy in each test. Unlike EIT studies involving image reconstruction – *e.g.*, Lin *et al.* (1993) – the emphasis here is not on <u>visualization</u> or on the accurate determination of interfaces between spatially segregated phases, but on the <u>quantitative</u> measurement of mixture properties for dispersed multiphase flows. The quantities determined by both GDT and EIT are averages over length scales larger than that of the dispersed phase but smaller than the measurement domain.

4.1. The Sandia GDT System

Many tomographic methods applied to multiphase flows involve radiation, such as neutrons, gamma rays, or x-rays, that is partially attenuated by the flow. Information about local density or phase distributions can be obtained by measuring the attenuation of radiation through the domain or by triangulating the locations of radiation sources within the flow (George *et al.*, 1998a). Gamma-densitometry tomography (GDT) uses collimated radiation sources to project beams of photons through the region of interest and detection systems to measure the fraction of photons that pass through the region unattenuated. Measurement of the attenuation along many different paths through the domain can be combined with linear tomographic reconstruction methods to produce an image of the phase distribution in the domain. GDT has been used for some time to investigate multiphase flows, and Munshi (1990), Beck *et al.* (1993), and Simons (1995) provide comprehensive reviews of the technique. GDT requires that the two phases have significantly different attenuation coefficients for good image resolution or useful quantitative results.

The GDT system developed at Sandia for studies of industrial-scale multiphase flows (Torczynski *et al.*, 1996b, 1997; Shollenberger *et al.*, 1997a) is shown in Figure 4.1. Figure 4.2 shows a schematic of the traverse as viewed from above, and Figure 4.3 presents a block diagram of the complete system. The system employs a 5-curie ^{137}Cs gamma source, a sodium-iodide scintillation detector system, computer-controlled traverses to position the source and detector on opposite sides of the test object, and hardware and software for data acquisition and system control. Attenuation of the 0.662-MeV photons is measured along many parallel beam paths, then attenuation by the testbed walls is subtracted from the raw data, and a path-averaged gamma attenuation coefficient, $\hat{\mu}$, is determined for each path within the domain.

Since GDT measurements require from 15 to 30 minutes, GDT essentially yields time-averaged results. In the bubble-column flows examined in this work, checks of the GDT attenuation profiles consistently revealed side-to-side symmetry across the column diameter over this time period, which justifies the assumption of an axisymmetric phase distribution. Using this assumption, the Abel transform (Vest, 1985) is used to convert the path-averaged attenuation coefficients $\hat{\mu}$ to a time-averaged, normalized, radial attenuation distribution, $\mu(r)$, in the circular domain. The attenuation distribution must then be transformed to a phase distribution through a mathematical relationship involving the attenuation coefficients of the separate phases. For the case of gas-liquid flows, the attenuation information may be converted to a gas volume fraction profile $\varepsilon_G(r)$ through the linear formula

$$\varepsilon_G^{GDT}(r) = \frac{\mu_L - \mu(r)}{\mu_L - \mu_G}, \tag{4.1}$$

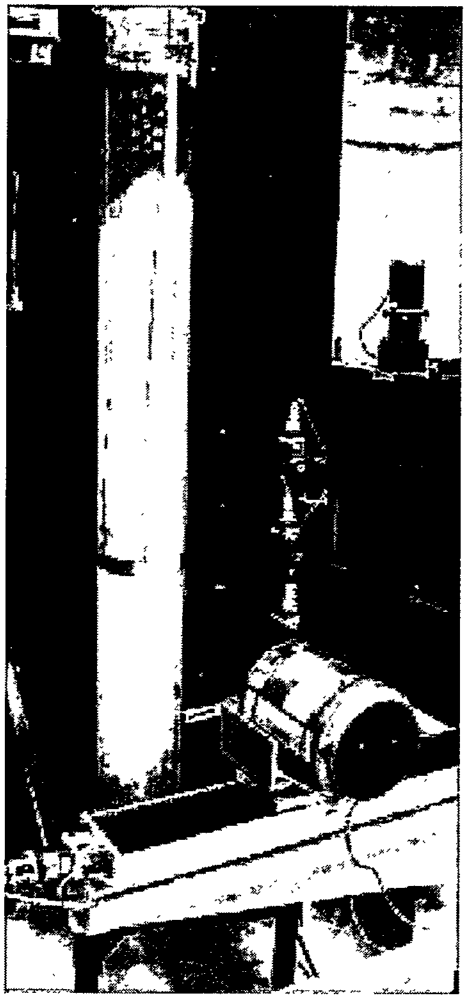


Figure 4.1. The Sandia GDT system and the transparent bubble column used in gas-liquid experiments. The ^{137}Cs source is in the foreground, shown on its traverse arm. The motor that positions the arms vertically is at top right.

where *r* is the radial coordinate, and μ_L and μ_G, the gamma ray attenuation coefficients of the liquid and gas, respectively, have been previously measured or are known *a priori*. The liquid volume fraction profile, $\varepsilon_L(r)$, may also be determined by subtracting the local value of $\varepsilon_G(r)$ from unity. More details on the GDT system and reconstruction algorithm may be found in Torczynski *et al.* (1997); a listing of the current version of the reconstruction code, `GDTAXI`, appears in Appendix I.

For the flows examined in this report, both the attenuation and phase distributions are modeled by the GDT reconstruction algorithm as polynomials with only even powers of *r*, based on the observation that the measured attenuation profiles are axisymmetric. The EIT reconstruction algorithm also assumes axisymmetric material distributions in these tests, allowing straightforward comparisons of EIT and GDT results. By substituting a cross-sectionally-averaged gamma attenuation coefficient $\langle\mu\rangle$ for $\mu(r)$ in Eq. 4.1, a cross-sectionally-averaged phase volume fraction, $\langle\varepsilon_G\rangle$, may be obtained. Also, as will be demonstrated in the

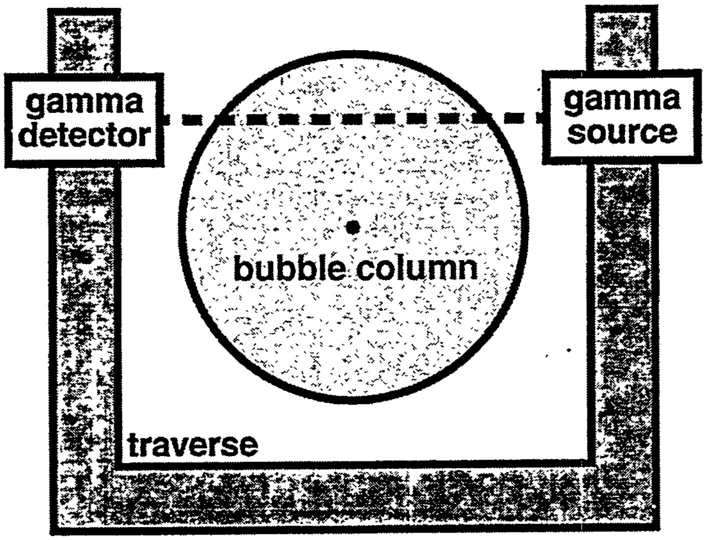


Figure 4.2. Conceptual diagram of the Sandia GDT system as viewed from above.

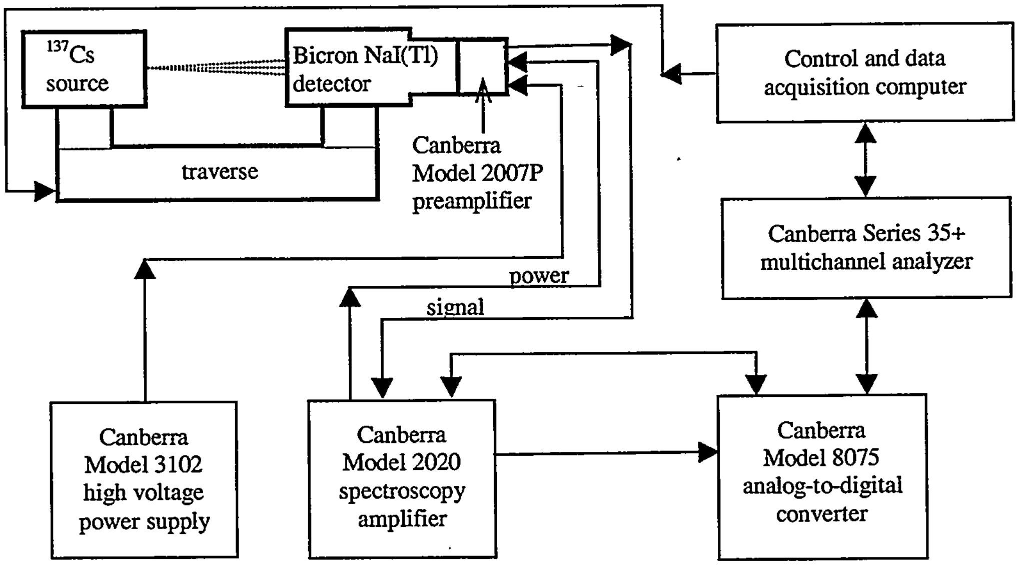

^{137}Cs source
Bicron NaI(Tl) detector
Canberra Model 2007P preamplifier
traverse
Control and data acquisition computer
Canberra Series 35+ multichannel analyzer
power
signal
Canberra Model 3102 high voltage power supply
Canberra Model 2020 spectroscopy amplifier
Canberra Model 8075 analog-to-digital converter

Figure 4.3. Block diagram of the Sandia GDT system.

next section, GDT data may be used with the same algorithm and a similar formula to measure phase distributions in solid-liquid or solid-gas flows if the attenuation coefficients of the two phases differ significantly (George *et al.*, 1999b).

4.2. Solid-Liquid Experiments

The first two-phase EIT validation experiments involved a closed container of conducting liquid with a flow of insulating solid particles driven by an impeller. This solid-liquid flow was chosen for two reasons. First, the amount of solids introduced into the container can be carefully controlled, providing an independent check of the tomography results. Second, unlike gas bubbles, solid particles can be small, uniform spheres that do not change shape during the experiment, so long as conditions are not harsh enough to fracture the particles. The solid particles and the liquid medium can be chosen to have significantly different gamma attenuation coefficients and electrical conductivities, so that GDT and EIT can be applied separately and independently to determine the solid volume fraction. However, liquids and insulating solids that have significantly different attenuation coefficients typically have substantially different densities as well, making the solids difficult to loft uniformly (Shollenberger *et al.*, 1997b).

A schematic diagram and a photograph of the experimental setup are shown in Figures 4.4 and 4.5. The testbed consisted of the strip-electrode cylinder stacked between two Lexan cylinders of the same inner diameter and wall thickness. This testbed, 81.3 cm in height, was found to be tall enough to encompass all the significant electric field lines from the strip electrodes. The three-part cylinder was capped at the bottom and the top, and a mixer was inserted into the cylinder through a 2.5-cm-diameter hole in the center of the top cap. The mixer consisted of a compact impeller assembly, a 1.5-watt Sargent-Welch motor (model S-76509-80B) mounted above the testbed, and a two-piece shaft 0.8 cm in diameter extending from the impeller to the motor. Since the mixer's original shaft was not long enough to reach out of the cylinder, an insulating Lexan extender joined the end of the mixer shaft to a second shaft attached to the mixer motor. The mixer was used to generate a relatively uniform solids distribution inside the cylindrical testbed. To prevent settling of the solids, the impeller was positioned 1.27 cm above the testbed floor. An overflow volume was placed around the top cap to eliminate free-surface effects — *e.g.*, a vortical "funnel" — in the cylinder interior during mixing.

Although the mixer shaft was small compared to the test cylinder (the ratio of their diameters was 0.042), the presence of the shaft posed problems for EIT. Placing an unshielded electrical conductor in the center of the testbed has the potential to significantly distort the electric field lines, so the steel shaft and impeller were coated with a layer of insulating paint to mitigate this effect. The Lexan extender also served to further insulate the motor assembly from the testbed. Earlier EIT measurements with and without the mixer shaft in place, with no particles present, verified that the insulated shaft had only a small effect on EIT measurements (Shollenberger *et al.*, 1997b).

The liquid phase was deionized water with a small amount of saturated sodium nitrate solution added for conductivity control. Liquid conductivity values ranged from 265 to 275 μS/cm over the course of experiments. For the solid phase, polystyrene spheres with a density of ρ_S = 1.04 g/cm^3 and a range of diameters from 200 to 700 μm were used (Figure 4.6). Polystyrene was chosen for this experiment because its specific gravity is nearly that of water,

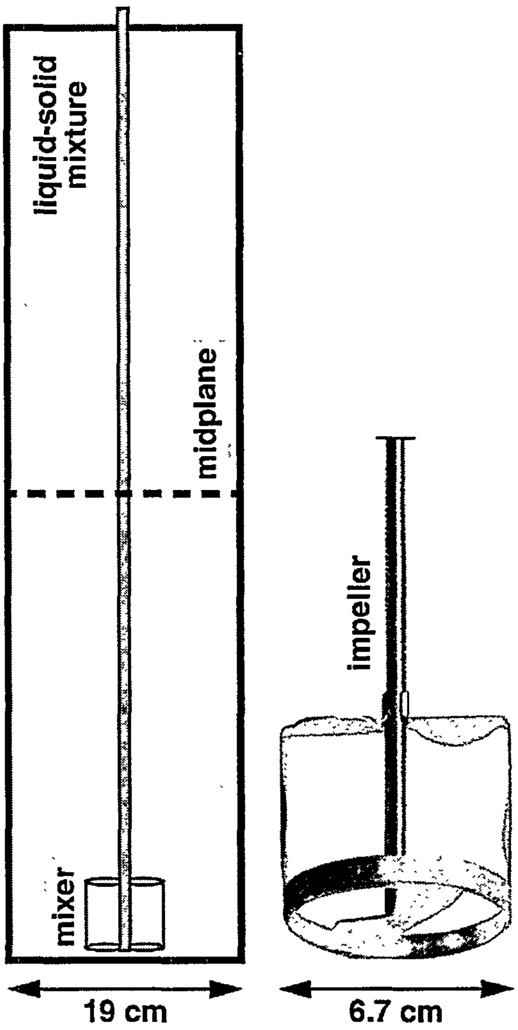


Figure 4.4. Schematic of test geometry for the solid-liquid experiments showing the impeller geometry and location within the testbed (adapted from Shollenberger *et al.*, 1997b).

making it easier to loft uniformly within the flow than heavier solids. Polystyrene is also an insulator with respect to water, so that EIT is able to discriminate between the solid and liquid phases. GDT is not useful in this case because the attenuation coefficients of polystyrene and water are too similar for the phases to be distinguished [$\mu_L = 0.0856\ \text{cm}^{-1}$, $\mu_S = 0.0866\ \text{cm}^{-1}$ (Thoraeus, 1965)]. Instead, EIT measurements were compared to the nominal cylinder-averaged solid volume fraction $\bar{\varepsilon}_S^{NOM}$ chosen for each test.

For each chosen value of $\bar{\varepsilon}_S^{NOM}$, the appropriate mass of polystyrene was computed from its density and the volume of the test cylinder. The required mass of spheres was introduced into the testbed, and liquid was added to occupy the remaining volume. A mixer speed of 420 rpm was applied for 30 minutes to all solid loadings, with the result that a roughly uniform distribution of particles was observed visually within the liquid during measurements (Figure 4.7). EIT voltage measurements were taken periodically over the thirty-minute stirring period to

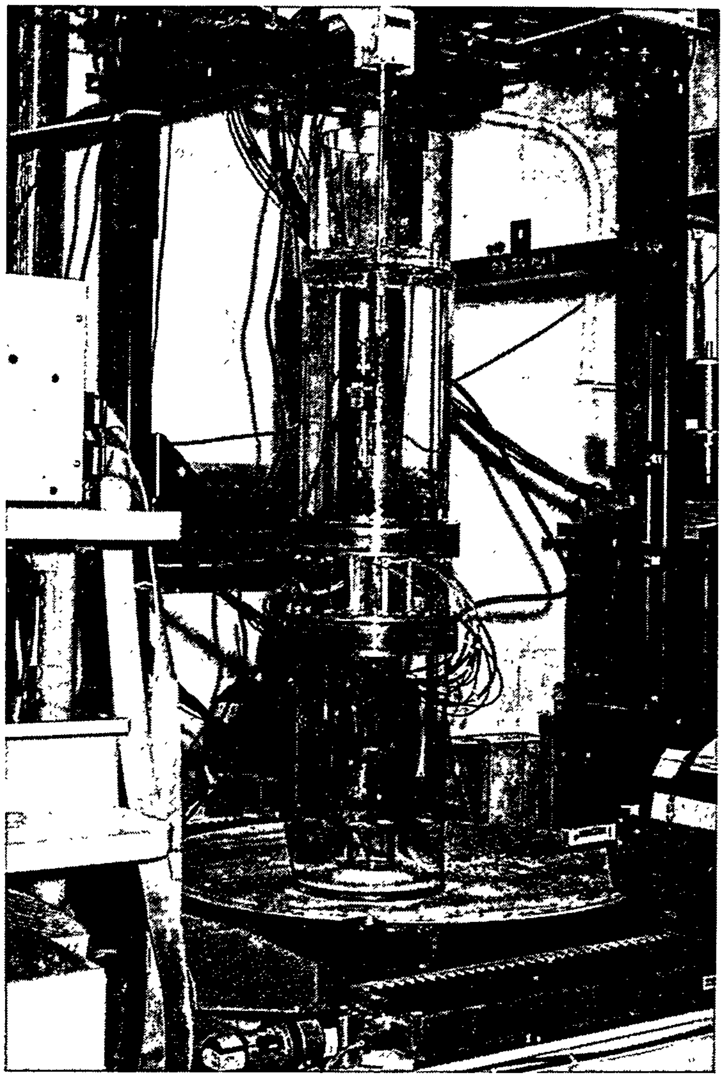


Figure 4.5. Photograph of the testbed for polystyrene-water solid-liquid experiments. The impeller motor is visible at the top; the EIT cabinet and GDT source are on the left and right sides of the photo, respectively.

verify that a steady-state distribution had been attained. After EIT measurements were taken of the uniform equilibrium distribution, mixing was terminated, the spheres were allowed to settle to the bottom of the cylinder, and EIT was applied again. The second EIT measurement was necessary to determine a baseline liquid conductivity for calibration: despite attempts to wash and rinse the polystyrene particles before the experiments, soluble contaminants were introduced with the particles that altered the conductivity of the water. The magnitude of this effect on the water conductivity was comparable to that of the suspended solid particles during mixing. By comparison, changes in water temperature and associated conductivity changes were negligible during mixing. In both the calibration measurements and the measurements of the uniform solid-liquid flow, 25 complete projection sets were averaged over a period of about 20 seconds to average out the effects of any fluctuations in the flow and thereby justify the assumption of axisymmetry in reconstructions.

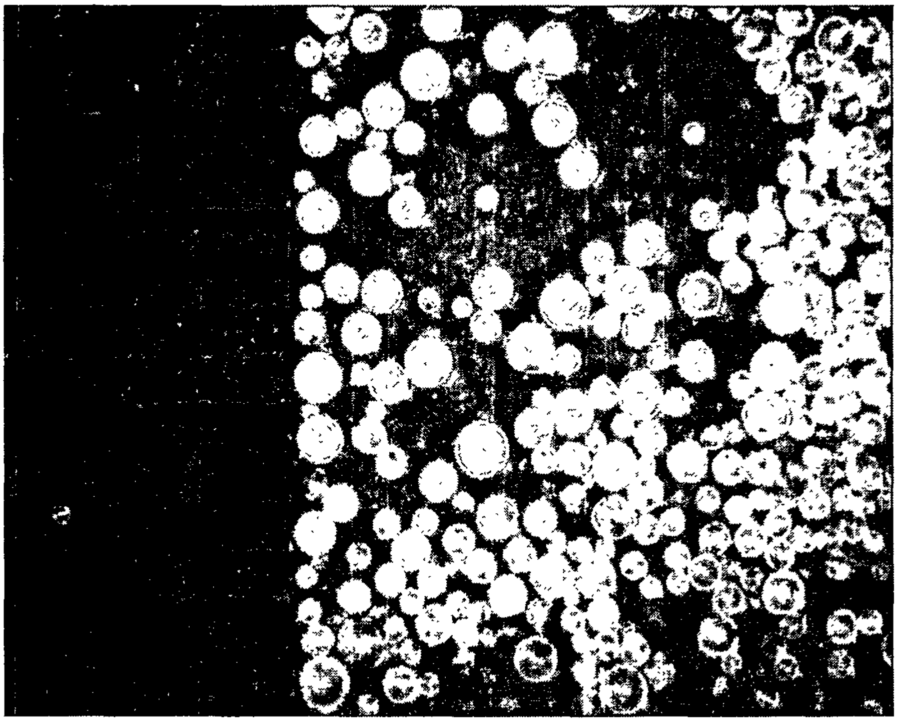


Figure 4.6. Photograph of polystyrene particles used as the solid phase. The smallest division on the scale is 0.5 mm.

Analysis of the solid-liquid data with EITAXI demonstrated that the radial solids profiles were uniform to within experimental uncertainty, as expected. Radial variations in the local solids density ε_S were less than 0.01 across the domain, changing over length scales much larger than the diameters of the particles themselves. The radial profiles were analytically integrated to obtain cross-sectionally-averaged solid volume fractions $\bar{\varepsilon}_S^{EIT}$, plotted in Figure 4.8. The solid volume fractions determined by EIT are seen to be in close agreement in all cases with the nominal values computed from the mass of added particles, validating the EIT method for solid-liquid measurements.

The difference between measured and nominal solid volume fractions becomes more negative with increasing solid volume fraction, approaching $\bar{\varepsilon}_S^{EIT} - \bar{\varepsilon}_S^{NOM} = -0.003$ when $\bar{\varepsilon}_S^{NOM} = 0.050$. It is possible that the ability of the mixer to produce a uniform axial distribution dropped with increased solid loadings, so that the local solid volume fraction was slightly lower than the nominal value near the top of the cylinder and slightly higher at the bottom. Another possible explanation for the trend stems from the observation that a small amount of solids escaped through the hole in the top cap over the course of each test. These solids passed into the overflow volume and reduced the actual amount of solids visible to EIT in the measurement volume. The fraction of total solids lost to the overflow volume was observed to increase with nominal loading, possibly causing the observed trend.

Before the validation tests in Chapter 3 were performed to determine an optimum electrode geometry, an EIT experiment was conducted by Sandia researchers in a solid-liquid

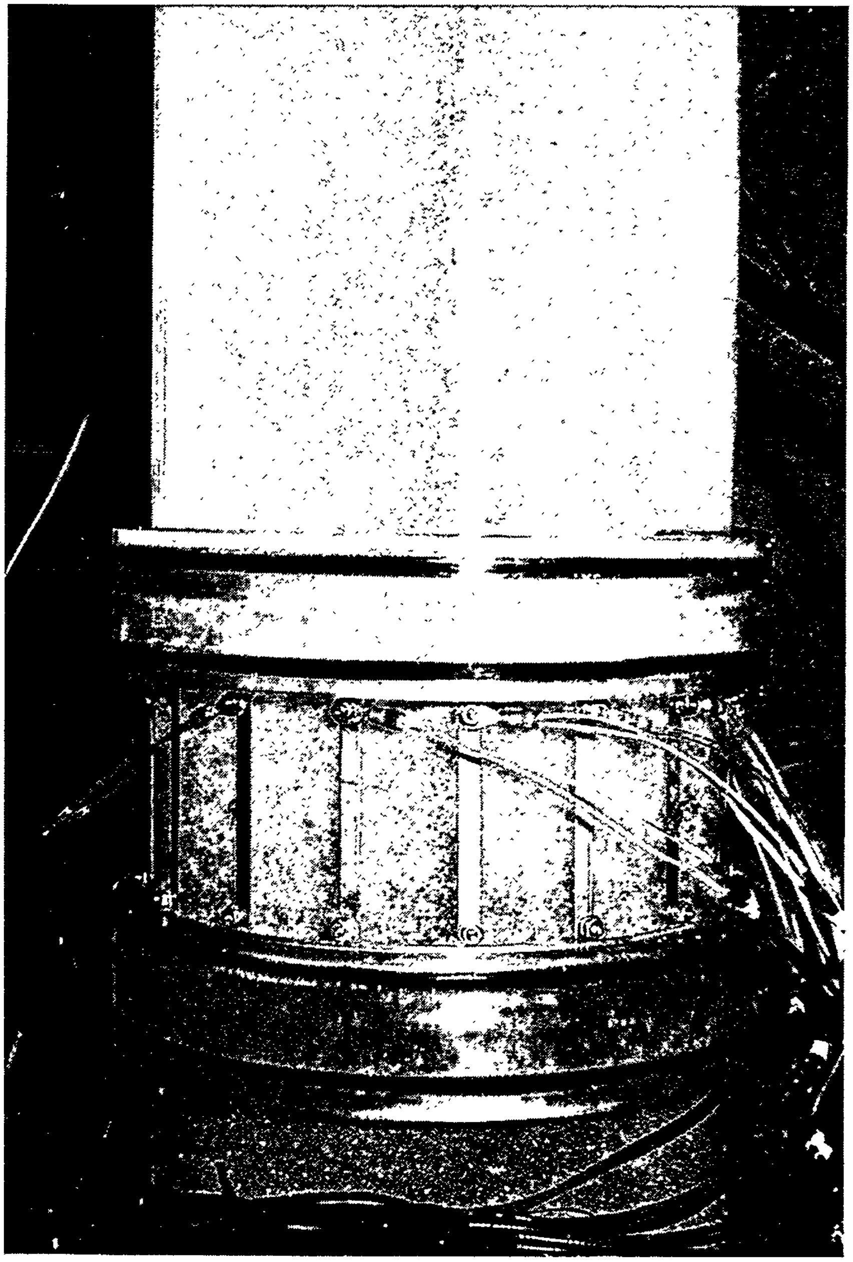


Figure 4.7. Photograph of flow conditions during solid-liquid experiments with $\bar{\varepsilon}_S^{NOM}$ = 0.01.

flow employing glass spheres with a mean diameter of 80 μm as the solid phase (Shollenberger *et al.*, 1997b). This experiment involved an electrode ring composed of sixteen disk electrodes 3 mm in diameter. Though the geometry tests later found strip electrodes to be superior to the point-like electrodes, these test results are significant because GDT was used to validate the EIT measurements. The experiments were very similar in setup and execution to the polystyrene-water experiments, but some differences should be noted, particularly the different collection times for EIT data and the higher electrical conductivity of the liquid phase required by the

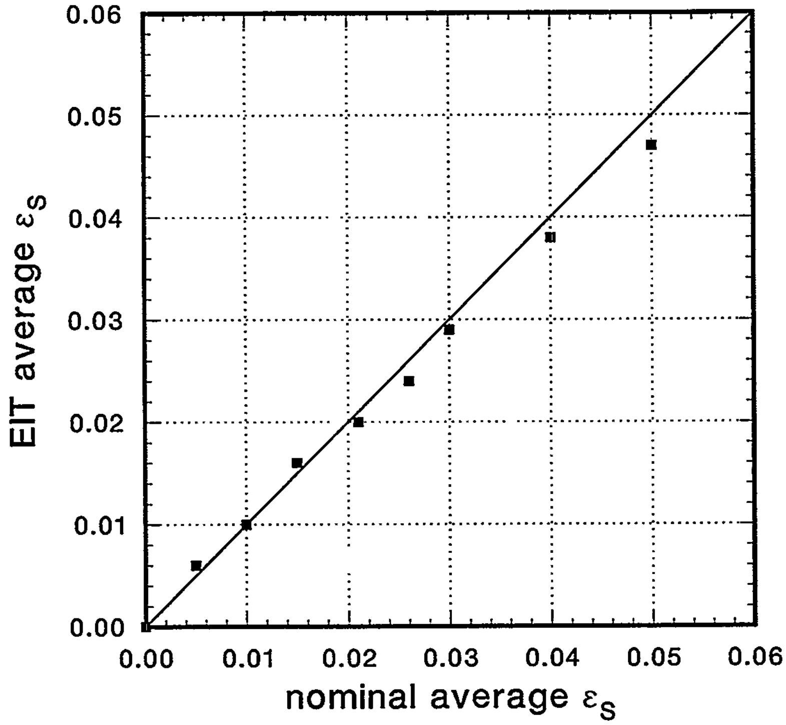


Figure 4.8. Solid volume fractions measured by EIT in polystyrene-water solid-liquid mixtures.

smaller size of the electrodes. The reader is referred to Torczynski *et al.* (1997) for details of these glass-water tests.

Analysis of the GDT glass-water data demonstrated that the time-averaged radial solids profiles were relatively uniform across the testbed. Using the assumption of a spatially uniform gamma attenuation coefficient $\langle\mu\rangle$, GDT data were converted to cross-sectionally-averaged solid volume fractions through the formula

$$\langle\varepsilon_S\rangle^{GDT} = \frac{\langle\mu\rangle - \mu_L}{\mu_S - \mu_L}, \tag{4.2}$$

where the attenuation coefficients μ_L and μ_S of the water and glass spheres were previously measured (Torczynski *et al.*, 1996b). Similarly, nearly identical domain-averaged conductivities $\overline{\sigma}$ were obtained from the EIT algorithm when using either a spatially uniform conductivity distribution or a radially parabolic distribution, so the simpler uniform distribution was employed. This simplification allowed the transformation of average EIT data directly to an average solid volume fraction, without integration of the volume fraction profile. The three-dimensional Maxwell-Hewitt relation, Eq. 3.13, was used with the solid-liquid continuity equation to arrive at the final formula,

$$\overline{\varepsilon}_S^{EIT} = \frac{1-[\overline{\sigma}/\sigma_L]}{1+\frac{1}{2}[\overline{\sigma}/\sigma_L]}. \tag{4.3}$$

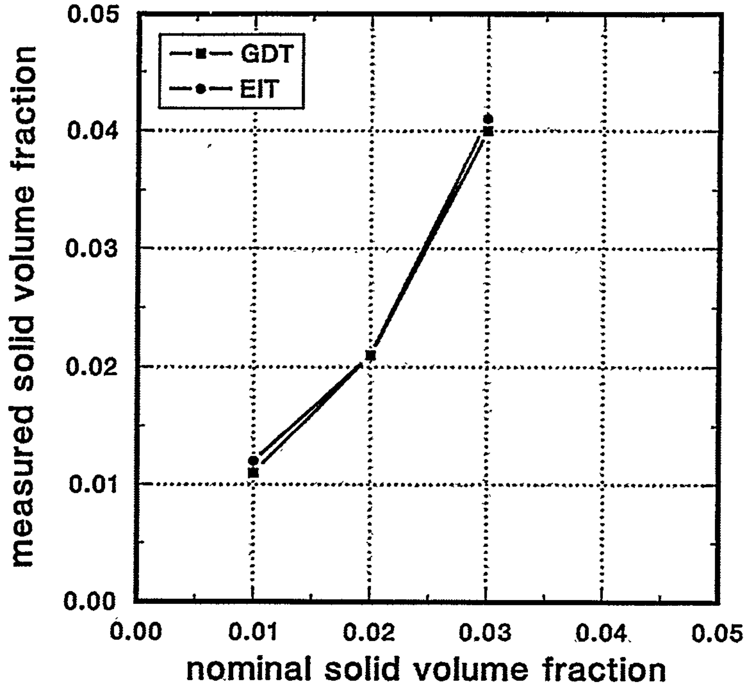


Figure 4.9. Comparison of testbed-averaged solid volume fractions measured by EIT and GDT in glass-water mixtures (from Shollenberger *et al.*, 1997b).

The solid volume fractions determined by GDT and EIT were in close agreement with each other for all three cases (Figure 4.9) and with the nominal values for the two lower solids loadings. The case of $\bar{\varepsilon}_S^{NOM}$ = 0.03 is interesting in that the GDT and EIT values were in agreement with each other but were somewhat higher than the nominal value. It was conjectured (as in the polystyrene tests) that the mixing in the latter case may not have been strong enough to overcome buoyancy-driven stratification, and thus did not produce a uniform solids distribution (Shollenberger *et al.*, 1997b). However, the close agreement between GDT and EIT in all three cases further validates the Sandia/Michigan EIT system in the measurement of solid-liquid phase distributions.

4.3. Gas-Liquid Experiments

4.3.1. Experimental Setup and Conditions

In the next phase of EIT evaluation, gas volume fraction spatial distributions were measured in air-water flows with both EIT and GDT. These gas-liquid tests were conducted in a transparent Lexan bubble column assembled at Sandia as a testbed for optical, electrical, and radiation-based multiphase flow diagnostics (Torczynski *et al.*, 1997). Figure 4.10 shows the EIT and GDT systems in place around the bubble column. The column itself, also shown

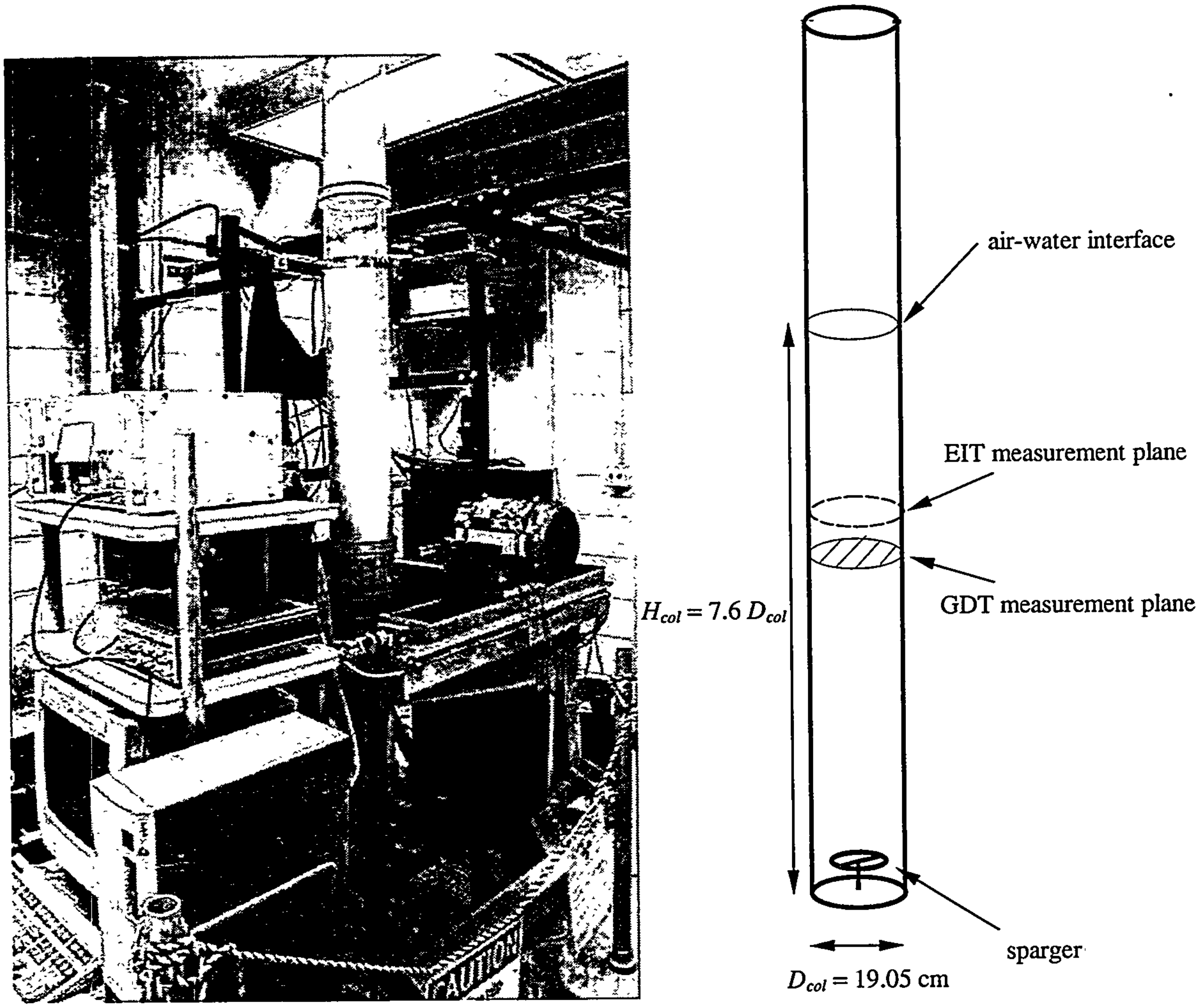


Figure 4.10. Experimental setup for EIT validation experiments in the Lexan bubble column. Left, a photograph of the EIT system connected to the strip electrode ring in the center of the column. The ^{137}Cs source and GDT traverse are visible on the right side of the photo. Right, a schematic diagram of the bubble column indicating measurement locations (not to scale).

schematically in Figure 4.10, has an inner diameter D_{col} of 19.05 cm, a wall thickness of 0.64 cm, and is built from interchangeable sections so that different diagnostic tools can be placed in the column. In this study, the EIT electrode section was placed near the center of the column. The 16 strip electrodes were centered on a plane $L = 97.16$ cm above the base, at a height-to-diameter ratio of $L / D_{col} = 5.1$. The column was filled with water to a depth H_{col} of 1.45 m, for a height-to-diameter ratio of 7.6; this depth placed the region of EIT sensitivity completely under water.

Dry air can be introduced at volumetric flow rates up to 600 L/min (superficial gas velocities up to 35 cm/s) through one of several interchangeable spargers at the base of the column. The spargers are electrically isolated to prevent interference with the EIT system. The

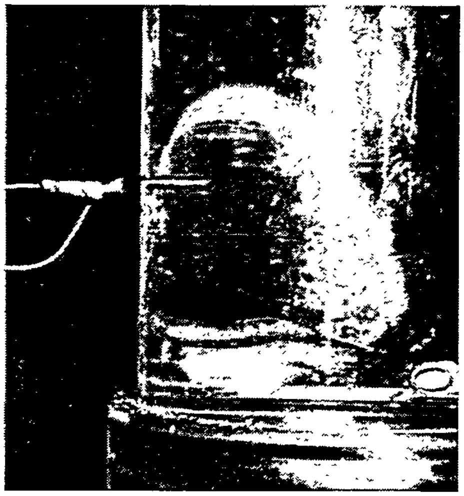


Figure 4.11. Typical cartridge heater used to maintain constant temperature within bubble-column flows.

column operates at ambient conditions, but the water is subject to evaporative cooling as air is bubbled through. To mitigate the effects of cooling on conductivity, heat energy was added to the flow by three cartridge heaters mounted at the column walls (Figure 4.11). A portable computer monitored the flow temperature via thermocouples mounted flush with the column walls, and the rate of heat addition was actively controlled by a rheostat to maintain a constant temperature in the flow. The liquid temperature was held constant by active heating to within ±0.2°C during experiments, which limited variations in conductivity with temperature to ±0.3%.

The goal for the gas-liquid tests was to validate EIT against GDT in a flow with variations on a larger scale than the solid-liquid flows. First, to investigate such a flow with smoothly varying properties, it was decided to create a homogeneous bubbly flow within the column, a flow in which bubbles do not coalesce and have minimal influence on one another's motion. Two different sparger designs for homogeneous flows (see Figure 4.12) were tested, and air was introduced at low superficial gas velocities ranging from 0.3 to 1.5 cm/s. Also, controlled amounts of electrolyte and Triton X-100 surfactant were added to the working liquid to reduce surface tension, prevent coalescence and delay the transition to churn-turbulent flow (*e.g.*, see Wilkinson *et al.*, 1992). Despite these efforts, a truly homogeneous bubbly flow could not be attained with either sparger. GDT reconstructions of the radial gas volume fraction profiles from both sparger designs were best described by fourth-order polynomials with maxima located at $r > 0$, instead of at the column centerline. Vortical motions were also observed at the higher flow rates, indicating a transition from homogeneous to churn-turbulent flow (George *et al.*, 1998b).

Consequently, a comparison of the two tomographic methods was made in churn-turbulent column flows. Such vertical flows normally have time-averaged volume fraction profiles that are approximately parabolic and can therefore be reconstructed by both the GDT and EIT algorithms. The sparger used to produce churn-turbulent flows is shown in Figure 4.13. This sparger is a hollow stainless steel toroid with a 10.16-cm centerline diameter, an inner tube diameter of 0.95 cm, and ten holes of diameter 0.16 cm facing downward. Spargers are often

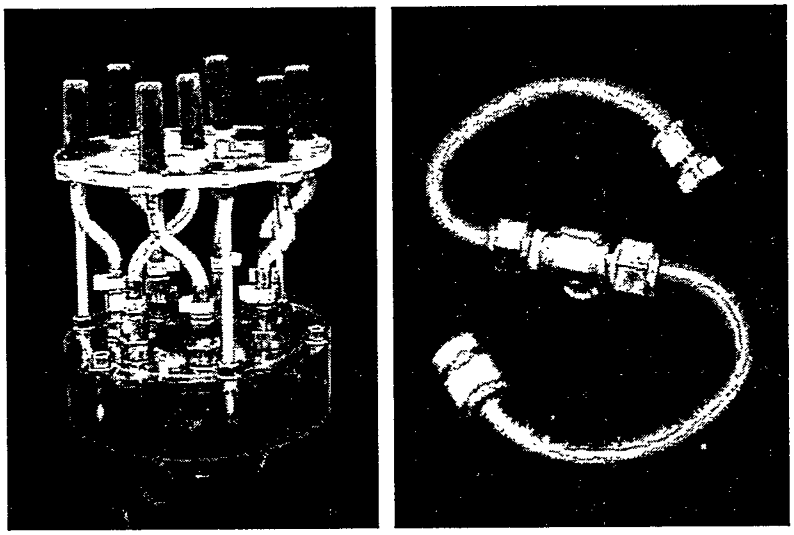


Figure 4.12. Sparger designs used in the unsuccessful homogeneous bubbly flow experiments. The sparger on the left injects air into the column through seven "airstone" frits made of sintered glass particles averaging 0.026 cm in diameter. The plenum and frit support are both 15.2 cm in diameter. The S-shaped sparger on the right is made of 0.95-cm I.D. copper tubing, and has an approximate diameter of 13 cm. Sixteen holes were drilled on each arm using a #80 (0.343 mm-diameter) drill bit.

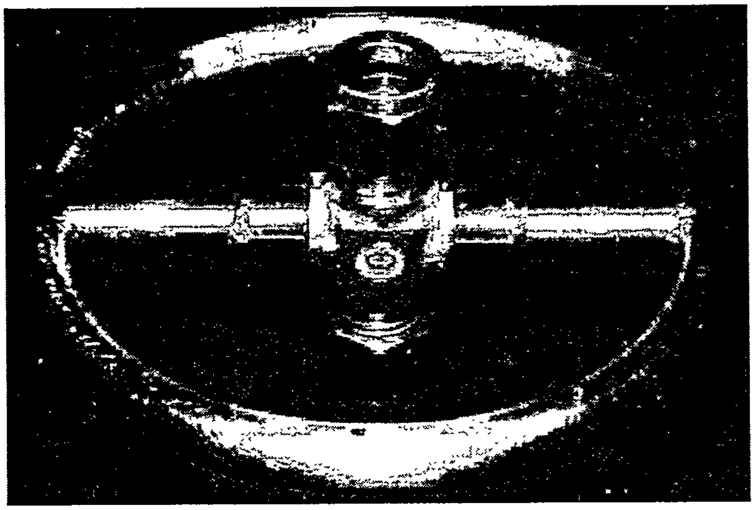


Figure 4.13. Ring sparger used in churn-turbulent flow experiments.

quantified by their porosity, the ratio of the total area of the sparger holes to the cross-sectional area of the column. In the Lexan bubble column, this sparger has a porosity of 0.00069.

Churn-turbulent experiments were performed using five air volumetric flow rates: Q_G = 25, 50, 75, 100 and 150 L/min. The corresponding superficial gas velocities U_G ranged from 1.5 to 8.8 cm/s. Figure 4.14 shows the conditions in the column for the minimum and maximum gas flow rates. At the lowest flow condition, a range of spherical and coalesced nonspherical bubble

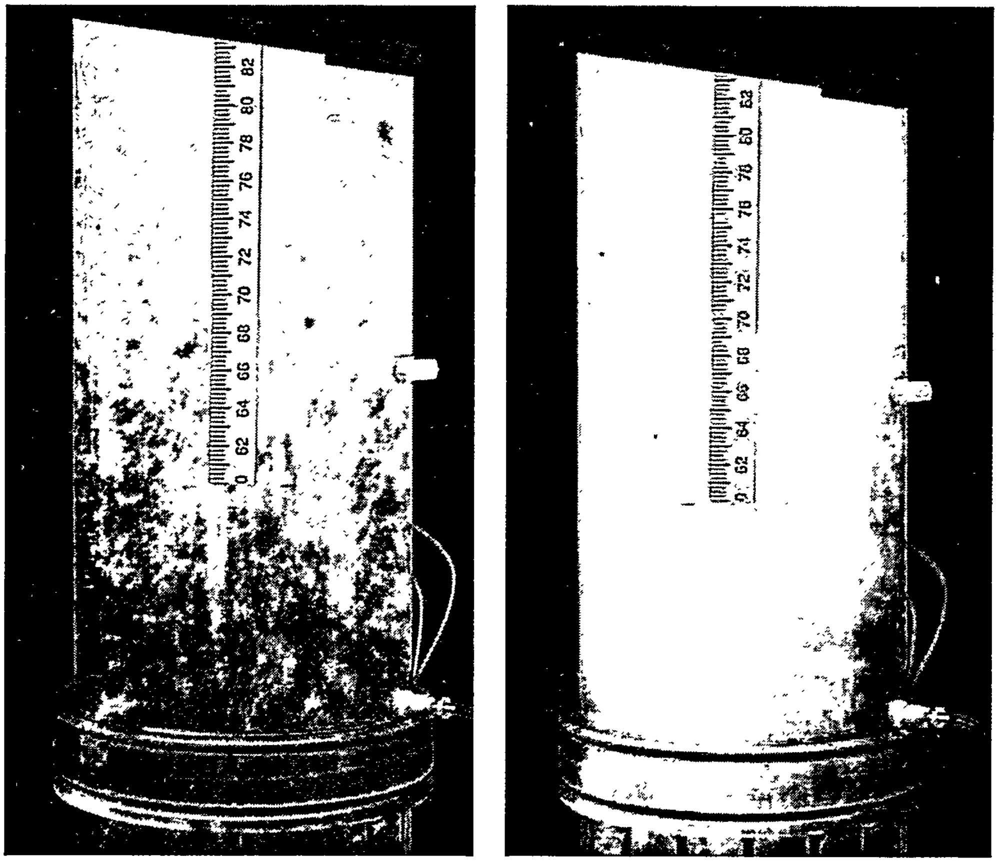


Figure 4.14. Churn-turbulent flow conditions in the transparent bubble column at minimum and maximum volumetric flow rates: (left) Q_G = 25 L/min, (right) Q_G = 150 L/min. The vertical scale is in centimeters.

sizes was evident, as was a central helical bubble stream. At rates of 75 L/min and above, the flow in the column was completely opaque and turbulent to the naked eye.

For each flow condition, 25 full EIT projection sets taken over a period of less than 20 seconds were averaged to obtain the voltage data for reconstructions. Multiple datasets were averaged to enable direct comparison with GDT results, which are inherently time-averaged, and to allow use of axisymmetric reconstruction algorithms. The conductivity of the water used in the gas-liquid experiments was 285±5 μS/cm. GDT measurements were taken in a plane 81.0 cm above the column floor (L/D_{col} = 4.25). This location, 16.2 cm below the center plane of the EIT electrodes, was chosen to avoid electrode obstruction of the photon paths. The GDT measurement chords were parallel and spaced 1 cm apart; detector counts were collected for 60 seconds along each chord.

In the EIT reconstructions, the three-dimensional code `EITAXI` was used with the assumption of a parabolic conductivity profile, given by Eq. 3.8 with C_2 = 0 (at this time, the quartic reconstruction option had not yet been added to `EITAXI`). Chord-averaged attenuation coefficients computed from GDT data were fit to a second or fourth-order polynomial before the Abel transform was taken. The choice of polynomial for each fit was made based upon the least-squares error between the fit and the attenuation data. It should be noted that GDT reconstructions represent the phase distribution over a very thin, nearly two-dimensional cross section of the column, while EIT measures phase distributions within a three-dimensional volume extending above and below the electrode strips. However, with both the electrode center plane and GDT scan plane located well into the fully developed flow region ($L/D_{col} > 2$), variations in the phase distributions between the GDT and EIT measurement domains were expected to be negligible.

4.3.2. Initial Results

Figure 4.15 shows the gas volume profiles measured by both methods for all volumetric flow rates. The EIT profile tends to lie slightly above the GDT profile at each flow rate; however, the GDT and EIT gas volume fractions are in excellent agreement, falling within 0.01 of each other at all radial positions. Since the change in the local volume fraction across the column is as large as 0.20, this level of agreement strongly validates the EIT method. Values of the domain-averaged gas volume fraction $\bar{\varepsilon}_G$ were also determined from EIT and GDT by analytically averaging the profiles of Figure 4.15 over the column area. Figure 4.16 shows these cross-sectionally-averaged gas volume fractions as a function of superficial gas velocity U_G. Again, the GDT and EIT values are in very good agreement, differing by no more than 0.01 in $\bar{\varepsilon}_G$. This degree of consistency between GDT and EIT suggests that both methods are accurate for measuring axisymmetric profiles so long as temporal-averaging effects are benign.

4.3.3. Uncertainty Analysis

Because of the difference in collection times for the two methods (about 23 minutes for GDT but less than 20 seconds for EIT), it was decided to assess the effect of temporal averaging on the EIT and GDT reconstructions. Difficulties could arise, for example, from slow oscillations in the flows that could be averaged out by GDT but not by EIT. Separate flow measurements were taken under the same churn-turbulent flow conditions with an impedance-based bulk void fraction meter described in detail by Torczynski *et al.* (1997). This instrument has two rectangular electrodes 3.8 cm high, which subtend 120° each on opposite sides of the column (Figure 4.17). An impedance phase meter circuit operating at 50 kHz is used to excite the electrodes. In earlier work, this instrument was used to measure bulk gas volume fractions in the bubble column through comparisons of signals from flow and no-flow conditions. Here, the voltage across the electrodes was recorded with a Tektronix model TDS 640 digitizing oscilloscope for analysis of fluctuations in gas volume fraction. The meter is capable of detecting fluctuations with time scales on the order of 40 μs or longer. Since the time scales for churn-turbulent flows in the bubble column are expected to be much larger than this value, the meter was used to resolve temporal variations in the flow.

The voltage across the bulk meter electrodes was recorded over a period of ten minutes at each flow condition, and Fourier spectra were computed from the voltage data to determine the frequencies of gas volume fraction fluctuations. Signal spectra revealed no coherent periodic

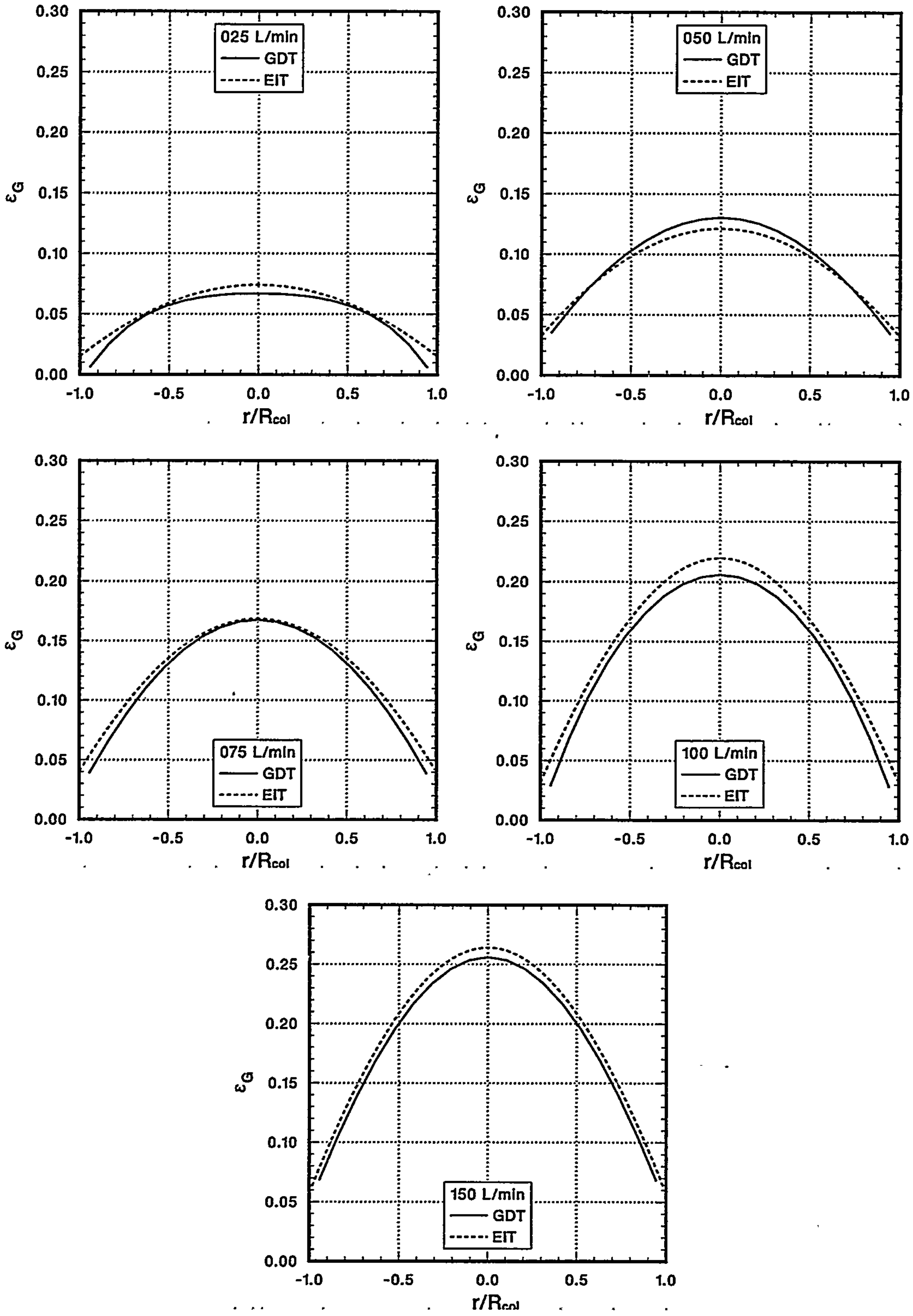


Figure 4.15. Comparisons of symmetric radial gas volume fraction profiles from GDT and EIT.

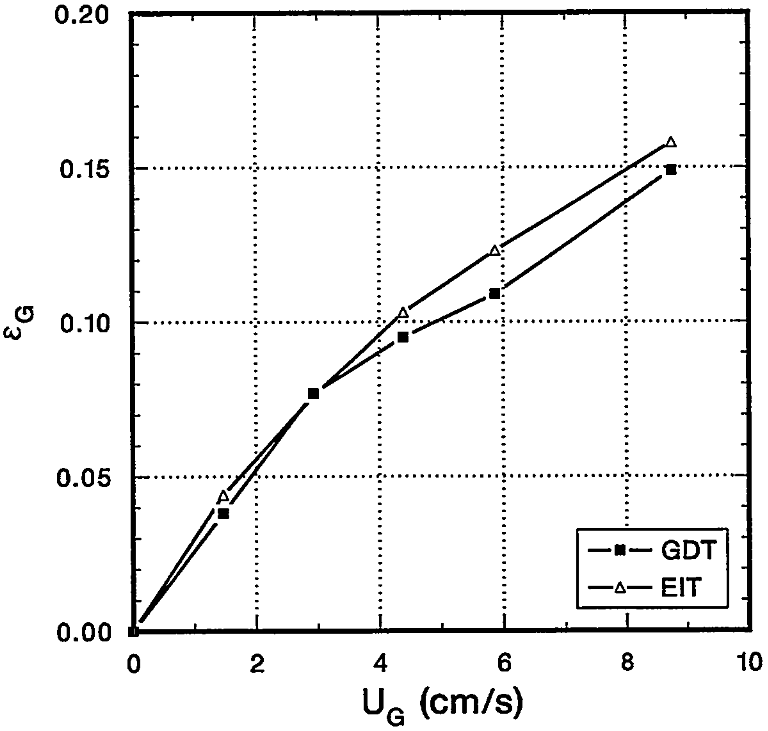


Figure 4.16. Comparison of cross-sectionally-averaged gas volume fractions measured by GDT and EIT.

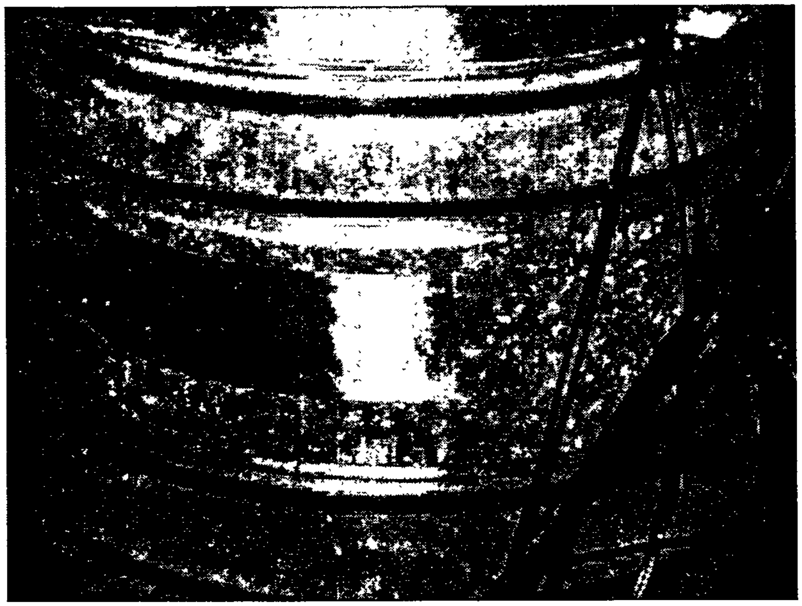


Figure 4.17. Impedance-based bulk void fraction meter used to investigate fluctuations in gas volume fraction.

flow behavior with a frequency less than 12.5 Hz; this alleviated the concern that EIT measurements were capturing data over only a portion of slow flow cycles. The bulk meter voltage signals were also analyzed to estimate the magnitude of variations in gas volume fraction

Table 4.1. Comparison of differences between EIT and GDT measurements with temporal fluctuations in gas volume fraction measured by the bulk void fraction meter.

U_G (cm/s)	$\overline{\varepsilon}_G^{EIT}$	$\langle\varepsilon_G\rangle^{GDT}$	$\overline{\varepsilon}_G^{EIT} - \langle\varepsilon_G\rangle^{GDT}$	$\Delta\overline{\varepsilon}_G^{Bulk}$
1.5	0.044	0.038	0.006	0.015
2.9	0.076	0.077	-0.001	0.016
4.4	0.103	0.095	0.008	0.018
5.9	0.123	0.109	0.014	0.020
8.8	0.158	0.149	0.009	0.023

for all flow conditions. Using a sum-of-squares method, the rms voltage variation measured without gas flow was subtracted from the rms variation at each flow condition to estimate variations in the signals due solely to variations in gas volume fraction. Time-averaged voltages $\overline{V}$ were correlated with cross-sectionally-averaged gas volume fractions $\langle\varepsilon_G\rangle^{GDT}$ determined from concurrent GDT scans, and temporal variations in $\overline{\varepsilon}_G^{Bulk}$ at each flow condition were estimated from the voltage variations and the correlation of $\langle\varepsilon_G\rangle^{GDT}$ versus $\overline{V}$.

Table 4.1 compares $\Delta\overline{\varepsilon}_G^{Bulk}$, the estimate of these temporal variations, with $\overline{\varepsilon}_G^{EIT} - \langle\varepsilon_G\rangle^{GDT}$, the difference between the average values measured by EIT and GDT. Since $\Delta\overline{\varepsilon}_G^{Bulk}$ and $\overline{\varepsilon}_G^{EIT} - \langle\varepsilon_G\rangle^{GDT}$ are comparable, the different methods of temporal averaging by EIT and GDT may account for some of the difference between the average values measured by these two techniques. However, since the difference $\overline{\varepsilon}_G^{EIT} - \langle\varepsilon_G\rangle^{GDT}$ and the variation $\Delta\overline{\varepsilon}_G^{Bulk}$ are small compared to both $\overline{\varepsilon}_G^{EIT}$ and $\langle\varepsilon_G\rangle^{GDT}$, both methods of temporal averaging appear to be benign. The relatively small difference between the methods may also be taken as evidence that the difference in measurement domains (negligible vertical thickness for GDT versus a few diameters in the vertical direction for EIT) is also benign. These conclusions are probably valid only for the fully developed flow region; it is likely that the different temporal averaging methods and measurement domains would be more important for strongly developing flows. Future investigations will examine the effects of temporal averaging by EIT and GDT more closely.

4.3.4. Modified Maxwell-Hewitt Relation and Revised Results

Another possible explanation for the difference in results between GDT and EIT, one related to the issue of temporal averaging, is the small but finite vertical motion of bubbles during EIT measurements. One assumption inherent in EIT reconstructions is that the impedance distribution in the domain does not change significantly over the course of voltage measurements (Ceccio and George, 1996). Any movement or change in shape of an insulating void over the course of a projection set will yield inconsistent voltages that may produce a conductivity reconstruction "smeared" over time. While the three-dimensional Maxwell-Hewitt relation (Eq. 3.13) assumes three-dimensional insulating inclusions in the domain (*e.g.*, spherical

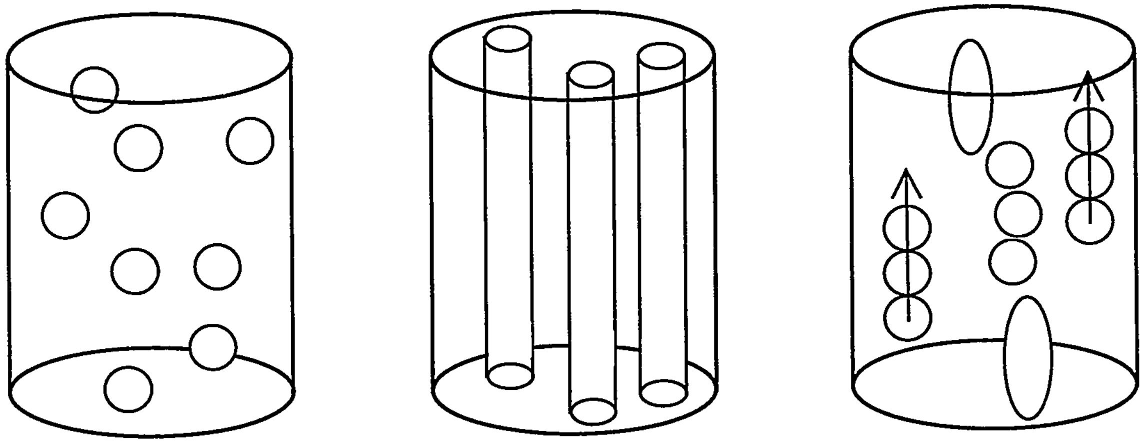


Figure 4.18. Concept of the modified Maxwell-Hewitt relation. Left, spherical insulators in a three-dimensional reconstruction; center, cylindrical insulators in a two-dimensional reconstruction; right, "time-exposure" of moving and deformed bubbles in a realistic bubble-column flow.

bubbles), the two-dimensional version (Eq. 3.14) assumes invariance in the axial direction, and the domain it describes may be thought of as infinitely high and containing cylindrical objects of infinite height (Figure 4.18). Vertical motion or axial elongation of bubbles during voltage acquisitions in the column may produce an effective conductivity distribution somewhere between the ideal three-dimensional and two-dimensional cases, so that Eq. 3.13 is not strictly applicable. (Recall that the electric fields in the column are three-dimensional, although the reconstructed distribution is two-dimensional.) Since the only difference between the two Maxwell-Hewitt relations is the factor of ½ versus 1 in the denominator, adjustment of this factor to a value between ½ and 1 could approximately account for this axial motion of the gas phase during EIT measurements.

At the same time the EIT reconstruction algorithm was modified to include quartic conductivity profiles, empirical tests were conducted to find a value of the Maxwell-Hewitt coefficient that further improved agreement between EIT and GDT. Figure 4.19 shows GDT and EIT reconstructions of the gas volume fraction profile from a preliminary test using both quadratic and quartic profiles for the gamma attenuation and the electrical conductivity. The top plot is a comparison when the electrical conductivity is converted into gas volume fraction using the original Maxwell-Hewitt relation. Agreement is acceptable, although the EIT values are always somewhat greater than the GDT values. The bottom plot is a comparison when Eq. 3.13 is modified in the following manner:

$$\varepsilon_G(r) = 1 - \varepsilon_L(r) = \frac{1 - [\sigma(r)/\sigma_L]}{1 + \frac{3}{5}[\sigma(r)/\sigma_L]} \text{ (modified Maxwell-Hewitt).} \tag{4.4}$$

The value of $3/5$ for the Maxwell-Hewitt coefficient is reasonable, based on the discussion above and the study of the accuracy of the Maxwell-Hewitt relation in Chapter 3. The agreement

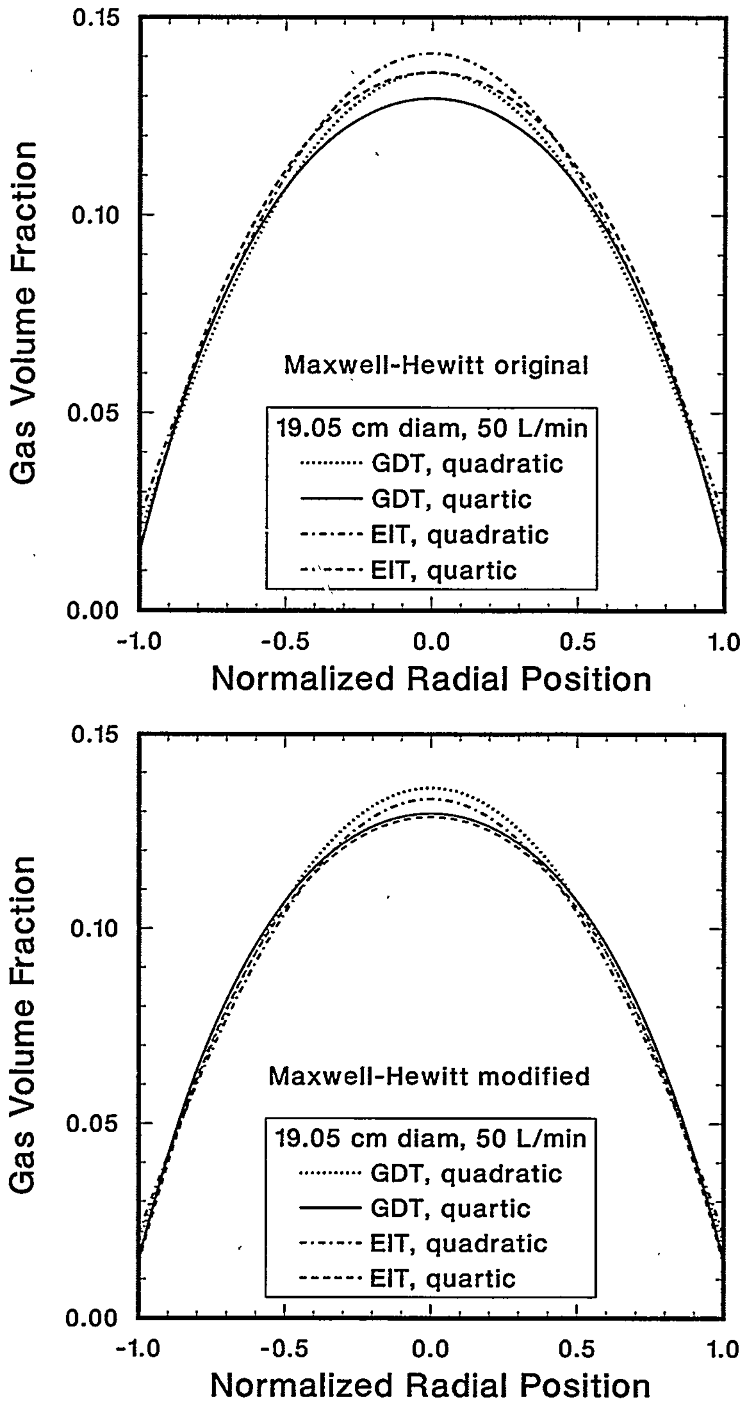


Figure 4.19. GDT and EIT reconstructions using quadratic and quartic profiles. Top, original Maxwell-Hewitt relation; bottom, modified Maxwell-Hewitt relation.

between GDT and EIT is substantially improved with the application of Eq. 4.4 to EIT reconstruction. Although no systematic study has yet been performed, spot checks on several other cases appear to indicate a similar improvement between GDT and EIT gas volume fraction profiles. This will be a topic of future EIT investigations.

4.4. Conclusions

The Sandia/Michigan EIT system and reconstruction algorithm have been validated for the quantitative measurement of solid volume fractions in solid-liquid flows and gas volume fractions and radial profiles in gas-liquid flows. EIT was used to measure solid volume fractions of up to 0.05, with excellent agreement between EIT results and the nominal values determined from the known mass of solids in each flow. Gas-liquid bubble-column flows were measured simultaneously with EIT and GDT; for cross-sectionally-averaged gas volume fractions up to 0.15 the average values and radial profiles from EIT and GDT agreed to within 0.01, despite large radial variations across the column.

As demonstrated through the "phantom" validations in Chapter 3 and the two-phase tests in this chapter, the EIT hardware and reconstruction algorithm have the ability to reproduce parabolic gas volume fraction distributions and the size and location of an insulating cylinder centered off the domain axis. With the recent addition of quartic conductivity profiles to the reconstruction library, the EIT system can also be applied to the study of the effects of solids on vertical churn-turbulent column flows. Two-phase tests employing both GDT and EIT indicate that the quartic terms in both methods have about the same importance. A slight modification of the Maxwell-Hewitt relation, also suggested by the results, appears to improve agreement between GDT and EIT.

In the next chapter, EIT and GDT are combined to measure properties of gas-liquid-solid flows in the bubble column in order to examine the effect of the solid phase on hydrodynamic behavior, particularly its effect on the distribution of the gas and liquid phases. These studies employ several different materials for the solid phase, all of which have conductive properties similar to air so that EIT can detect both the solid and gas phases. The density of each solid phase is significantly different from air, however, so that GDT can provide the information needed to determine the radial distributions of all three phases. The solid volume fraction in these three-phase tests is chosen to be well above 0.01, the uncertainty in EIT gas volume fractions based on the difference between GDT and EIT results in two-phase tests.

5. Investigation of Three-Phase Flows with EIT

With the electrical-impedance tomography system validated for quantitative measurements of two-phase flows, EIT and GDT were used in conjunction to make measurements of average and spatially resolved material distributions in a three-phase vertical bubble-column flow. This investigation served two purposes: (1) to further determine the usefulness of EIT in industrial multiphase flows, and (2) to investigate the effect of solids on slurry bubble-column reactor hydrodynamics and provide information which might be useful in the design and scaleup of such reactors (George *et al.*, 1999b).

For the two techniques to be used together, each phase in the gas-solid-liquid flow must have a unique combination of conductivity and attenuation properties. This study employed dry air as the gas phase and water with sodium nitrate as the liquid phase. For the solid phase, both polystyrene and glass were chosen. Polystyrene is an electrical insulator like air and has an attenuation coefficient similar to water, so that EIT is influenced by both the solid and gas phases but GDT is primarily sensitive to the gas alone. This marked difference in sensitivity of the two methods permits a first-order approximation, if desired, in reconstructions: the gas volume fraction profile from GDT may be subtracted from the insulating phase profile determined by EIT, yielding the solid and liquid phase distributions directly. Glass is also an insulator like air, but has an attenuation coefficient significantly different from both air and water, so that the first-order approximation does not hold in this case. Instead, the gamma attenuation formulas and the Maxwell-Hewitt relation must be solved as a set of simultaneous equations to reconstruct the radial volume fraction profiles of all three phases. This more accurate approach is used in all reconstructions presented in this chapter.

5.1. Experimental Setup

5.1.1. Testbed Setup

The three-phase experiments were conducted in the same Lexan bubble column used for the air-water tests described in Chapter 4. In each three-phase experiment, the column was initially filled with water and solids to a depth H_{col} of 144.8 cm, for a height-to-diameter ratio of $H_{col} / D_{col} = 7.6$. The 16 strip electrodes were centered on a plane L = 82.0 cm above the base ($L / D_{col} = 4.3$), again placing the region of EIT sensitivity completely under water. GDT scans were taken on two source-detector planes, located at L = 70.6 and 93.4 cm above the column base. These positions above and below the midplane of the EIT electrode ring were chosen to prevent obstruction by the electrodes, the cables exterior to the electrode ring, and the O-ring

Table 5.1. Properties of the materials used in the three-phase experiments.

material	ρ (g/cm^3)	μ (cm^{-1})	σ (μS/cm)	ω (μF/cm)
polystyrene	1.04	0.0866	$< 10^{-10}$	2.3×10^{-7}
glass	2.41	0.209	5.9×10^{-11}	$\approx 5\times10^{-7}$
water/NaNO$_3$	0.997	0.0856	242 – 432	$\approx 7\times10^{-6}$
air	0.00106	0.0000819	$\approx 10^{-10}$	8.86×10^{-8}

collars that joined the electrode ring to the remainder of the column. These locations also allowed verification that fully developed flow had been achieved at the measurement region.

The ring sparger used in the two-phase experiments (Figure 4.13) was again used to introduce dry air into the flow and to loft the solids from the column floor. To improve the lofting efficiency of the sparger for the denser glass spheres, the sparger was lowered (after the first experiments with polystyrene) to a position 6 cm above the base of the column. All tests were conducted at ambient pressure. The three cartridge heaters along the height of the column were again used to counteract evaporative cooling and maintain a constant temperature. The liquid temperature was held constant to within ±0.4ºC during experiments, which limited conductivity variations induced by temperature to ±0.6%.

5.1.2. Material Properties

The material properties of the phases used in these experiments are presented in Table 5.1. Deionized water was again used in the liquid phase, with controlled amounts of saturated sodium nitrate solution added to increase conductivity well above the lower limit of 200 μS/cm (Eq. 3.5) and provide a useful dynamic range of measured voltages. The initial DC conductivity of the aqueous sodium nitrate solution, measured before the first test at each solids loading, was $\sigma_L = 261\pm19$ μS/cm. As will be explained in Section 5.1.3, the conductivity consistently increased over the course of experiments as turbulence removed more coating material from the solids. The attenuation coefficient of glass in Table 5.1 was measured previously with the GDT system (Shollenberger *et al.*, 1997b), while the attenuation coefficients of polystyrene, air and water were obtained from the literature (Lamarsh, 1983; Thoraeus, 1965).

To determine whether the density and the size distribution of the solid particles can affect the behavior of the other phases, experiments were performed with a total of four different types of particles. The particles were chosen so that the effects of size and density could be observed separately. In the first series of three-phase tests, the polystyrene spheres described in the previous chapter (400 μm nominal diameter, 1.04 g/cm^3) were again used as the solid phase. Other particles used were polystyrene spheres of a smaller diameter (200 μm nominal) than the original polystyrene particles, and glass beads with smaller diameters (80 and 200 μm nominal) and a higher density (2.41 g/cm^3). The solid densities were measured by a volumetric method.

Photographs of each kind of particles used in this study and plots of their size distributions are shown in Figures 5.1 through 5.4. From Figure 5.4, it is apparent that some of

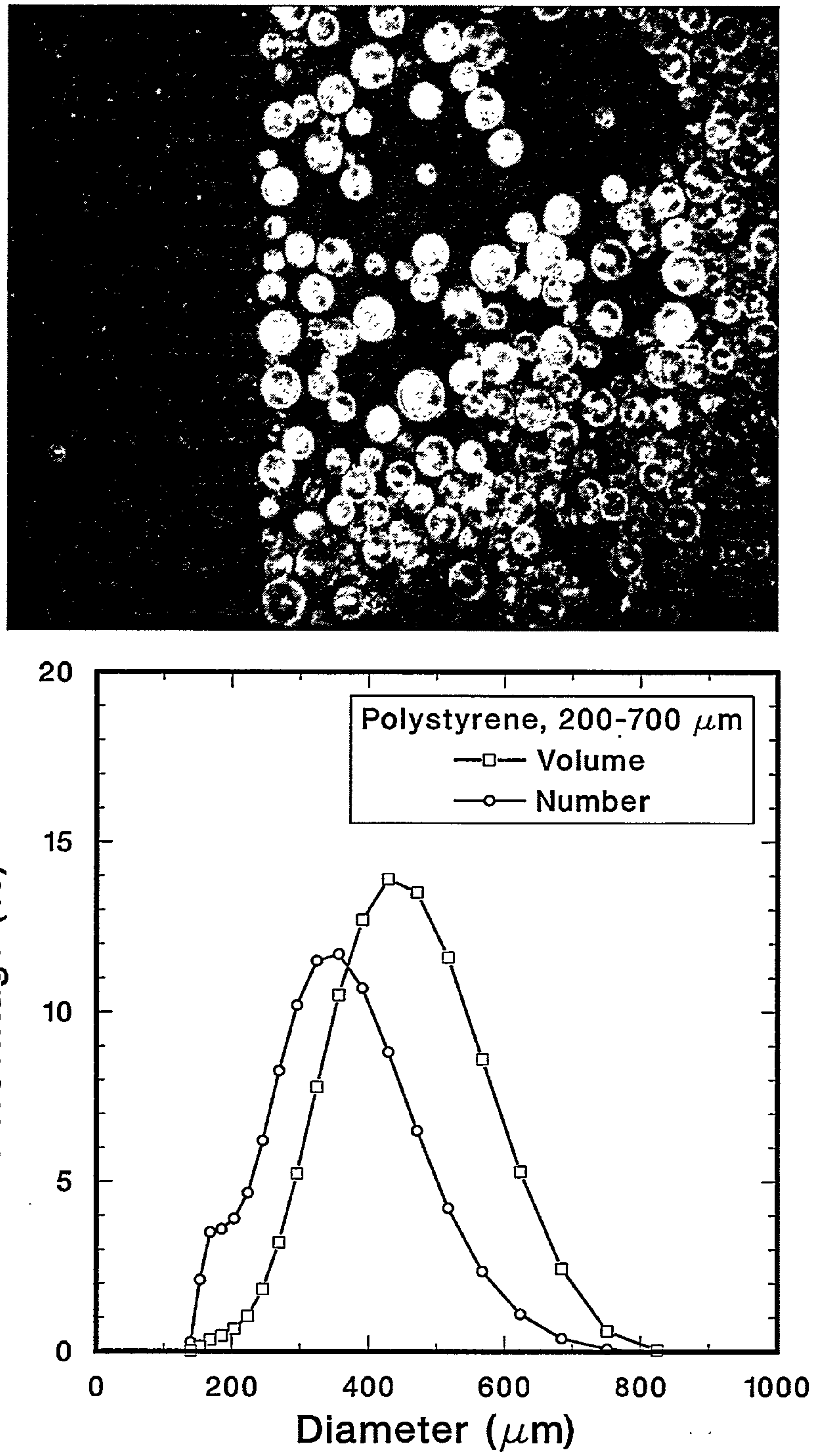


Figure 5.1. Photograph and size distribution of large polystyrene beads (diameter 200 – 700 μm, density 1.04 g/cm^3). The smallest ruler division in the photo is 500 μm.

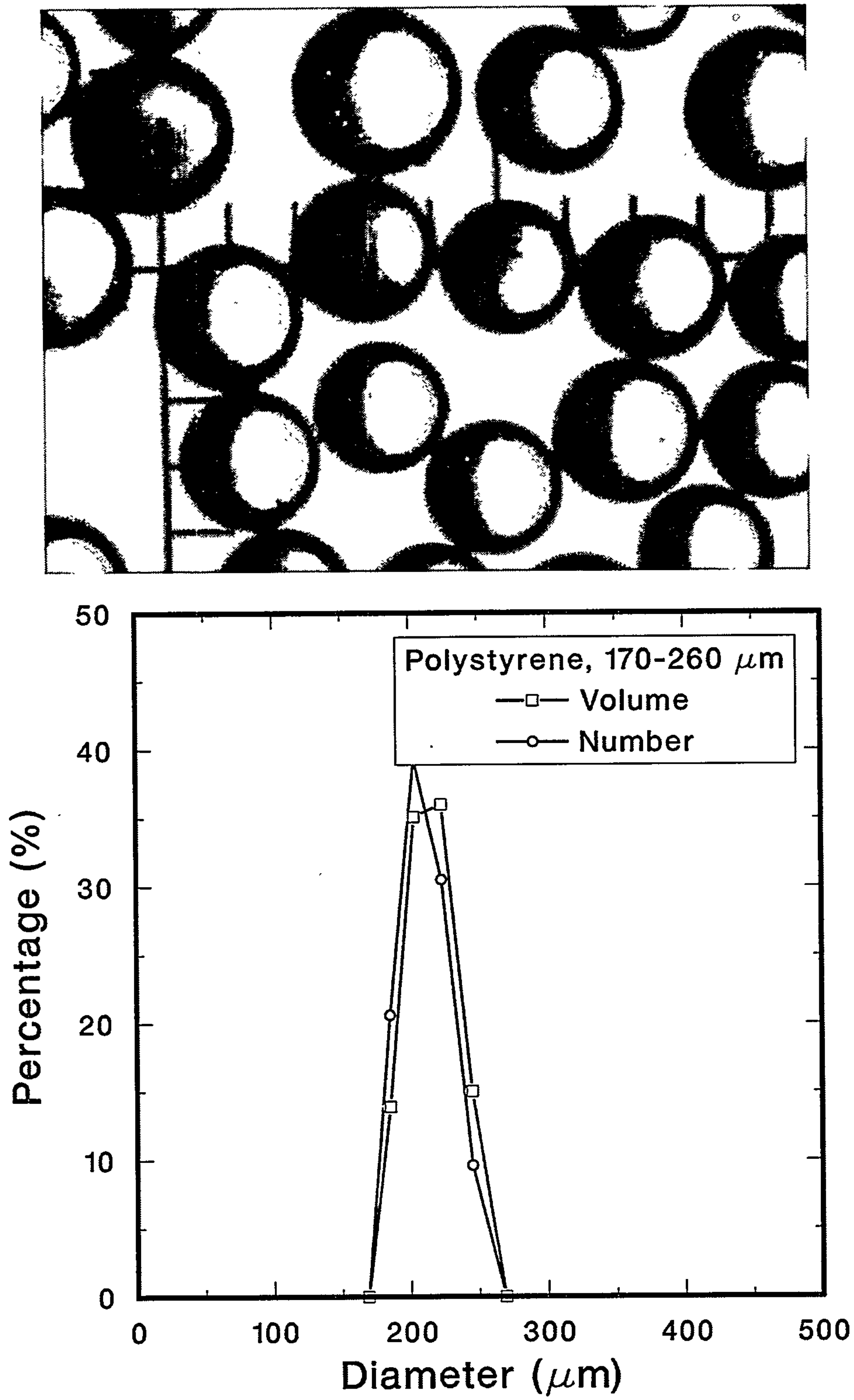


Figure 5.2. Photograph and size distribution of medium polystyrene beads (diameter 170 – 260 μm, density 1.04 g/cm^3). The smallest ruler division in the photo is 100 μm.

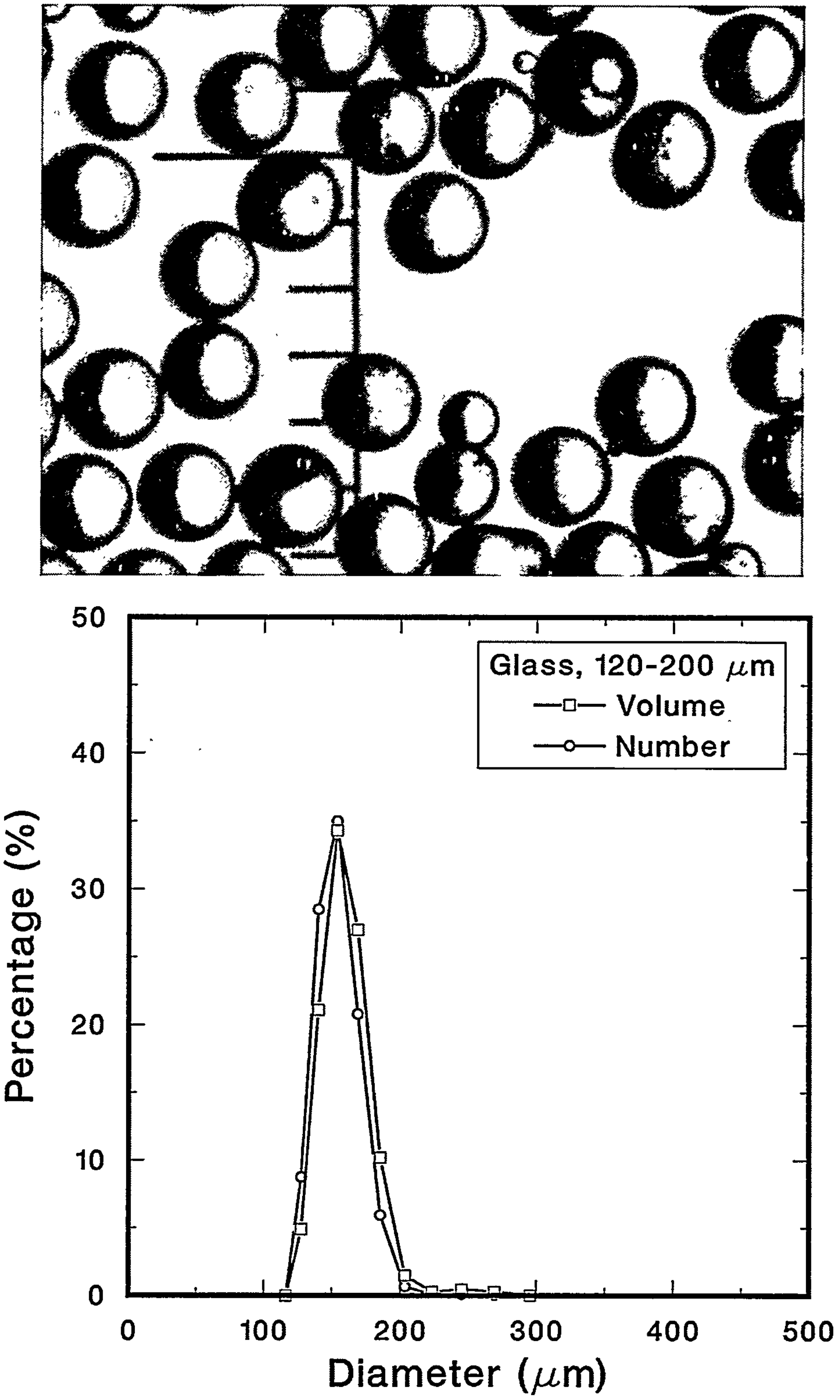


Figure 5.3. Photograph and size distribution of medium glass beads (diameter 120 – 200 μm, density 2.41 g/cm^3). The smallest ruler division is the photo is 100 μm.

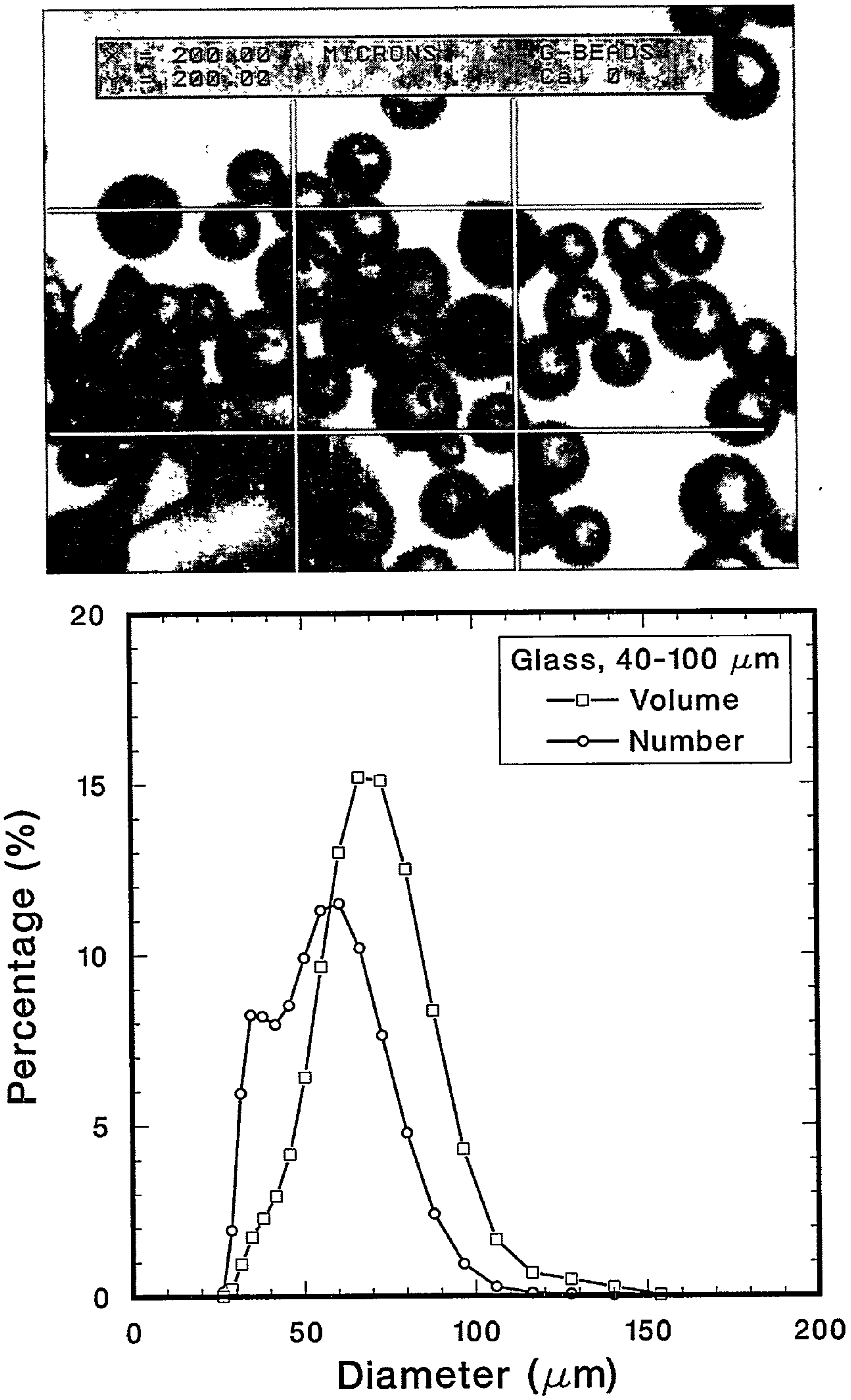


Figure 5.4. Photograph and size distribution of small glass beads (diameter 40 – 100 μm, density 2.41 g/cm^3). The central square in the photo is 200 μm on a side.

the "small" glass particles are decidedly aspherical; most are globular, but some elongated particles and shards are present. By comparison, the other particles are almost uniformly spherical. The graphs in each figure show the size distribution of each particle type, both in terms of number percentage and volume percentage, as determined by R. R. Lagasse using a Mie-scattering technique (Lagasse, 1999). The size distribution of the "large" polystyrene particles is seen to be broadest by far, with 98% of the volume of solids consisting of particles with diameters ranging from 200 to 700 µm. The "small" glass beads, the second type of particles to be investigated, are seen to have diameters generally in the range of 40–100 µm. The "medium" glass and polystyrene particles were both identified by their respective manufacturers as nominally 200 µm in diameter, but had notably different distributions: 120–200 µm for the glass spheres, 170–260 µm for the polystyrene.

5.1.3. Experimental Procedure

Before the first set of three-phase experiments, which involved polystyrene, the particles were washed and rinsed repeatedly to minimize the amount of soluble contamination carried with the solids into the liquid. Despite these efforts, noticeable amounts of opaque coating material were transferred from the polystyrene into the liquid by the turbulent flow. The opaque coating acted as a surfactant, as evidenced by foam at the top of the liquid column during gas flow. Electrolytes present in the coating material also altered the liquid conductivity over the course of experiments. To compensate for this effect of the contaminants, the standard procedure introduced in Chapter 4 of using the EIT system itself to measure the baseline liquid conductivity between tests was followed here as well.

The glass spheres also shed opaque material into the liquid during experiments. Although the glass particles were not washed before tests, the conductive effects of their contaminant were much less than that of the polystyrene, suggesting that a different coating was involved or perhaps that the "contaminant" was very small shards of broken particles. Again, baseline EIT conductivity measurements taken between flow conditions were used to compensate.

Table 5.2 lists the combinations of solid materials, solids loadings and gas volumetric flow rates at which measurements of phase distributions were made. Experiments were broken into subsets, each involving one nominal solid volume fraction $\bar{\varepsilon}_S^{NOM}$ defined as the ratio of the volume of solids to the combined reference volume of solids and liquid. Solid loadings up to $\bar{\varepsilon}_S^{NOM}$ = 0.30 were included in the tests with large polystyrene particles; loadings for the other solids were limited to $\bar{\varepsilon}_S^{NOM}$ = 0.15, the highest loading of glass particles that could be completely lofted at these gas flow rates. To obtain the desired solids loading for each subset, the total mass of required particles was computed from the reference volume (41.26 L, based upon a depth of H_{col} = 144.8 cm) and the known density of the solid material. Any amount of solids already present in the column from previous experiments was subtracted from this total to find the mass to be added. This mass was weighed out, introduced into the column, and allowed to settle to the bottom. After settling of the solids, liquid was drained from the column to return

Table 5.2. Three-phase experimental conditions analyzed with the EIT/GDT system.

Particle		Q_G (L/min)						
type	$\bar{\varepsilon}_S^{NOM}$	50	100	150	200	300	400	500
80 µm glass	0			X	X	X	X	
	5	a	a	X	X	X	X	
	10	a	a	X	X	X	X	
	15	a	a	a	X	X	X	
200 µm glass	0					X	X	X
	5	a	a	a	a	X	X	X
	10	a	a	a	a	X	X	X
	15	a	a	a	a	X	X	X
200 µm polystyrene	0	X	X	X	X	X	X	
	5	X	X	X	X	X	X	
	10	X	X	X	X	X	X	
	15	f	X	X	X	X	X	
400 µm polystyrene	0	X	X	X	X			
	5	X	X	X	X			
	10	X	X	X	X			
	22	X	X	X	X			
	30	f	X	X	X			

X = condition successfully measured
a = air flow rate inadequate to loft all solids
f = head of foam from surfactant in particle coating expanded to top of column

the total solid-liquid volume to the reference value, and an EIT scan was performed to measure the baseline liquid conductivity.

For each solids loading, air was introduced through the ring sparger at the chosen volumetric flow rate, Q_G, and the cartridge heaters were turned on during the flow to compensate for evaporative cooling. Over the course of all experiments, Q_G ranged from 50 to 500 L/min, corresponding to superficial gas velocities, U_G, of 2.9 to 29.2 cm/s. The flows were all churn-turbulent and opaque, even at the lowest solids loading and lowest gas flow rate (see Figure 5.5). At higher gas flow rates, non-spherical voids were occasionally visible at the column walls.

In each successful test, the downward flow of air from the sparger was sufficient to loft all the solids from the bottom and produce a three-phase flow throughout the column. No measurements were made for certain combinations of gas flow rate and solid volume fraction, marked "a" in Table 5.2, because an appreciable amount of solid particles remained at the bottom of the column under these conditions. Although these particles could be lofted by temporarily increasing the air flow rate, they settled to the bottom of the column when the air flow rate was returned to the prescribed value. It should be noted that cases of "0%" nominal

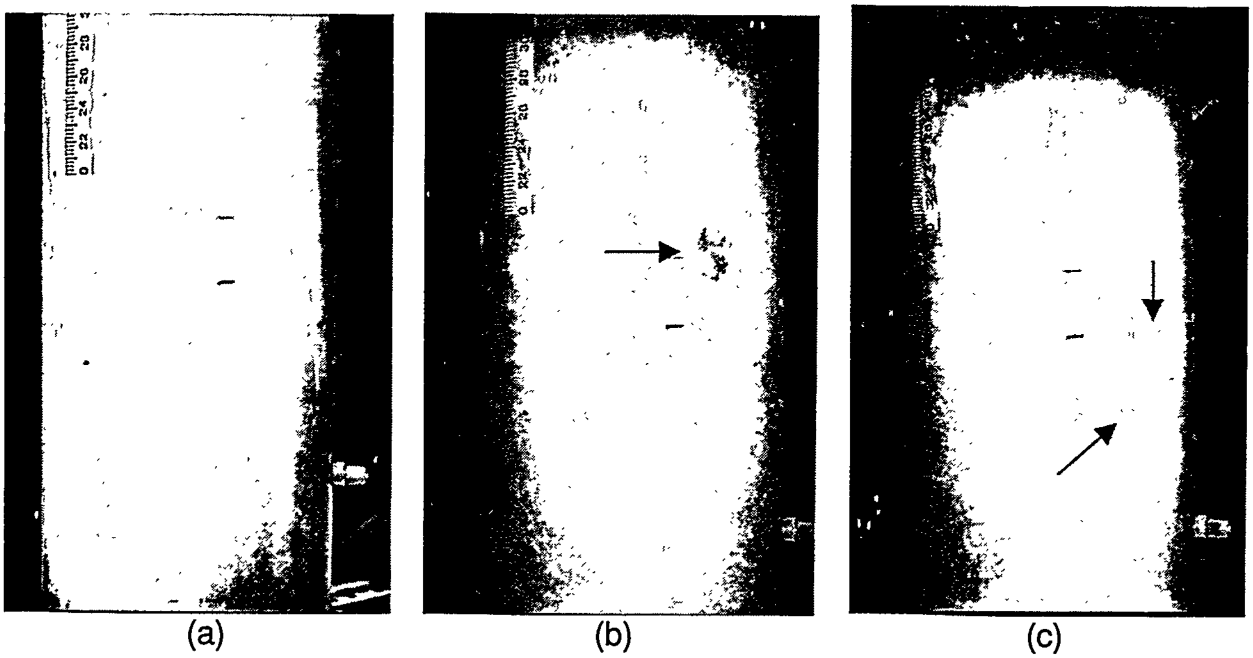


Figure 5.5. Photographs of flow conditions in three-phase, polystyrene-air-water flows in the transparent bubble column. (a) $\bar{\varepsilon}_S^{NOM}$ = 0.05, Q_G = 50 L/min; (b) $\bar{\varepsilon}_S^{NOM}$ = 0.10, Q_G = 200 L/min; (c) $\bar{\varepsilon}_S^{NOM}$ = 0.22, Q_G = 100 L/min. Note the air voids at the walls (marked by arrows) at the higher gas flow rates, indicating a churn-turbulent gas flow regime. The solid phase is 400 μm polystyrene particles; the vertical scale is in centimeters.

solid volume fraction employ water in which solids previously added were removed for the test, leaving coating material in the water. This was necessary to ensure that the surface tension was the same for all experiments, both with and without solids present. Table 5.2 also notes flow conditions which were attempted but not completed because the surfactant foam above the solid-liquid slurry expanded and threatened to overflow the top of the column.

As shown in Table 5.1 and discussed in the introduction to this chapter, the polystyrene and water attenuation coefficients for ^{137}Cs gamma rays are nearly the same. Since gamma attenuation and density are roughly proportional for the lighter elements, it is not unexpected that polystyrene is nearly neutrally buoyant in water, and that it is easily lofted and mixed with the other phases by the downward air jets from the sparger. Glass, however, has both a density and gamma attenuation coefficient significantly higher than those of water. Because of the higher density, some air flow rates that easily lofted an entire polystyrene loading were inadequate to loft the same volume of negatively buoyant glass. As a result, minimum and maximum flow rates for three-phase tests involving glass were higher than those in polystyrene tests. Lofting efficiency was also influenced to some extent by particle diameter; tests with 200 μm glass spheres required higher minimum flow rates than tests with 80 μm glass spheres.

Profile measurements at each flow condition consisted of a GDT scan at the lower scan plane, followed by an EIT scan, followed by a second GDT scan at the upper plane. After the profile measurements were completed at each condition, gas flow was terminated, and the solids were allowed to settle beyond the EIT measurement region to the bottom of the column. This settling process required 15 to 60 minutes, depending upon the amount and type of solids present. Following the settling period, EIT was applied again to measure changes in the baseline conductivity of the liquid. The values of liquid conductivity measured before and after each set of flow conditions were averaged and used to compute volume fraction profiles under those conditions. In the experiments involving polystyrene, a fraction of the particles floated to the free surface of the liquid during the post-test settling period. This layer of solids was not disturbed during the settling period or the EIT baseline measurements, but was forced back into the liquid before the next test by a single Taylor bubble created from a momentary air flow.

For each flow condition, the information from the pair of GDT scans was averaged and combined with the EIT data to determine the radial volume fraction profiles at the EIT electrode plane. As shown in the results later in this chapter, the two GDT scans did not differ from each other by more than 0.01 in ε_G for all cases tested. For the first experiments with 400-μm polystyrene particles, the original 486 PC was used to operate the EIT package. Twenty-five EIT projection sets were acquired in fast mode during each flow condition over a period of about 20 seconds and averaged to yield voltage data for reconstructions. This averaging was required in order to combine the EIT results with the average of the two GDT scans. After the first polystyrene experiments, several changes were made to the EIT system, including the replacement of the 486 with a Pentium PC and the addition of the quartic reconstruction algorithm. Uncertainty analyses of the quartic algorithm (described in Section 3.3.4) mandated that a total of 100 projection sets be collected and averaged for the remaining tests, while difficulties with memory management on the Pentium restricted data acquisition to slow mode. As a result, EIT reconstructions for the remaining three-phase tests represented averages over a period of 2 ½ minutes, instead of the twenty-second period originally required. By comparison, the GDT scans involved data acquisitions over 60 seconds along each of 19 beam paths, so that their reconstructions were averages over a period of 23 minutes. However, since the measurements of two-phase flows in Chapter 4 demonstrated that any differences in temporal averaging by GDT and EIT do not significantly affect results, it is believed that combining these data averaged over different periods to determine the three-phase distributions does not introduce unwarranted error.

5.2. Calculation of Phase Volume Fraction Profiles

The data from the GDT and EIT scans may be combined for three-phase flows by modifying the basic equations used with each method to account for the presence of all three phases. In the case of a purely two-phase, gas-liquid flow, the two-phase continuity equation and the Maxwell-Hewitt relation are used with conductivity distributions from the EIT reconstruction code to determine liquid and gas volume fraction profiles. In the more general three-phase case, the quantity $(1-\varepsilon_L)$ in Eq. 3.13 is equivalent to the total insulating phase fraction ε_I, which includes both the solid and gas phases in this study. The three-phase continuity equation is therefore

$$\varepsilon_S + \varepsilon_L + \varepsilon_G = \varepsilon_I + \varepsilon_L = 1 . \tag{5.1}$$

The solid and gas volume fractions must be separated from ε_I using GDT attenuation information.

This section presents the relations used in this study to determine the radial variations of ε_G, ε_L and ε_S from the radial variations of the normalized gamma attenuation f_μ and the normalized electrical conductivity $\sigma(r)/\sigma_L$. Each term in Eq. 5.1 is implicitly a function of position in the flow. In this investigation only radial variations in the phase distributions are considered, since the reconstruction codes EITAXI and GDTAXI assume invariance in θ and z in the conductivity and attenuation distributions. The derivations will therefore explicitly show dependencies only upon *r*.

In either the two-phase or three-phase case, the liquid profile may be determined directly from the reconstructed conductivity profile by solving the Maxwell-Hewitt relation for ε_L. In the original three-dimensional formulation, for example, Eq. 3.13 becomes

$$\varepsilon_L(r) = \frac{3\sigma(r)}{2\sigma_L + \sigma(r)}. \tag{5.2}$$

The GDT reconstruction algorithm, initially developed for use with gas-liquid flows, produces a normalized radial attenuation distribution of the form

$$f_\mu(r) = \frac{\mu(r) - \mu_G}{\mu_L - \mu_G}. \tag{5.3}$$

The measured local value of $\mu(r)$ is a weighted average of the attenuation of all phases over a differential volume at *r*, so that in the three-phase case,

$$\mu(r) = \mu_S\varepsilon_S(r) + \mu_L\varepsilon_L(r) + \mu_G\varepsilon_G(r). \tag{5.4}$$

Combining Eqs. 5.1 through 5.4 allows the distributions of all three phases in the flow to be solved for. The only pieces of information required to compute $\varepsilon_L(r)$, $\varepsilon_G(r)$ and $\varepsilon_S(r)$ are $f_\mu(r)$, determined from GDT data; $\sigma(r)$, determined from EIT data; and four physical properties known *a priori*: the phase attenuation coefficients μ_S, μ_L, and μ_G and the liquid conductivity σ_L.

While the approach of Eqs. 5.1 through 5.4 is a general one, the equation for $\varepsilon_L(r)$ is subject to change, specifically with the assumptions inherent in the Maxwell-Hewitt relation. Recall that the relation was successfully modified in Chapter 4 (see Eq. 4.4) to improve agreement between GDT and EIT in gas-liquid flows, based on the assumption of slight vertical motion of the bubbles during EIT measurements. The Maxwell-Hewitt relation was also shown in Chapter 3 to be acceptably accurate for volume fractions of bimodal bubble sizes, although insulators of a single size are assumed in its derivation. With these facts in mind, empirical adjustments were made to the original Maxwell-Hewitt relation with the intent of extending its applicability to the three-phase flows of this chapter and improving reconstruction accuracy.

The changes involve two assumptions related to these three-phase flows. First, the solid and the gas are treated by EIT as a single insulating phase. Second, a general Maxwell-Hewitt relation is assumed that involves a coefficient, α, dependent upon the domain geometry: $\alpha = 1$ for two-dimensional cylindrical inclusions or $\alpha = ½$ for three-dimensional spherical bubbles. Under these assumptions, Eqs. 3.13 and 3.14 may be generalized to the following equation for normalized conductivity:

$$\sigma/\sigma_L = E[\varepsilon_G + \varepsilon_S, \alpha]. \tag{5.5}$$

Here, σ, ε_G and ε_S are implicitly functions of r. The form of the function E is defined by the Maxwell-Hewitt relation as

$$E(\xi, \alpha) = \frac{1-\xi}{1+\alpha\xi}. \tag{5.6}$$

Note also that

$$E[E(\xi, \alpha), \alpha] = \xi. \tag{5.7}$$

As discussed in Section 3.3.3 and George *et al.* (1999b), the first assumption can be modified to represent the bimodal distribution of "particle" sizes (*i.e.*, bubbles and solid particles) by treating the three-phase flow as gas bubbles within a liquid-solid mixture. In this case the modified Maxwell-Hewitt relation is applied recursively, first to the solid particles in the liquid phase, then to the gas in the liquid-solid mixture. As suggested in Chapter 4, it may also be appropriate to empirically modify the value of α from $\frac{1}{2}$ to $\frac{3}{5}$ for time-averaged three-dimensional reconstructions. The following relation then replaces Equation 5.5:

$$\sigma/\sigma_L = E(\varepsilon_G, \tfrac{3}{5}) E[\varepsilon_S/(\varepsilon_S + \varepsilon_L), \tfrac{3}{5}]. \tag{5.8}$$

This modified relation is used in all three-phase analyses in the remainder of this chapter.

The approach outlined in this section is incorporated into the reconstruction code `GDTEIT`, listed in Appendix J. In the reconstructions, the conductivity distributions determined by `EITAXI` and input to `GDTEIT` were modeled as a quartic function in r, symmetric about the column axis. Similarly, chord-averaged attenuation coefficients computed from GDT data were fit to a fourth-order polynomial by `GDTAXI` before the Abel transform was taken so that the distribution data input to `GDTEIT` from both methods were of the same order.

5.3. Results and Discussion

5.3.1. Three-Phase Results and Observed Trends

The complete set of phase volume fraction profiles for all solids loadings and gas flow rates may be found in Appendix K. Relevant graphs from the appendix are repeated in Figures 5.6 and 5.7 to illustrate points of discussion later in this section. Several general observations can be made about the results. First, as expected, the gas volume fraction increases as the superficial gas velocity is increased. The gas volume fraction profiles behave like those found in gas-liquid flows, increasing with U_G preferentially on the column axis. Generally, the profiles $\varepsilon_S(r)$ and $\varepsilon_L(r)$ of the other phases decrease with rising U_G, as the volume of gas in the column increases and the constant amounts of solid and liquid occupy a smaller fraction of the total volume. Second, the combined reconstruction algorithm produces a small but nonzero solid-volume-fraction profile even when no solid particles have been added to the column. Based on the individual uncertainties of the separate EIT and GDT methods, an uncertainty around ±0.02 for volume-fraction profiles has been assigned to the combined algorithm. The fact that the computed solids profiles are generally less than 0.01 when no solids are present is in harmony with this combined uncertainty. Third, the solid-volume-fraction profiles have a variety of shapes, including concave upward, a central maximum, and a maximum at a radial location between the axis and the wall. In most cases, these variations lie within or close to the indicated uncertainty and should not be considered significant. Possible exceptions are the profiles of 400 μm polystyrene particles at volume fractions of 0.22 and 0.30; these profiles were calculated from 25 EIT projection sets rather than 100, and so are expected to have higher uncertainties.

A significant observation is demonstrated in Figure 5.6, which shows the gas volume fraction profiles for a superficial gas velocity of 8.8 cm/s and four different solid volume fractions of 400 μm polystyrene particles. The gas profile is identical in all four plots to well within the ±0.02 accuracy limit, suggesting that replacement of up to 30 volume percent of the liquid with solids of similar density has little or no effect on the gas distribution. Figure 5.7 shows similar radial volume-fraction profiles for the 200 μm glass beads. The addition of glass particles consistently reduces the gas volume fraction by a small amount, but for solid volume fractions as high as 15%, the decrease is comparable to the experimental uncertainty. Indeed, for all of the particle types and gas flow rates examined, the gas volume fraction radial profiles were unaffected by the addition of solids to within an uncertainty band equal to the experimental uncertainty of ±0.02 in volume fraction.

Domain-averaged phase volume fractions were also computed from the reconstructions. The effect of the presence of solids on average gas volume fraction is quantified in Figures 5.8 through 5.11, which show plots of $\bar{\varepsilon}_G$ for all four particle types. In all cases $\bar{\varepsilon}_G$ is seen to increase monotonically with U_G, as expected. Decreases in average gas volume fraction with increasing solids density (glass versus polystyrene) are strongly suggested by the plots. The

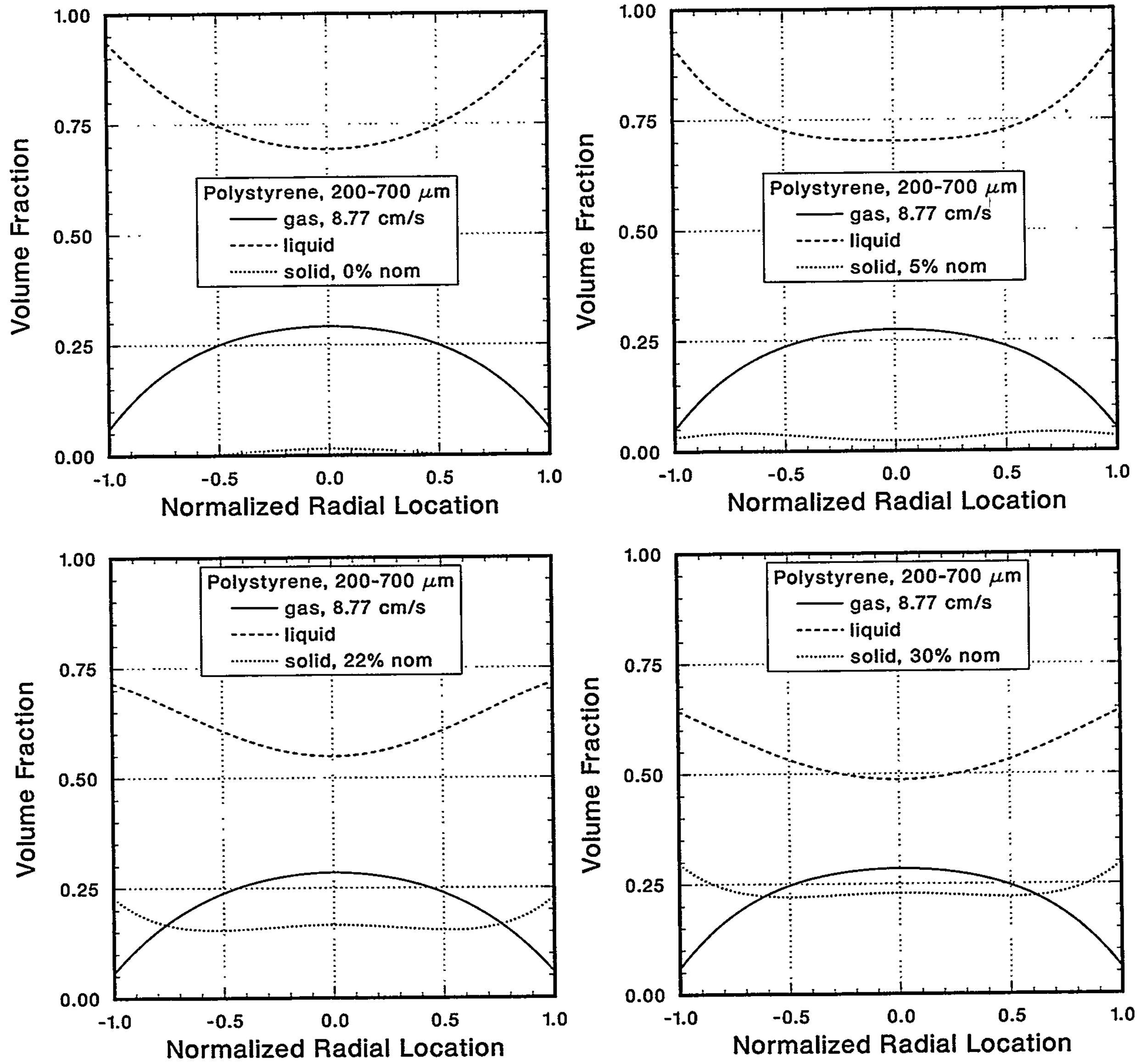


Figure 5.6. Phase volume fraction profiles determined from EIT and GDT reconstructions for a superficial gas velocity of U_G = 8.8 cm/s and a solid phase of 400 μm polystyrene particles. Nominal solid volume fractions $\bar{\varepsilon}_S^{NOM}$ are listed in each plot.

graphs involving glass particles also show a clear decrease in gas volume fraction (by as much as 0.04 in $\bar{\varepsilon}_G$) with increasing solid volume fraction, but no such trend is observed for the polystyrene spheres. As with Table 5.2, a nominal solid volume fraction of 0% in these graphs indicates cases in which particles loaded for previous experiments have been removed from the water, leaving coating material behind.

Figure 5.12 is a compilation of average gas volume fractions for all two-phase and three-phase experiments performed in the Lexan bubble column to date and presented separately in Figures 5.8 through 5.11. For comparison, Figure 5.12 also shows the average gas volume fractions obtained using clean, deionized water and no solids. These "clean" volume fractions

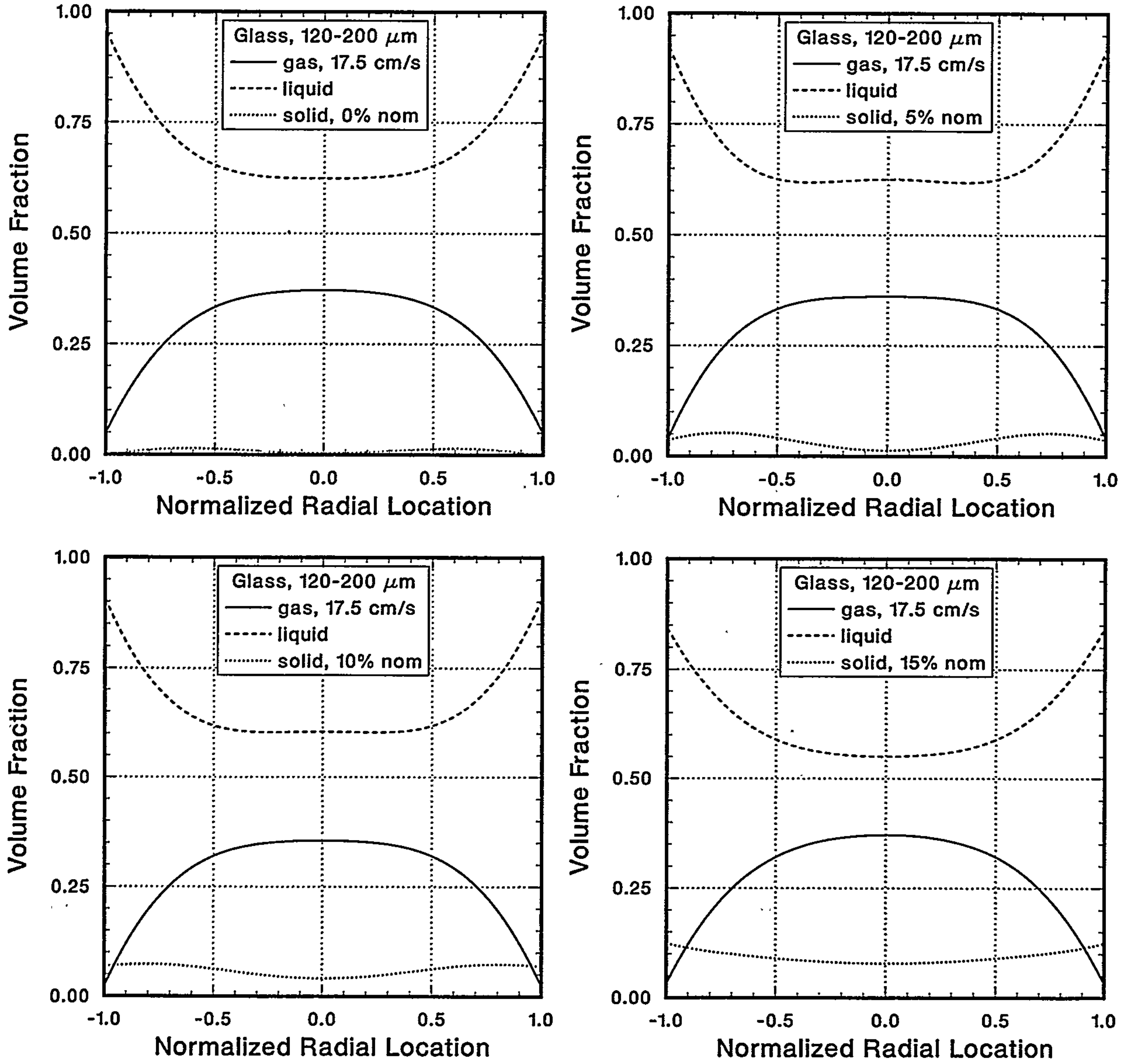


Figure 5.7. Phase volume fraction profiles determined from EIT and GDT reconstructions for a superficial gas velocity of U_G = 17.5 cm/s and a solid phase of 200-µm glass particles. Nominal solid volume fractions $\bar{\varepsilon}_S^{NOM}$ are listed in each plot.

are considerably lower than for the cases in which particles are employed, but all results can be described to within ±0.04 by a curve passing through the middle of the data. As noted earlier, the addition of particles also unavoidably adds surfactants, which reduce surface tension and change the gas volume fraction $\bar{\varepsilon}_G$ by +0.02 to +0.06, substantially more than the change produced by the particles themselves. This indicates that caution should be exercised in determining "three-phase" effects, which may result from spurious surface-tension changes instead of the presence of particles.

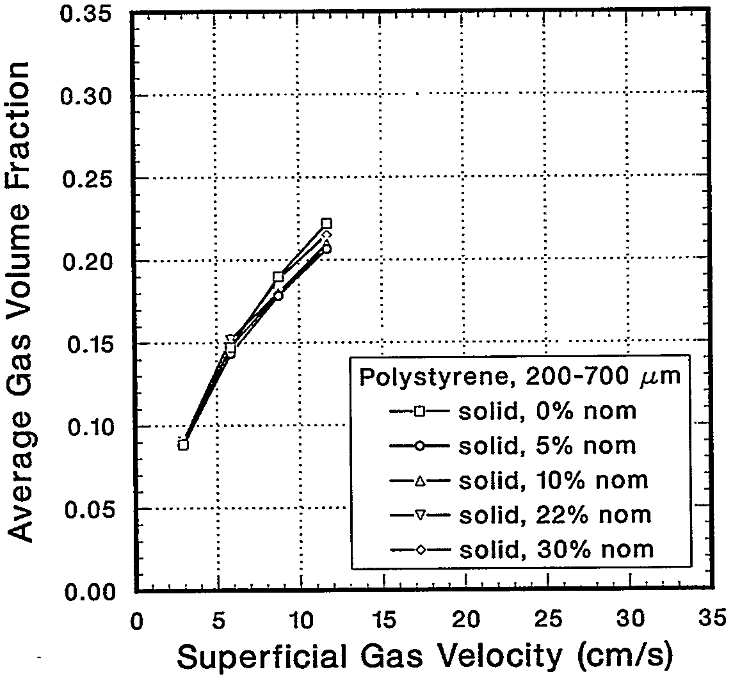


Figure 5.8. Average gas volume fractions for large (400 μm nominal diameter) polystyrene beads.

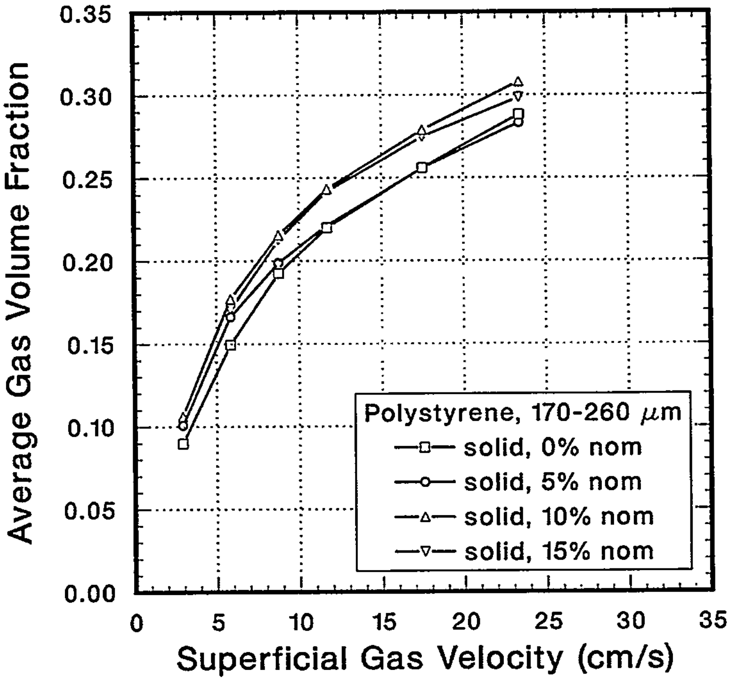


Figure 5.9. Average gas volume fractions for medium (200 μm nominal diameter) polystyrene beads.

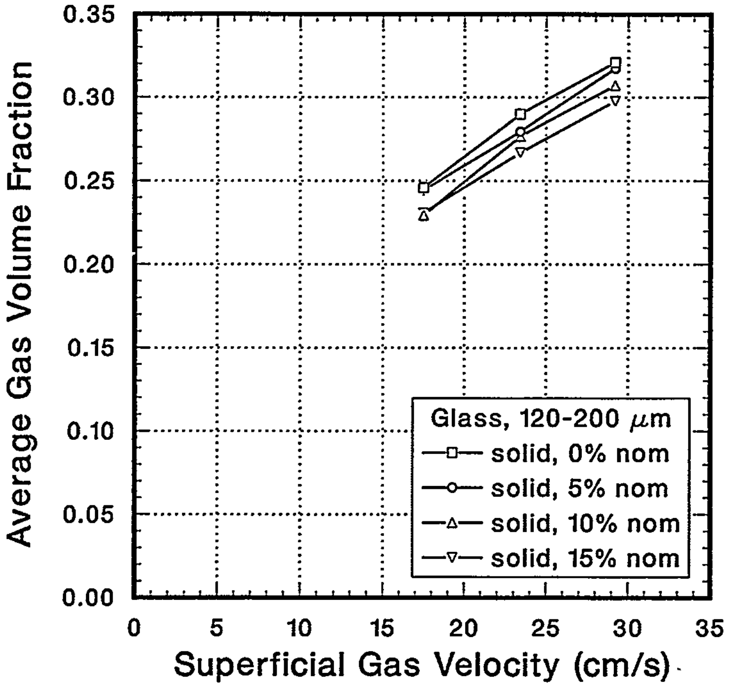


Figure 5.10. Average gas volume fractions for medium (200 μm nominal diameter) glass beads.

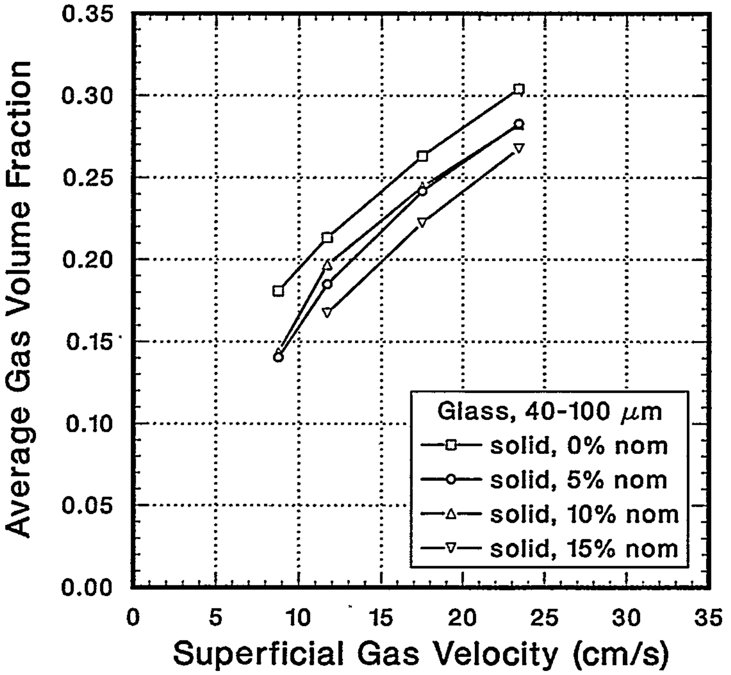


Figure 5.11. Average gas volume fractions for small (80 μm nominal diameter) glass beads.

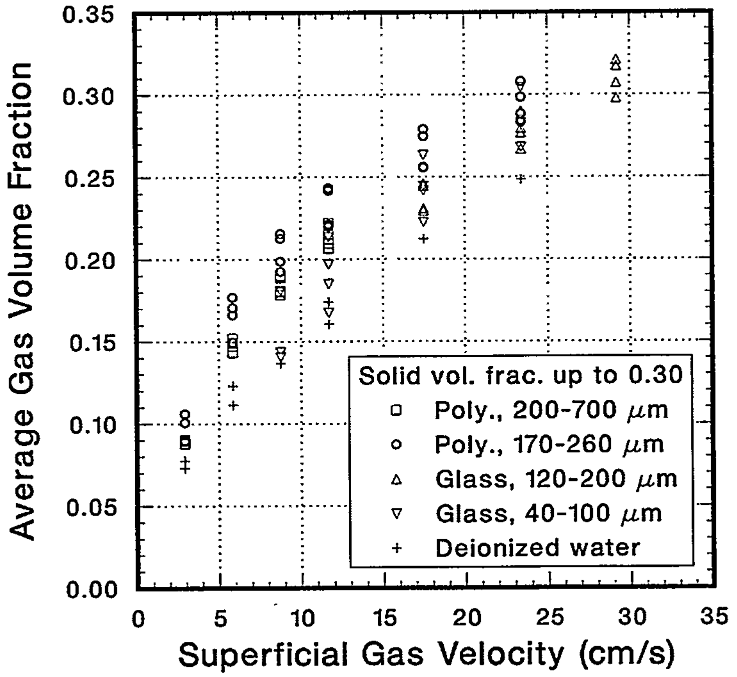


Figure 5.12. Average gas volume fractions for all two-phase and three-phase experiments to date.

As observed earlier, solid phase density appears to influence gas distributions. For glass particles, particularly the 80 μm particles, the gas-volume-fraction profiles decrease slightly in magnitude as the solids loading is increased. This is quantified in Figure 5.13, where the average gas volume fraction is plotted against the nominal solid volume fraction of glass particles for all values of the superficial gas velocity. Despite the uncertainty around ±0.02 in the average values, $\bar{\varepsilon}_G$ clearly decreases with increasing nominal solid volume fraction of glass. The corresponding graph for the less dense polystyrene spheres, Figure 5.14, shows that changes in the gas volume fraction with solid volume fraction are smaller and fall within the experimental uncertainty about the data. While the gas volume fraction appears to increase for $\bar{\varepsilon}_S^{NOM} \leq 0.1$ of the medium polystyrene, $\bar{\varepsilon}_G$ decreases as the solid fraction increases further, and remains within the ±0.02 uncertainty band. Direct comparison of the gas volume fractions for the cases of 200 μm polystyrene and 200 μm glass (at $Q_G = 300$ and 400 L/min) shows that for $\bar{\varepsilon}_S^{NOM} \geq 0.1$, $\bar{\varepsilon}_G$ differs by 0.035 to 0.04, twice the measurement uncertainty. Though not conclusive, these observed effects of solids density and loading are in good agreement with an effective two-phase model for these flows discussed in the next section.

A small diameter effect is noted in that values of $\bar{\varepsilon}_G$ are consistently slightly higher for the "medium" (200 μm) polystyrene spheres than for the "large" (400 μm) polystyrene spheres, but again, the differences are comparable to the uncertainty of ±0.02. No significant diameter effect is observable between the "small" and "medium" glass particles, however. Overall, the maximum change in average gas volume fraction with both particle density and size is ±0.04, which is less important than changes from contaminants and the resulting surface-tension changes evidenced in Figure 5.12.

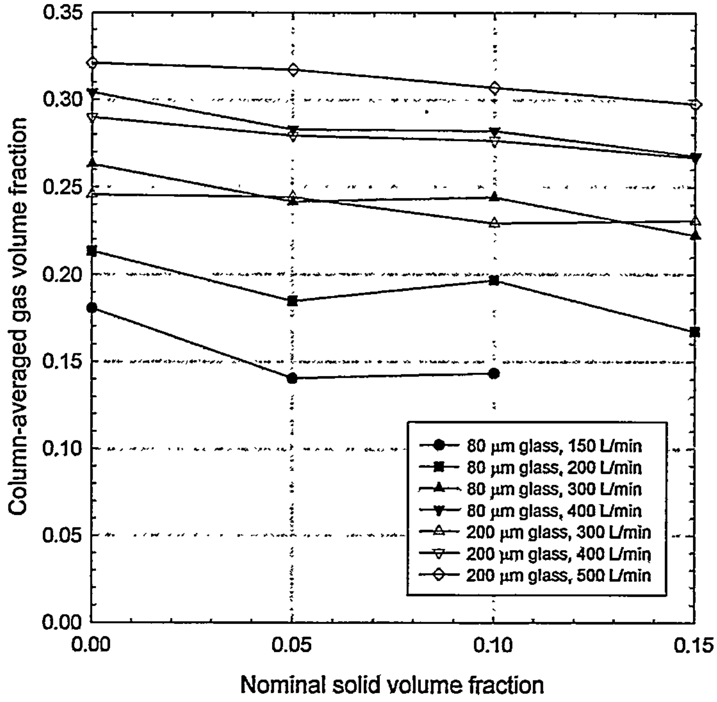


Figure 5.13. Dependence of average gas volume fraction on nominal solid volume fraction for all three-phase experiments involving glass particles.

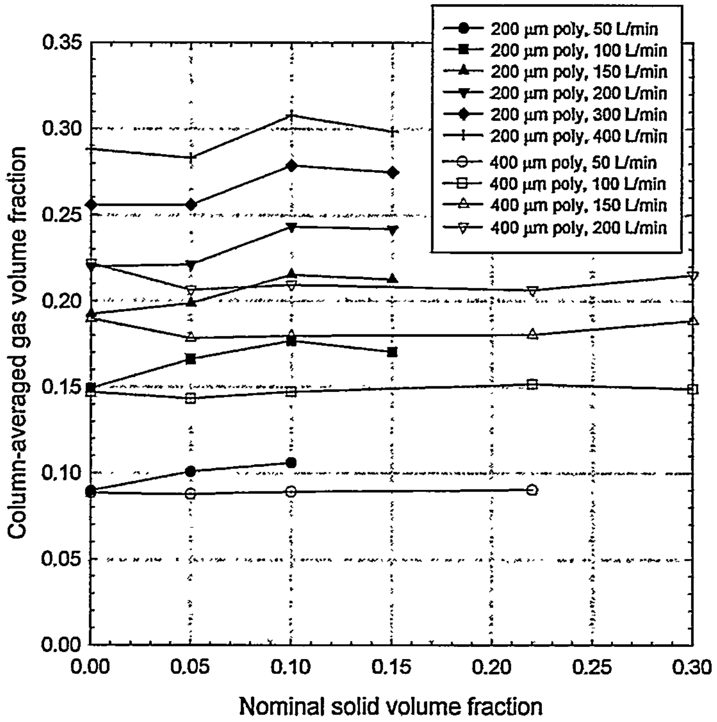


Figure 5.14. Dependence of average gas volume fraction on nominal solid volume fraction for all three-phase experiments involving polystyrene particles.

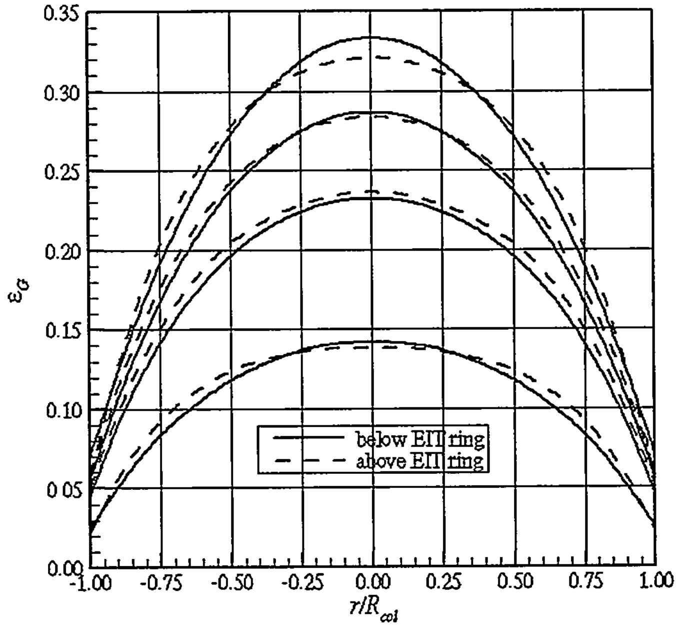


Figure 5.15. Comparison of gas profiles computed from GDT attenuation data taken above and below the EIT electrode ring for $\bar{\varepsilon}_S^{NOM}$ = 0.22. From bottom to top, the curve pairs correspond to superficial gas velocities U_G of 2.9, 5.9, 8.8, and 11.7 cm/s.

5.3.2. Error Analysis and Model Refinement

A brief analysis was performed to quantify the potential error due to vertical averaging. For each flow condition, a single liquid-phase profile was determined from the EIT projection sets, whereas two attenuation profiles were determined with GDT — one above the electrode ring, the other below. The average of the two attenuation profiles was combined with the liquid profile to obtain phase volume fraction profiles at the electrode plane. Figure 5.15 presents preliminary gas profiles computed separately from each attenuation profile for the case of 400 μm polystyrene spheres, $\bar{\varepsilon}_S^{NOM}$ = 0.22 and all gas flow rates. The gas profile pairs are identical to within 0.01, the accuracy of EIT determined during validation tests. This agreement suggests that at this solids loading, the flow was approximately fully developed by the time it reached the measurement region. Comparisons were also made for other randomly chosen polystyrene loadings. For $\bar{\varepsilon}_S^{NOM}$ < 0.22, the difference in profiles is smaller, while for $\bar{\varepsilon}_S^{NOM}$ = 0.30, the difference approaches 0.04 in ε_G. This variation with vertical location may indicate that the flow was still developing at the measurement region, although inadequacies in the original Maxwell-Hewitt relation discussed below are more likely explanations for the difference.

An improvement to the constitutive model was made after comparison of the data to a simple phase distribution model. Because of the similarity in two-phase and three-phase experimental results, it is appealing to attempt to interpret gas-liquid-solid experiments in terms of gas-liquid experiments in which the "liquid" (more properly, the "dense phase") is taken to have effective properties corresponding to the actual liquid-solid mixture. Such models often assume that the solid phase is dispersed homogeneously and entirely within the liquid phase, so that the spatial distributions of the two dense phases are proportional to one another for all gas flow rates at a constant solids loading. The uncertainty in the reconstructions makes it difficult to test the assumption of homogeneity by directly comparing solid and liquid profiles. Instead, the ratio of the domain-averaged solid volume fraction $\bar{\varepsilon}_S$ to the domain-averaged dense phase volume fraction $(\bar{\varepsilon}_S + \bar{\varepsilon}_L)$ was computed for representative test conditions. If the above assumption about solid and liquid distribution were true, the ratio $\bar{\varepsilon}_S /(\bar{\varepsilon}_S + \bar{\varepsilon}_L)$ would equal $\bar{\varepsilon}_S^{NOM}$ at all gas flow rates, even in the presence of significant axial and radial variations in the phase distributions.

Figure 5.16 presents the ratio $\bar{\varepsilon}_S /(\bar{\varepsilon}_S + \bar{\varepsilon}_L)$ as a function of superficial gas velocity and nominal column-averaged solids loading $\bar{\varepsilon}_S^{NOM}$ (denoted by the dashed lines) for the 400 µm polystyrene particles, the first solids used. These values were computed from the known masses of solid and liquid phases in the column. The curves were produced using the original Maxwell-Hewitt relation (Eq. 3.13), before the recursive relation or the modified coefficient of $\alpha = \frac{3}{5}$ were developed. At all solids loadings, the ratio appears to increase above the nominal value as the superficial gas velocity U_G increases, which is counter-intuitive. Two possible causes for this trend were identified. The first was EIT measurement inaccuracies resulting from the increasing fraction of insulating phases (solid and gas) in the flow, which approached $\varepsilon_I = 0.50$ at the highest solids loading and gas flow rate. The second explanation was the inadequacies in the original Maxwell-Hewitt relation that led to the derivation of the modified relation.

Figure 5.17 shows the results when the original Maxwell-Hewitt relation is replaced with the recursive, bimodal Maxwell-Hewitt relation (Eq. 5.8). The values of $\bar{\varepsilon}_S /(\bar{\varepsilon}_S + \bar{\varepsilon}_L)$ now generally lie below the values of $\bar{\varepsilon}_S^{NOM}$, the ideal values if the solid were distributed uniformly throughout the liquid, and approach these nominal values as the superficial gas velocity is increased. Since larger superficial gas velocities agitate the medium more severely, it would not be surprising to find that the solid-liquid mixture became more uniform at larger superficial gas velocities. In the interest of expediency, average values were used instead of profiles to produce this plot, so it is suggested that this analysis be repeated in the future using the profiles themselves to verify these observations. Nevertheless, these results suggest that the modified Maxwell-Hewitt relation has improved further the accuracy of material-distribution profile measurements, and that the high insulating volume fractions did not pose problems for EIT in these experiments.

5.3.3. Correlation of Data

Typically, experimental results are correlated using three liquid properties: density, viscosity and surface tension. Recall from Section 2.2.1 that greater liquid density increases the

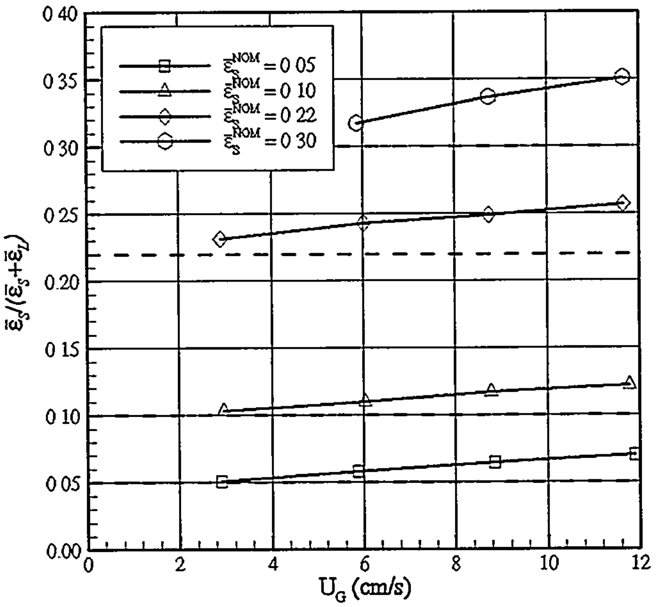


Figure 5.16. Dependence of the ratio $\bar{\varepsilon}_S/(\bar{\varepsilon}_S + \bar{\varepsilon}_L)$ on superficial gas velocity, calculated using the original Maxwell-Hewitt relation. The dashed lines indicate values of $\bar{\varepsilon}_S^{NOM}$ for each solids loading, the values that would be achieved if the solid were uniformly distributed in the liquid.

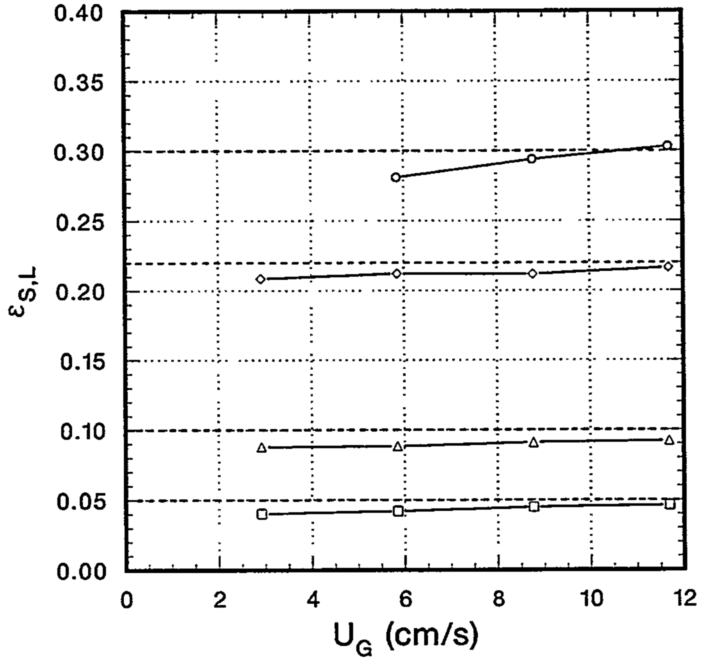


Figure 5.17. Dependence of the ratio $\bar{\varepsilon}_S/(\bar{\varepsilon}_S + \bar{\varepsilon}_L)$ on superficial gas velocity, calculated using the modified Maxwell-Hewitt relation.

buoyant force on a bubble, which increases its velocity relative to the liquid and thereby tends to decrease the gas volume fraction. However, greater liquid density can also increase bubble breakup, which reduces bubble size and velocity relative to the liquid and thereby tends to increase the gas volume fraction. A majority of the empirical gas-liquid correlations in the exhaustive review by Joshi *et al.* (1998) indicate that increasing liquid density generally increases gas volume fraction. This suggests that the bubble breakup effect of ρ_L tends to dominate in bubble column reactors. Similarly, greater liquid viscosity decreases bubble velocity but increases bubble size; in this case, the latter effect dominates and the net result is generally a reduction in gas volume fraction. Greater surface tension generally increases bubble size and reduces gas volume fraction as well. These opposing trends with liquid properties are supported by the common dependence of gas volume fraction upon the Bond and Archimedes numbers in empirical correlations (see, for example, Eqs. 2.11 and 2.12).

The experimental values in Figures 5.13 and 5.14 were fit using a dimensional density-based correlation:

$$\bar{\varepsilon}_G = \frac{0.0555 U_G^{0.549}}{\left[\rho_L\left(1 - \bar{\varepsilon}_S^{NOM}\right) + \rho_S \bar{\varepsilon}_S^{NOM}\right]^{1.017}}, \tag{5.9}$$

where U_G is in cm/s, ρ_L and ρ_S are in g/cm^3, $\bar{\varepsilon}_S^{NOM}$ is the nominal solid volume fraction, and $\bar{\varepsilon}_G$ is the average gas volume fraction. The bracketed term in this correlation is the density of an effective "dense-phase" medium composed of the liquid and the solid. For the glass beads and the 400 μm polystyrene beads, the correlation generally reproduces the data to within the experimental uncertainty of ±0.02 (Figures 5.18 and 5.19). The data from the experiments with 200 μm polystyrene particles are not fit well by the correlation, however, suggesting that other physical phenomena (such as particle diameter or surface tension) must be considered in describing the behavior of the gas phase. Note also that the trend of decreasing gas volume fraction with increasing medium density runs counter to most gas-liquid correlations, as explained in the previous paragraph. Further work to parameterize the data and compare results to existing two-phase correlations is recommended.

5.4. Conclusions

Material distributions have been measured successfully in many three-phase bubble-column flows using the combined EIT/GDT technique. The experiments involved both glass particles with a density substantially greater than the liquid density and polystyrene particles that were nearly neutrally buoyant. The gas volume fraction radial profiles were found to be relatively insensitive to the particle loading for all of the particle types examined, although the glass particles appeared to produce lower gas volume fractions at higher solids loadings. For solid volume fractions up to 0.30, the observed variation of gas volume fraction distribution with particle density and diameter was no more than ±0.04. While the observed variations can be explained with some accuracy by an effective gas/dense phase model, other physical mechanisms not yet accounted for may also be involved. Further analysis of the data will be

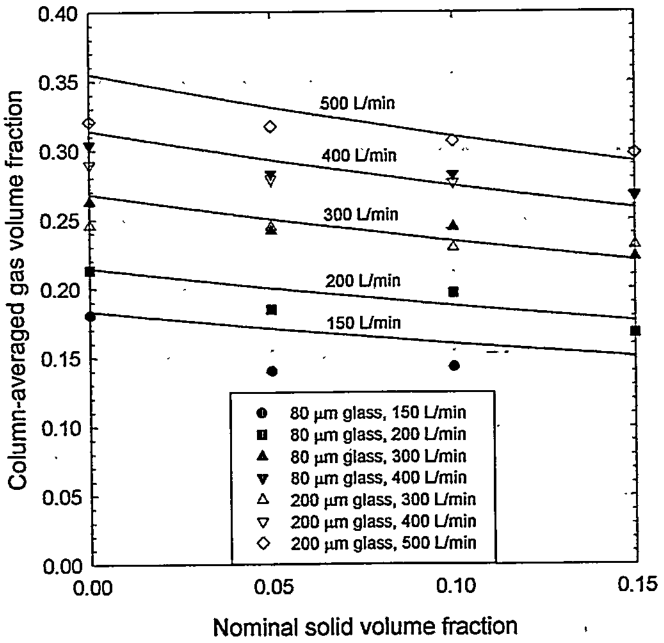


Figure 5.18. Comparison of measured and predicted average gas volume fractions in three-phase flows involving glass particles. Symbols are measured values; curves are predicted by Eq. 5.9.

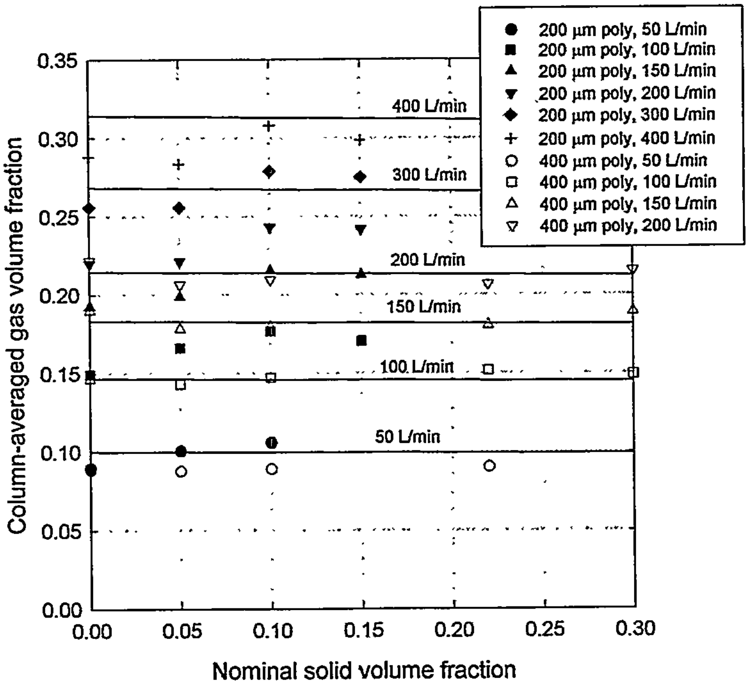


Figure 5.19. Comparison of measured and predicted average gas volume fractions in three-phase flows involving polystyrene particles. Symbols are measured values; curves are predicted by Eq. 5.9.

performed and may determine whether the effective two-phase model is an adequate description of three-phase flows.

With respect to bubble-column hydrodynamics, several conclusions may be drawn from these results about the changes (or lack of changes) to gas-liquid flows with the addition of a solid phase. For all solids loadings and gas flow rates examined, the time-averaged gas volume fraction profiles were still found to be roughly parabolic, indicating (along with visual observations) that the flow was still in the churn-turbulent regime. No dense layer of the solid phase was observed along the sides of the vessel; the particles appear to be approximately uniformly distributed in the liquid. Similarly, small but observable changes in the dense phase ratio $\bar{\varepsilon}_S/(\bar{\varepsilon}_S + \bar{\varepsilon}_L)$ with superficial gas velocity U_G suggest that at lower gas flow rates, the solid phase is not homogeneously distributed in the liquid phase, as assumed in some simplified three-phase models. However, as the gas flow rate increases and lofting efficiency is improved, the dense phase ratio approaches values determined from the homogeneous assumption.

6. Conclusions

The application of tomographic techniques to multiphase flows continues to be a significant research topic, where goals include improvements in temporal and spatial resolution and quantitative accuracy. In a contribution to this field, an electrical-impedance tomography (EIT) system, developed collaboratively by Sandia National Laboratories and the University of Michigan, has been validated for quantitative measurements of solid volume fractions in solid-liquid flows and gas volume fractions and radial profiles in axisymmetric gas-liquid flows. Many of these validation studies were performed through comparison of EIT results to measurements from an established gamma-densitometry tomography (GDT) system. In solid-liquid flows, EIT was used to measure solid volume fractions up to 0.05, with excellent agreement between EIT results and the nominal values determined from the mass of solids in each flow. Very good agreement was also obtained between EIT and GDT in those solid-liquid flows when the latter technique could be applied. Air-water bubble-column flows in a 19-cm ID bubble column were also measured with both EIT and GDT. For cross-sectionally-averaged gas volume fractions up to 0.15, the average gas volume fraction values and radial profiles from EIT and GDT agreed to within the EIT measurement uncertainty of ±0.01, even with large radial variations across the column. The Maxwell-Hewitt relations used to convert conductivity information to conducting phase distributions were found to be accurate for cases where the assumptions inherent in their derivation were valid. A minor, physically reasonable modification of the Maxwell-Hewitt relations was found to improve the level of agreement between GDT and EIT.

After validation of the EIT method, the EIT and GDT systems were successfully combined to accurately measure radial phase distributions in a series of three-phase, solid-gas-liquid bubble-column flows resembling those in slurry bubble-column reactors. The goal of these experiments was to examine the effect of the solid phase on the hydrodynamic behavior of the remaining phases, representing another step toward the application of EIT to industrial multiphase flows. The study employed solid phases with conductive properties similar to air but densities on the same order of magnitude as water, so that each of the three phases involved a unique combination of attenuating and conductive properties. This also required that the constitutive equations for both measurement methods be solved simultaneously to determine the distribution of all three phases. Four particle types were examined: 200-700 micron polystyrene spheres (specific gravity 1.04), 40-100 micron glass spheres (specific gravity 2.4), 120-200 micron glass spheres, and 170-260 micron polystyrene spheres. Solid volume fractions of up to 0.30 were examined over superficial gas velocities in the 2-30 cm/s range. This approach of repeating the experiments with solid particles of different diameters and specific gravities was taken to determine the influences of these quantities on radial phase profiles.

Over the range of solid volume fractions from 0 to 0.30, the gas distribution for each gas flow rate was relatively insensitive to the amount of solids present in the mixture. For all of the particle types that were examined, a maximum variation in gas volume fraction of ±0.04 was observed for a given superficial gas velocity. Neutrally buoyant particles were found to have almost no effect on the gas distribution. A slight decrease in gas volume fraction with nominal solid volume fraction was observed with the glass particles, which were substantially more dense than the liquid. The physical mechanism responsible for this effect is not yet clear. Particle diameter was found to have only a small effect on gas distribution. By comparison, surface-tension changes from contaminants added with the particles were more important than changes from the presence of the particles themselves. Finally, no dense layer of particles along the side walls of the vessel was observed; the particles appear to be approximately uniformly distributed in the column. Results such as these are essential to the development of accurate phase interaction models for three-phase flows. The implication of the present results for flow modeling is that an effective two-phase description may be accurate for describing the three-phase flows studied here.

The results of the bubble column experiments indicate that the assumption of a parabolic or quartic phase distribution is reasonably accurate when information is averaged over time periods as short as twenty seconds. Since the existing EIT system can clearly determine time-averaged phase distributions in a short time, the question of its potential to measure instantaneous phase distributions arises. For nearly instantaneous measurements, data must be acquired over periods approaching the time scale of the turbulent two-phase flow, estimated for the conditions of Chapters 4 and 5 to be on the order of a few milliseconds. Even with the recent conversion to a Pentium computer to control the EIT system, a single projection set cannot be obtained this quickly. In theory, the minimum time required for an AC electrical instrument to acquire a single data point is two cycles of the excitation voltage, so that the maximum data acquisition rate is half the excitation frequency. Based on a sixteen electrode system operating at 50 kHz, and assuming the minimum number of voltage values required for a reconstruction is given by Eq. 3.3, an instantaneous measurement of the phase distribution could theoretically be obtained in 5 ms, comparable to the flow time scale. However, the uncertainty of such a reconstruction must be considered. Early experiments with Lexan "phantom" inclusions similar to those in Chapter 3 suggest that ten projection sets with 1920 data points each is the minimum necessary to produce a reconstruction of a stationary distribution with acceptably low uncertainty. Tests using the quartic reconstruction algorithm in Chapter 4 suggest that 100 projection sets are required for accurate reconstructions of phase profiles in dynamic flows. Thus, while the present system can produce time-averaged phase profiles quickly and accurately, time-resolved information in dynamic flows appears to be beyond its capabilities.

Future efforts will focus on two areas. The first area involves improvements to the EIT system described in this work. As this report is being completed, the EIT system is being integrated into the LabView environment to standardize data acquisition. The time necessary to acquire projection sets may be reduced by as much as an order of magnitude with improved PC hardware, and measurement accuracy can be increased by a conversion of data acquisition hardware from 12-bit to 16-bit precision. The second area of interest involves extensions of the EIT technique to Sandia's slurry bubble-column reactor (SBCR) testbed and to other industrial-scale systems (*e.g.*, conducting vessels with insulating liquids). With the successful combination of EIT and GDT reported here, additional investigations are now possible to help develop three-phase flow models for bubble-column reactor design. Three-phase experiments in the 48.26-cm ID SBCR testbed using the LabView EIT and GDT systems are planned upon completion of a sparger-parameter study currently under way. It has also been noted that the existing EIT hardware could be modified to collect voltage projections from flows in which the phases are predominantly capacitive. Rather than modify the existing system, consideration is being given to a new capacitive tomography system that would determine three-dimensional solids distributions in a dynamic gas-solid flow. The new system would be evaluated in an industrial-scale, gas-solid reactor recently constructed at Sandia. Other proposed enhancements (of lower priority) to the existing EIT system include a reconstruction algorithm that would resolve non-axisymmetric phase distributions, and the combination of two or more separate EIT systems to measure phase velocities.

References

Adkins, D. R., Shollenberger, K. A., O'Hern, T. J., and Torczynski, J. R., 1996, "Pressure Effects on Bubble-Column Flow Characteristics," *ANS Proceedings of the 1996 National Heat Transfer Conference*, American Nuclear Society, LaGrange Park, Illinois, THD-Vol. 9, pp. 318-325.

Akita, K., and Yoshida, F., 1973, "Gas Holdup and Volumetric Mass Transfer Coefficient in Bubble Columns," *Industrial and Engineering Chemistry Process Design and Development*, Vol. 12, pp. 76-80.

Beck, M. S., Campogrande, E., Morris, M., Williams, R. A., and Waterfall, R. C., eds., 1993, *Tomographic Techniques for Process Design and Operation*, Computational Mechanics Publications, Southampton, England.

Biesheuvel, A., and Gorissen, W. C. M., 1990, "Void Fraction Disturbances in a Uniform Bubbly Fluid," *International Journal of Multiphase Flow*, Vol. 16, pp. 211-231.

Carey, V. P., 1992, *Liquid-Vapor Phase Change Phenomena: an Introduction to the Thermophysics of Vaporization and Condensation Processes in Heat Transfer Equipment*, Taylor and Francis, Bristol, Pennsylvania.

Ceccio, S. L., and George, D. L., 1996, "A Review of Electrical Impedance Techniques for the Measurement of Multiphase Flows," *ASME Journal of Fluids Engineering*, Vol. 118, pp. 391-399.

Chen, R. C., Reese, J., and Fan, L.-S., 1994, "Flow Structure in a Three-Dimensional Bubble Column and Three-Phase Fluidized Bed," *AIChE Journal*, Vol. 40, pp. 1093-1104.

Darton, R. C., LaNauze, R. D., Davidson, J. F., and Harrison, D., 1977, "Bubble Growth Due to Coalescence in Fluidized Beds," *Transactions of the Institute of Chemical Engineering*, Vol. 55, pp. 274.

Data Translation, Inc., 1994, *Data Translation® SP0131 Software Toolkit for DOS User Manual*, Data Translation, Incorporated, Marlboro, Massachusetts.

Deckwer, W.-D., and Schumpe, A., 1993, "Improved Tools For Bubble Column Reactor Design and Scale-Up," *Chemical Engineering Science*, Vol. 48, pp. 889-911.

Degaleesan, S., Dudukovic, M. P., Toseland, B. A., and Bhatt, B. L., 1997, "A Two-Compartment Convective-Diffusion Model for Slurry Bubble Column Reactors," *Industrial and Engineering Chemistry Research*, Vol. 36, pp. 4670-4680.

Dickin, F. J., Williams, R. A., and Beck, M. S., 1993, "Determination of Composition and Motion of Multicomponent Mixtures in Process Vessels using Electrical Impedance Tomography - I. Principles and Process Engineering Applications," *Chemical Engineering Science*, Vol. 48, pp. 1883-1897.

Duraiswami, R., Chahine, G. L., and Sarkar, K., 1997, "Boundary Element Techniques for Efficient 2-D and 3-D Electrical Impedance Tomography," *Chemical Engineering Science*, Vol. 52, pp. 2185-2196.

Fan, L.-S., and Zhu, C., 1998, *Principles of Gas-Solid Flows*, Cambridge University Press, New York, p. 24.

Fluid Dynamics International, 1996, *FIDAP Users Manual*, versions 7.0, 7.5 and 7.6, Fluid Dynamics International, Evanston, IL.

George, D. L., Ceccio, S. L., O'Hern, T. J., Shollenberger, K. A., and Torczynski, J. R., 1998a, "Advanced Material Distribution Measurement in Multiphase Flows: a Case Study," *Proceedings of the 1998 ASME International Mechanical Engineering Conference and Exposition*, American Society of Mechanical Engineers, New York, FED-Vol. 247, pp. 31-42.

George, D. L., Ceccio, S. L., Shollenberger, K. A., Torczynski, J. R., and O'Hern, T. J., 1998b, "Comparison of Electrical-Impedance Tomography and Gamma-Densitometry Tomography for the Measurement of Gas Volume Fraction Profiles in a Bubble Column," *Proceedings of the 1998 ASME Fluids Engineering Division Summer Meeting*, American Society of Mechanical Engineers, New York, FED-Vol. 245, No. 98-5081.

George, D. L., 1999a, *Experimental Studies of Cavitation and Vertical Multiphase Flows Using Electrical Impedance Techniques*, Ph.D. Thesis, University of Michigan, Ann Arbor, Michigan.

George, D. L., Ceccio, S. L., Torczynski, J. R., Shollenberger, K. A., and O'Hern, T. J., 1999b, "Quantitative Phase Distribution Measurements in Three-Phase Flows," *Proceedings of the 3rd ASME/JSME Joint Fluids Engineering Conference*, American Society of Mechanical Engineers, San Francisco, FED-Vol. 248, No. 99-7375.

George, D. L., Torczynski, J. R., Shollenberger, K. A., O'Hern, T. J., and Ceccio, S. L., 1999c, "Validation of Electrical-Impedance Tomography for Measurements of Material Distribution in Two-Phase Flows," accepted for publication in *International Journal of Multiphase Flow*.

Hewitt, G. F., 1978, *Measurement of Two-Phase Flow Parameters*, Academic Press, London, England.

Hughmark, G. A., 1967, "Holdup and Mass Transfer in Bubble Columns," *Industrial and Engineering Chemistry Process Design and Development*, Vol. 6, pp. 218-220.

Ishii, M., and Zuber, N., 1979, "Drag Coefficient and Relative Velocity in Bubbly, Droplet or Particulate Flows," *AIChE Journal*, Vol. 25, pp. 843-855.

Jackson, N. B., Torczynski, J. R., Shollenberger, K. A., O'Hern, T. J., and Adkins, D. R., 1996, "Hydrodynamic Characterization of Slurry Bubble-Column Reactors for Fischer-Tropsch Synthesis," *Proceedings of the Thirteenth Annual International Pittsburgh Coal Conference, Vol. 2: Coal-Energy and the Environment*, S.-H. Chiang, ed., University of Pittsburgh Center for Energy Research, Pittsburgh, pp. 1226-1231.

Jones, O. C., Lin, J. T., and Ovacik, L., 1992, "Investigation of Electrical Impedance Imaging Relative to Two-Phase, Gas-Liquid Flows," *Chemical Engineering Communications*, Vol. 118, pp. 299-325.

Jones, O. C., Lin, J. T., Ovacik, L., and Shu, H., 1993, "Impedance Imaging Relative to Gas-Liquid Systems," *Nuclear Engineering and Design*, Vol. 141, pp. 159-176.

Jones, O. C., Lin., J. T., Shu, H., Ovacik., L., and He, Y., 1994, "Impedance Imaging Relative to Binary Mixtures," *Fifth International Symposium on Liquid-Solid Flows,* ASME, Lake Tahoe, NV.

Joshi, J. B., Parasu Veera, U., Prasad, Ch. V., Phanikumar, D. V., Deshphande, N. S., Thakre, S. S., and Thorat, B. N., 1998, "Gas Hold-Up Structure in Bubble Column Reactors," *Proceedings of the Indian National Science Academy*, Part A, Vol. 64, pp. 441-567.

Krishna, R., de Swart, J. W. A., Hennephof, D. E., Ellenberger, J., and Hoefsloot, H. C. J., 1994, "Influence of Increased Gas Density on Hydrodynamics of Bubble-Column Reactors," *AIChE Journal*, Vol. 40, pp. 112-119.

Krishna, R., and Ellenberger, J., 1996, "Gas Holdup in Bubble Column Reactors Operating in the Churn-Turbulent Flow Regime," *AIChE Journal*, Vol. 42, pp. 2627-2634.

Kumar, S. B., Devanathan, N., Moslemian, D., and Dudukovic, M. P., 1994, "Effect of Scale on Liquid Recirculation in Bubble Columns," *Chemical Engineering Science*, Vol. 49, pp. 5637-5652.

Kumar, S. B., Moslemian, D., and Dudukovic, M. P., 1997, "Gas-Holdup Measurements in Bubble Columns Using Computed Tomography," *AIChE Journal*, Vol. 43, pp. 1414-1425.

Lagasse, R. R., 1999, private communication.

Lamarsh, J. R., 1983, *Introduction to Nuclear Engineering,* Addison-Wesley, Reading, MA, pp. 78-88, 472-488, 648-649.

Lin, J.-T., Jones, O. C., Ovacik, L., and Shu, H.-J., 1993, "Advances in Impedance Imaging Relative to Two-Phase Flow," *ANS Proceedings of the 1993 National Heat Transfer Conference*, American Nuclear Society, LaGrange Park, Illinois, Vol. THD-7, pp. 68-75.

Lin, T.-J., Tsuchiya, K., and Fan, L.-S., 1998, "Bubble Flow Characteristics in Bubble Columns at Elevated Pressure and Temperature," *AIChE Journal*, Vol. 44, pp. 545-560.

Maxwell, J. C., 1881, *A Treatise on Electricity and Magnetism*, Clarendon Press, Oxford, England.

Microsoft Corporation, 1991, *Microsoft® QuickBasic™*, Microsoft Corporation, Redmond, Washington.

Munshi, P., 1990, "A Review of Computerized Tomography with Applications to Two-Phase Flows," *Sadhana*, Vol. 15, Part 1, pp. 43-55.

O'Hern, T. J., Torczynski, J. R., Ceccio, S. L., Tassin, A. L., Chahine, G. L., Duraiswami, R., and Sarkar, K., 1995, "Development of an Electrical Impedance Tomography System for an Air-Water Vertical Bubble Column," *Forum on Measurement Techniques in Multiphase Flows*, T. J. O'Hern, A. Naqwi, C. Presser, and R. D. Skocypec, eds., ASME, New York, FED-Vol. 233, pp. 531-537.

Öztürk, S. S., Schumpe, A., and Deckwer, W.-D., 1987, "Organic Liquids in a Bubble Column: Hold-ups and Mass Transfer Coefficients," *AIChE Journal*, Vol. 33, pp. 1473-1480.

Plaskowski, A., Beck, M. S., Thorn, R., and Dyakowski, T., 1995, *Imaging Industrial Flows: Applications of Electrical Process Tomography*, Institute of Physics Publishing, Bristol, England, pp. 2-5, 127-141.

Reilly, I. G., Scott, D. S., de Bruijn, T. J. W., and MacIntyre, D., 1994, "The Role of Gas Phase Momentum in Determining Gas Holdup and Hydrodynamic Flow Regimes in Bubble Column Operations," *Canadian Journal of Chemical Engineering*, Vol. 72, pp. 3-12.

Shah, Y. T., Kelkar, B. G., Godbole, S. P., and Deckwer, W.-D., 1982, "Design Parameter Estimations for Bubble Column Reactors," *AIChE Journal*, Vol. 28, pp. 353-379.

Shah, Y. T., and Deckwer, W.-D., 1983, "Hydrodynamics of Bubble Columns," *Handbook of Fluids in Motion*, Cheremisinoff, N. P., and Gupta, R., eds., Ann Arbor Science, Ann Arbor, MI, pp. 583-620.

Shollenberger, K. A., Torczynski, J. R., Adkins, D. R., and O'Hern, T. J., 1995, "Bubble Column Measurements Using Gamma Tomography," *Fluid Measurement and Instrumentation*, G. L. Morrison, M. Nishi, T. B. Morrow, and R. A. Gore, eds., American Society of Mechanical Engineers, New York, FED-Vol. 211, pp. 25-30.

Shollenberger, K. A., Torczynski, J. R., Adkins, D. R., O'Hern, T. J., and Jackson, N. B., 1997a, "Gamma-Densitometry Tomography of Gas Holdup Spatial Distribution in Industrial-Scale Bubble Columns," *Chemical Engineering Science*, Vol. 52, pp. 2037-2048.

Shollenberger, K. A., Torczynski, J. R, O'Hern, T. J., Adkins, D. R., Ceccio, S. L., and George, D. L., 1997b, "Comparison of Gamma-Densitometry Tomography and Electrical-Impedance Tomography for Determining Material Distribution in Liquid-Solid Flows," *Proceedings of the 1997 ASME Fluids Engineering Division Summer Meeting*, J. Katz and K. J. Farrell, eds., American Society of Mechanical Engineers, New York, Vol. FEDSM '97, No. 97-3690.

Simons, S. J. R., 1995, "Imaging Techniques for Fluidized Bed Systems: a Review," *Chemical Engineering Journal*, Vol. 56, pp. 83-93.

Somersalo, E., Cheney, M., and Issacson, D., 1992, "Existence and Uniqueness for Electrode Models for Electric Current Computed Tomography," *SIAM Journal of Applied Mathematics*, Vol. 52, pp. 1023-1040.

Thoraeus, R, 1965, "Attenuation of Gamma Radiation from ^{60}Co, ^{137}Cs, ^{192}Ir, and ^{226}Ra in Various Materials Used in Radiology," *Acta Radiologica*, Vol. 3, pp. 81-86.

Torczynski, J. R., O'Hern, T. J., Shollenberger, K. A., Ceccio, S. L., and Tassin, A. L., 1996a, "Finite Element Method Electrical Impedance Tomography for Phase Distribution Determination in Multiphase Flows: Validation Calculations and Experiments," *Cavitation and Multiphase Flow Forum*, J. Katz and K. J. Farrell, eds., ASME, New York, FED-Vol. 236, pp. 497-501.

Torczynski, J. R., Adkins, D. R., Shollenberger, K. A., and O'Hern, T. J., 1996b, "Application of Gamma-Densitometry Tomography to Determine Phase Spatial Variation in Two-Phase and Three-Phase Bubbly Flows," *ASME Cavitation and Multiphase Flow Forum*, J. Katz and K. J. Farrell, eds., American Society of Mechanical Engineers, New York, FED-Vol. 236, pp. 503-508.

Torczynski, J. R., O'Hern, T. J., Adkins, D. R., Jackson, N. B., and Shollenberger, K. A., 1997, *Advanced Tomographic Flow Diagnostics for Opaque Multiphase Fluids*, Report SAND97-1176, Sandia National Laboratories, Albuquerque, NM.

Torczynski, J. R., George, D. L., Shollenberger, K. A., and O'Hern, T. J., 1998, "Validation of Electrical-Impedance Tomography by Comparison to Gamma-Densitometry Tomography for Bubble-Column Flows," Sandia internal memorandum to distribution.

Ueyama, K., and Miyauchi, T., 1979, "Properties of Recirculating Turbulent Two Phase Flow in Gas Bubble Columns," *AIChE Journal*, Vol. 25, pp. 258-266.

Vest, C. M., 1985, "Tomography for Properties of Materials that Bend Rays: a Tutorial," *Applied Optics,* Vol. 24, pp. 4089-4094.

Wallis, G. B., 1969, *One Dimensional Two-Phase Flow*, McGraw-Hill, New York, NY.

Webster, J. G., ed., 1990, *Electrical Impedance Tomography,* Adam Hilger, Bristol, England, pp. 1-20, 75-86, 175-200.

White, F. M., 1991, *Viscous Fluid Flow*, McGraw-Hill, New York, NY, pp. 437-438.

Wilkinson, P. M., 1991, *Physical Aspects and Scale-Up of High-Pressure Bubble Columns*, Ph.D. Thesis, University of Groningen, The Netherlands.

Wilkinson, P. M., Spek, A. P., and van Dierendonck, L. L., 1992, "Design Parameters Estimation for Scale-Up of High-Pressure Bubble Columns," *AIChE Journal*, Vol. 38, pp. 544-554.

Wilkinson, P. M., Haringa, H., van Dierendonck, L. L., 1994, "Mass Transfer and Bubble Size in a Bubble Column Under Pressure," *Chemical Engineering Science*, Vol. 49, pp. 1417-1427.

Wolfram, S., 1996, *The Mathematica Book,* Cambridge University Press, Cambridge, MA.

Yorkey, T. J., Webster, J. G., and Tompkins, W. J., 1987, "Comparing Reconstruction Algorithms for Electrical Impedance Tomography," *IEEE Transactions on Biomedical Engineering*, Vol. 11, pp. 843-852.

Zou, R., Jiang, X., Li, B., Zu, Y., and Zhang, L., 1988, "Studies on Gas Holdup in a Bubble Column Operated at Elevated Temperatures," *Industrial and Engineering Chemistry Research*, Vol. 27, pp. 1910-1916.

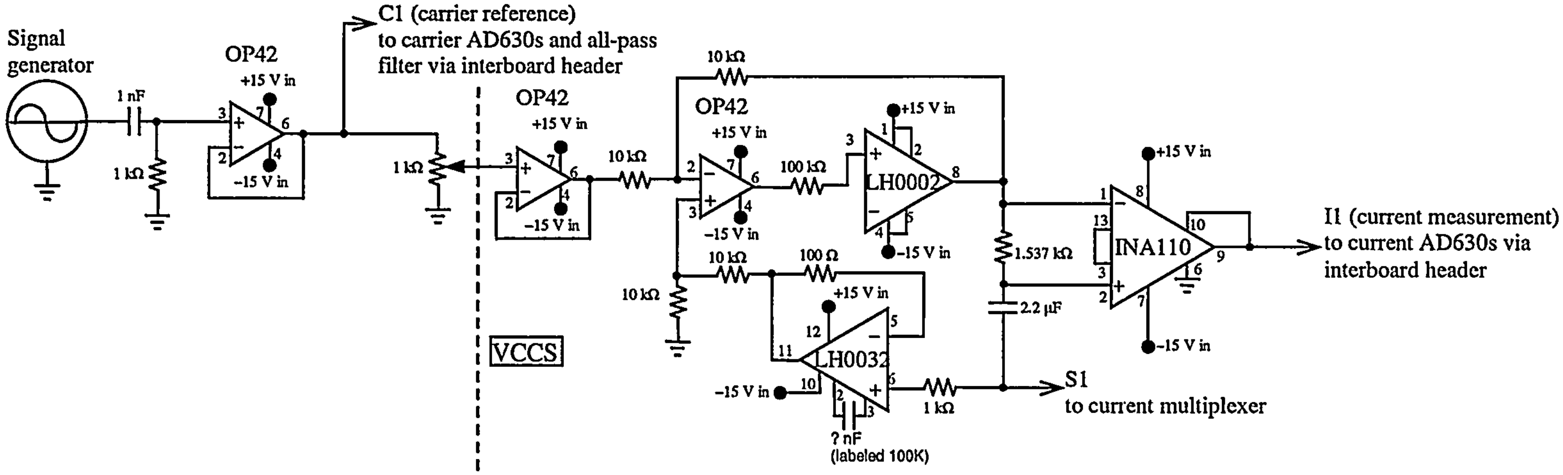
Zuber, N., and Findlay, J. A., 1965, "Average Volumetric Concentration in Two-Phase Flow Systems," *ASME Journal of Heat Transfer*, Vol. 87, pp. 453-468.

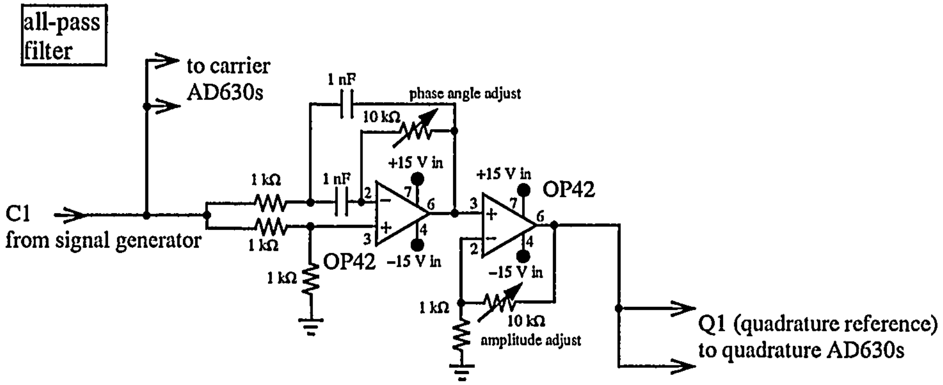
Appendix A
Circuit Diagrams of the Sandia/Michigan EIT System

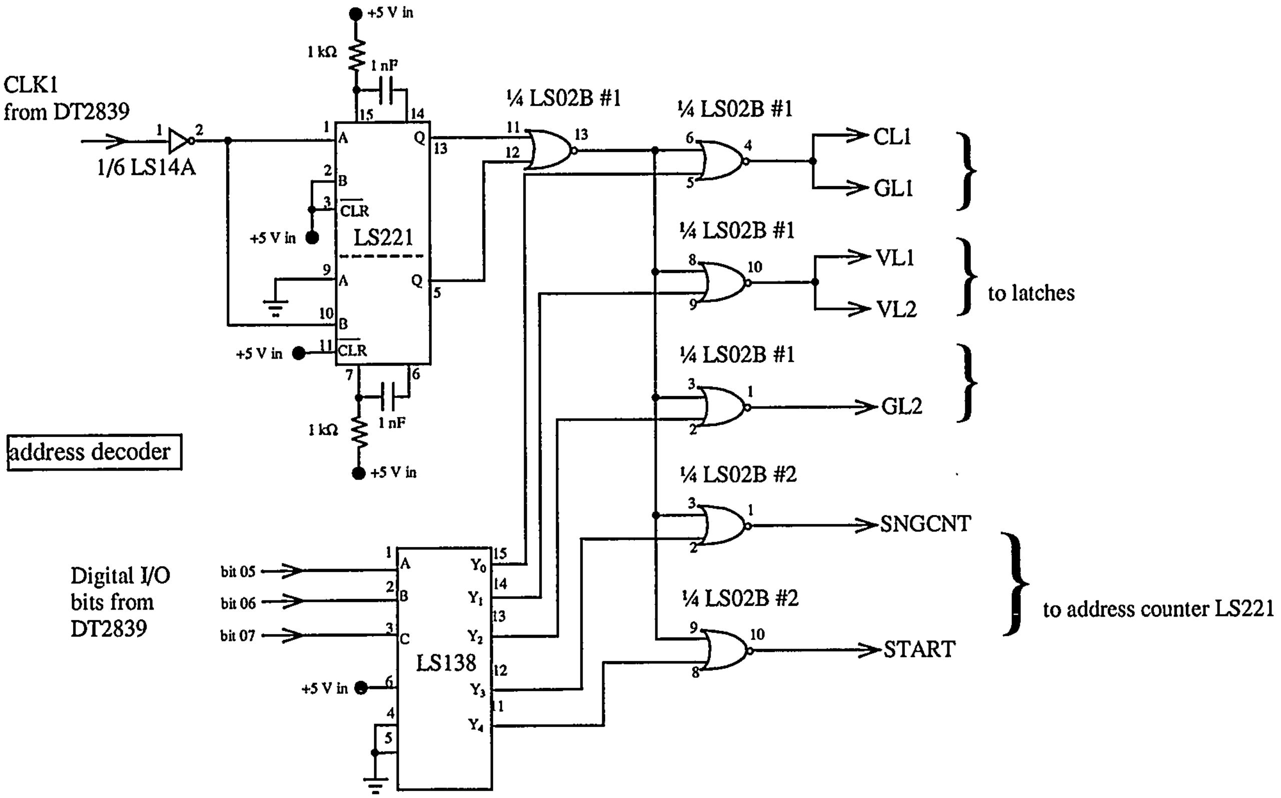
The following pages present circuit diagrams for the custom-built portion of the Sandia/Michigan EIT system. The first portion contains diagrams for circuitry in the EIT package, including:

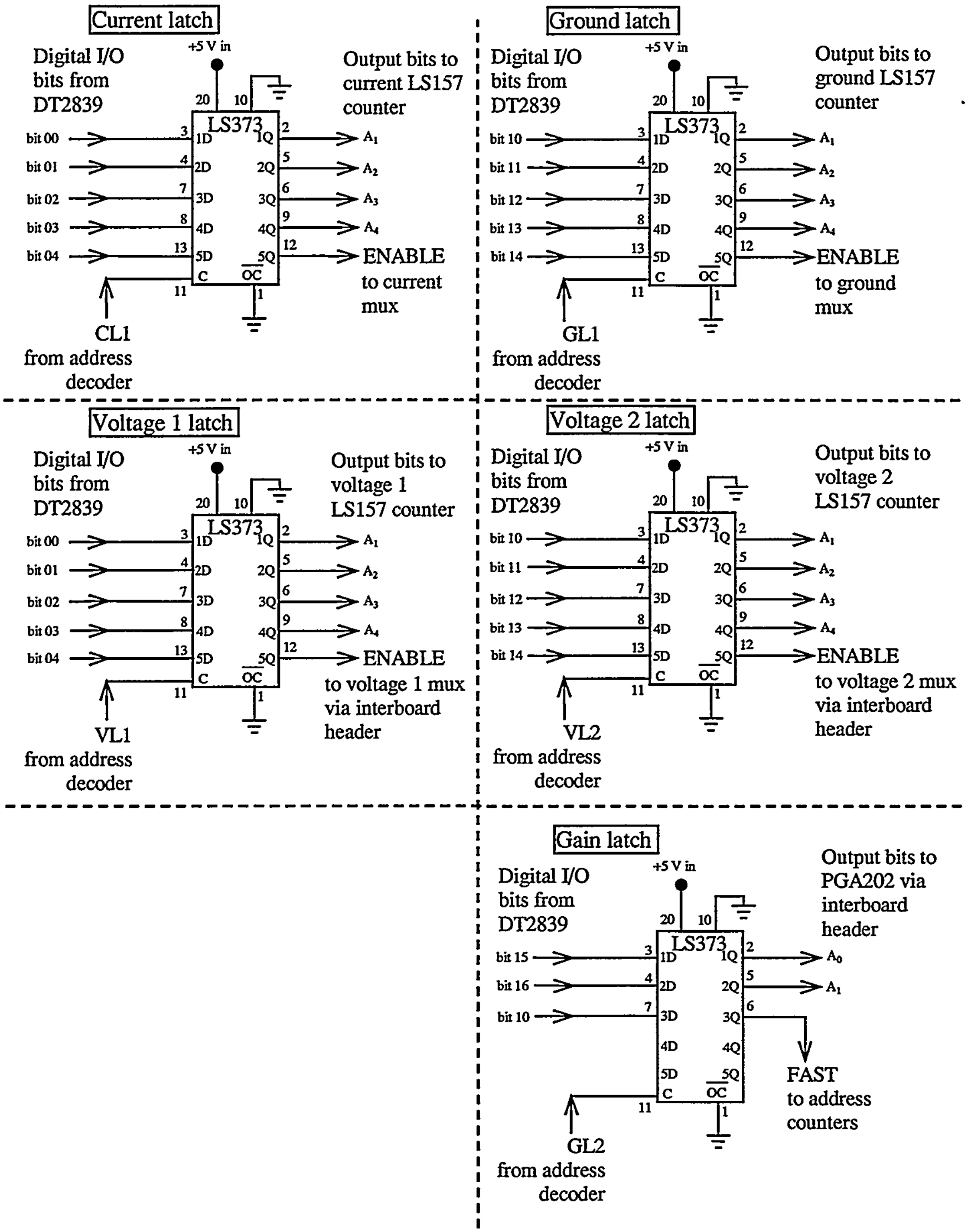
- circuitry to generate reference signals and injection current
- address decoders to initialize latches and electrode counters
- latches to select injection, ground and measurement electrodes from "slow mode" computer commands
- internal counters to select electrodes in "fast mode"
- multiplexers and buffers to and from electrodes
- amplifier, demodulators and low pass filters to process measured voltage signals

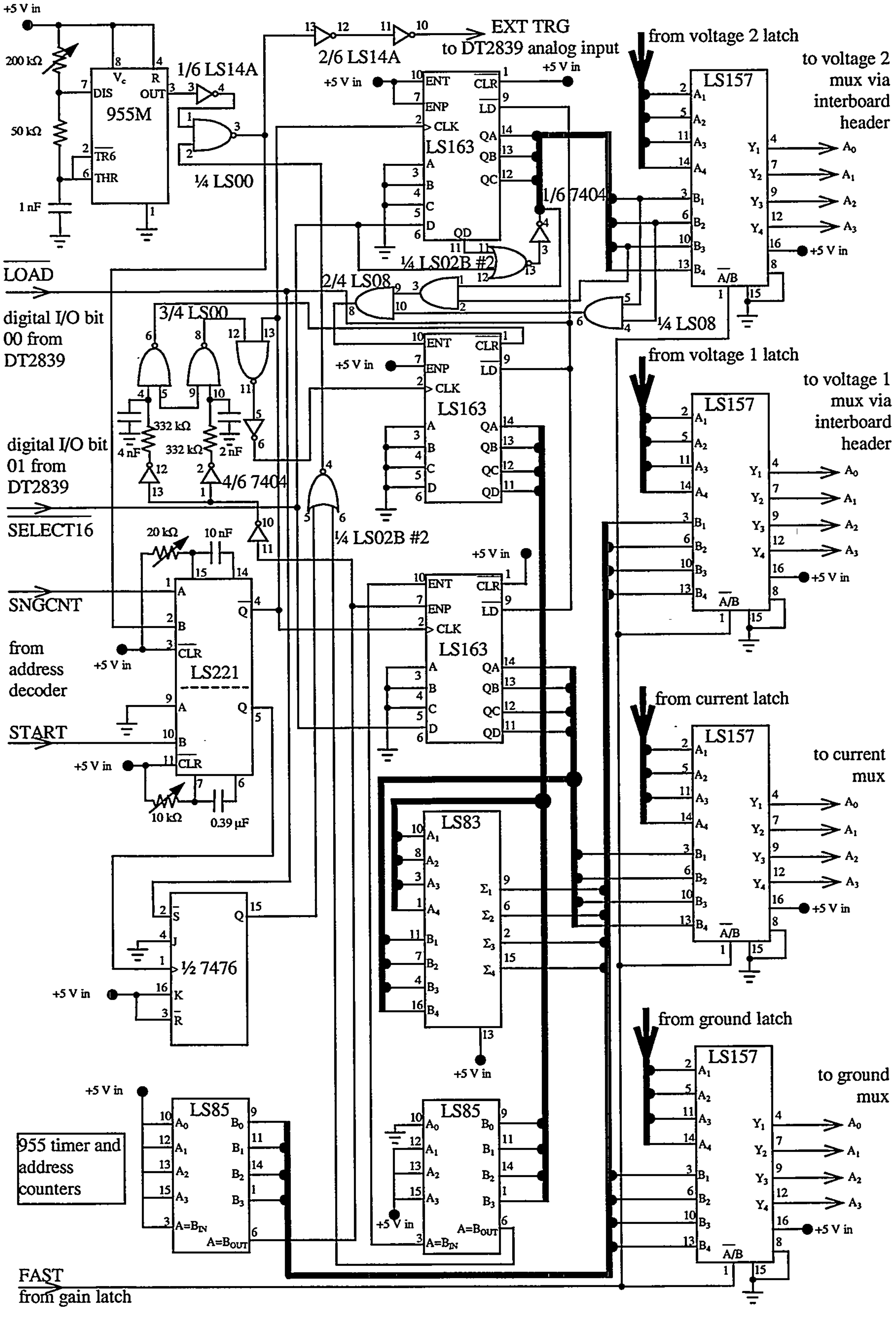
Labeled arrows mark connections between circuits on different pages. The circuit schematics are followed by diagrams of header connections between circuit boards, electrodes and the DT2839 data acquisition board.

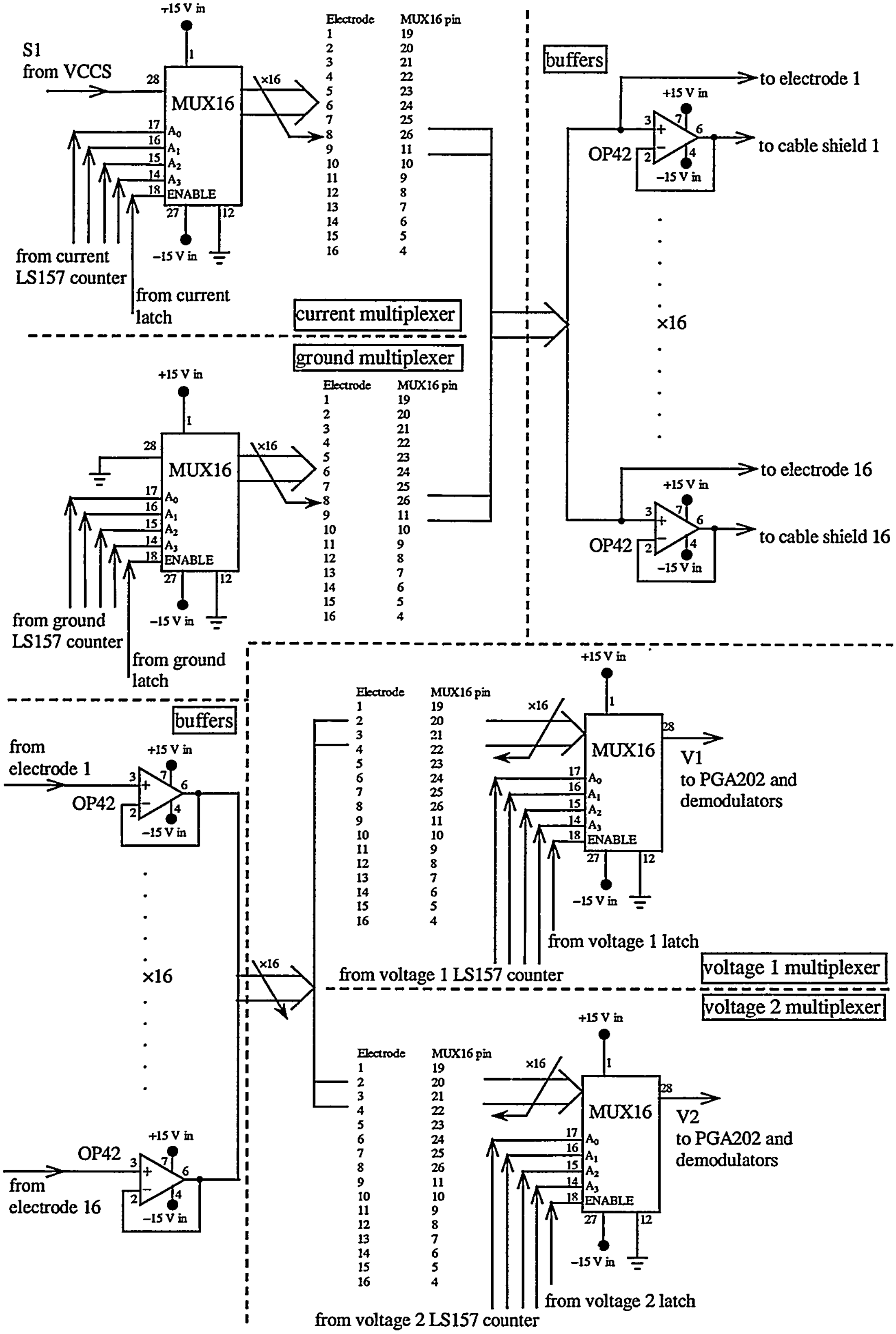

Signal generator
1 nF
1 kΩ
OP42
+15 V in
-15 V in
C1 (carrier reference)
to carrier AD630s and all-pass filter via interboard header
1 kΩ
OP42
+15 V in
-15 V in
VCCS
10 kΩ
10 kΩ
OP42
+15 V in
-15 V in
100 kΩ
LH0002
+15 V in
-15 V in
10 kΩ
10 kΩ
100 Ω
+15 V in
LH0032
-15 V in
? nF
(labeled 100K)
1 kΩ
1.537 kΩ
2.2 µF
S1
to current multiplexer
INA110
+15 V in
-15 V in
I1 (current measurement)
to current AD630s via interboard header


all-pass filter
to carrier AD630s
C1
from signal generator
1 nF
phase angle adjust
10 kΩ
1 kΩ
1 nF
1 kΩ
1 kΩ
+15 V in
OP42
-15 V in
+15 V in
OP42
-15 V in
1 kΩ
10 kΩ
amplitude adjust
Q1 (quadrature reference)
to quadrature AD630s

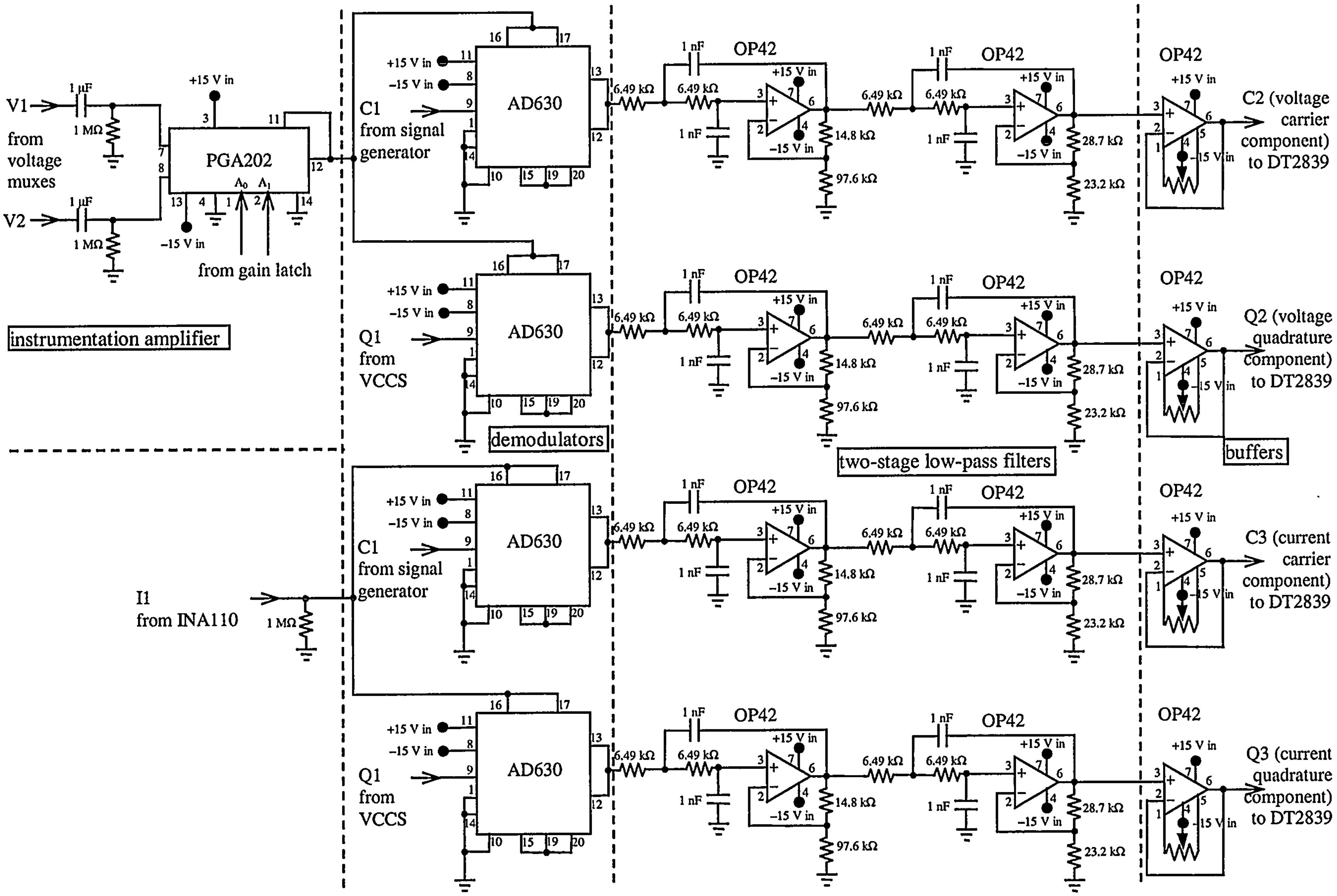

CLK1
from DT2839
1/6 LS14A
+5 V in
1 kΩ
1 nF
LS221
A
B
CLR
Q
+5 V in
1 kΩ
1 nF
+5 V in
+5 V in
¼ LS02B #1
¼ LS02B #1
¼ LS02B #1
¼ LS02B #1
¼ LS02B #2
¼ LS02B #2
CL1
GL1
VL1
VL2
GL2
SNGCNT
START
to latches
to address counter LS221
address decoder
Digital I/O
bits from
DT2839
bit 05
bit 06
bit 07
LS138
A
B
C
Y_0
Y_1
Y_2
Y_3
Y_4
+5 V in


Current latch
Digital I/O bits from DT2839
+5 V in
20
10
LS373
bit 00
bit 01
bit 02
bit 03
bit 04
3 1D
4 2D
7 3D
8 4D
13 5D
C
11
1Q 2
2Q 5
3Q 6
4Q 9
5Q 12
OC 1
Output bits to current LS157 counter
A_1
A_2
A_3
A_4
ENABLE to current mux
CL1 from address decoder
Ground latch
Digital I/O bits from DT2839
+5 V in
20
10
LS373
bit 10
bit 11
bit 12
bit 13
bit 14
3 1D
4 2D
7 3D
8 4D
13 5D
C
11
1Q 2
2Q 5
3Q 6
4Q 9
5Q 12
OC 1
Output bits to ground LS157 counter
A_1
A_2
A_3
A_4
ENABLE to ground mux
GL1 from address decoder
Voltage 1 latch
Digital I/O bits from DT2839
+5 V in
20
10
LS373
bit 00
bit 01
bit 02
bit 03
bit 04
3 1D
4 2D
7 3D
8 4D
13 5D
C
11
1Q 2
2Q 5
3Q 6
4Q 9
5Q 12
OC 1
Output bits to voltage 1 LS157 counter
A_1
A_2
A_3
A_4
ENABLE to voltage 1 mux via interboard header
VL1 from address decoder
Voltage 2 latch
Digital I/O bits from DT2839
+5 V in
20
10
LS373
bit 10
bit 11
bit 12
bit 13
bit 14
3 1D
4 2D
7 3D
8 4D
13 5D
C
11
1Q 2
2Q 5
3Q 6
4Q 9
5Q 12
OC 1
Output bits to voltage 2 LS157 counter
A_1
A_2
A_3
A_4
ENABLE to voltage 2 mux via interboard header
VL2 from address decoder
Gain latch
Digital I/O bits from DT2839
+5 V in
20
10
LS373
bit 15
bit 16
bit 10
3 1D
4 2D
7 3D
4D
5D
C
11
1Q 2
2Q 5
3Q 6
4Q
5Q
OC 1
Output bits to PGA202 via interboard header
A_0
A_1
FAST to address counters
GL2 from address decoder


+5 V in
200 kΩ
50 kΩ
1 nF
955M
DIS
OUT
TR6
THR
1/6 LS14A
¼ LS00
2/6 LS14A
EXT TRG
to DT2839 analog input
LS163
ENT
ENP
CLK
CLR
LD
QA
QB
QC
QD
1/6 7404
¼ LS02B #2
2/4 LS08
¼ LS08
from voltage 2 latch
LS157
to voltage 2
mux via
interboard
header
A₀
A₁
A₂
A₃
LOAD
digital I/O bit
00 from
DT2839
3/4 LS00
332 kΩ
4 nF
2 nF
4/6 7404
digital I/O bit
01 from
DT2839
SELECT16
from voltage 1 latch
to voltage 1
mux via
interboard
header
20 kΩ
10 nF
SNGCNT
from
address
decoder
LS221
START
10 kΩ
0.39 μF
½ 7476
from current latch
to current
mux
LS83
from ground latch
to ground
mux
LS85
A=B_IN
A=B_OUT
955 timer and
address
counters
FAST
from gain latch



Electrode	MUX16 pin
1	19
2	20
3	21
4	22
5	23
6	24
7	25
8	26
9	11
10	10
11	9
12	8
13	7
14	6
15	5
16	4


V1
from voltage muxes
V2
1 µF
1 MΩ
+15 V in
−15 V in
PGA202
A0 A1
from gain latch
instrumentation amplifier
I1 from INA110
C1 from signal generator
Q1 from VCCS
AD630
+15 V in
−15 V in
demodulators
6.49 kΩ
1 nF
OP42
14.8 kΩ
97.6 kΩ
28.7 kΩ
23.2 kΩ
two-stage low-pass filters
buffers
C2 (voltage carrier component) to DT2839
Q2 (voltage quadrature component) to DT2839
C3 (current carrier component) to DT2839
Q3 (current quadrature component) to DT2839

Interboard header

Numbered boxes represent header pins; DGND = digital ground; AGND = analog ground

Board 1 header

	From voltage 1 LS157 counter				From voltage 1 latch		From voltage 2 LS157 counter				From voltage 2 latch		From gain latch			I1 (from INA110)		C1 (from signal generator)		
IC pin	4	7	9	12	12		4	7	9	12	12		2	5		9				
	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
	DGND	DGND	DGND	DGND	DGND		DGND	DGND	DGND	DGND	DGND		DGND	DGND				AGND	AGND	AGND

Board 2 header

	To voltage 1 multiplexer						To voltage 2 multiplexer						To PGA202			I1 (to current AD630s)		C1 (to all-pass filter, carrier AD630s)		To DT2839 analog header (unused)
IC pin	17	16	15	14	18		17	16	15	14	18		1	2						
	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
	DGND	DGND	DGND	DGND	DGND		DGND	DGND	DGND	DGND	DGND		DGND	DGND				AGND	AGND	AGND

Headers between multiplexers and electrodes

Header from board 1 to electrode cable screw terminals

Lines 17 – 48 unused

From current and ground multiplexers

mux pin					4	5	6	7	8	9	10	11	26	25	24	23	22	21	20	19
	·	·	·	·	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	·	·	·	·	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
electrode					16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

From shield drivers

Header from electrode cable screw terminals to board 2

Lines 1 – 48 unused

	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
	·	·	·	·	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
electrode					16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

To voltage buffers

Headers between DT2839 and EIT electronics

Header between DT2839 digital I/O terminals and board 1

	EXT TRG pulse to DT2839		CLK1 pulse to address decoder			To LS138 (address decoder)			To latches											
Digital I/O bit						07	06	05	16	15	14	13	12	11	10	04	03	02	01	00
	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
	DGND				DGND	DGND	DGND	DGND	DGND	DGND	DGND	DGND	DGND	DGND	DGND	DGND	DGND	DGND	DGND	DGND

Header between DT2839 analog I/O terminals and board 2

To pin 40, board 2, interboard header	1	16	To pin 1, board 2, interboard header	
	2	15		
	3	14	Q3, current quadrature	Voltage components measured by DT2839
	4	13	C3, current carrier	
AGND	5	12	Q2, voltage quadrature	
	6	11		
	7	10		
AGND	8	9	C2, voltage carrier	

Appendix B

EIT Data Acquisition Code `FASTEIT.BAS`

This program, written in Microsoft® QuickBasic™ (Microsoft Corporation, 1991), initializes the EIT system for operation with the internal electrode counters. After initializing the DT2839 data acquisition board, the code dynamically allocates buffers in the PC memory to store voltage measurements, then arms the EIT electronics. Multiplexers in the electronics select the electrodes that inject and ground current, and also select the electrodes whose voltages will be measured by the data acquisition card. This version of the code activates a series of internal counters that control the multiplexers, so that the card is only responsible for measuring electrode voltages. All voltages are stored in the buffers until all projection sets are collected, then the voltage values are transferred to arrays for averaging calculations.

Several subroutines from the Data Translation SP0131 Software Toolkit (Data Translation, Inc., 1994) are used to initialize the data acquisition board, issue commands to the EIT electronics, and acquire voltage measurements. These subroutines are proprietary and are not listed here, but may be recognized by the prefix "`dt.`" For example, the subroutine that opens communications with the DT2839 board is called by the statement:

```
istat = dt.initialize(SADD(idrivername$), VARSEG(idrivername$), ihandle).
```

The arguments in parentheses may be instructions to the board or data returned by the board; a nonzero value of `istat` identifies an error in execution.

```
'***
'***  Steve and Darin's FAST EIT code
'***    Written 7/97-12/97 by dlg and slc
'***    Electrode selection is "hard-wired" through IC counters;
'***    data acquisition board only starts counters and collects
'***    voltage data.  To conserve memory, dynamic allocation is
'***    used.  Arrays are dimensioned when first used and deallocated
'***    when they are no longer needed.
'***

'$DYNAMIC

DECLARE SUB exit.error (status AS INTEGER, message AS STRING)
DECLARE SUB cleanup (dummy)
DECLARE SUB eistats (elec%, proj%, slength%, carrsum&(), quadsum&(), Vsum!(), resp$)

'*****************************************************************
' Definition files for extended memory manager, error codes, counter/
'   timer subroutines and software tools; these files define several
'   constants used in subroutine calls, which are found in ALL CAPS in
'   the code
```

```
'$INCLUDE: 'd:\toolkit\dtst_xmm.bi'
'$INCLUDE: 'd:\toolkit\dtst_err.bi'
'$INCLUDE: 'd:\toolkit\dtst_ctr.bi'
'$INCLUDE: 'd:\toolkit\dtst_tls.bi'

DIM digitalio AS DIGITALIOSTRUCT
DIM sap AS SETACQPARAMSSTRUCT
DIM board AS BOARDSTRUCT

DEFINT I-N
DEFSNG A-H
DEFLNG O-Z

CONST slength% = 16       ' number of oversamples on each channel

CONST currlatch% = 0      ' addresses of counters, muxes, etc., and
CONST voltlatch% = 32     '   commands to arm them
CONST gainlatch% = 64
CONST fastflag% = 1
CONST enable% = 16

CLS
PRINT
PRINT "This EIT program arms the internal counters, which select the current,"
PRINT "ground and measurement electrodes.  The A/D card records the output"
PRINT "of the demodulators when triggered by the EIT electronics."
PRINT

DO
  INPUT "Enter the number of electrodes (8 or 16): ", elec%
  IF elec% <> 8 AND elec% <> 16 THEN
    PRINT "Incorrect input."
  END IF
LOOP UNTIL elec% = 8 OR elec% = 16

resetcntr% = 96 + (16 - elec%) / 4      ' commands to reset and start
startcntr% = 128 + (16 - elec%) / 4 + 1  '   hardware counters with
                                         '   desired number of electrodes

'**********************************************************************
' Compute quantities for memory and buffer allocation

icombs = elec% ^ 2 * (elec% - 1) / 2  ' number of current, ground and
                      '   measurement combinations in one projection
                      '   set; one scan of the channel/gain list is
                      '   made for each electrode combination

icglsize = 2 * slength%   ' size of channel/gain list in number of samples;
                      '   list is scanned once for each electrode
                      '   combination, and specifies two channels
                      '   (Ch. 0 = carrier, Ch. 1 = quadrature)

isamptot& = CLNG((icombs - 1)) * CLNG(icglsize) ' total samples in one
                      '   projection set, minus one electrode combination
                      '   (in fast mode, system never sends a trigger
                      '   pulse for the final combination)

iblksize = 32         ' number of kbytes per extended memory block
                      '   (this number found to work best through
                      '   trial and error; required to be a multiple
                      '   of 2)
```

```
DO

'*********************************************************************
' Initialize system for data acquisition

  INPUT "Enter the number of consecutive projections (1 to 100): ", proj%
  IF proj% < 1 THEN proj% = 1
  IF proj% > 100 THEN proj% = 100

  INPUT "Do you wish to change the amplifier gain from the default (Y/N) (default N)?
       ", gainset$

  IF gainset$ = "Y" OR gainset$ = "y" THEN

    PRINT "Input the index for the amplifier gain:"
    PRINT "   N = 0       Gain =    1"
    PRINT "       1                10"
    PRINT "       2               100"
    PRINT "       3              1000"
    INPUT "Recommended value is N = 1 (default). ", igainpwr

    IF igainpwr < 0 OR igainpwr > 3 THEN igainpwr = 1

  ELSE

    igainpwr = 1

  END IF

  gain% = 32 * igainpwr

'*********************************************************************
' Initialize and reset the DT283X board
'*** SADD and VARSEG return addresses, not string expresions themselves

  idrivername$ = "DT283X$0" + CHR$(0)
  iunit = 0       'Board is unit 0

  istat = dt.initialize(SADD(idrivername$), VARSEG(idrivername$), ihandle)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.initialize")

  istat = dt.ct31.initialize(ihandle)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.ct31.initialize")

  istat = dt.reset(iunit, ihandle)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.reset")

'*********************************************************************
' Set the board parameters

  board.dmachannel1 = 5              'DMA channel 1 selection
  board.dmachannel2 = 6              'DMA channel 2 selection
  board.interruptlevel = 10          'interrupt level
  board.boardtimeout = 10            '10-second wait before timeout error returned
'  board.boardtimeout = 0             'infinite patience
  board.adsetupbits = ADTPOL + ADTRIGGER + ADSCANENABLE
                               'a-to-d setup codes, external triggered
                               '  scan (see 'dtst_tls.bi' for codes)
  board.dasetupbits = NODACS         'd-to-a setup code (no d-to-a)

  istat = dt.set.board(iunit, ihandle, board)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.set.board")
```

```
'*********************************************************************
' Create extended memory blocks and buffers

 isection = ADSECTION

 istat = dt.xm.initialize
 IF istat <> 0 THEN CALL exit.error(istat, "dt.xm.initialize")

' Select trial number of data samples per buffer (two bytes per sample;
'   best initial guess for buffer size arrived at by trial and error)

 IF elec% = 16 THEN
   isamptrial = 4 * 1024
 ELSE
   isamptrial = 2 * 1024
 END IF

' Allocate arrays:
'   ixmhdl() = extended memory (XM) block handle
'   ihdl() = buffer handle
'   xmaddr&() = memory address of XM block
'   isamp(i) = number of samples in buffer i
'   ibuff(,) = conventional memory array to receive data from XM buffers

 DIM ixmhdl(1 TO 16), ihdl(1 TO 16), xmaddr&(1 TO 16), isamp(1 TO 16),
     ibuff(isamptrial, 1 TO 16)

 ihdlcnt = 1            ' tally of buffer handles
 isampcnt& = 0          ' tally of samples for which buffer space has been
                        '   allocated

 isamp(ihdlcnt) = isamptrial

' Begin allocation

 DO

' Allocate extended memory block, get its info, and lock it

   istat = dt.xm.allocate.block(iblksize, ixmhdl(ihdlcnt))
   IF istat <> 0 AND istat <> 24579 THEN CALL exit.error(istat,
     "dt.xm.allocate.block")

   istat = dt.xm.get.emb.info(ixmhdl(ihdlcnt), ilockcnt, ifrembs, iblklen) '***
     diagnostic
'   PRINT "Lock flag, remaining free XM blocks, block length in kbytes, istat =" '***
     diagnostic
'   PRINT ilockcnt, ifrembs, iblklen, istat '*** diagnostic

   istat = dt.xm.lock.block(ixmhdl(ihdlcnt), xmaddr&(ihdlcnt))
   IF istat <> 0 THEN CALL exit.error(istat, "dt.xm.lock.block")

' Create a single buffer from the extended memory block; if the buffer
'   crosses a DMA page boundary, recreate the buffer using the maximum
'   buffer size that fits within the page

   istat = dt.xm.create.buffer(iunit, isection, isamp(ihdlcnt), xmaddr&(ihdlcnt),
     ihdl(ihdlcnt))
   IF istat = 12294 THEN
     isamp(ihdlcnt) = isamp(ihdlcnt) - (isamp(ihdlcnt) MOD icglsize)
     istat = dt.xm.create.buffer(iunit, isection, isamp(ihdlcnt), xmaddr&(ihdlcnt),
      ihdl(ihdlcnt))
     IF istat <> 0 THEN
```

```
       PRINT "Error during re-creation of buffer #"; ihdlcnt
       CALL exit.error(istat, "dt.xm.create.buffer")
      END IF
    ELSEIF istat <> 0 THEN
      PRINT "Error during creation of buffer #"; ihdlcnt
      CALL exit.error(istat, "dt.xm.create.buffer")
    END IF
'    PRINT "Buffer "; ihdlcnt; " created with "; isamp(ihdlcnt); " samples" '***
      diagnostic

    isampcnt& = isampcnt& + CLNG(isamp(ihdlcnt))
'    PRINT "Total samples allocated = "; isampcnt& '*** diagnostic
'    INPUT resp$

' Choose a trial size for the next buffer, based on number of samples
'   still to be allocated

    IF isampcnt& < isamptot& AND (isamptot& - isampcnt&) >= CLNG(isamptrial) THEN
      ihdlcnt = ihdlcnt + 1
      isamp(ihdlcnt) = isamptrial
    ELSEIF isampcnt& < isamptot& AND (isamptot& - isampcnt&) < CLNG(isamptrial) THEN
      ihdlcnt = ihdlcnt + 1
      isamp(ihdlcnt) = CINT(isamptot& - isampcnt&)
    ELSEIF isampcnt& = isamptot& THEN
'      PRINT "All buffers created" '*** diagnostic
    ELSE
      PRINT "Error in creation of buffers: samples created, samples required = "
      PRINT isampcnt&, isamptot&
      CALL exit.error(666, "MAIN")
    END IF

  LOOP UNTIL isampcnt& >= isamptot&

'*** diagnostic print of XM block info

'  PRINT "XM block", "handle" '***
'  FOR iii = 1 TO ihdlcnt '***
'    PRINT iii, ixmhdl(iii) '***
'  NEXT iii '***
'  INPUT resp$

'**********************************************************************
' Create a channel/gain list (since the board can apply a different
'   gain to each channel)
'
'       DT2839 channel gains:   gain    igain()=
'                                 1        0
'                                 2        1
'                                 4        2
'                                 8        3
'
'   Board output range is set at +/- 10V.

  DIM ichan(icglsize - 1)              ' channel array for channel/gain table
  DIM igain(icglsize - 1)              ' gain array for channel/gain table
'  PRINT "Channel/gain table dimensioned..." '*** diagnostic

  FOR jj = 1 TO slength%
    ichan(2 * jj - 2) = 0: ichan(2 * jj - 1) = 1
    igain(2 * jj - 2) = 0: igain(2 * jj - 1) = 0
  NEXT jj

' Subroutine 'dt.create.cgl' expects ichan() and igain() arrays to be
'   passed by reference, not by value, so last two arguments must be the
'   first elements in each array

  istat = dt.create.cgl(icglsize, ichan(0), igain(0))
  IF istat <> 0 THEN CALL exit.error(istat, "dt.create.cgl")

'**********************************************************************
' Set the acquisition parameters
'*** Constants defined in library 'dtst_tls.bi' --- dlg, 7/97

' Clock rate must be reduced from 416 kHz to 320 kHz when channel gain > 1.
'   If value of arate is not a possible conversion rate, driver rounds up
'   to the next highest possible rate.

  IF igain(0) = 0 AND igain(1) = 0 THEN
    arate = 400000  ' rounded up to 416000
  ELSE
    arate = 300000  ' rounded up to 320000
  END IF

  digitalio.command = SETXFER
  digitalio.direction = DIOOUTPUT

  sap.section = isection
  sap.transfertype = BDUALTRIGSCAN
  sap.clockrate = arate
  sap.cutoff = 0

  istat = dt.set.acq(iunit, ihandle, isection, sap)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.set.acq")

'**********************************************************************
' Initialize raw data arrays

  DIM carrsum&(1 TO (elec% - 1), 2 TO elec%, 1 TO elec%) ' sums of carrier voltages
  DIM quadsum&(1 TO (elec% - 1), 2 TO elec%, 1 TO elec%) ' sums of quadrature voltages
  DIM Vsum!(1 TO 100, 1 TO elec%) ' sums of off-centerline electrode voltages
                              '   for 180-degree injection/ground cases
                              '   indexed by projection and relative
                              '   location
'  PRINT "carrsum&, quadsum&, Vsum! dimensioned..." '*** diagnostic

  FOR icurrent = 1 TO elec% - 1         'icurrent = index of current injection
       electrode
    FOR iground = icurrent + 1 TO elec% 'iground = index of current return electrode
      FOR ivolt2 = 1 TO elec%           'ivolt2 = index of voltage meas. electrode
       carrsum&(icurrent, iground, ivolt2) = 0
       quadsum&(icurrent, iground, ivolt2) = 0
      NEXT ivolt2
    NEXT iground
  NEXT icurrent

  FOR i = 1 TO proj%
    FOR ivolt2 = 1 TO elec%
      Vsum!(i, ivolt2) = 0!
    NEXT ivolt2
  NEXT i

'**********************************************************************
'
' Arrangement of ports which communicate to the EIT electronics in
'   fast mode:
'
```

```
' Port 0 Bit:  00      01      02      03      04      05      06      07
'            |count | 16/8 |  not used      | mux  |   latch address,     |
'            |start | flag |                |enable|  counter controls    |
'
' Port 1 Bit:  10      11      12      13      14      15      16      17
'            | slow/|       not used        | mux  | PGA202 gain | not   |
'            | fast |                       |enable|             | used  |
'            | flag |
'
' To send instructions to the correct mux, the latch address I/O lines
'   must be set as follows:
'
'                               Port 0 Bit:   07   06   05  In decimal
' latch current injection & ground counters  L    L    L        0
' latch voltage measurement counters         L    L    H       32
' latch gain mux                             L    H    L       64
' reset electrode counters                   L    H    H       96
' start electrode counters                   H    L    L      128
'
'   To enable the current muxes, bit 04 must be set high (16 in decimal)
'   as the current counters are latched; likewise for bit 14 and the
'   voltage muxes.  Bit 10 must be set high (1 in decimal) to start the
'   fast electrode counters.
'
' To select the number of electrodes, the following bit must be set
'   at the same time the electrodes counters are reset and started:
'
' Port 0 Bit:      01    In decimal  Electrodes
'                   L        0           16
'                   H        2            8
'
' To set the PGA202 amplifier gain, I/O lines must be set as follows:
'
' Port 1 Bit:      16     15     In decimal  Gain
'                   L      L         0          1
'                   L      H        32         10
'                   H      L        64        100
'                   H      H        96       1000
'
'*****

' Send commands to set gain mux and activate fast hardware counters.  The
'   'dt.ct31.gate.delay' command sends a pulse on CLK1 to "latch" the
'   gain mux (i.e., prompt it to accept instructions).

  digitalio.dioport = PORT0
  digitalio.diovalue = gainlatch%
  istat = dt.set.dio(iunit, ihandle, digitalio)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.set.dio (PGA202)")

  digitalio.dioport = PORT1
  digitalio.diovalue = fastflag% + gain%
  istat = dt.set.dio(iunit, ihandle, digitalio)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.set.dio (gain value)")

  istat = dt.ct31.gate.delay(iunit, 0, 0, 1, 2)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.ct31.gate.delay (PGA202)")

' Send commands to latch current counters and enable current muxes

  digitalio.dioport = PORT0
  digitalio.diovalue = currlatch% + enable%
  istat = dt.set.dio(iunit, ihandle, digitalio)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.set.dio (current mux PORT 0)")

  digitalio.dioport = PORT1
  digitalio.diovalue = enable%
  istat = dt.set.dio(iunit, ihandle, digitalio)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.set.dio (current mux PORT 1)")

  istat = dt.ct31.gate.delay(iunit, 0, 0, 1, 2)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.ct31.gate.delay (current muxes)")

' Send commands to latch voltage counters and enable voltage muxes

  digitalio.dioport = PORT0
  digitalio.diovalue = voltlatch% + enable%
  istat = dt.set.dio(iunit, ihandle, digitalio)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.set.dio (voltage mux PORT 0)")

  digitalio.dioport = PORT1
  digitalio.diovalue = enable%
  istat = dt.set.dio(iunit, ihandle, digitalio)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.set.dio (voltage mux PORT 1)")

  istat = dt.ct31.gate.delay(iunit, 0, 0, 1, 2)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.ct31.gate.delay (voltage muxes)")

'******************* START OF DATA ACQUISITION LOOP *******************

  CLS
  PRINT "Acquiring projections...": PRINT

'  OPEN "d:\data\diagnose.dat" FOR OUTPUT AS #2 '*** diagnostic dump file

  FOR projloop% = 1 TO proj%

' Get clock time at start of projection routine

    starttime! = TIMER

' Send command to reset address counters (LS163's) and choose number of
'   electrodes.  The 'dt.ct31.gate.delay' call sends a pulse on CLK1 to
'   latch the LS163's.

    digitalio.dioport = PORT0
    digitalio.diovalue = resetcntr%
    istat = dt.set.dio(iunit, ihandle, digitalio)
    IF istat <> 0 THEN CALL exit.error(istat, "dt.set.dio (reset counters)")

    istat = dt.ct31.gate.delay(iunit, 0, 0, 1, 2)
    IF istat <> 0 THEN CALL exit.error(istat, "dt.ct31.gate.delay (reset counters)")

' Reset A/D buffers

    FOR iii = 1 TO ihdlcnt
      istat = dt.reset.buffer(ihdl(iii))
      IF istat <> 0 THEN CALL exit.error(istat, "dt.reset.buffer")
    NEXT iii

' Start fast hardware counters

    digitalio.dioport = PORT0
    digitalio.diovalue = startcntr%
    istat = dt.set.dio(iunit, ihandle, digitalio)
    IF istat <> 0 THEN CALL exit.error(istat, "dt.set.dio (start counters)")
    istat = dt.ct31.gate.delay(iunit, 0, 0, 1, 2)
    IF istat <> 0 THEN CALL exit.error(istat, "dt.ct31.gate.delay (start counters)")

' Start the A/D process in the background

    istat = dt.start.acq(iunit, ihandle, isection)
'    PRINT " acq istat   ", istat '*** diagnostic
    IF istat <> 0 AND istat <> 1 THEN
      istat2 = dt.stop.acq(iunit, ihandle, isection)
      CALL exit.error(istat, "dt.start.acq")
    END IF

' Wait until all buffers are inactive; exit on any unexpected error code

'    prevhdl% = 0 '*** diagnostic setup

    DO
      istat = dt.check.buffer(iunit, isection, bufhdl%, bufstat%)

'      IF bufhdl% <> prevhdl% OR istat <> 1 THEN '*** diagnostic prints
'        PRINT #2, istat, bufhdl%, bufstat% '***
'        prevhdl% = bufhdl% '***
'      END IF '***

      IF istat > 1 AND istat <> 3 THEN
       istat2 = dt.stop.acq(iunit, ihandle, isection)
       CALL exit.error(istat, "dt.check.buffer")
      END IF
    LOOP UNTIL istat = 3

'    PRINT #2, "Buffer I/O complete" '*** diagnostic

' Release all buffers for transfer to conventional memory

    istat = dt.wait.buffer(iunit, isection, bufhdl%)
    IF istat <> 0 THEN CALL exit.error(istat, "dt.wait.buffer")

'    INPUT resp$

' Transfer buffer contents to DOS-accessible conventional memory; note
'   that buffers are used Last-In, First-Out

'    PRINT "Ready to transfer buffer to DOS memory..." '*** diagnostic
    FOR iii = ihdlcnt TO 1 STEP -1
      istat = dt.xm.move.xm.to.dos(ixmhdl(iii), 0&, ibuff(0, iii), CLNG(isamp(iii) *
       2))
'      PRINT "Buffer "; iii; " moved to DOS"'*** diagnostic
      IF istat <> 0 THEN CALL exit.error(istat, "dt.xm.move.xm.to.dos")
    NEXT iii

'*** diagnostic disk dump of averages from each channel/gain scan
'    FOR iii = ihdlcnt TO 1 STEP -1
'      FOR jjj = 0 TO isamp(iii) - 1 STEP icglsize
'        ocave& = 0: oqave& = 0
'        FOR kkk = 0 TO slength% - 1
'          ocave& = ocave& + ibuff(((2 * kkk) + jjj), iii)
'          oqave& = oqave& + ibuff(((2 * kkk + 1) + jjj), iii)
'        NEXT kkk
'        PRINT #2, iii, jjj, ocave& / CLNG(slength%), oqave& / CLNG(slength%)
'      NEXT jjj
'    NEXT iii
```

```
'*** diagnostic dump of raw contents of buffers (disk space hog!)
'    FOR iii = ihdlcnt TO 1 STEP -1
'      FOR jjj = 0 TO isamp(iii) - 1 STEP 2
'        PRINT #2, iii, jjj, ibuff(jjj, iii), ibuff(jjj + 1, iii)
'      NEXT jjj
'    NEXT iii

' Oversampled voltage measurements have been sent to DOS buffer arrays,
'   alternating between carrier and quadrature values; measurements are
'   now stripped from the buffers and placed in the 'carrsum&' and
'   'quadsum&' arrays.  Each channel/gain list scan of 2 * slength%
'   data points is associated with a distinct electrode combination.

    icurrent = 1
    iground = 2
    ivolt2 = 1

    FOR iii = ihdlcnt TO 1 STEP -1
      FOR jjj = 0 TO isamp(iii) - 1 STEP (slength% * 2)

       ocave& = 0: oqave& = 0

       FOR kkk = 0 TO slength% - 1
         ocave& = ocave& + ibuff(((2 * kkk) + jjj), iii)
         oqave& = oqave& + ibuff(((2 * kkk + 1) + jjj), iii)
       NEXT kkk

       carrsum&(icurrent, iground, ivolt2) = ocave& + carrsum&(icurrent, iground,
       ivolt2)
       quadsum&(icurrent, iground, ivolt2) = oqave& + quadsum&(icurrent, iground,
       ivolt2)

' Calculate average voltages for each projection set for each case where
'   the current injection and ground are 180 degrees opposed.  There are
'   (elec%/2) cases per projection set, and the voltages are recorded and
'   averaged for each electrode.

       IF iground = (icurrent + (elec% / 2)) THEN
         k = ivolt2 - icurrent + 1
         IF k < 1 THEN k = elec% + 1 - icurrent + ivolt2
         Vsum!(projloop%, k) = Vsum!(projloop%, k) + SQR(CSNG(ocave& ^ 2 + oqave& ^
       2))
       END IF

' Advance electrode indices for next scan's worth of data

       ivolt2 = ivolt2 + 1
       IF ivolt2 > elec% THEN
         ivolt2 = 1
         iground = iground + 1
         IF iground > elec% THEN
           icurrent = icurrent + 1
           iground = icurrent + 1
         END IF
       END IF

      NEXT jjj
    NEXT iii

' Copy data from similar case to "fudge" data for last electrode combination
'   (recall that in fast mode, the electronics shut down before the trigger
'   pulse for the last electrode combination makes it to the DT card)
```

```
    carrsum&(elec% - 1, elec%, elec%) = carrsum&(elec% - 2, elec% - 1, elec% - 1)
    quadsum&(elec% - 1, elec%, elec%) = quadsum&(elec% - 2, elec% - 1, elec% - 1)

    FOR k = 1 TO elec%
      Vsum!(projloop%, k) = Vsum!(projloop%, k) / CSNG((elec% / 2) * (slength%))
    NEXT k

' Record time at which projection work ends

    endtime = TIMER

    PRINT USING " Projection ### of ### acquired in ##.## seconds"; projloop%; proj%;
       endtime - starttime!
' proj% on previous line added by TJO 11/6/96
    PRINT USING " Mean Cross Electric Voltage, 1 to #, projection ### = #####.## ";
       (elec% / 2 + 1); projloop%; Vsum!(projloop%, 1)
    PRINT

  NEXT projloop%

'  CLOSE #2 '*** diagnostic dump file

'*********************************************************************
' Clean up extended memory and buffers, terminate communication with
'   DT board, and erase arrays no longer needed

  CALL cleanup(dummy)

  istat = dt.terminate(ihandle)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.terminate")

  ERASE ixmhdl, ihdl, xmaddr, isamp, ibuff
  ERASE ichan, igain

  INPUT " Hit <return> to continue. ", resp$

'*********************************************************************
' Call subroutines to compute voltage statistics and output results

  CALL eistats(elec%, proj%, slength%, carrsum&(), quadsum&(), Vsum!(), resp$)

  ERASE carrsum&, quadsum&, Vsum!

LOOP WHILE resp$ = "C" OR resp$ = "c"

PRINT : PRINT " Program stop.": PRINT

CLOSE
END

REM $STATIC
SUB cleanup (dummy)
' This subroutine unlocks and frees extended memory, deletes buffer
'   transfer lists, and stops the DT board's counter.  It can be called
'   after normal or abnormal termination of data acquisition.

SHARED iunit, isection, ihandle, ihdlcnt
SHARED ixmhdl(), ihdl()

  FOR iii = 1 TO ihdlcnt
'      PRINT "Buffer = "; iii'*** diagnostic
      istat = dt.xm.unlock.block(ixmhdl(iii))
      IF istat <> 0 THEN PRINT "dt.xm.unlock.block, istat = ", istat
      istat = dt.xm.free.block(ixmhdl(iii))
      IF istat <> 0 THEN PRINT "dt.xm.free.block, istat = ", istat
      istat = dt.delete.buffer(iunit, isection, ihdl(iii))
      IF istat <> 0 THEN PRINT "dt.delete.buffer, istat = ", istat
  NEXT iii

  istat = dt.ct31.terminate(ihandle)
  IF istat <> 0 THEN PRINT "dt.ct31.terminate, istat = ", istat

'  PRINT "Cleanup complete." '*** diagnostic

'  CLOSE #2 '*** diagnostic dump file

END SUB

SUB exit.error (status AS INTEGER, message AS STRING)
' This version of 'exit.error' copied from module 'bas_sub.bas' to
'   avoid loading the entire module and to speed up execution of
'   FASTMEM.BAS ---dlg, 9/11/97

  SHARED ihandle
  IF status = 0 THEN EXIT SUB

  PRINT : PRINT "Error "; status; " returned by routine "; message

'  Call subroutine to close buffers and clean up communications
  CALL cleanup(dummy)

  PRINT "Program terminating status="; dt.terminate(ihandle)
  STOP
END SUB
```

Appendix C
EIT Data Acquisition Code `SLOWEIT.BAS`

This Microsoft® QuickBasic™ program functions in a similar manner to FASTEIT.BAS, but initializes the EIT system so that multiplexer commands are issued by the data acquisition card instead of the internal counters. Memory is statically allocated by the code, and voltages are stored in arrays after each projection set is collected, rather than being held in buffers until all projection sets are complete.

```
'***
'***  Steve and Ann's EIT code
'***    Revised 9/19/96 by dlg
'***    Revised 11/1/96 by kas to add symmetry checks
'***    Revised 6/97 by dlg to adapt to eight electrodes
'***    Revised 7/97-8/97 by slc and dlg for speed
'***    Revised 11/97 by dlg to move statistics code to separate module
'***

'$DYNAMIC

DECLARE SUB exit.error (status AS INTEGER, message AS STRING)
DECLARE SUB cleanup (dummy)
DECLARE SUB eistats (elec%, proj%, slength%, carrsum&(), quadsum&(), Vsum!(), resp$)

' Definition files for extended memory manager, error codes, counter/
'   timer subroutines and software tools; these files define several
'   constants used in subroutine calls, which are found in BOLDFACE
'   in the code

'$INCLUDE: 'd:\toolkit\dtst_xmm.bi'
'$INCLUDE: 'd:\toolkit\dtst_err.bi'
'$INCLUDE: 'd:\toolkit\dtst_ctr.bi'
'$INCLUDE: 'd:\toolkit\dtst_tls.bi'

'DECLARE SUB featur (iunit AS INTEGER, ihandle AS INTEGER) ' diagnostic subroutine
'DECLARE SUB putboard (iunit AS INTEGER, ihandle AS INTEGER, idma1 AS INTEGER, idma2
       AS INTEGER, intrpt AS INTEGER, adsetup AS INTEGER, dasetup AS INTEGER, itmout
       AS INTEGER) '*** replaced by direct call to dt.set.board
'DECLARE SUB getboard (iunit AS INTEGER, ihandle AS INTEGER) ' diagnostic subroutine

DIM digitalio AS DIGITALIOSTRUCT
DIM sap AS SETACQPARAMSSTRUCT
DIM board AS BOARDSTRUCT

DEFINT I-N
DEFSNG A-H
DEFLNG O-Z
```

```
' Size of channel/gain table changed so that one pass through the
'   table completely fills the buffer --- slc & dlg, 7/97

CONST blength% = 64        ' this is the buffer length
DIM ibuff(blength% - 1)    ' buffer array
DIM ichan(blength% - 1)    ' channel array for channel/gain table
DIM igain(blength% - 1)    ' gain array for channel/gain table

' Number of electrodes can now be set at 8 or 16 --- dlg, 5/27/97
CLS
PRINT
PRINT "This EIT program sets the individual electrodes as current, ground, or"
PRINT "measurement.  The voltage output of the demodulators is recorded for"
PRINT "analysis. "
PRINT
DO
  INPUT "Enter the number of electrodes (8 or 16): ", elec%
  IF elec% <> 8 AND elec% <> 16 THEN
    PRINT "Incorrect input."
  END IF
LOOP UNTIL elec% = 8 OR elec% = 16

DIM carrsum&(1 TO (elec% - 1), 2 TO elec%, 1 TO elec%) ' sums of carrier voltages
DIM quadsum&(1 TO (elec% - 1), 2 TO elec%, 1 TO elec%) ' sums of quadrature voltages
DIM Vsum!(1 TO 100, 1 TO elec%)    ' sums of off-centerline electrode voltages
                             '   for 180-degree injection/ground cases
                             '   indexed by projection and relative
                             '   location

' Initialize system for data acquisition
' DO...LOOP WHILE added to eliminate bug in output options -- dlg, 5/28/97

DO
  INPUT "Input the number of projections to be taken (1 to 100): ", proj%
  IF proj% < 1 THEN proj% = 1
  IF proj% > 100 THEN proj% = 100

' PGA202 amplifier gain is now adjustable before execution --- dlg, 8/2/96
' Default gain changed to 10 with addition of AC coupling filters; gain
'   of 1 is now too low --- dlg, 8/97

  INPUT "Do you wish to change the amplifier gain from the default (Y/N) (default N)?
      ", gainset$

  IF gainset$ = "Y" OR gainset$ = "y" THEN

    PRINT "Input the index for the amplifier gain:"
    PRINT "   N = 0       Gain =    1"
    PRINT "       1                10"
    PRINT "       2               100"
    PRINT "       3              1000"
    INPUT "Recommended value is N = 1 (default). ", igainpwr

    IF igainpwr < 0 OR igainpwr > 3 THEN igainpwr = 1

  ELSE

    igainpwr = 1

  END IF

'******* Initialize summation and averaging arrays *******
```

```
  FOR icurrent = 1 TO elec% - 1          'icurrent = index of current injection
       electrode
    FOR iground = icurrent + 1 TO elec%  'iground = index of current return electrode
      FOR ivolt2 = 1 TO elec%            'ivolt2 = index of voltage meas. electrode
       carrsum&(icurrent, iground, ivolt2) = 0
       quadsum&(icurrent, iground, ivolt2) = 0
      NEXT ivolt2
    NEXT iground
  NEXT icurrent

  FOR i = 1 TO proj%
    FOR ivolt2 = 1 TO elec%
      Vsum!(i, ivolt2) = 0!
    NEXT ivolt2
  NEXT i

'******* Initialize and reset the DT283X board *******
'*** Diagnostic calls to 'exit.error' added after each 'dt.' call ---
'***   dlg, 8/25/97

  idrivername$ = "DT283X$0" + CHR$(0)
  iunit = 0       'Board is unit 0

' SADD and VARSEG functions pass the address of their arguments;
'   retained from example on p. 24 of software toolkit user's manual

  istat = dt.initialize(SADD(idrivername$), VARSEG(idrivername$), ihandle)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.initialize")
  istat = dt.ct31.initialize(ihandle)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.ct31.initialize")
  istat = dt.reset(iunit, ihandle)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.reset")

' Get the board's feature list
'  CALL featur(iunit, ihandle) '*** diagnostic

' Get/set the board parameters
'  CALL getboard(iunit, ihandle) '*** diagnostic

  board.dmachannel1 = 5             'DMA channel 1 selection
'  board.dmachannel2 = 6             'DMA channel 2 selection '*** removed for single
      DMA
  board.interruptlevel = 10         'interrupt level
  board.boardtimeout = 10           '10-second wait before timeout error returned
  board.adsetupbits = 0             'a-to-d setup code
  board.dasetupbits = NODACS        'd-to-a setup code

  istat = dt.set.board(iunit, ihandle, board)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.set.board")

'  CALL getboard(iunit, ihandle) '*** diagnostic

'******* Set board acquisition parameters *******

  isection = ADSECTION
  isize = blength%

  digitalio.command = SETXFER
  digitalio.direction = DIOOUTPUT

'******* Create a buffer *******
' 'dt.create.buffer' moved out from electrode loop, since it must be
'   created before channel-gain list --- dlg, 8/25/97
```

```
 istat = dt.create.buffer(iunit, isection, isize, ibuff(0), ibhandle)
 IF istat <> 0 THEN CALL exit.error(istat, "dt.create.buffer")

' Create a channel/gain list (since the board can apply a different
'  gain to each D/A channel)
'
'    DT2839 channel gains:   gain    igain()
'                             1       0
'                             2       1
'                             4       2
'                             8       3
'
'  Board range is set at +/- 10V.

 FOR jj = 0 TO 31
  ichan(2 * jj) = 0: ichan(2 * jj + 1) = 1
  igain(2 * jj) = 0: igain(2 * jj + 1) = 0
 NEXT jj

' Subroutine 'dt.create.cgl' expects ichan() and igain() arrays to be
'  passed by reference, not by value, so last two arguments are the
'  first elements in each array, not the arrays themselves
' Hardwired list size replaced by 'blength%' to match buffer and
'  list sizes --- dlg, 8/27/97

 istat = dt.create.cgl(blength%, ichan(0), igain(0))
 IF istat <> 0 THEN CALL exit.error(istat, "dt.create.cgl")

' Clock rate must be reduced from 416 kHz to 320 kHz when channel gain > 1.
'  If value of arate is not a possible conversion rate, driver rounds up
'  to the next highest available rate.

 IF igain(0) = 0 AND igain(1) = 0 THEN
  arate = 400000
 ELSE
  arate = 300000
 END IF

' Transfer mode changed from BDUAL (dual channel DMA) to BSINGLE (single
'  channel DMA) to speed things up --- slc and dlg, 7/97
' Maximum possible trigger rate to scan channel-gain list (CGL) is
'  clockrate/(CGL size + 2).  If sap.trigrate is set too high,
'  'dt.set.acq' will round down to the highest possible rate ---
'  dlg, 8/22/97

 sap.section = isection
 sap.transfertype = BSINGLE
 sap.clockrate = arate
 sap.trigrate = arate / 4
 sap.cutoff = 0

 istat = dt.set.acq(iunit, ihandle, isection, sap)
' PRINT "New clockrate = "; sap.clockrate '*** diagnostic
' PRINT "New trigger rate = "; sap.trigrate '*** diagnostic
 IF istat <> 0 THEN CALL exit.error(istat, "dt.set.acq")

'*****
'
' Rules to set current injection and ground electrodes and voltage
'  amplifier:
'
' There are (N-1)*N/2 possible projections, where N is the number of
'   electrodes.  "icurrent" is the injection electrode, and "iground"
'   is the return electrode.
'
' All linearly independent combinations of electrode pairs will be used.
'   When only eight electrodes are used, leads 9 through 16 must be
'   connected, with 1 through 8 isolated; this is needed for the
'   internal counters to operate in fast mode, so the same convention
'   was placed in this slow version of the code.
'
' Arrangement of ports which communicate to the EIT electronics:
'
' Port 0 Bit:  00     01     02     03     04     05     06     07
'            |      electrode address     |enable|  latch address    |
'
' Port 1 Bit:  10     11     12     13     14     15     16     17
'            |      electrode address     |enable| PGA202 gain | not  |
'                                                              | used |
'
' To ready electrodes for current injection or voltage measurement,
'   the digital I/O lines must be set as follows to commumicate with
'   the correct muxes:
'
'                                         Port 0: Bit 05  Bit 06  Bit 07
'   current injection or ground mux                L       L       L
'   voltage measurement or reference mux           H       L       L
'
' To select an electrode, the following i/o lines must be set:
'
'                                                                 Bit 04
'                          Port 0: Bit 00  Bit 01  Bit 02  Bit 03 (enable)
'   current injection              A0      A1      A2      A3      H
'   voltage reference (ground)     A0      A1      A2      A3      H
'                                                                 Bit 14
'                          Port 1: Bit 10  Bit 11  Bit 12  Bit 13 (enable)
'   current return (ground)        A0      A1      A2      A3      H
'   voltage measurement            A0      A1      A2      A3      H
'
' A0, A1, A2, A3 are set high or low according to the table below:
'
'     A3   A2   A1   A0   Electrode
'      L    L    L    L       1
'      L    L    L    H       2
'      L    L    H    L       3
'      L    L    H    H       4
'      L    H    L    L       5
'      L    H    L    H       6
'      L    H    H    L       7
'      L    H    H    H       8
'      H    L    L    L       9
'      H    L    L    H      10
'      H    L    H    L      11
'      H    L    H    H      12
'      H    H    L    L      13
'      H    H    L    H      14
'      H    H    H    L      15
'      H    H    H    H      16
'
' To set the PGA202 amplifier gain, the following digital i/o lines must
'   be set:
'           Port 0: Bit 05  Bit 06  Bit 07
'                     L       H       L
'           Port 1: Bit 15  Bit 16
'                     A0      A1
'   Port 0 values changed with the addition of fast selection counters
'   to the EIT hardware, 7/97.
'
' A0 and A1 are set high or low according to the table below:
'
'                            Value added to integer which
'       A1    A0      Gain   selects voltage channels
'        L     L        1       0
'        L     H       10      32
'        H     L      100      64
'        H     H     1000      96
'
'*****

'******* Set amplifier gain *******
' This set of commands latches the gain on the PGA202 amplifier and
'   instructs the system not to use the new internal electrode
'   counters.  The '64' in .diovalue selects the gain mux.

  digitalio.dioport = PORT0
  digitalio.diovalue = 64
  istat = dt.set.dio(iunit, ihandle, digitalio)  'sends word to di/o.
  IF istat <> 0 THEN CALL exit.error(istat, "dt.set.dio (PGA202 PORT0)")

  gain% = 32 * igainpwr

  digitalio.dioport = PORT1
  digitalio.diovalue = gain%
  istat = dt.set.dio(iunit, ihandle, digitalio)  'sends word to di/o.
  IF istat <> 0 THEN CALL exit.error(istat, "dt.set.dio (PGA202 PORT1)")

' The following function, dt.ct31.gate.delay, is used here to send
'   a pulse from clock CLK1.  The pulse will force the address decoder to
'   enable the gain latches, which will then set the gain muxes.

  istat = dt.ct31.gate.delay(iunit, 0, 0, 1, 2)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.ct31.gate.delay (PGA202)")

'****************** START OF DATA ACQUISITION LOOP ******************

  PRINT "Acquiring projections...": PRINT

'  OPEN "d:\data\diagnose.dat" FOR OUTPUT AS #2 '*** diagnostic

  FOR projloop% = 1 TO proj%

' Get clock time at start of projection routine

    starttime! = TIMER

' 'icurrent' is the array index for the injection electrode, while
'   'icelec' is the associated electrode number; in the case of eight
'   electrodes, 'icurrent' ranges from 1 to 8, while 'icelec' runs
'   from 9 to 16.  See note under "Rules to set electrodes" above.

    FOR icurrent = 1 TO elec% - 1
      icelec = 16 - elec% + icurrent

'  'iground' and 'igelec' have the same relationship as 'icurrent'
'    and 'icelec'

      FOR iground = icurrent + 1 TO elec%
       igelec = 16 - elec% + iground
```

```
' In the statement (digitalio.diovalue=icelec-1+16+0), icelec is the
'   electrode that is the current injector; '-1' sets the bit pattern
'   correctly, e.g., for current electrode 8, a 7 in binary needs to be
'   sent to the digitalio channels.  The '+16' sets bit 04 to high,
'   enabling the latches.  '+0' is a placeholder which represents
'   the selection of current muxes; see the later statements for
'   voltage muxes, where '+32' appears.

' Write address of injection electrode to port 0

       digitalio.dioport = PORT0
       digitalio.diovalue = icelec - 1 + 16 + 0
       istat = dt.set.dio(iunit, ihandle, digitalio)
       IF istat <> 0 THEN CALL exit.error(istat, "dt.set.dio (current mux PORT 0)")

' Write address of ground electrode to port 1

       digitalio.dioport = PORT1
       digitalio.diovalue = igelec - 1 + 16
       istat = dt.set.dio(iunit, ihandle, digitalio)
       IF istat <> 0 THEN CALL exit.error(istat, "dt.set.dio (current mux PORT 1)")

       istat = dt.ct31.gate.delay(iunit, 0, 0, 1, 2)
       IF istat <> 0 THEN CALL exit.error(istat, "dt.ct31.gate.delay (current mux)")

' The next command is used to set the reference voltage electrode,
'   ivolt1.  The reference electrode is chosen to be the ground electrode.
'   '+32' selects the voltage muxes.

       ivolt1 = igelec

       digitalio.dioport = PORT0
       digitalio.diovalue = ivolt1 - 1 + 16 + 32
       istat = dt.set.dio(iunit, ihandle, digitalio)  'sends word to di/o.
       IF istat <> 0 THEN CALL exit.error(istat, "dt.set.dio (voltage mux (PORT0)")

' This loop is used to choose the electrode for voltage measurement,
'   ivelec.  This electrode goes through all possible values.  'ivolt2'
'   and 'ivelec' have the same relationship as 'icurrent' and 'icelec'.

       FOR ivolt2 = 1 TO elec%
         ivelec = 16 - elec% + ivolt2

         digitalio.dioport = PORT1
         digitalio.diovalue = ivelec - 1 + 16
         istat = dt.set.dio(iunit, ihandle, digitalio)  'sends word to di/o.
         IF istat <> 0 THEN CALL exit.error(istat, "dt.set.dio (voltage mux PORT1)")

         istat = dt.ct31.gate.delay(iunit, 0, 0, 1, 2)
         IF istat <> 0 THEN CALL exit.error(istat, "dt.ct31.gate.delay (voltage mux)")

' Now data must be read from the A/D channels.  A/D Channel 0 is the
'   carrier signal, A/D channel 1 is the quadrature signal.
'
' Reset buffer

         istat = dt.reset.buffer(ibhandle)
         IF istat <> 0 THEN CALL exit.error(istat, "dt.reset.buffer")

' Take multiple measurements and average output values for in-phase
'   and quadrature signals.  Acquire blength%/2 samples on each channel.
```

```
         istat = dt.start.acq(iunit, ihandle, isection)
         IF istat <> 0 AND istat <> 1 THEN
           CALL exit.error(istat, "dt.start.acq")
           istat = dt.stop.acq(iunit, ihandle, isection)
         END IF

'*** Diagnostic check of buffer added --- dlg, 8/27/97
'recheck:
'          istat = dt.check.buffer(iunit, isection, ibhandle, ibufstat)
'          IF ibufstat <> 194 THEN PRINT "Buffer status = "; ibufstat
'          IF ibufstat = 194 THEN GOTO recheck

' Check that buffer has been processed
' Buffer reset commented out to ensure data survives until it is
'   read --- dlg, 8/27/97

         istat = dt.wait.buffer(iunit, isection, ibhandle)
         IF istat <> 0 THEN CALL exit.error(istat, "dt.wait.buffer")

'          istat = dt.reset.buffer(ibhandle)
'          PRINT "Handle of reset buffer = "; ibhandle
'          IF istat <> 0 THEN CALL exit.error(istat, "dt.reset.buffer")

' Oversampled voltage measurements have been sent to the buffer, alternating
'   between carrier and quadrature values, blength%/2 of each; measurements
'   are now stripped from the buffer and placed in the 'carrsum&' and
'   'quadsum&' arrays

         ocave& = 0: oqave& = 0

         FOR ib = 1 TO blength% / 2
'*** diagnostic prints added
'            PRINT #2, USING "& &"; HEX$(ibuff(2 * ib - 2) + 2048); HEX$(ibuff(2 * ib
       - 1) + 2048)'***
'            PRINT "carrier value = "; ibuff(2 * ib - 2), '***
'            PRINT "quadrature value = "; ibuff(2 * ib - 1) '***
           ocave& = ocave& + ibuff(2 * ib - 2)
           oqave& = oqave& + ibuff(2 * ib - 1)
         NEXT ib

'          WRITE #2, projloop%, icurrent, iground, ivolt2, (ocave& / (blength% / 2)),
       (oqave& / (blength% / 2))'***diagnostic
'          WRITE #2, icurrent, iground, ivolt1, ivolt2, icelec, igelec, ivelec,
       (ocave& / (blength% / 2)), (oqave& / (blength% / 2))'***diagnostic

         carrsum&(icurrent, iground, ivolt2) = ocave& + carrsum&(icurrent, iground,
       ivolt2)
         quadsum&(icurrent, iground, ivolt2) = oqave& + quadsum&(icurrent, iground,
       ivolt2)

' Calcuate average voltages for each projection for each case where
'   the current injection and ground are 180 degrees opposed. There are
'   (elec%/2) cases per projection and the voltages are recorded and
'   averaged for each electrode.

         IF iground = (icurrent + (elec% / 2)) THEN
           k = ivolt2 - icurrent + 1
           IF k < 1 THEN k = elec% + 1 - icurrent + ivolt2
           Vsum!(projloop%, k) = Vsum!(projloop%, k) + SQR(CSNG(ocave& ^ 2 + oqave& ^
       2))

'            PRINT USING "ocave(##) = ######   oqave(##) = ######"; k; ocave& /
       (blength% / 2); k; oqave& / (blength% / 2) '***diagnostic
```

```
        END IF

      NEXT ivolt2
     NEXT iground
   NEXT icurrent

   FOR k = 1 TO elec%
     Vsum!(projloop%, k) = Vsum!(projloop%, k) / CSNG((elec% / 2) * (blength% / 2))
   NEXT k

' Record time at which projection work ends

   endtime = TIMER

   PRINT USING " Projection ### of ### acquired in ##.## seconds"; projloop%; proj%;
      endtime - starttime!
' proj% printout on previous line added by TJO 11/6/96

   PRINT USING " Mean Cross Electric Voltage, injection at 1 = #####.## ";
      Vsum!(projloop%, 1)
   PRINT

  NEXT projloop%

'  CLOSE #2 '*** diagnostic file

' Set bit 04 to zero to disable muxes and shut off current injection at
'   electrodes --- dlg, 11/6/97

  digitalio.dioport = PORT0
  digitalio.diovalue = 0
  istat = dt.set.dio(iunit, ihandle, digitalio)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.set.dio (current mux PORT 0)")

  digitalio.dioport = PORT1
  digitalio.diovalue = 0
  istat = dt.set.dio(iunit, ihandle, digitalio)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.set.dio (current mux PORT 1)")

  istat = dt.ct31.gate.delay(iunit, 0, 0, 1, 2)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.ct31.gate.delay (current mux)")

' Delete buffers and terminate communication with board
' 'dt.delete.buffer' and 'dt.ct31.terminate' moved outside electrode
'   loop, into subroutine where they can be called under normal or
'   abnormal conditions (in 'exit.error') --- dlg, 9/11/97

  CALL cleanup(dummy)

  istat = dt.terminate(ihandle)
  IF istat <> 0 THEN CALL exit.error(istat, "dt.terminate")

  INPUT " Hit <return> to continue. ", resp$

'****************** END OF DATA ACQUISITION LOOP ********************

' Call subroutines to compute voltage statistics and output results

  CALL eistats(elec%, proj%, (blength% / 2), carrsum&(), quadsum&(), Vsum!(), resp$)

' DO...LOOP WHILE added to eliminate bug in output options -- dlg, 5/28/97
```

```
LOOP WHILE resp$ = "C" OR resp$ = "c"

PRINT : PRINT " Program stop.": PRINT

CLOSE
END

REM $STATIC
SUB cleanup (dummy)
SHARED iunit, isection, ibhandle, ihandle

istat = dt.delete.buffer(iunit, isection, ibhandle)
IF istat <> 0 THEN PRINT "dt.delete.buffer, istat = "; istat

istat = dt.ct31.terminate(ihandle)
IF istat <> 0 THEN PRINT "dt.ct31.terminate, istat = "; istat

END SUB
```

Appendix D
EIT Data Acquisition Subroutines

This appendix lists Microsoft® QuickBasic™ subroutines called by FASTEIT.BAS and SLOWEIT.BAS. These subroutines perform statistical checks on electrode voltages, display symmetry and antisymmetry checks for cases where injection and ground electrodes are directly opposite each other, and save the measured voltages in a variety of output formats. The file format used by EIT reconstruction codes is described in Appendix E; other output formats include an ASCII file readable by Microsoft Excel and a Postscript file which prints the normalized electrodes voltages in a graphic "clock face" format.

```
DECLARE SUB datmake (elec%, carr!(), quad!(), volt!(), file$)
DECLARE SUB symmake (elec%, symstd!, sym!(), asymstd!, asym!(), file$)
DECLARE SUB difmake (elec%, carr!(), quad!(), volt!())
DECLARE SUB xlsmake (elec%, volt!())
DECLARE SUB linmake (elec%, std.L!, volt.L!(), file$)
DECLARE SUB psmake (elec%, carr!(), quad!(), volt!())
DECLARE SUB promake (elec%, proj%, Vsum!(), czero!, qzero!, CEVmean!, CEVmax!,
       CEVmin!, CEVstd!, Vmean!, Vmax!, Vmin!, Vstd!, file$)

DEFINT I-N
DEFLNG O-Z



SUB datmake (elec%, carr!(), quad!(), volt!(), file$)

PRINT
PRINT "Input file name ('.dat' is appended automatically) or "
INPUT "type 'ls' to see a listing of current files: ", file$
IF file$ = "ls" OR file$ = "LS" THEN
  FILES "d:\data\*.dat"
  INPUT "Input file name ('.dat' is appended automatically): ", file$
END IF

OPEN "d:\data\" + file$ + ".dat" FOR OUTPUT AS #1
FOR iv1 = 1 TO elec% - 1
  FOR iv2 = iv1 + 1 TO elec%
    FOR iv3 = 1 TO elec%
      PRINT #1, USING "## ## ## ####.## ####.## ####.##"; iv1; iv2; iv3; volt!(iv1,
       iv2, iv3); carr!(iv1, iv2, iv3); quad!(iv1, iv2, iv3)
    NEXT iv3
  NEXT iv2
NEXT iv1
CLOSE #1

PRINT "Data saved to file d:\data\"; file$; ".dat"
```

```
END SUB


SUB difmake (elec%, carr!(), quad!(), volt!())

PRINT
PRINT "Use a reference file that involves the same number of electrodes as"
PRINT "the current data."
INPUT "Input the reference file ('.dat' is appended automatically): ", file$
OPEN "d:\data\" + file$ + ".dat" FOR INPUT AS #3

PRINT
INPUT "Input difference file name ('.dif' is appended automatically): ", file$
OPEN "d:\data\" + file$ + ".dif" FOR OUTPUT AS #1

' Take difference of data values

FOR iv1 = 1 TO elec% - 1
  FOR iv2 = iv1 + 1 TO elec%
    FOR iv3 = 1 TO elec%
      INPUT #3, i1temp%, i2temp%, i3temp%, vtemp!, ctemp!, qtemp!

      IF i3temp% <> iv3 THEN PRINT "Discrepancy in electrode count between files!"

      ctemp! = carr!(iv1, iv2, iv3) - ctemp!
      qtemp! = quad!(iv1, iv2, iv3) - qtemp!
      vtemp! = volt!(iv1, iv2, iv3) - vtemp!

      PRINT #1, USING "## ## ## #####.## #####.## #####.##"; iv1; iv2; iv3; vtemp!;
       ctemp!; qtemp!

    NEXT iv3
  NEXT iv2
NEXT iv1

CLOSE #3
CLOSE #1

PRINT "Difference file saved as d:\data\"; file$; ".dif"

END SUB


SUB eistats (elec%, proj%, slength%, carrsum&(), quadsum&(), Vsum!(), resp$)

DIM Vsum.elc!(1 TO elec%)         ' averages of Vsum!() for each electrode
DIM Vsum2.elc!(1 TO elec%)        ' standard deviations of Vsum!() for each
                                  '   electrode

'*** diagnostic to print free memory
'PRINT "Free memory in bytes = "; FRE(-1)

DIM carr!(1 TO (elec% - 1), 2 TO elec%, 1 TO elec%) 'averages of voltages saved
DIM quad!(1 TO (elec% - 1), 2 TO elec%, 1 TO elec%) '  in real arrays
DIM volt!(1 TO (elec% - 1), 2 TO elec%, 1 TO elec%)

DIM sym!(1 TO elec% / 2, 1 TO 2, 1 TO (elec% / 2 - 1)) 'symmetry data
DIM save!(1 TO elec%)                   'averages of symmetry data
DIM ssum!(1 TO (elec% - 1))             'sums of symmetry data
DIM ssum2!(1 TO (elec% - 1))            'sum of symmetry data squared
DIM asym!(1 TO elec% / 2, 1 TO 4, 1 TO elec% / 4) 'antisymmetry data
DIM asum!(1 TO elec% / 4)               'sums of antisymmetry data
DIM asum2!(1 TO elec% / 4)              'sum of antisymmetry data squared

DIM volt.L!(1 TO (elec% / 2 - 1), 2 TO elec% - 1, 1 TO elec%) 'linearity check arrays
DIM sum2.L!(1 TO (elec% / 2 - 1), 2 TO elec% - 1)

' Subtract off baseline voltage from projections and average;
'   since the ground electrode is used as a voltage reference,
'   carrsum(i,n,n) and quadsum(i,n,n) should equal zero for any
'   i and n (in theory)

  carrzero& = 0: quadzero& = 0

  FOR iv1 = 1 TO elec% - 1
    FOR iv2 = iv1 + 1 TO elec%

      carrzero& = carrzero& + carrsum&(iv1, iv2, iv2)
      quadzero& = quadzero& + quadsum&(iv1, iv2, iv2)

    NEXT iv2
  NEXT iv1

  nz! = (elec%) * (elec% - 1) / 2
  fact! = 1! / CSNG(proj% * slength%)
  czero! = CSNG(carrzero&) * (fact! / nz!)
  qzero! = CSNG(quadzero&) * (fact! / nz!)

  FOR iv1 = 1 TO elec% - 1
    FOR iv2 = iv1 + 1 TO elec%
      FOR iv3 = 1 TO elec%

        carr!(iv1, iv2, iv3) = fact! * CSNG(carrsum&(iv1, iv2, iv3)) - czero!
        quad!(iv1, iv2, iv3) = fact! * CSNG(quadsum&(iv1, iv2, iv3)) - qzero!
        volt!(iv1, iv2, iv3) = SQR(carr!(iv1, iv2, iv3) ^ 2 + quad!(iv1, iv2, iv3) ^
       2)

'        PRINT
'        PRINT "carr, quad, volt   "; carr!(iv1, iv2, iv3), quad!(iv1, iv2, iv3),
       volt!(iv1, iv2, iv3)
'        PRINT


      NEXT iv3
    NEXT iv2
  NEXT iv1

'************* CALCULATE STATISTICS AND OUTPUT TO SCREEN *************

' Calculate mean and extremes of voltages recorded for each projection
'   for 180 degree current injection and ground cases. Also, calculate
'   standard deviation between projections. NOTE: CEV stands for cross
'   electrode voltage and is the voltage between the current and the ground.
'   The other statistics are for the remaining voltages.

  FOR iv3 = 1 TO elec%
    Vsum.elc!(iv3) = 0!
    Vsum2.elc!(iv3) = 0!
  NEXT iv3

  CEVmean! = 0!: CEVmin! = 100000!: CEVmax! = 0!: CEVstd! = 0!
  Vmean! = 0!: Vmin! = 100000!: Vmax! = 0!: Vstd! = 0!

  FOR i = 1 TO proj%
```

```
    IF Vsum!(i, 1) > CEVmax! THEN CEVmax! = Vsum!(i, 1)
    IF Vsum!(i, 1) < CEVmin! THEN CEVmin! = Vsum!(i, 1)
    CEVmean! = CEVmean! + Vsum!(i, 1)
    CEVstd! = CEVstd! + Vsum!(i, 1) ^ 2

    FOR iv3 = 2 TO elec%
      IF iv3 <> (elec% / 2 + 1) THEN
        IF Vsum!(i, iv3) > Vmax! THEN Vmax! = Vsum!(i, iv3)
        IF Vsum!(i, iv3) < Vmin! THEN Vmin! = Vsum!(i, iv3)
        Vsum.elc!(iv3) = Vsum.elc!(iv3) + Vsum!(i, iv3)
        Vsum2.elc!(iv3) = Vsum2.elc!(iv3) + Vsum!(i, iv3) ^ 2
      END IF
    NEXT iv3

  NEXT i

  IF proj% > 1 THEN
    temp! = CSNG(proj%) * CEVstd! - CEVmean! ^ 2
    IF temp! < 0 THEN
      PRINT "ERROR computing standard deviation in CEV"
      PRINT "N sum(Xi^2) - (sum(Xi))^2 = "; temp!
      INPUT "Hit <return> to continue. ", resp$
      CEVstd! = 0!
    ELSE
      CEVstd! = SQR(temp! / CSNG(proj% * (proj% - 1)))
    END IF
    CEVmean! = CEVmean! / CSNG(proj%)
    FOR iv3 = 2 TO elec%
      IF iv3 <> (elec% / 2 + 1) THEN
        temp! = CSNG(proj%) * Vsum2.elc!(iv3) - Vsum.elc!(iv3) ^ 2
        IF temp! < 0 THEN
          PRINT "ERROR computing standard deviation in voltages"
          PRINT "N sum(Xi^2) - (sum(Xi))^2 = "; temp!
          INPUT "Hit <return> to continue. ", resp$
        ELSE
          Vstd! = Vstd! + SQR(temp! / CSNG(proj% * (proj% - 1))) / CSNG(elec% - 2)
        END IF
        Vmean! = Vmean! + Vsum.elc!(iv3) / CSNG((elec% - 2) * proj%)
      END IF
    NEXT iv3
  ELSE
    CEVstd! = 0!
    Vstd! = 0!
    FOR iv3 = 2 TO elec%
      IF iv3 <> (elec% / 2 + 1) THEN
        Vmean! = Vmean! + Vsum.elc!(iv3) / CSNG(elec% - 2)
      END IF
    NEXT iv3
  END IF

' Print data to screen and file

  CLS
  PRINT " Average Data Set Characteristics"
  PRINT
  PRINT USING " Carrier Zero Offset             #####"; czero!
  PRINT USING " Quad Zero Offset                #####"; qzero!
  PRINT
  PRINT " Statistics for cross electrode voltages comparing data"
  PRINT " recorded for each projection:"
  PRINT
  PRINT USING " MEAN Cross Electrode Voltage    #####.##"; CEVmean!
  PRINT USING " MAX Cross Electrode Voltage     #####.##"; CEVmax!
  PRINT USING " MIN Cross Electrode Voltage     #####.##"; CEVmin!
  PRINT USING " STDEV Cross Electrode Voltage    ####.##"; CEVstd!
  PRINT
  PRINT " Statistics for electrode voltages (not including current injection"
  PRINT " and ground electrodes) for 180 degree injection/ground cases"
  PRINT " comparing data recorded for each projection"
  PRINT
  PRINT USING " MEAN Electrode Voltage          #####.##"; Vmean!
  PRINT USING " MAX Electrode Voltage           #####.##"; Vmax!
  PRINT USING " MIN Electrode Voltage           #####.##"; Vmin!
  PRINT USING " STDEV Electrode Voltage          ####.##"; Vstd!
  PRINT
  INPUT " Hit <return> to continue. ", resp$

'*********** CHECK SYMMETRY AND ANTISYMMETRY ************************

  FOR iv1 = 1 TO elec% / 2
    iv2 = iv1 + elec% / 2
    save!(iv1) = 0!
    FOR k = 1 TO (elec% / 2 - 1)
      k1 = iv1 + k
      k2 = iv1 - k
      IF k2 < 1 THEN k2 = elec% - (k - iv1)
      sym!(iv1, 1, k) = volt!(iv1, iv2, k1)
      sym!(iv1, 2, k) = volt!(iv1, iv2, k2)
      save!(iv1) = save!(iv1) + (sym!(iv1, 1, k) + sym!(iv1, 2, k)) / CSNG(elec% - 2)
    NEXT k
  NEXT iv1

  FOR k = 1 TO (elec% / 2 - 1)
    ssum!(k) = 0!
    ssum2!(k) = 0!
  NEXT k
  smax! = 0!: smin! = 10000!
  FOR iv1 = 1 TO elec% / 2
    FOR j = 1 TO 2
      FOR k = 1 TO (elec% / 2 - 1)
        sym!(iv1, j, k) = sym!(iv1, j, k) - save!(iv1)
        IF sym!(iv1, j, k) > smax! THEN smax! = sym!(iv1, j, k)
        IF sym!(iv1, j, k) < smin! THEN smin! = sym!(iv1, j, k)
        ssum!(k) = ssum!(k) + sym!(iv1, j, k)
        ssum2!(k) = ssum2!(k) + sym!(iv1, j, k) ^ 2
      NEXT k
    NEXT j
  NEXT iv1

  symstd! = 0!
  FOR k = 1 TO (elec% / 2 - 1)
    temp! = CSNG(elec%) * ssum2!(k) - ssum!(k) ^ 2
    IF temp! < 0 THEN
      PRINT "ERROR computing standard deviation in symmetry data"
      PRINT "N sum(Xi^2) - (sum(Xi))^2 = "; temp!
      INPUT "Hit <return> to continue. ", resp$
    ELSE
      symstd! = symstd! + SQR(temp! / CSNG(elec% * (elec% - 1))) / CSNG(elec% / 2 - 1)
    END IF
  NEXT k

  FOR k = 1 TO elec% / 4
    asum!(k) = 0!
    asum2!(k) = 0!
  NEXT k
  amin! = 10000!: amax! = 0!
  FOR iv1 = 1 TO elec% / 2
    iv2 = iv1 + elec% / 2
    n1 = (iv1 + iv2) / 2                     'antisymmetry nodes
    IF n1 <= (elec% / 2) THEN
      n2 = n1 + elec% / 2
      C! = 1!
    ELSE
      ntemp = n1
      n1 = n1 - elec% / 2
      n2 = ntemp
      C! = -1!
    END IF
    FOR k = 1 TO elec% / 4
      k1 = n1 - (k - 1)
      IF k1 < 1 THEN k1 = elec% + n1 - (k - 1)
      k2 = n1 + (k - 1)
      k3 = n2 - (k - 1)
      k4 = n2 + (k - 1)
      IF k4 > elec% THEN k4 = (k - 1) - (elec% - n2)
      asym!(iv1, 1, k) = C! * (volt!(iv1, iv2, k1) - save!(iv1))
      asym!(iv1, 2, k) = C! * (save!(iv1) - volt!(iv1, iv2, k2))
      asym!(iv1, 3, k) = C! * (save!(iv1) - volt!(iv1, iv2, k3))
      asym!(iv1, 4, k) = C! * (volt!(iv1, iv2, k4) - save!(iv1))
      FOR j = 1 TO 4
        IF asym!(iv1, j, k) > amax! THEN amax! = asym!(iv1, j, k)
        IF asym!(iv1, j, k) < amin! THEN amin! = asym!(iv1, j, k)
        asum!(k) = asum!(k) + asym!(iv1, j, k)
        asum2!(k) = asum2!(k) + asym!(iv1, j, k) ^ 2
      NEXT j
    NEXT k
  NEXT iv1

  asymstd! = 0!
  FOR k = 1 TO (elec% / 4)
    temp! = CSNG(2 * elec%) * asum2!(k) - asum!(k) ^ 2
    IF temp! < 0 THEN
      PRINT "ERROR computing standard deviation in axisymmetry data"
      PRINT "N sum(Xi^2) - (sum(Xi))^2 = "; temp!
      INPUT "Hit <return> to continue. ", resp$
    ELSE
      asymstd! = asymstd! + SQR(temp! / CSNG((2 * elec%) * (2 * elec% - 1))) /
       CSNG(elec% / 4)
    END IF
  NEXT k

' Display symmetry data to screen

  CLS

  SCREEN 12                        'set screen to 640x480 resolution
  dt.p! = 5!                       'length of tick marks in pixels
  view.p! = 240!                   'viewport length and width in pixels

  x1.p% = (640! - 2! * view.p!) / 3! - 1
  y1.p% = (480! - view.p!) / 2! - 1
  x2.p% = 2 * x1.p% + view.p!
  y2.p% = y1.p%

'****************** SYMMETRY PLOT ********************

' Find limits for y axis
```

```
  ymax! = 10 * CINT((smax! + 10) / 10)
  ymin! = 10 * CINT((smin! - 10) / 10)
  n = CINT((ymax! - ymin!) / 10)
  dy! = (ymax! - ymin!) / n

  VIEW (x1.p%, y1.p%)-(x1.p% + view.p!, y1.p% + view.p!)
  WINDOW (0, ymin!)-(elec% / 2, ymax!)

  LINE (0, ymin!)-(elec% / 2, ymax!), , B        'draw box around graph

  dt.x! = (dt.p! / view.p!) * (ymax! - ymin!)    'calculate length of x-ticks
  FOR i = 1 TO (elec% / 2 - 1)                   'draw x ticks
    LINE (i, ymin!)-(i, ymin! + dt.x!)
  NEXT i

  dt.y! = (dt.p! / view.p!) * CSNG(elec% / 2)    'calculate length of y-ticks
  FOR i = 1 TO n - 1                             'draw y ticks
    y! = ymin! + CSNG(i) * dy!
    LINE (0, y!)-(0 + dt.y!, y!)
  NEXT i

  FOR i = 1 TO elec% / 2
    FOR j = 1 TO 2
      PSET (1, sym!(i, j, 1)), i                 'plot first point
      FOR k = 2 TO (elec% / 2 - 1)
        LINE -(k, sym!(i, j, k)), i              'join remaining points
      NEXT k
    NEXT j
  NEXT i

  COL% = (x1.p% / 640!) * 80 + 1
  ROW% = (y1.p% / 480!) * 30 - 1
  LOCATE ROW%, COL%
  PRINT "SYMMETRY PLOT"
  LOCATE ROW% + 1, COL%
  PRINT USING "(####.## STANDARD DEVIATION)"; symstd!

  COL% = (x1.p% / 640!) * 80 - 5
  ROW% = (y1.p% / 480!) * 30 + 1
  LOCATE ROW%, COL%
  PRINT USING "####"; ymax!

  COL% = (x1.p% / 640!) * 80 - 5
  ROW% = ((y1.p% + view.p!) / 480!) * 30 + 1
  LOCATE ROW%, COL%
  PRINT USING "####"; ymin!

  FOR i = 1 TO (elec% / 2 - 1)
    COL% = ((x1.p% + i * (view.p! / CSNG(elec% / 2))) / 640!) * 80 + 1
    ROW% = ((y1.p% + view.p!) / 480!) * 30 + 2
    LOCATE ROW%, COL%
    PRINT USING "#"; i
  NEXT i

'****************** ANTISYMMETRY PLOT *******************

' Find limits for y axis

  ymax! = 10 * CINT((amax! + 10) / 10)
  ymin! = 10 * CINT((amin! - 10) / 10)
  n = CINT((ymax! - ymin!) / 10)
  dy! = (ymax! - ymin!) / n
  VIEW (x2.p%, y2.p%)-(x2.p% + view.p!, y2.p% + view.p!)
  WINDOW (0, ymin!)-((elec% / 4 + 1), ymax!)

  LINE (0, ymin!)-((elec% / 4 + 1), ymax!), , B  'draw box around graph

  dt.x! = (dt.p! / view.p!) * (ymax! - ymin!)     'calculate length of x-ticks
  FOR i = 1 TO elec% / 4                          'draw x ticks
    LINE (i, ymin!)-(i, ymin! + dt.x!)
  NEXT i

  dt.y! = (dt.p! / view.p!) * CSNG(elec% / 4 + 1) 'calculate length of y-ticks
  FOR i = 1 TO n - 1                              'draw y ticks
    y! = ymin! + CSNG(i) * dy!
    LINE (0, y!)-(0 + dt.y!, y!)
  NEXT i

  FOR i = 1 TO elec% / 2
    FOR j = 1 TO 4
      PSET (1, asym!(i, j, 1)), i                 'plot first point
      FOR k = 2 TO elec% / 4
        LINE -(k, asym!(i, j, k)), i              'join remaining points
      NEXT k
    NEXT j
  NEXT i

  COL% = (x2.p% / 640!) * 80 + 1
  ROW% = (y2.p% / 480!) * 30 - 1
  LOCATE ROW%, COL%
  PRINT "ANTISYMMETRY PLOT"
  LOCATE ROW% + 1, COL%
  PRINT USING "(####.## STANDARD DEVIATION)"; asymstd!

  COL% = (x2.p% / 640!) * 80 - 5
  ROW% = (y2.p% / 480!) * 30 + 1
  LOCATE ROW%, COL%
  PRINT USING "####"; ymax!

  COL% = (x2.p% / 640!) * 80 - 4
  ROW% = ((y2.p% + view.p!) / 480!) * 30 + 1
  LOCATE ROW%, COL%
  PRINT USING "###"; ymin!

  FOR i = 1 TO elec% / 4
    COL% = ((x2.p% + i * (view.p! / CSNG(elec% / 4 + 1))) / 640!) * 80 + 1
    ROW% = ((y2.p% + view.p!) / 480!) * 30 + 2
    LOCATE ROW%, COL%
    PRINT USING "#"; i
  NEXT i

  LOCATE 28, 1
  INPUT " Hit <return> to continue. ", resp$

'**************** CHECK LINEARITY OF DATA ******************

  max.L! = 0!: min.L! = 10000!
  FOR iv1 = 1 TO elec% / 2 - 1
    FOR iv2 = iv1 + 1 TO elec% - iv1
      avg.L! = 0!
      FOR iv3 = 1 TO elec%
        IF iv3 <> iv1 AND iv3 <> iv2 THEN
          temp! = volt!(iv1, elec% + 1 - iv1, iv3) - volt!(iv2, elec% + 1 - iv1, iv3)
          volt.L!(iv1, iv2, iv3) = volt!(iv1, iv2, iv3) - temp!
          avg.L! = avg.L! + volt.L!(iv1, iv2, iv3) / CSNG(elec% - 2)
        END IF
      NEXT iv3
      sum2.L!(iv1, iv2) = 0!
      FOR iv3 = 1 TO elec%
        IF iv3 <> iv1 AND iv3 <> iv2 THEN
          volt.L!(iv1, iv2, iv3) = volt.L!(iv1, iv2, iv3) - avg.L!
          IF volt.L!(iv1, iv2, iv3) > max.L! THEN max.L! = volt.L!(iv1, iv2, iv3)
          IF volt.L!(iv1, iv2, iv3) < min.L! THEN min.L! = volt.L!(iv1, iv2, iv3)
          sum2.L!(iv1, iv2) = sum2.L!(iv1, iv2) + volt.L!(iv1, iv2, iv3) ^ 2
        ELSE
          volt.L!(iv1, iv2, iv3) = 0!
        END IF
      NEXT iv3
    NEXT iv2
  NEXT iv1
  std.L! = 0!
  idenom = 0
  FOR iv1 = 1 TO elec% / 2 - 1
    idenom = idenom + iv1
  NEXT iv1
  denom = CSNG((elec% * (elec% / 2 - 1)) - 2 * idenom)
  FOR iv1 = 1 TO elec% / 2 - 1
    FOR iv2 = iv1 + 1 TO elec% - iv1
      std.L! = std.L! + SQR(sum2.L!(iv1, iv2) / CSNG(elec% - 3)) / denom
    NEXT iv2
  NEXT iv1

'****************** LINEARITY PLOT ********************

  CLS 0
  dt.p! = 5!                       'length of tick marks in pixels
  view.p! = 240!                   'viewport length and width/2 in pixels

  xoff.p% = (640! - 2! * view.p!) / 2! - 1
  yoff.p% = (480! - view.p!) / 2! - 1

' Find limits for y axis

  ymax! = 10 * CINT((max.L! + 10) / 10)
  ymin! = 10 * CINT((min.L! - 10) / 10)
  n = CINT((ymax! - ymin!) / 10)
  dy! = (ymax! - ymin!) / n

  VIEW (xoff.p%, yoff.p%)-(xoff.p% + 2 * view.p!, yoff.p% + view.p!)
  WINDOW (0, ymin!)-(elec% + 1, ymax!)

  LINE (0, ymin!)-(elec% + 1, ymax!), , B         'draw box around graph

  dt.x! = (dt.p! / view.p!) * (ymax! - ymin!)     'calculate length of x-ticks
  FOR i = 1 TO elec%                              'draw x ticks
    LINE (i, ymin!)-(i, ymin! + dt.x!)
  NEXT i

  dt.y! = (dt.p! / (2 * view.p!)) * CSNG(elec% + 1) 'calculate length of y-ticks
  FOR i = 1 TO n - 1                              'draw y ticks
    y! = ymin! + CSNG(i) * dy!
    LINE (0, y!)-(0 + dt.y!, y!)
  NEXT i

  FOR iv1 = 1 TO elec% / 2 - 1
    FOR iv2 = iv1 + 1 TO elec% - iv1
      PSET (1, volt.L!(iv1, iv2, 1)), iv1         'plot first point
      FOR iv3 = 2 TO elec%
        LINE -(iv3, volt.L!(iv1, iv2, iv3)), iv1  'join remaining points
      NEXT iv3
    NEXT iv2
  NEXT iv1

  COL% = (xoff.p% / 640!) * 80 + 1
  ROW% = (yoff.p% / 480!) * 30 - 1
  LOCATE ROW%, COL%
  PRINT "LINEARITY PLOT"
  LOCATE ROW% + 1, COL%
  PRINT USING "(####.## STANDARD DEVIATION)"; std.L!

  COL% = (xoff.p% / 640!) * 80 - 5
  ROW% = (yoff.p% / 480!) * 30 + 1
  LOCATE ROW%, COL%
  PRINT USING "####"; ymax!

  COL% = (xoff.p% / 640!) * 80 - 4
  ROW% = ((yoff.p% + view.p!) / 480!) * 30 + 1
  LOCATE ROW%, COL%
  PRINT USING "###"; ymin!

  FOR i = 1 TO elec%
    COL% = ((xoff.p% + i * (2 * view.p! / CSNG(elec% + 1))) / 640!) * 80
    ROW% = ((yoff.p% + view.p!) / 480!) * 30 + 2
    LOCATE ROW%, COL%
    PRINT USING "##"; i
  NEXT i

  LOCATE 28, 1
  INPUT " Hit <return> to continue. ", resp$

'***************** PRINT DATA TO FILES ********************

  SCREEN 0
  CLS

' Output options DO loop corrected to ensure data is saved -- dlg, 5/28/97

  ftest% = 0
  cont$ = "N"

  DO

    PRINT : PRINT : PRINT "Output options:"
    PRINT "  Input 'F' to save data to files."
    PRINT "  Input 'D' to generate difference files."
    PRINT "  Input 'E' to generate an ASCII file for Excel."
    PRINT "  Input 'P' to generate a graphical output file."
    PRINT "  Input 'C' to collect a new data set."
    PRINT "  Input 'Q' to quit."
    INPUT resp$
    IF resp$ = "F" OR resp$ = "f" THEN
      CALL datmake(elec%, carr!(), quad!(), volt!(), file$)
      CALL symmake(elec%, symstd!, sym!(), asymstd!, asym!(), file$)
      CALL linmake(elec%, std.L!, volt.L!(), file$)
      CALL promake(elec%, proj%, Vsum!(), czero!, qzero!, CEVmean!, CEVmax!, CEVmin!,
       CEVstd!, Vmean!, Vmax!, Vmin!, Vstd!, file$)
      ftest% = 1
    ELSEIF resp$ = "D" OR resp$ = "d" THEN
      CALL difmake(elec%, carr!(), quad!(), volt!())
    ELSEIF resp$ = "E" OR resp$ = "e" THEN
```

```
      CALL xlsmake(elec%, volt!())
      ftest% = 1
    ELSEIF resp$ = "P" OR resp$ = "p" THEN
      CALL psmake(elec%, carr!(), quad!(), volt!())
    ELSEIF resp$ = "C" OR resp$ = "c" OR resp$ = "Q" OR resp$ = "q" THEN
      IF ftest% = 0 THEN
        PRINT "The data set has not been saved to a file yet."
        INPUT "Are you sure you want to do this (Y/N - default N) ?", cont$
      ELSE
        cont$ = "Y"
      END IF
    END IF

  LOOP UNTIL cont$ = "Y" OR cont$ = "y"

  ERASE carr!, quad!, volt!
  ERASE Vsum.elc!, Vsum2.elc!, sym!, save!, ssum!, ssum2!
  ERASE asym!, asum!, asum2!, volt.L!, sum2.L!

END SUB



SUB linmake (elec%, std.L!, volt.L!(), file$)

PRINT
INPUT "Do you want to save linearity data? (default Y):", save$

IF save$ <> "N" AND save$ <> "n" THEN

  OPEN "d:\data\" + file$ + ".lin" FOR OUTPUT AS #1
  PRINT #1, USING "Standard Deviation                #####.## "; std.L!
  PRINT #1, ""
  FOR iv1 = 1 TO (elec% / 2 - 1)
    FOR iv2 = iv1 + 1 TO elec% - iv1
      PRINT #1, USING "## , ## , "; iv1; iv2;
      FOR iv3 = 1 TO elec%
        PRINT #1, USING "####.## , "; volt.L!(iv1, iv2, iv3);
      NEXT iv3
      PRINT #1, ""
    NEXT iv2
  NEXT iv1
  CLOSE #1

  PRINT "Linearity data saved to d:\data\"; file$; ".lin"

END IF

END SUB



SUB promake (elec%, proj%, Vsum!(), czero!, qzero!, CEVmean!, CEVmax!, CEVmin!,
       CEVstd!, Vmean!, Vmax!, Vmin!, Vstd!, file$)

PRINT
INPUT "Do you want to save projection statistics data? (default Y):", save$

IF save$ <> "N" AND save$ <> "n" THEN

  OPEN "d:\data\" + file$ + ".pro" FOR OUTPUT AS #1
  PRINT #1, " Average Data Set Characteristics"
  PRINT #1, ""
  PRINT #1, USING " Carrier Zero Offset              #####"; czero!
  PRINT #1, USING " Quad Zero Offset                 #####"; qzero!
  PRINT #1, ""
  PRINT #1, " Statistics for cross-electrode voltages comparing data"
  PRINT #1, " recorded for each projection:"
  PRINT #1, ""
  PRINT #1, USING " MEAN Cross Electrode Voltage     #####.##"; CEVmean!
  PRINT #1, USING " MAX Cross Electrode Voltage      #####.##"; CEVmax!
  PRINT #1, USING " MIN Cross Electrode Voltage      #####.##"; CEVmin!
  PRINT #1, USING " STDEV Cross Electrode Voltage     ####.##"; CEVstd!
  PRINT #1, ""
  PRINT #1, " Statistics for electrode voltages (not including current injection"
  PRINT #1, " and ground electrodes) for 180 degree injection/ground cases"
  PRINT #1, " comparing data recorded for each projection:"
  PRINT #1, ""
  PRINT #1, USING " MEAN Electrode Voltage           #####.##"; Vmean!
  PRINT #1, USING " MAX Electrode Voltage            #####.##"; Vmax!
  PRINT #1, USING " MIN Electrode Voltage            #####.##"; Vmin!
  PRINT #1, USING " STDEV Electrode Voltage           ####.##"; Vstd!
  PRINT #1, ""
  PRINT #1, "Average data for each projection recorded. Voltages are"
  PRINT #1, "averages for each electrode for cases where current and"
  PRINT #1, USING "ground are 180 degrees opposed (# total). Column 1 is the"; elec% /
       2
  PRINT #1, USING "projection number and columns 2 through ## are data from"; elec% +
       1
  PRINT #1, USING "electrodes 1 through ## (1 current and # ground)."; elec%; elec% /
       2 + 1
  PRINT #1, ""
  FOR i = 1 TO proj%
    PRINT #1, USING "### , "; i;
    FOR k = 1 TO elec% - 1
      PRINT #1, USING "#####.##, "; Vsum!(i, k);
    NEXT k
    PRINT #1, USING "#####.##"; Vsum!(i, elec%)
  NEXT i
  CLOSE #1

  PRINT "Projection statistics saved to d:\data\"; file$; ".pro"

END IF

END SUB



SUB psmake (elec%, carr!(), quad!(), volt!())
DIM x!(15), y!(15), cur%(3850), gnd%(3850)
DIM mag%(3850)

DO

  PRINT "Enter 'M' for magnitude, 'C' for carrier, or 'Q' for quadrature signal plot.
       "
  INPUT "  Hit <return> for default (magnitude plot). ", pl$
  IF pl$ = "" THEN pl$ = "M"

  INPUT "Enter Postscript data file name ('.ps' appended automatically): ", file$

  file$ = "d:\data\" + file$ + ".ps"
  OPEN file$ FOR OUTPUT AS #2

  icount = 1
```

```
FOR iv1 = 1 TO elec% - 1
  FOR iv2 = iv1 + 1 TO elec%
    FOR iv3 = 1 TO elec%

      cur%(icount) = iv1: gnd%(icount) = iv2

      IF pl$ = "M" OR pl$ = "m" THEN
        mag%(icount) = volt!(iv1, iv2, iv3)
      ELSEIF pl$ = "C" OR pl$ = "c" THEN
        mag%(icount) = carr!(iv1, iv2, iv3)
      ELSEIF pl$ = "Q" OR pl$ = "q" THEN
        mag%(icount) = quad!(iv1, iv2, iv3)
      END IF

      icount = icount + 1

    NEXT iv3
  NEXT iv2
NEXT iv1

max% = 0
min% = 5000

FOR i = 1 TO icount - 1

  IF mag%(i) > max% THEN max% = mag%(i)
  IF mag%(i) < min% THEN min% = mag%(i)

NEXT i

x!(1) = 1.5
x!(2) = 2.2
x!(3) = 2.2
x!(4) = -3.3
x!(5) = 2.2
x!(6) = 2.2
x!(7) = -5.5
x!(8) = 2.2
x!(9) = 2.2
x!(10) = -3.3
x!(11) = 2.2
x!(12) = 2.2
x!(13) = -5.5
x!(14) = 2.2
x!(15) = 2.2

y!(1) = 9.5
y!(2) = 0!
y!(3) = 0!
y!(4) = -2!
y!(5) = 0!
y!(6) = 0!
y!(7) = -2!
y!(8) = 0!
y!(9) = 0!
y!(10) = -2!
y!(11) = 0!
y!(12) = 0!
y!(13) = -2!
y!(14) = 0!
y!(15) = 0!
```

```
  PRINT #2, "%!PS-Adobe-3.0 EPSF-3.0"
  PRINT #2, "%%Page: 8"
  PRINT #2, "/inch { 72 mul } bind def"
  PRINT #2, "/inner 0.80 inch def"
  PRINT #2, "/outer 1.0 inch def"

  l = 0

' Total number of voltage circles is 1+2+...+(elec%-1); indices
'   changed to keep 15 voltage circles per page.  Actual LSB values
'   are printed now without being normalized to a maximum of 1000 ---
'   dlg, 5/28/97

  icrcls = 0
  icrcct = 0
  FOR n = 1 TO (elec% - 1)
    icrcls = icrcls + n
  NEXT n
  npages = FIX((icrcls + 14) / 15)

  FOR i = 1 TO npages

    FOR k = 1 TO 15

      IF icrcct < icrcls THEN

        PRINT #2, USING "##.# inch ##.# inch translate"; x!(k); y!(k)
        PRINT #2, "/Times-Roman findfont 0.15 inch scalefont setfont"

        FOR j = 1 TO elec%

          l = l + 1
          linew! = .015 + mag%(l) * .3 / (max% - min%)

' Normalization to 1000 can be restored on the next line if needed

          PRINT #2, USING "0.5 inch -0.05 inch moveto (####) show"; mag%(l)
          PRINT #2, USING "#.### inch setlinewidth"; linew!
          PRINT #2, "0 cos inner mul 0 sin inner mul moveto"
          PRINT #2, "0 cos outer mul 0 sin outer mul lineto"
          PRINT #2, "stroke"
          PRINT #2, USING "###.# rotate"; -360! / CSNG(elec%)

        NEXT j

        cangle! = (cur%(l) - 1) * -360! / CSNG(elec%)
        cnangle! = -1! * cangle!

        gangle! = (gnd%(l) - 1) * -360! / CSNG(elec%)
        gnangle! = -1! * gangle!

        PRINT #2, "/Times-Roman findfont 0.2 inch scalefont setfont"

        PRINT #2, USING "####.# rotate 1.0 inch -0.05 inch moveto (J) show"; cangle!
        PRINT #2, USING "####.# rotate"; cnangle!

        PRINT #2, USING "####.# rotate 1.0 inch -0.05 inch moveto (G) show"; gangle!
        PRINT #2, USING " ####.# rotate"; gnangle!

        PRINT #2, "/Times-Roman findfont 0.15 inch scalefont setfont"

        icrcct = icrcct + 1
      ELSE
      END IF

    NEXT k

    PRINT #2, "showpage"
    PRINT #2, USING "%%Page: ## ##"; i; i

  NEXT i

  CLOSE #2

  PRINT "Postscript file saved to "; file$
  INPUT "Do you wish to make another plot? (Y/N) ", rs$

LOOP WHILE rs$ = "y" OR rs$ = "Y"

END SUB



SUB symmake (elec%, symstd!, sym!(), asymstd!, asym!(), file$)

PRINT
INPUT "Do you want to save symmetry & antisymmetry data? (default Y):", save$

IF save$ <> "N" AND save$ <> "n" THEN

  OPEN "d:\data\" + file$ + ".sym" FOR OUTPUT AS #1
  PRINT #1, "This file contains data used to determine if the test case"
  PRINT #1, USING "has the appropriate symmetry and antisymmetry. The # cases tested";
       elec% / 2
  PRINT #1, "are those where the current and ground are 180 degrees opposed."
  PRINT #1, "The first column is the number of electrode 'steps' from the injection"
  PRINT #1, "point to the measurement point.  For each data point, the mean voltage"
  PRINT #1, "has been subtracted to allow for differences in the ground"
  PRINT #1, "electrode's contact resistance."
  PRINT #1, ""
  PRINT #1, "Symmetry Data: The data is given for each case, first clockwise,"
  PRINT #1, "then counter-clockwise from the injection point for a total of"
  PRINT #1, USING "## curves. The STDEV between the curves is: ###.##"; elec%; symstd!
  PRINT #1, ""
  PRINT #1, "Injection pt:";
  FOR i = 1 TO elec% / 2
    PRINT #1, USING "#                   "; i;
  NEXT i
  PRINT #1, ""
  PRINT #1, "Steps";
  FOR i = 1 TO elec% / 2
    PRINT #1, "   CW        CCW    ";
  NEXT i
  PRINT #1, ""
  FOR k = 1 TO elec% / 2 - 1
    PRINT #1, USING "   # , "; k;
    FOR i = 1 TO elec% / 2
      FOR j = 1 TO 2
        PRINT #1, USING "#####.##, "; sym!(i, j, k);
      NEXT j
    NEXT i
    PRINT #1, ""
  NEXT k
  PRINT #1, ""
  PRINT #1, "Antisymmetry Data: The data is given for each case, first counter-"
  PRINT #1, "clockwise, then clockwise from both antisymmetry electrode nodes for"
  PRINT #1, USING "a total of ## curves.  Antisymmetry nodes are 90 degrees from the";
      elec% * 2
  PRINT #1, "injection and ground points.  The first values are measured at the
      asymmetry"
  PRINT #1, "points, the second from one step away, etc.  The STDEV between the"
  PRINT #1, USING "curves is: ###.##"; asymstd!
  PRINT #1, ""
  PRINT #1, "Injection pt:";
  FOR i = 1 TO elec% / 4
    PRINT #1, USING " #                                    "; i;
  NEXT i
  PRINT #1, ""
  PRINT #1, "From inj. ";
  FOR i = 1 TO elec% / 4
    PRINT #1, "  CW        CW       CCW      CCW     ";
  NEXT i
  PRINT #1, ""
  PRINT #1, "From asym.";
  FOR i = 1 TO elec% / 4
    PRINT #1, " CCW        CW       CCW       CW     ";
  NEXT i
  PRINT #1, ""
  FOR k = 1 TO elec% / 4
    PRINT #1, USING "# , "; k;
    FOR i = 1 TO elec% / 2
      FOR j = 1 TO 4
        PRINT #1, USING "#####.##, "; asym!(i, j, k);
      NEXT j
    NEXT i
    PRINT #1, ""
  NEXT k
  CLOSE #1

  PRINT "Symmetry and antisymmetry data saved to d:\data\"; file$; ".sym"

END IF

END SUB



SUB xlsmake (elec%, volt!())
  INPUT "Input Excel file name ('.xls' is appended automatically): ", file$
  OPEN "d:\data\" + file$ + ".xls" FOR OUTPUT AS #1

  FOR iv1 = 1 TO elec% - 1
    FOR iv2 = iv1 + 1 TO elec%

      PRINT #1, iv1; "to"; iv2; ",";

      FOR iv3 = 1 TO elec% - 1
        PRINT #1, USING "#####.## ,"; volt!(iv1, iv2, iv3)
      NEXT iv3

      PRINT #1, USING "#####.##"; volt!(iv1, iv2, elec%)

    NEXT iv2
  NEXT iv1

  PRINT "Data saved in excel format as d:\data\"; file$; ".xls"
  CLOSE #1
END SUB
```

Appendix E
EIT Reconstruction Code FEMEIT.F

The reconstruction code FEMEIT, written in Fortran 77 by J. R. Torczynski, generates and solves the finite-element representation of the voltage equations (Eq. 3.1) for two-dimensional arbitrary domains, including multiply connected domains and geometries with electrodes on the domain boundary or within the domain itself. Electrodes themselves are represented by mesh nodes, essentially mathematical points. FEMEIT uses global conductivity functions selected from a subroutine library and applies a Newton-Raphson algorithm to find the conductivity parameters that most closely reproduce the measured voltages. Conductivity functions in the library include a constant conductivity, a single insulating bubble, a series of radial insulating annuli centered at an arbitrary position, and conductivity distributions described by Cartesian and radial polynomials.

FEMEIT reads from and writes to the following files:

femeit_inp.dat	general input parameters (input)
femeit_exp.dat	file of experimental voltages (input)
femeit_nod.dat	finite element mesh information (input)
femeit_elc.dat	table associating electrodes with mesh nodes (input)
femeit_log.dat	convergence and conductivity parameters after each iteration (diagnostic output)
femeit_con.dat	conductivity values at mesh nodes (output)
femeit_out.dat	general output parameters (output)

Examples of some file formats follow.

femeit_inp.dat:

1.	Length scale
0.001	Initial value of liquid conductivity, σ_L
50	Maximum number of iterations
0.8	}
0.1	} Dampening coefficients in Newton-Raphson algorithm
1.	}
0.00001	Convergence criterion for conductivity parameters
0.00001	Convergence criterion for residuals of voltage equation
4	}
4	} Parameters to select conductivity function
2	}

`0.3` } C_1

`0.3` } C_2

`0.` } C_3 Initial values of conductivity parameters

`0.` } C_4

`0.99` } K_1

`0.03` } K_2

`femeit_out.dat` echoes the input values in `femeit_inp.dat`, then reports the converged conductivity parameters.

`femeit_exp.dat:`

The input file `femeit_exp.dat` is a primary output file from the EIT data acquisition codes, listed in Appendices B and C. Columns 1 through 3 identify the injection, ground and measurement electrodes, respectively; columns 4 through 6 list the magnitude, carrier component and quadrature component of the measured voltages.

```
1  2  1  858.63  847.32  138.91
1  2  2    0.18    0.18    0.02
1  2  3  359.77  343.14  108.11
.  .  .     .       .       .
.  .  .     .       .       .
.  .  .     .       .       .
```

`femeit_nod.dat:`

The first line contains the number of nodes in the finite-element representation of the domain. Subsequent lines are composed of three columns containing, in order, the number of each node, its *x* coordinate and its *y* coordinate. The *xy* coordinate system is defined so that a circular domain is centered on the origin and has a radius of 1 unit. After the node list, the number of elements in the mesh is given, then each element is listed along with the three nodes defining its boundaries (all elements are triangular).

```
441
  1   0.996917    7.84591E-02
  2   0.987688    0.156434
  3   0.972370    0.233445
  .     .            .
  .     .            .
  .     .            .
800
   1   80    1  152
   2    1    2   81
   3    2    3   82
   .    .    .    .
   .    .    .    .
   .    .    .    .
```

femeit_elc.dat:

The first line contains the number of electrodes; the remaining lines list each electrode and its node number in the finite element mesh.

```
16
   1   5
   2  10
   3  15
   .   .
   .   .
   .   .
  16  80
```

```
c
c23456789012345678901234567890123456789012345678901234567890123456789012
c
      program femeit

c
      implicit double precision (a-h,o-z)
c
      parameter (nnodem=441, nelemm=800, ngparm=15, nvert=3)
      parameter (nconnm=9)
      parameter (nprobm=16)
      parameter (npgm=2*ngparm)
      dimension xnode(nnodem), ynode(nnodem)
      dimension xelem(nelemm), yelem(nelemm)
      dimension condnd(nnodem), dconnd(nnodem)
      dimension ndelem(nelemm,nvert)
      dimension xvert(nvert), yvert(nvert)
      dimension ivert1(nvert), ivert2(nvert)
      dimension a1(nvert), ax(nvert), ay(nvert)
      dimension gigjda(nelemm,nvert,nvert)
      dimension cdnode(nnodem,0:ngparm), cdelem(nelemm,0:ngparm)
      dimension cdfcn(0:ngparm)
      dimension cond(ngparm), dcon(ngparm)
      dimension pg(npgm)
      dimension ndprob(nprobm), vnrm(nprobm,nprobm,nprobm)
      dimension femmat(nnodem,nnodem), feminv(nnodem,nnodem)
      dimension fempar(nnodem,nconnm,0:ngparm)
      dimension nconn(nnodem), nd21(nnodem,nconnm)
      dimension soln(nprobm), sjac(nprobm,ngparm)
      dimension psoln(nprobm,nprobm), psjac(nprobm,nprobm,ngparm)
      dimension amat(ngparm,ngparm), bvec(ngparm)
      data ivert1 / 2, 3, 1/
      data ivert2 / 3, 1, 2/
c
      wtints = 1.D+00 / dfloat(nvert+1)
      wtintc = 1.D+00 - 1.D+00 / dfloat(nvert+1)
c
 1001 format (i4,3(1x,e7.2),1x,e11.6,5(1x,e7.2))
 1002 format (1x,e11.5)
 1003 format (1x,i4)
 1004 format (2(1x,e11.5),i4)
 1005 format (i4,6(1x,e11.5))
 1006 format (3x,e11.5)
c
```

```
c *** Read in the mesh, first dimension, then nodes, then elements.
c
      write (6,*) 'Reading file femeit_nod.dat...'
      open (unit=21, status='unknown', file='femeit_nod.dat')
      read (21,*,end=998,err=998) nnode
      nnode1 = nnode - 1
      if (nnode.gt.nnodem) then
         write (6,*) 'Max no. nodes exceeded: ', nnode, nnodem
         close (unit=21)
         go to 998
         end if
      do 010 in = 1, nnode, 1
         read (21,*,end=998,err=998) nmnode, xnode(in), ynode(in)
         if (nmnode.ne.in) then
            write (6,*) 'Nodes not numbered in correct order.'
            close (unit=21)
            go to 998
            end if
 010     continue
      read (21,*,end=998,err=998) nelem
      if (nelem.gt.nelemm) then
         write (6,*) 'Max no. elems exceeded: ', nelem, nelemm
         close (unit=21)
         go to 998
         end if
      do 020 ie = 1, nelem, 1
         read (21,*,end=998,err=998) nmelem, (ndelem(ie,kv),kv=1,nvert)
         if (nmelem.ne.ie) then
            write (6,*) 'Elements not numbered in correct order.'
            close (unit=21)
            go to 998
            end if
 020     continue
      close(unit=21)
c
c *** Read in conductivity fitting information.
c
      write (6,*) 'Reading file femeit_inp.dat...'
      open (unit=22, status='unknown', file='femeit_inp.dat')
      read (22,*,end=998,err=998) slen
      read (22,*,end=998,err=998) sigma0
      read (22,*,end=998,err=998) niter
      read (22,*,end=998,err=998) damp0
      read (22,*,end=998,err=998) damp1
      read (22,*,end=998,err=998) damp2
      read (22,*,end=998,err=998) tolc
      read (22,*,end=998,err=998) tolr
      read (22,*,end=998,err=998) kctype
      read (22,*,end=998,err=998) ngpar
      read (22,*,end=998,err=998) npg
      call cdchk(kctype,ngpar,npg,ngparx,npgx,ichk)
      if ((ngpar.ne.ngparx).or.(npg.ne.npgx).or.(ichk.ne.1)) then
         write (6,*) 'No. parameters not as expected: '
         write (6,*) 'If ichk = 0, model not available: ', kctype
         write (6,*) 'ngpar, ngparx ', ngpar, ngparx
         write (6,*) 'npg,   npgx   ', npg,   npgx
         close (unit=22)
         go to 998
         end if
      if (ngpar.gt.ngparm) then
         write (6,*) 'Max no. gpars exceeded: ', ngpar, ngparm
         close (unit=22)
         go to 998
         end if
      do 030 ig = 1, ngpar, 1
         read (22,*,end=998,err=998) cond(ig)
  030    continue
      if (npg.gt.npgm) then
         write (6,*) 'Max no. gfcn pars exceeded: ', npg, npgm
         close (unit=22)
         go to 998
         end if
      if ((npg.gt.0).and.(npg.le.npgm)) then
         read (22,*,end=998,err=998) (pg(ipg),ipg=1,npg,1)
         end if
      close (unit=22)
c
c *** Read in electrode numbers and nodes.
c
      write (6,*) 'Reading file femeit_elc.dat...'
      open (unit=23, status='unknown', file='femeit_elc.dat')
      read (23,*,end=998,err=998) nprob
      if (nprob.gt.nprobm) then
         write (6,*) 'Max no. electrodes exceeded: ', nprob, nprobm
         close (unit=23)
         go to 998
         end if
      nexpt = nprob * (nprob - 1) / 2
      do 040 ip = 1, nprob, 1
         read (23,*,end=998,err=998) iprob, ndprob(ip)
         if (iprob.ne.ip) then
            write (6,*) 'Electrodes out of order:', ip, iprob
            close (unit=23)
            go to 998
            end if
  040    continue
      close (unit=23)
c
c *** Read in experimental data and normalize voltage by current.
c
      write (6,*) 'Reading file femeit_exp.dat...'
      open (unit=24, status='unknown', file='femeit_exp.dat')
      svcex2 = 0.D+00
      do 070 ip1 = 1, nprob-1, 1
      do 070 ip2 = ip1+1, nprob, 1
         wtotal = 0.D+00
         svcexa = 0.D+00
         svcexb = 0.D+00
         do 060 ip = 1, nprob, 1
            curr12 = 1.D+00
            read (24,*) ipa, ipb, ipc, vmagn, vcarr, vquad
            vcex = vmagn * sigma0 * slen / curr12
            wtex = wt(ip,ip1,ip2)
            wtotal = wtotal + wtex
            svcexa = svcexa + wtex * vcex
            svcexb = svcexb + wtex * vcex ** 2
            vnrm(ip,ip1,ip2) = vcex
  060       continue
         svcexa = svcexa / wtotal
         svcexb = svcexb / wtotal
         svcex2 = svcex2 + svcexb - svcexa ** 2
  070    continue
      svcex1 = sqrt(svcex2 / dfloat(nexpt))
      close (unit=24)
c
c *** Compute nontrivial node pairs.
```

```
c
      write (6,*) 'Computing nontrivial node pairs...'
      do 080 in1 = 1, nnode, 1
      do 080 in2 = 1, nnode, 1
         femmat(in1,in2) = 0.D+00
  080    continue
c
      do 110 ie = 1, nelem, 1
         do 100 iv1 = 1, nvert, 1
            in1 = ndelem(ie,iv1)
         do 090 iv2 = 1, nvert, 1
            in2 = ndelem(ie,iv2)
            femmat(in1,in2) = 1.D+00
  090       continue
  100       continue
  110    continue
c
      do 130 in1 = 1, nnode, 1
         nconn(in1) = 0
      do 120 in2 = 1, nnode, 1
         if (femmat(in1,in2).gt.0.5D+00) then
            nconn(in1) = nconn(in1) + 1
            nd21(in1,nconn(in1)) = in2
            if (nconn(in1).gt.nconnm) then
               write (6,*) '*** INSUFFICIENT LINKS ***'
               go to 998
               end if
            end if
  120    continue
  130    continue
c
c *** Compute element quantities: Int. grad phi_i . grad phi_j dV.
c
      write (6,*) 'Computing element quantities...'
      inotcc = 0
      do 170 ie = 1, nelem, 1
         xsum = 0.D+00
         ysum = 0.D+00
         do 140 iv = 1, nvert, 1
            xvert(iv) = xnode(ndelem(ie,iv))
            yvert(iv) = ynode(ndelem(ie,iv))
            xsum = xsum + xvert(iv)
            ysum = ysum + yvert(iv)
  140       continue
         xelem(ie) = xsum / dfloat(nvert)
         yelem(ie) = ysum / dfloat(nvert)
         size = 0.D+00
         do 150 iv = 1, nvert, 1
            a1(iv) = xvert(ivert1(iv)) * yvert(ivert2(iv))
     1             - xvert(ivert2(iv)) * yvert(ivert1(iv))
            ax(iv) = yvert(ivert1(iv)) - yvert(ivert2(iv))
            ay(iv) = xvert(ivert2(iv)) - xvert(ivert1(iv))
            size = size + 0.5D+00 * a1(iv)
  150       continue
         if (size.le.0) then
            write (6,*) 'Nodes not counterclockwise for elt: '
            write (6,*) ie, size
            inotcc = 1
            if (size.eq.0) go to 998
            end if
         do 160 iv1 = 1, nvert, 1
         do 160 iv2 = 1, nvert, 1
            gigj = ax(iv1) * ax(iv2) + ay(iv1) * ay(iv2)
            gigjda(ie,iv1,iv2) = gigj * 0.25D+00 / size
  160       continue
  170    continue
      if (inotcc.gt.0) go to 998
c
c *** Begin major iteration loop.
c
      write (6,*) 'Beginning major iteration loop...', niter
      write (6,*) 'Writing file femeit_log.dat...'
      write (6,*) ' '
      write (6,*)  ' it  tolc    snorm   sdcon   scond       '//
     1             'tolr    snvec   sbvec   rmsnrm  corr'
      open (unit=25, status='unknown', file='femeit_log.dat')
      write (25,*) ' it  tolc    snorm   sdcon   scond       '//
     1             'tolr    snvec   sbvec   rmsnrm  corr'
      do 500 it = 1, niter, 1
c
c *** Compute average conductivity and derivatives for each element.
c
      do 190 in = 1, nnode, 1
         xn = xnode(in)
         yn = ynode(in)
         call cdsub(xn,yn,slen,kctype,npg,pg,ngpar,cond,cdfcn)
         do 180 ig = 0, ngpar, 1
            cdnode(in,ig) = cdfcn(ig)
  180       continue
  190       continue
c
      do 198 ie = 1, nelem, 1
         xe = xelem(ie)
         ye = yelem(ie)
         call cdsub(xe,ye,slen,kctype,npg,pg,ngpar,cond,cdfcn)
         do 196 ig = 0, ngpar, 1
            cdelem(ie,ig) = cdfcn(ig)
  196       continue
  198    continue
c
      do 220 ie = 1, nelem, 1
      do 210 ig = 0, ngpar, 1
         cdfcn(ig) = 0.D+00
         do 200 iv = 1, nvert, 1
            cdfcn(ig) = cdfcn(ig) + cdnode(ndelem(ie,iv),ig)
  200       continue
         cdfcn(ig) = cdfcn(ig) / dfloat(nvert)
         cdelem(ie,ig) = wtintc * cdelem(ie,ig) + wtints * cdfcn(ig)
  210    continue
  220    continue
c
c *** Assemble the derivative FEM matrices.
c
      do 230 in1 = 1, nnode, 1
      do 230 ic = 1, nconnm, 1
      do 230 ig = 0, ngpar, 1
         fempar(in1,ic,ig) = 0.D+00
  230    continue
c
      do 250 ie = 1, nelem, 1
      do 250 iv1 = 1, nvert, 1
         in1 = ndelem(ie,iv1)
         nconn1 = nconn(in1)
      do 250 iv2 = 1, nvert, 1
         in2 = ndelem(ie,iv2)
         ic = 0
         do 240 ic0 = 1, nconn1, 1
            if (nd21(in1,ic0).eq.in2) ic = ic0
  240       continue
      do 250 ig = 0, ngpar, 1
         fempar(in1,ic,ig) = fempar(in1,ic,ig)
     1      + cdelem(ie,ig) * gigjda(ie,iv1,iv2)
  250    continue
c
c *** Assemble FEM matrix.
c
      do 260 in1 = 1, nnode, 1
      do 260 in2 = 1, nnode, 1
         femmat(in1,in2) = 0.D+00
         feminv(in1,in2) = 0.D+00
  260    continue
c
      do 280 in1 = 1, nnode, 1
         nconn1 = nconn(in1)
      do 280 ic = 1, nconn1, 1
         in2 = nd21(in1,ic)
         femmat(in1,in2) = fempar(in1,ic,0)
         feminv(in1,in2) = femmat(in1,in2)
  280    continue
c
c *** Invert FEM matrix.
c
      call zmtinv(femmat,feminv,nnode1,rcond,info)
c
c *** Loop over electrode-electrode combinations to calculate
c *** partial solution vector and portion of electrode Jacobian.
c     Could speed up using psoln(ipk,ipm) = psoln(ipm,ipk)
c     and psjac(ipk,ipm,ig) = psjac(ipm,ipk,ig).
c
      do 380 ipm = 1, nprob, 1
         inm = ndprob(ipm)
      do 380 ipk = 1, nprob, 1
         ink = ndprob(ipk)
c
         psoln(ipm,ipk) = feminv(ink,inm)
c
      do 360 ig = 1, ngpar, 1
         psj = 0.D+00
      do 340 in1 = 1, nnode1, 1
         nconn1 = nconn(in1)
      do 340 ic = 1, nconn1, 1
         in2 = nd21(in1,ic)
         psj = psj+feminv(ink,in1)*fempar(in1,ic,ig)*feminv(in2,inm)
  340    continue
         psjac(ipm,ipk,ig) = - psj
  360    continue
c
  380    continue
c
c ** Calculate the least-squares matrix and vector.
c
      do 390 ig1 = 1, ngpar, 1
         bvec(ig1) = 0.D+00
      do 390 ig2 = 1, ngpar, 1
         amat(ig1,ig2) = 0.D+00
  390    continue
c
c *** Calculate the least-squares matrix and vector.
c     Could speed up using amat(ig2,ig1) = amat(ig1,ig2).
```

```
c
      svrms2 = 0.D+00
      do 460 ipm = 1, nprob-1, 1
      do 460 ipn = ipm+1, nprob, 1
      do 410 ipk = 1, nprob, 1
         soln(ipk) = psoln(ipm,ipk) - psoln(ipn,ipk)
      do 400 ig1 = 1, ngpar, 1
         sjac(ipk,ig1) = psjac(ipm,ipk,ig1) - psjac(ipn,ipk,ig1)
  400    continue
  410    continue
         wtotal = 0.D+00
         svrmsa = 0.D+00
         svrmsb = 0.D+00
      do 450 ipk = 1, nprob, 1
         bterm = vnrm(ipk,ipm,ipn) - soln(ipk)
         wtipk = wt(ipk,ipm,ipn)
         wtotal = wtotal + wtipk
         svrmsa = svrmsa + wtipk * bterm
         svrmsb = svrmsb + wtipk * bterm * bterm
      do 440 ipl = 1, nprob, 1
         wtipl = wt(ipl,ipm,ipn)
         wtterm = wtipk * wtipl
      do 430 ig1 = 1, ngpar, 1
         sjterm = wtterm * ( sjac(ipk,ig1) - sjac(ipl,ig1) )
         bvec(ig1) = bvec(ig1) + sjterm * bterm
      do 420 ig2 = 1, ngpar, 1
         amat(ig1,ig2) = amat(ig1,ig2) + sjterm * sjac(ipk,ig2)
  420    continue
  430    continue
  440    continue
  450    continue
         svrmsa = svrmsa / wtotal
         svrmsb = svrmsb / wtotal
         svrms2 = svrms2 + svrmsb - svrmsa ** 2
  460    continue
      svrms1 = sqrt(svrms2 / dfloat(nexpt))
      rmsnrm = svrms1 / svcex1
c
c *** Solve matrix equation for conductivity parameter increments.
c
      call zlnsol(amat,ngpar,bvec,dcon,indwrn)
c
c *** Determine nodal conductivity, change thereof, damping parameter.
c     Update conductivity. Compute changes. Test tolerances.
c
      damp = damp0
      do 480 in = 1, nnode, 1
         condnd(in) = cdnode(in,0)
         dconnd(in) = 0.D+00
         do 470 ig = 1, ngpar, 1
            dconnd(in) = dconnd(in) + cdnode(in,ig) * dcon(ig)
  470       continue
         damp = max(damp,abs(dconnd(in)/condnd(in)))
  480    continue
      corr = max(damp1,min(damp2,damp0/damp))
c
      scond = 0.D+00
      sdcon = 0.D+00
      sbvec = 0.D+00
      do 490 ig = 1, ngpar, 1
         cond(ig) = cond(ig) + corr * dcon(ig)
         scond = scond + cond(ig)*cond(ig)
         sdcon = sdcon + dcon(ig)*dcon(ig)
         sbvec = sbvec + bvec(ig)*bvec(ig)
  490    continue
      scond = sqrt( scond / dfloat(ngpar) )
      sdcon = sqrt( sdcon / dfloat(ngpar) )
      snorm = sdcon / scond
      sbvec = sqrt( sbvec / dfloat(ngpar) )
      if (it.eq.1) svscal = sbvec
      snvec = sbvec / svscal
      write (25,1001) it, tolc, snorm, sdcon, scond,
     1                tolr, snvec, sbvec, rmsnrm, corr
      write (6,1001)  it, tolc, snorm, sdcon, scond,
     1                tolr, snvec, sbvec, rmsnrm, corr
      if ((snorm.lt.tolc).and.(snvec.lt.tolr)) go to 800
c
c *** End major iteration loop.
c
  500 continue
      write (6,*) 'Exceeded iteration limit without convergence.'
c
c *** Write solution and close files.
c
  800 continue
      close (unit=25)
      write (6,*) ' '
c
      write (6,*) 'Writing file femeit_out.dat...'
      open (unit=26, status='unknown', file='femeit_out.dat')
      write (26,1002) slen
      write (26,1002) sigma0
      write (26,1003) niter
      write (26,1002) damp0
      write (26,1002) damp1
      write (26,1002) damp2
      write (26,1002) tolc
      write (26,1002) tolr
      write (26,1003) kctype
      write (26,1003) ngpar
      write (26,1003) npg
      do 810 ig = 1, ngpar, 1
         write (26,1004) cond(ig), dcon(ig), ig
  810    continue
      if (npg.gt.0) then
         do 820 ipg = 1, npg, 1
            write (26,1006) pg(ipg)
  820       continue
         end if
      close (unit=26)
c
      write (6,*) 'Writing file femeit_con.dat...'
      do 840 in = 1, nnode, 1
         xn = xnode(in)
         yn = ynode(in)
         call cdsub(xn,yn,slen,kctype,npg,pg,ngpar,cond,cdfcn)
         condnd(in) = cdfcn(0)
         dconnd(in) = 0.D+00
         do 830 ig = 1, ngpar, 1
            dconnd(in) = dconnd(in) + cdnode(in,ig) * dcon(ig)
  830       continue
         damp = max(damp,abs(dconnd(in)/condnd(in)))
  840    continue
      open (unit=27, status='unknown', file='femeit_con.dat')
      do 850 in = 1, nnode, 1
         write (27,1005) in, condnd(in), dconnd(in)
  850    continue
      close (unit=27)
c
c *** Completed, stop.
c
      go to 999
  998 write (6,*) '*** ABNORMAL STOP ***'
  999 stop 'femeit'
      end
c
c23456789012345678901234567890123456789012345678901234567890123456789012
c
      subroutine cdchk(k,ng,np,ngx,npx,ichk)
c
      implicit double precision (a-h,o-z)
c
c *** Set default to abort.
c
      ngx = -1
      npx = -1
      ichk = 0
c
c *** Cartesian polynomials.
c
      if (k.eq.0) then
         ngx = max(0,ng)
         npx = 2 * ngx
         ichk = 1
         end if
c
c *** Radial polynomials and angular sine and cosine.
c
      if (k.eq.1) then
         ngx = max(0,ng)
         npx = 2 * ngx
         ichk = 1
         end if
c
c *** Piecewise linear interpolation over ng radial annuli.
c
      if (k.eq.2) then
         ngx = max(1,ng)
         npx = ngx
         ichk = 1
         end if
c
c *** Bubble centered at origin.
c
      if (k.eq.3) then
         ngx = 2
         npx = 2
         ichk = 1
         end if
c
c *** Bubble centered arbitrarily.
c
      if (k.eq.4) then
         ngx = 4
         npx = 2
         ichk = 1
         end if
c
c *** Bubble centered arbitrarily with known conductivity.
```

```
c
      if (k.eq.5) then
         ngx = 3
         npx = 3
         ichk = 1
         end if
c
      return
      end
c
c2345678901234567890123456789012345678901234567890123456789012345678901
c
      subroutine cdsub(x,y,sl,k,np,p,ng,cn,cdfcn)
c
c *** Determines conductivity and derivatives at (x,y).
c     Function is type k, with np internal parameters in p,
c     ng fitting parameters in cn, and length scale sl.
c     cdfcn(0) = conductivity
c     cdfcn(ig) = d conductivity / d cn(ig)
c
      implicit double precision (a-h,o-z)
      dimension p(np)
      dimension cn(ng)
      dimension cdfcn(0:ng)
c
      zero = 0.D+00
      unit = 1.D+00
      two2 = 2.D+00
      pi = 2.D+00 * asin(1.D+00)
c
      radius = sqrt(x**2+y**2)
      angle = 0.D+00
      if ((x.ne.zero).or.(y.ne.zero)) angle = atan2(y,x)
      xsl = x / sl
      ysl = y / sl
      rsl = radius / sl
      do 010 ig = 0, ng, 1
         cdfcn(ig) = 0.D+00
  010    continue
c
c *** Cartesian polynomials.
c     p(2*ig-1) = m = x exponent, p(2*ig) = n = y exponent.
c     cn(ig) is coefficient multiplying xsl**m * ysl**n.
c
      if (k.eq.0) then
         cdfcn(0) = 0.D+00
         do 100 ig = 1, ng, 1
            m = nint(p(2*ig-1))
            xfcn = 1.D+00
            if (m.ne.0) xfcn = xsl ** m
            n = nint(p(2*ig))
            yfcn = 1.D+00
            if (n.ne.0) yfcn = ysl ** n
            cdfcn(ig) = xfcn * yfcn
            cdfcn(0) = cdfcn(0) + cn(ig) * cdfcn(ig)
  100       continue
         end if
c
c *** Radial polynomials and angular sine and cosine.
c     p(2*ig-1) = m = r exponent, p(2*ig) = n = theta harmonic.
c     If n < 0, cosine. If n > 0, sine. If n = 0, unity.
c     cn(ig) is coefficient multiplying rsl**m * trig(n*theta).
c
```

```
      if (k.eq.1) then
         cdfcn(0) = 0.D+00
         do 110 ig = 1, ng, 1
            m = nint(p(2*ig-1))
            rfcn = 1.D+00
            if (m.ne.0) rfcn = rsl ** m
            n = nint(p(2*ig))
            zn = dfloat(abs(n))
            tfcn = 1.D+00
            if (n.lt.0) tfcn = cos(zn*angle)
            if (n.gt.0) tfcn = sin(zn*angle)
            cdfcn(ig) = rfcn * tfcn
            cdfcn(0) = cdfcn(0) + cn(ig) * cdfcn(ig)
  110       continue
         end if
c
c *** Piecewise linear interpolation over ng radial annuli.
c     p(ig) are the right-hand radii of the annuli, ng values.
c     cn(ig) are the values at these radii.
c     If they existed, p(0) = 0 (i.e. r=0), and cn(0) = cn(1)
c     (i.e. the value at r = 0 is taken to be cn(1)).
c
      if (k.eq.2) then
         iannu = 0
         do 140 ig = ng, 1, -1
            cdfcn(ig) = 0.D+00
            rrght = p(ig) / sl
            if (rsl.le.rrght) iannu = ig
  140       continue
         if (iannu.eq.0) iannu = ng
         if (iannu.eq.1) then
            frght = 1.D+00
            cdfcn(0) = frght * cn(1)
            cdfcn(1) = frght
            end if
         if (iannu.ne.1) then
            ig = iannu
            rleft = p(ig-1) / sl
            rrght = p(ig) / sl
            frght = ( rsl - rleft ) / ( rrght - rleft )
            frght = max(zero,min(unit,frght))
            fleft = 1.D+00 - frght
            cdfcn(0) = fleft * cn(ig-1) + frght * cn(ig)
            cdfcn(ig-1) = fleft
            cdfcn(ig) = frght
            end if
         end if
c
c *** Bubble model: nearly zero inside, nearly constant outside.
c     Smooth transition between 2 constant regions: 1 circular
c     region embedded eccentrically in another circular region.
c     cn(1) is overall amplitude.
c     cn(2) is normalized radius for changeover.
c     Center: (0,0) for k = 3, (cn(3),cn(4)) for k = 4.
c     p(1) is degree of variation, strictly 0 < p(1) < 1.
c     p(2) is normalized width of changeover, strictly p(2) > 0.
c
      if (k.eq.3) then
         arg1 = ( rsl - cn(2) ) / p(2)
         arg2 = ( rsl + cn(2) ) / p(2)
         cdfcn(1) = 1.D+00 + 0.5D+00 * p(1) * ( tanh(arg1)-tanh(arg2) )
         cdfcn(0) = cn(1) * cdfcn(1)
         sech12 = 1.D+00 / cosh(arg1)**2
         sech22 = 1.D+00 / cosh(arg2)**2
         cdfcn0 = cn(1) * 0.5D+00 * p(1) / p(2)
         cdfcn(2) = - cdfcn0 * ( sech12 + sech22 )
         end if
c
      if (k.eq.4) then
         rlr = sqrt((xsl-cn(3))**2+(ysl-cn(4))**2)
         arg1 = ( rlr - cn(2) ) / p(2)
         arg2 = ( rlr + cn(2) ) / p(2)
         cdfcn(1) = 1.D+00 + 0.5D+00 * p(1) * ( tanh(arg1)-tanh(arg2) )
         cdfcn(0) = cn(1) * cdfcn(1)
         sech12 = 1.D+00 / cosh(arg1)**2
         sech22 = 1.D+00 / cosh(arg2)**2
         cdfcn0 = cn(1) * 0.5D+00 * p(1) / p(2)
         cdfcn(2) = - cdfcn0 * ( sech12 + sech22 )
         cdfcn(3) = 0.D+00
         cdfcn(4) = 0.D+00
         if (rlr.ne.0.D+00) then
            cdfcn5 = cdfcn0 * ( sech22 - sech12 )
            cdfcn(3) = ( xsl - cn(3) ) / rlr * cdfcn5
            cdfcn(4) = ( ysl - cn(4) ) / rlr * cdfcn5
            end if
         end if
c
      if (k.eq.5) then
         rlr = sqrt((xsl-cn(2))**2+(ysl-cn(3))**2)
         arg1 = ( rlr - cn(1) ) / p(2)
         arg2 = ( rlr + cn(1) ) / p(2)
         cdfcn(0) = p(3) * (1.D+00+0.5D+00*p(1)*(tanh(arg1)-tanh(arg2)))
         sech12 = 1.D+00 / cosh(arg1)**2
         sech22 = 1.D+00 / cosh(arg2)**2
         cdfcn0 = p(3) * 0.5D+00 * p(1) / p(2)
         cdfcn(1) = - cdfcn0 * ( sech12 + sech22 )
         cdfcn(2) = 0.D+00
         cdfcn(3) = 0.D+00
         if (rlr.ne.0.D+00) then
            cdfcn5 = cdfcn0 * ( sech22 - sech12 )
            cdfcn(2) = ( xsl - cn(2) ) / rlr * cdfcn5
            cdfcn(3) = ( ysl - cn(3) ) / rlr * cdfcn5
            end if
         end if
c
      return
      end
c
c23456789012345678901234567890123456789012345678901234567890123456789012
c
      function wt(i,i1,i2)
c
      implicit double precision (a-h,o-z)
      wt = 1.D+00
      if ((i.eq.i1).or.(i.eq.i2)) wt = 0.D+00
c
      return
      end
c
c23456789012345678901234567890123456789012345678901234567890123456789012
c
      subroutine zmtinv(amat,ainv,n,rcond,info)
c
c *** Drives SLATEC routines DPOCO, DPODI.
c
      implicit double precision (a-h,o-z)
      parameter (nnodem=441)
      parameter (lda=nnodem)
      dimension amat(lda,lda), ainv(lda,lda), work(lda)
      dimension det(2)
c
      do 100 i = 1, lda, 1
      do 100 j = 1, lda, 1
         ainv(i,j) = amat(i,j)
  100    continue
c
      call dpoco(ainv,lda,n,rcond,work,info)
c
      call dpodi(ainv,lda,n,det,11)
c
      do 200 i = 1, lda, 1
      do 200 j = 1, i, 1
         ainv(i,j) = ainv(j,i)
  200    continue
c
      return
      end
c
c23456789012345678901234567890123456789012345678901234567890123456789012
c
      subroutine zlnsol(amat,n,bvec,xvec,ind)
c
c *** Drives SLATEC routine DGEFS.
c
      implicit double precision (a-h,o-z)
      parameter (ngparm=15)
      parameter (lda=ngparm)
      dimension amat(lda,lda), bvec(lda), xvec(lda)
      dimension awrk(lda,lda), work(lda), iwork(lda)
c
      do 100 i = 1, lda, 1
         xvec(i) = bvec(i)
      do 100 j = 1, lda, 1
         awrk(i,j) = amat(i,j)
  100    continue
c
      call dgefs(awrk,lda,n,xvec,1,ind,work,iwork)
c
      return
      end
c
c23456789012345678901234567890123456789012345678901234567890123456789012
c
```

Appendix F
EIT Reconstruction Code EITAXI.F

This Fortran 77 program by J. R. Torczynski determines the radial conductivity distribution in a three-dimensional cylindrical domain from voltage measurements at electrodes on the domain boundary. EITAXI uses library files that map the electrode voltages as a function of the conductivity distribution within the domain. The libraries are generated by the commercial code FIDAP (Fluid Dynamics International, 1996), which solves the finite-element representation of the voltage equations (Eq. 3.1). Using cubic-spline interpolation and a Newton-Raphson algorithm, EITAXI determines the best-fit parameters for a user-selected conductivity distribution that most closely reproduce the measured voltages. This latest version of EITAXI includes the option to convert voltage data to quartic radial conductivity profiles, but permits quadratic and constant conductivity profiles as simpler cases.

EITAXI reads from and writes to the following files:

eitaxi_inp.dat	general input parameters (input)
eitaxi_exp.dat	file of experimental voltages (input)
eitaxi_coe.dat	file of coefficients fit to fundamental voltage solutions (input)
eitaxi_log.dat	convergence and conductivity parameters after each iteration (diagnostic output)
eitaxi_out.dat	general output parameters (output)
eitaxi_sol.dat	converged fundamental voltage solution (output)

The file eitaxi_exp.dat is identical in format to femeit_exp.dat in Appendix E. Examples of pertinent file formats unique to EITAXI follow.

eitaxi_inp.dat:

1.0E-04	Conversion from least significant bits (EIT output units) to volts; nominal value is 0.0001 V/LSB
1.0	Ratio of electrode height to domain radius (used only in two-dimensional reconstructions, where input data is based on linear current density instead of total current; set to 1.0 for correct normalization in three-dimensional reconstructions)
0.09525	Domain radius, R_{col} (m)
125.0E-06	Injection current (A)
0.0600	Baseline liquid conductivity, σ_L (S/m)
20	Maximum number of iterations
0.5	}
0.	} Iterative damping coefficients in Newton-Raphson algorithm (dimensionless)

1.0	}
0.0000001	Convergence criterion for conductivity parameters C_1 and C_2
0.0000001	Convergence criterion for conductivity parameter C_0
1.0	}C_0
1.0	}C_1 Initial values of conductivity parameters
0.0	}C_2
2	Number of fit parameters: 0 = constant profile, 1 = quadratic, 2 = quartic

eitaxi_out.dat echoes the contents of eitaxi_inp.dat but replaces the initial values of the conductivity parameters with the final, converged values.

eitaxi_coe.dat:

This library file contains the fitting coefficients, one per line, for all reciprocal nontrivial voltages of the fundamental voltage solution. The outermost loop is the electrode number, the middle loop is the power of C_2, and the innermost loop is the power of C_1.

```
c
c2345678901234567890123456789012345678901234567890123456789012345678901
c
      program eitaxi
c
c     Revision 19990419
c
c *** Finds best fit axisymmetric conductivity profile.
c     Uses 2 shape parameters.
c     Uses analytical representation of fundamental solution.
c
c     Currently eitaxi_coe.dat has fit coefficients for a quartic:
c     (1/c0)(1+c1*(2r^2-1)+c2*(1-6r^2+6r^4))
c     Electrodes are 3" x 0.25", 16 at 22.5 degrees, 7.5" ID.
c     Must recompute eitaxi_coe.dat for different electrode geometry.
c
      implicit double precision (a-h,o-z)
c
      parameter (nfun=8)
      parameter (ncoe=24)
      dimension vcoe0(0:ncoe,0:ncoe,0:nfun)
      dimension vcoe1(0:ncoe,0:ncoe,0:nfun)
      dimension vcoe2(0:ncoe,0:ncoe,0:nfun)
      dimension vfun0(0:nfun)
      dimension vfun1(0:nfun)
      dimension vfun2(0:nfun)
c
      parameter (nelc=2*nfun)
      dimension wt(nelc,nelc,nelc)
      dimension vk0(nelc)
      dimension vkm0(nelc,nelc)
      dimension vkmn0(nelc,nelc,nelc)
      dimension vk1(nelc)
      dimension vkm1(nelc,nelc)
      dimension vkmn1(nelc,nelc,nelc)
      dimension vk2(nelc)
      dimension vkm2(nelc,nelc)
      dimension vkmn2(nelc,nelc,nelc)
      dimension vnrm(nelc,nelc,nelc)
c
 1001 format (1x,d18.12)
 1002 format (1x,i4)
 1003 format (i4,3(1x,d11.5,1x,d8.2),2(1x,d8.2))
 1004 format (3(1x,d11.5))
 2000 format (1x,a)
 2001 format (1x,a12,d18.12)
 2002 format (1x,a12,i4)
c
c *** Initialize the weights.
c
      do 0020 i1 = 1, nelc, 1
      do 0020 i2 = 1, nelc, 1
      do 0020 i3 = 1, nelc, 1
         wt(i1,i2,i3) = 1.
         if ((i1.eq.i2).or.(i1.eq.i3).or.(i2.eq.i3)) wt(i1,i2,i3)=0.
 0020    continue
c
c *** Read in input parameters.
c
      write (6,2000) 'Reading input parameters from eitaxi_inp.dat'
      open (unit=23, status='old', file='eitaxi_inp.dat')
      read (23,*) convrt
      read (23,*) hoverr
      read (23,*) radius
      read (23,*) curr12
      read (23,*) sigma0
      read (23,*) niter
      read (23,*) damp0
      read (23,*) damp1
      read (23,*) damp2
      read (23,*) tolc
      read (23,*) tolr
      read (23,*) c0val
      read (23,*) c1val
      read (23,*) c2val
      read (23,*) nshape
      close (unit=23)
c
      vltref = curr12 / ( convrt * hoverr * sigma0 * radius )
      vltcon = 1. / vltref
      write (6,2001) '   convrt = ', convrt
      write (6,2001) '   hoverr = ', hoverr
      write (6,2001) '   radius = ', radius
      write (6,2001) '   curr12 = ', curr12
      write (6,2001) '   sigma0 = ', sigma0
      write (6,2002) '   niter  = ', niter
      write (6,2001) '   damp0  = ', damp0
      write (6,2001) '   damp1  = ', damp1
      write (6,2001) '   damp2  = ', damp2
      write (6,2001) '   tolc   = ', tolc
      write (6,2001) '   tolr   = ', tolr
      write (6,2001) '   c0val  = ', c0val
      write (6,2001) '   c1val  = ', c1val
      write (6,2001) '   c2val  = ', c2val
      write (6,2002) '   nshape = ', nshape
```

```
c
c *** Read in experimental voltages and normalize.
c
      write (6,2000) ’Reading experimental data from eitaxi_exp.dat’
      open (unit=24, status=’old’, file=’eitaxi_exp.dat’)
      svcex2 = 0.
      wnexp  = 0.
      do 0050 ip1 = 1, nelc-1, 1
      do 0050 ip2 = ip1+1, nelc, 1
         wtotal = 0.
         svcexa = 0.
         svcexb = 0.
         do 0040 ip = 1, nelc, 1
            read (24,*) ipa, ipb, ipc, vmagn, vcarr, vquad
            vcex = vmagn * vltcon
            wtex = wt(ip,ip1,ip2)
            wtotal = wtotal + wtex
            svcexa = svcexa + wtex * vcex
            svcexb = svcexb + wtex * vcex * vcex
            vnrm(ip,ip1,ip2) = vcex
 0040       continue
         svcexa = svcexa / wtotal
         svcexb = svcexb / wtotal
         svcex2 = svcex2 + svcexb - svcexa ** 2
         wnexp  = wnexp + 1.
 0050    continue
      svcex1 = sqrt(svcex2 / wnexp)
      close (unit=24)
c
c *** Initialize the fundamental solution and derivative parameters.
c     The coefficients vcoe0(*,*,0) must all be zero.
c
      write (6,2000) ’Reading coefficients from eitaxi_coe.dat’
      open (unit=20, status=’old’, file=’eitaxi_coe.dat’)
c
      do 0110 ifun = 0, nfun, 1
      do 0105 ico2 = 0, ncoe, 1
      do 0100 ico1 = 0, ncoe, 1
         read (20,*) vcoe0(ico1,ico2,ifun)
 0100    continue
 0105    continue
 0110    continue
c
      close (unit=20)
c
      do 0115 ico2 = 0, ncoe, 1
      do 0115 ico1 = 0, ncoe, 1
         if (vcoe0(ico1,ico2,0).ne.0.) go to 0998
 0115    continue
c
      do 0130 ifun = 0, nfun, 1
      do 0120 ico2 = 0, ncoe, 1
      do 0120 ico1 = 0, ncoe-1, 1
         vcoe1(ico1,ico2,ifun) = dfloat(ico1+1)*vcoe0(ico1+1,ico2,ifun)
 0120    continue
         vcoe1(ncoe,ico2,ifun) = 0.
 0130    continue
c
      do 0150 ifun = 0, nfun, 1
      do 0140 ico1 = 0, ncoe, 1
      do 0140 ico2 = 0, ncoe-1, 1
         vcoe2(ico1,ico2,ifun) = dfloat(ico2+1)*vcoe0(ico1,ico2+1,ifun)
 0140    continue
```

```
         vcoe2(ico1,ncoe,ifun) = 0.
 0150    continue
c
c *** Begin the iterative least-squares fit.
c
      write (6,2000) 'Writing iterations to eitaxi_log.dat'
      open (unit=25, status='unknown', file='eitaxi_log.dat')
      write (6,2000) ' '
      write (6,2000) ' it   c0val       c0rel'//
     1                  '   c1val       c1inc'//
     2                  '   c2val       c2inc'//
     3                  '   corr     rmsnrm'
      write  (6,1003) 0, c0val, 0., c1val, 0., c2val, 0., 0., 0.
      write (25,1003) 0, c0val, 0., c1val, 0., c2val, 0., 0., 0.
c
      do 0500 it = 1, niter, 1
c
c *** Find the fundamental solution and derivative.
c
      do 0200 ifun = 0, nfun, 1
         vfun0(ifun) = 0.
         vfun1(ifun) = 0.
         vfun2(ifun) = 0.
 0200    continue
c
      c2arg = 1.
      do 0230 ico2 = 0, ncoe, 1
      c12arg = c2arg
      do 0220 ico1 = 0, ncoe, 1
      do 0210 ifun = 1, nfun, 1
         vfun0(ifun) = vfun0(ifun) + vcoe0(ico1,ico2,ifun) * c12arg
         vfun1(ifun) = vfun1(ifun) + vcoe1(ico1,ico2,ifun) * c12arg
         vfun2(ifun) = vfun2(ifun) + vcoe2(ico1,ico2,ifun) * c12arg
 0210    continue
         c12arg = c12arg * c1val
 0220    continue
         c2arg = c2arg * c2val
 0230    continue
c
      do 0240 ifun = 1, nfun, 1
         vfun0(ifun) = 1. / max(vfun0(ifun),1.D-03)
         vfun1(ifun) = - vfun1(ifun) * ( vfun0(ifun) ** 2 )
         vfun2(ifun) = - vfun2(ifun) * ( vfun0(ifun) ** 2 )
 0240    continue
c
c *** Find voltages and derivatives for all pairwise combinations.
c
      do 0310 ip = 1, 1+nfun, 1
         ifun = 1 + nfun - ip
         vk0(ip) = vfun0(ifun)
         vk1(ip) = vfun1(ifun)
         vk2(ip) = vfun2(ifun)
 0310    continue
c
      do 0320 ip = 2+nfun, nelc, 1
         ifun = ip - ( 1 + nfun )
         vk0(ip) = vfun0(ifun)
         vk1(ip) = vfun1(ifun)
         vk2(ip) = vfun2(ifun)
 0320    continue
c

      do 0330 ipm = 1, nelc, 1
      do 0330 ipk = 1, nelc, 1
         ip = ipk + 1 - ipm
         if (ip.le.0) ip = ip + nelc
         vkm0(ipk,ipm) = vk0(ip)
         vkm1(ipk,ipm) = vk1(ip)
         vkm2(ipk,ipm) = vk2(ip)
 0330    continue
c
      do 0340 ipk = 1, nelc, 1
      do 0340 ipm = 1, nelc, 1
      do 0340 ipn = 1, nelc, 1
         vkmn0(ipk,ipm,ipn) = vkm0(ipk,ipm) - vkm0(ipk,ipn)
         vkmn1(ipk,ipm,ipn) = vkm1(ipk,ipm) - vkm1(ipk,ipn)
         vkmn2(ipk,ipm,ipn) = vkm2(ipk,ipm) - vkm2(ipk,ipn)
 0340    continue
c
c *** Find the matrix and vector for least-squares fit.
c
      e00 = 0.
      e10 = 0.
      e20 = 0.
      c00 = 0.
      c01 = 0.
      c02 = 0.
      c10 = 0.
      c11 = 0.
      c12 = 0.
      c20 = 0.
      c21 = 0.
      c22 = 0.
c
      svrms2 = 0.
      do 0400 ipm = 1, nelc-1, 1
      do 0400 ipn = ipm+1, nelc, 1
         wtotal = 0.
         svrmsa = 0.
         svrmsb = 0.
      do 0390 ipk = 1, nelc, 1
         verr = vnrm(ipk,ipm,ipn) - vkmn0(ipk,ipm,ipn) * c0val
         wtipk = wt(ipk,ipm,ipn)
         wtotal = wtotal + wtipk
         svrmsa = svrmsa + wtipk * verr
         svrmsb = svrmsb + wtipk * verr * verr
      do 0380 ipl = 1, nelc, 1
         weight = wtipk * wt(ipl,ipm,ipn)
         vdif0 = vkmn0(ipk,ipm,ipn) - vkmn0(ipl,ipm,ipn)
         vdif1 = vkmn1(ipk,ipm,ipn) - vkmn1(ipl,ipm,ipn)
         vdif2 = vkmn2(ipk,ipm,ipn) - vkmn2(ipl,ipm,ipn)
         edif0 = vnrm(ipk,ipm,ipn) - vnrm(ipl,ipm,ipn)
         c00 = c00 + weight * vdif0 * vdif0
         c01 = c01 + weight * vdif0 * vdif1
         c02 = c02 + weight * vdif0 * vdif2
c         c10 = c10 + weight * vdif1 * vdif0
         c11 = c11 + weight * vdif1 * vdif1
         c12 = c12 + weight * vdif1 * vdif2
c         c20 = c20 + weight * vdif2 * vdif0
c         c21 = c21 + weight * vdif2 * vdif1
         c22 = c22 + weight * vdif2 * vdif2
         e00 = e00 + weight * edif0 * vdif0
         e10 = e10 + weight * edif0 * vdif1
         e20 = e20 + weight * edif0 * vdif2
 0380    continue
 0390    continue
```

```
         svrmsa = svrmsa / wtotal
         svrmsb = svrmsb / wtotal
         svrms2 = svrms2 + svrmsb - svrmsa ** 2
 0400    continue
      svrms1 = sqrt(svrms2 / wnexp)
      rmsnrm = svrms1 / svcex1
c
      c10 = c01
      c20 = c02
      c21 = c12
      e01 = e00
      e02 = e00
      e11 = e10
      e12 = e10
      e21 = e20
      e22 = e20
c
c *** Solve for c0inc, c1inc, c2inc.
c
      dt012 = 1.
      s0inc = c0val
      s1inc = 0.
      s2inc = 0.
c
      if (nshape.eq.0) then
         dt012 = c00
         s0inc = e00
         s1inc = 0.
         s2inc = 0.
         end if
c
      if (nshape.eq.1) then
         dt012 = c00*c11 - c01*c10
         s0inc = e00*c11 - c01*e10
         s1inc = c00*e11 - e01*c10
         s2inc = 0.
         end if
c
      if (nshape.eq.2) then
         dt012 = c00*c11*c22 + c02*c10*c21 + c01*c12*c20
     1         - c00*c12*c21 - c02*c11*c20 - c01*c10*c22
         s0inc = e00*c11*c22 + c02*e10*c21 + c01*c12*e20
     1         - e00*c12*c21 - c02*c11*e20 - c01*e10*c22
         s1inc = c00*e11*c22 + c02*c10*e21 + e01*c12*c20
     1         - c00*c12*e21 - c02*e11*c20 - e01*c10*c22
         s2inc = c00*c11*e22 + e02*c10*c21 + c01*e12*c20
     1         - c00*e12*c21 - e02*c11*c20 - c01*c10*e22
         end if
c
      s0inc = s0inc / dt012
      s1inc = s1inc / dt012
      s2inc = s2inc / dt012
      c0tmp = s0inc
      c0inc = c0tmp - c0val
      c1inc = s1inc / c0tmp
      c2inc = s2inc / c0tmp
c
c *** Compute adaptive damping.
c
      damp = damp0
      damp = max(damp,abs(c0inc/c0val))
      corr = max(damp1,min(damp2,damp0/damp))
c
```

```
c *** Update c0val, c1val.
c
      c0val = c0val + c0inc * corr
      c1val = c1val + c1inc * corr
      c2val = c2val + c2inc * corr
c
c *** Write result to monitor and file.
c
      c0rel = c0inc / c0val
c
      write  (6,1003) it,c0val,c0rel,c1val,c1inc,c2val,c2inc,corr,rmsnrm
      write (25,1003) it,c0val,c0rel,c1val,c1inc,c2val,c2inc,corr,rmsnrm
c
c *** Branch out of loop if tolerances satisfied.
c
      if ((abs(c0rel).lt.tolr).and.(abs(c1inc).lt.tolc).and.
     1    (abs(c2inc).lt.tolc)) go to 0600
c
c *** Iteration finished.
c
 0500 continue
c
c *** Close file.
c
 0600 continue
      close (unit=25)
c
c *** Write out fundamental solution.
c
      write (6,2000) ' '
      write (6,2000) 'Writing fundamental solution to eitaxi_sol.dat'
      open (unit=27, status='unknown', file='eitaxi_sol.dat')
      do 0700 ifun = 0, nfun, 1
         write (27,1001) vfun0(ifun) * c0val
 0700    continue
      close (unit=27)
c
c *** Write out input parameters.
c
      write (6,2000) 'Writing output parameters to eitaxi_out.dat'
      open (unit=26, status='unknown', file='eitaxi_out.dat')
      write (26,1001) convrt
      write (26,1001) hoverr
      write (26,1001) radius
      write (26,1001) curr12
      write (26,1001) sigma0
      write (26,1002) niter
      write (26,1001) damp0
      write (26,1001) damp1
      write (26,1001) damp2
      write (26,1001) tolc
      write (26,1001) tolr
      write (26,1001) c0val
      write (26,1001) c1val
      write (26,1001) c2val
      write (26,1002) nshape
      close (unit=26)
c
      write (6,*) ' '
      write (6,1004) c0val, c1val, c2val
      write (6,*) ' '
c
c *** Completed, stop.
```

```
c
      go to 0999
 0998 write (6,2000) '*** ABNORMAL STOP ***'
 0999 continue
c
      stop 'eitaxi'
      end
c
c2345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
c
```

Appendix G
Supplementary EIT Code `EITFUN.F`

The supplementary Fortran 77 code `EITFUN` by J. R. Torczynski computes fundamental voltage solutions from axisymmetric experimental datasets. These may be compared with computational fundamental voltages to evaluate the sensitivity of different electrode geometries. Both input and output are in "least significant bit" (LSB) units. For data taken with the original Data Translation® card, the data can be converted to units of volts by the equality 10^{-4} V = 1 LSB, determined during validation experiments.

The `EITFUN` algorithm begins by calculating weighted voltage "measurements" $\hat{V}_m^{(ij)}$ over all combinations of injection electrode i, withdrawal electrode j, and measurement electrode m. The weighted "measurements" are computed directly from the actual measured voltages $\tilde{V}_m^{(ij)}$ by the formula

$$\hat{V}_m^{(ij)} = \left(\frac{\sigma_L R_{col}}{I}\right)\left[\tilde{V}_m^{(ij)} - \frac{1}{N-2}\sum_{m=1}^{N} w_m^{(ij)}\tilde{V}_m^{(ij)}\right]. \tag{G.1}$$

The weights are equal to unity if all three electrode indices are different, and zero whenever at least two of the indices are equal.

$$w_m^{(ij)} = \begin{cases} 1 \text{ if } i \neq j, i \neq m, j \neq m \\ 0 \text{ otherwise} \end{cases}. \tag{G.2}$$

Next, weight functions F_n are computed as a summation of the weighted measurements over all possible combinations of i, j, and m:

$$F_n = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\sum_{m=1}^{N} w_m^{(ij)}\hat{V}_m^{(ij)}\left(\delta_{pn} - \delta_{qn}\right),\ n = 1,\mathrm{K},(N/2)-1. \tag{G.3}$$

Here, δ_{pn} is the Kronecker delta function, and the indices p and q are defined as

$$p = \left|\frac{N}{2} - |i-m|\right|, \tag{G.4}$$

$$q = \left|\frac{N}{2} - |j-m|\right|. \tag{G.5}$$

Finally, the fundamental voltages are computed. At the non-current-bearing electrodes, the fundamental voltages V_n are determined by the formula

$$V_n = \frac{1}{2N(N-1)}\left[F_n + 2\left(\sum_{s=1}^{\frac{N}{2}-1} F_s\right)\right],\ n = 1,\mathrm{K}\,,(N/2)-1. \tag{G.6}$$

By definition, the fundamental voltage V_0 at the reference electrode is zero, while the fundamental voltage at the injection electrode is computed using the fundamental voltages at the other electrodes:

$$V_{\frac{N}{2}} = \frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\frac{1}{2}\left(\hat{V}_i^{(ij)} - \hat{V}_j^{(ij)}\right) + \sum_{i=1}^{N}\sum_{j=i+1}^{N} V_{\left|\frac{N}{2}-|i-j|\right|}}{N(N-1)/2}. \tag{G.7}$$

The input file `eitfun_inp.dat` is identical to the first five lines of `eitaxi_inp.dat` in Appendix F. The data file `eitfun_exp.dat` is produced by the data acquisition codes in Appendices B and C and is identical in format to `femeit_exp.dat` in Appendix E. The output file `eitfun_sol.dat` lists the computed fundamental voltages in the order V_0, V_1, ..., $V_{\frac{N}{2}}$ (injection electrode last).

```
c
c23456789012345678901234567890123456789012345678901234567890123456789012
c
      program eitfun
c
c     Revision 19990419
c
c *** Computes fundamental voltage solution from experimental data.
c
      implicit double precision (a-h,o-z)
c
      parameter (nfun=8)
      dimension f(0:nfun)
      dimension v(0:nfun)
c
      parameter (nelc=2*nfun)
      dimension wt(nelc,nelc,nelc)
      dimension ve(nelc,nelc,nelc)
      dimension vo(nelc,nelc)
c
 1001 format (1x,d18.12)
 1002 format (1x,i4)
 2000 format (1x,a)
 2001 format (1x,a12,d18.12)
 2002 format (1x,a12,i4)
c
c *** Initialize the weights.
```

```
c
      do 0020 i1 = 1, nelc, 1
      do 0020 i2 = 1, nelc, 1
      do 0020 i3 = 1, nelc, 1
         wt(i1,i2,i3) = 1.
         if ((i1.eq.i2).or.(i1.eq.i3).or.(i2.eq.i3)) wt(i1,i2,i3) = 0.
 0020    continue
c
c *** Read in input parameters.
c
      write (6,2000) 'Reading input parameters from eitfun_inp.dat'
      open (unit=23, status='old', file='eitfun_inp.dat')
      read (23,*) convrt
      read (23,*) hoverr
      read (23,*) radius
      read (23,*) curr12
      read (23,*) sigma0
      close (unit=23)
c
      vltref = curr12 / ( convrt * hoverr * sigma0 * radius )
      vltcon = 1. / vltref
      write (6,2001) '   convrt = ', convrt
      write (6,2001) '   hoverr = ', hoverr
      write (6,2001) '   radius = ', radius
      write (6,2001) '   curr12 = ', curr12
      write (6,2001) '   sigma0 = ', sigma0
c
c *** Read in experimental voltages and normalize.
c
      write (6,2000) 'Reading experimental voltages from eitfun_exp.dat'
      open (unit=24, status='old', file='eitfun_exp.dat')
      do 0050 ip1 = 1, nelc-1, 1
      do 0050 ip2 = ip1+1, nelc, 1
         do 0040 ip = 1, nelc, 1
            read (24,*) m, n, k, vm, vr, vq
            ve(m,n,k) = vm * vltcon
 0040       continue
 0050    continue
      close (unit=24)
c
c *** Find mean voltages for each (m,n) projection.
c
      do 0150 m = 1, nelc-1, 1
      do 0140 n = m+1, nelc, 1
         vo(m,n) = 0.
         wo = 0.
         do 0130 k = 1, nelc, 1
            wo = wo + wt(m,n,k)
            vo(m,n) = vo(m,n) + wt(m,n,k) * ve(m,n,k)
 0130       continue
         vo(m,n) = vo(m,n) / wo
 0140    continue
 0150    continue
c
c *** Subtract mean voltages from experimental voltages.
c
      do 0250 m = 1, nelc-1, 1
      do 0240 n = m+1, nelc, 1
      do 0230 k = 1, nelc, 1
         ve(m,n,k) = ve(m,n,k) - vo(m,n)
 0230    continue
 0240    continue
 0250    continue
```

```
c
c *** Find the RHS vector and its sum.
c
      do 0350 m = 1, nelc-1, 1
      do 0340 n = m+1, nelc, 1
      do 0330 k = 1, nelc, 1
         i = abs(nfun-abs(m-k))
         j = abs(nfun-abs(n-k))
         f(i) = f(i) + ve(m,n,k) * wt(m,n,k)
         f(j) = f(j) - ve(m,n,k) * wt(m,n,k)
 0330    continue
 0340    continue
 0350    continue
c
      fs = 0.
      do 0380 i = 1, nfun-1, 1
         fs = fs + f(i)
 0380    continue
c
c *** Find the solution vector.
c
      znelc = dfloat(nelc)
      fac = 0.5 / ( znelc * ( znelc - 1. ) )
      v(0) = 0.
      do 0400 i = 1, nfun-1, 1
         v(i) = fac * ( f(i) + 2. * fs )
 0400    continue
c
      vnfun = 0.
      do 0450 m = 1, nelc-1, 1
      do 0440 n = m+1, nelc, 1
         ij = abs(nfun-abs(m-n))
         vnfun = vnfun + 0.5 * ( ve(m,n,m) - ve(m,n,n) ) + v(ij)
 0440    continue
 0450    continue
      v(nfun) = vnfun / dfloat(nelc*(nelc-1)/2)
c
c *** Write out fundamental solution.
c
      write (6,2000) 'Writing fundamental solution to eitfun_sol.dat'
      open (unit=27, status='unknown', file='eitfun_sol.dat')
      do 0700 ifun = 0, nfun, 1
         write (27,1001) v(ifun)
 0700    continue
      close (unit=27)
      do 0750 ifun = 0, nfun, 1
         write  (6,2001) '   v(i)   = ', v(ifun)
 0750    continue
c
c *** Write out input parameters.
c
      write (6,2000) 'Writing output parameters to eitfun_out.dat'
      open (unit=26, status='unknown', file='eitfun_out.dat')
      write (26,1001) convrt
      write (26,1001) hoverr
      write (26,1001) radius
      write (26,1001) curr12
      write (26,1001) sigma0
      close (unit=26)
c
c *** Completed, stop.
c
      stop 'eitfun'
      end
c
c23456789012345678901234567890123456789012345678901234567890123456789012
c
```

Appendix H
Supplementary EIT Code EITFUL.F

The supplementary Fortran 77 code EITFUL by J. R. Torczynski produces complete, axisymmetric "experimental" datasets from fundamental voltage solutions. These datasets may be used to validate reconstruction algorithms such as EITAXI or may be used for other verification work.

EITFUL uses two properties of the fundamental voltages to compute axisymmetric datasets. First, because the fundamental voltage solutions are computed from axisymmetric domains, the voltages are rotationally invariant. For the case of current injection at electrode *i*, withdrawal at electrode *j*, and voltage determined at electrode *m*, the domain conductivity distribution (or, equivalently, each of the electrode indices) may be azimuthally rotated by any value without altering the boundary voltages.

$$V_m^{(ij)} = V_{m+1}^{(i+1,j+1)} = V_{m+2}^{(i+2,j+2)} \ldots \tag{H.1}$$

Second, the system of FEM equations describing the boundary voltages is linear, so that fundamental voltage sets for two related injection-withdrawal combinations may be added to obtain the fundamental voltage sets for a third combination. If the fundamental voltages are known for the case of injection at electrode *a* and withdrawal at electrode *b*, and also for injection at *b* and withdrawal at *c*, then

$$V_m^{(ab)} + V_m^{(bc)} = V_m^{(ac)} + C \tag{H.2}$$

Given a combination of values for *a*, *b* and *c*, a single value of the linearity constant *C* applies for all measurements *m* in the dataset.

The input file eitful_inp.dat is identical to the first five lines of eitaxi_inp.dat in Appendix F. The data file eitful_sol.dat contains the fundamental voltages in the order V_0, V_1, ..., V_8 (injection electrode last). The output file eitful_exp.dat contains the "experimental" dataset and follows the standard format shown as femeit_exp.dat in Appendix E.

```
c
c23456789012345678901234567890123456789012345678901234567890123456789012
c
      program eitful
c
```

```
c     Revision 19990419
c
c *** Produces full EIT experimental data set
c     from axisymmetric fundamental solution.
c
      implicit double precision (a-h,o-z)
c
      parameter (nfun=8)
      dimension vfun0(0:nfun)
c
      parameter (nelc=2*nfun)
      dimension vk0(nelc)
      dimension vkm0(nelc,nelc)
      dimension vkmn0(nelc,nelc,nelc)
c
 1001 format (1x,d18.12)
 1002 format (1x,i4)
 1003 format (3(i2,1x),3(d18.12,1x))
 2000 format (1x,a)
 2001 format (1x,a12,d18.12)
 2002 format (1x,a12,i4)
c
c *** Read in input parameters.
c
      write (6,2000) 'Reading input parameters from eitful_inp.dat'
      open (unit=23, status='old', file='eitful_inp.dat')
      read (23,*) convrt
      read (23,*) hoverr
      read (23,*) radius
      read (23,*) curr12
      read (23,*) sigma0
      close (unit=23)
c
      vltref = curr12 / ( convrt * hoverr * sigma0 * radius )
      vltcon = 1. / vltref
      write (6,2001) '   convrt = ', convrt
      write (6,2001) '   hoverr = ', hoverr
      write (6,2001) '   radius = ', radius
      write (6,2001) '   curr12 = ', curr12
      write (6,2001) '   sigma0 = ', sigma0
c
c *** Read in the fundamental solution.
c
      write (6,2000) 'Reading fundamental solution from eitful_sol.dat'
      open (unit=27, status='old', file='eitful_sol.dat')
      do 0110 ifun = 0, nfun, 1
         read (27,*) vfun0(ifun)
 0110    continue
      close (unit=27)
c
c *** Find all experimental voltages.
c
      do 0310 ip = 1, 1+nfun, 1
         ifun = 1 + nfun - ip
         vk0(ip) = vfun0(ifun)
 0310    continue
c
      do 0320 ip = 2+nfun, nelc, 1
         ifun = ip - ( 1 + nfun )
         vk0(ip) = vfun0(ifun)
 0320    continue
c
      do 0330 ipm = 1, nelc, 1
      do 0330 ipk = 1, nelc, 1
         ip = ipk + 1 - ipm
         if (ip.le.0) ip = ip + nelc
         vkm0(ipk,ipm) = vk0(ip)
 0330    continue
c
      do 0340 ipk = 1, nelc, 1
      do 0340 ipm = 1, nelc, 1
      do 0340 ipn = 1, nelc, 1
         vkmn0(ipk,ipm,ipn) = vkm0(ipk,ipm) - vkm0(ipk,ipn)
 0340    continue
c
c *** Write out unnormalized experimental voltages.
c
      write (6,2000) 'Writing experimental data to eitful_exp.dat'
      open (unit=24, status='unknown', file='eitful_exp.dat')
      vquad = 0.
      do 0050 ip1 = 1, nelc-1, 1
      do 0050 ip2 = ip1+1, nelc, 1
         do 0040 ip = 1, nelc, 1
            vmagn = vkmn0(ip,ip1,ip2) * vltref
            vcarr = vmagn
            write (24,1003) ip1, ip2, ip, vmagn, vcarr, vquad
 0040       continue
 0050    continue
      close (unit=24)
c
c *** Write out input parameters.
c
      write (6,2000) 'Writing output parameters to eitful_out.dat'
      open (unit=26, status='unknown', file='eitful_out.dat')
      write (26,1001) convrt
      write (26,1001) hoverr
      write (26,1001) radius
      write (26,1001) curr12
      write (26,1001) sigma0
      close (unit=26)
c
c *** Completed, stop.
c
      go to 999
  998 write (6,2000) '*** ABNORMAL STOP ***'
  999 continue
c
      stop 'eitful'
      end
c
c23456789012345678901234567890123456789012345678901234567890123456789012
c
```

Appendix I
GDT Reconstruction Code GDTAXI.F

This Fortran 77 program by J. R. Torczynski determines the radial conductivity distribution in a two-dimensional cylindrical domain from gamma-ray attenuation measurements along many parallel beam paths through the domain. A complete description of the algorithm may be found in Torczynski *et al.* (1997).

GDTAXI reads from and writes to the following files:

gdtaxi_inp.dat	general input parameters (input)
gdtaxi_ful.dat	photon count rate data from full column with no gas present (input)
gdtaxi_emp.dat	photon count rate data from empty column with no liquid present (input)
gdtaxi_flo.dat	photon count rate data from column during flow of interest (input)
gdtaxi_out.dat	general output parameters and coefficients of volume fraction profiles (output)
gdtaxi_gas.dat	gas volume fraction profile (output)
gdtaxi_liq.dat	liquid volume fraction profile (output)

Examples of some input file formats follow.

gdtaxi_inp.dat:

9.525	column inner radius (cm)
19.79	column x-midpoint (cm)
0.1	thickness of boundary layer within which data is discarded (cm)
0	clipping suppressed or enabled (0 or 1, respectively)
0.	lower value for clipped gas volume fraction (usually 0)
1.	upper value for clipped gas volume fraction (usually 1)
0.0E-06	time constant for nonlinear detector response (s), usually set to 0
4	degree of polynomial fit (even integer: 0 = constant, 2 = quadratic, 4 = quartic, etc.)

gdtaxi_out.dat echoes the contents of gdtaxi_inp.dat, then reports the column-averaged gas and liquid volume fractions, and finally prints the polynomial coefficients of the computed volume fraction distributions.

`gdtaxi_ful.dat, gdtaxi_emp.dat, gdtaxi_flo.dat:`

These are primary output files from the LabView program that controls the GDT system. The first line of each file describes the horizontal and vertical motion of the source and detector during the scan; `GDTAXI` compares this information in all three files to verify that they are measurements of the same column geometry. The second line, which is not used by `GDTAXI`, reports parameters used by LabView to compute count rates from the output of the multichannel analyzer. The remaining lines of each input file contain gamma beam positions and the count rates at each position. Columns 1 and 2 list the horizontal and vertical coordinates, respectively, of the source and detector; column 3 contains the count rate at the detector at those coordinates. Columns 4 through 6 are not used.

```
10.780 1.000  18.000 26.650 0.000  0.000  0.000  0.000
60     280    480    4.000  5.000  0.100  0.100
10.78  26.65  8208.94       148.49 55.28  104.45
11.78  26.65  6295.64       114.12 55.17  104.74
12.78  26.65  5314.41        96.14 55.28  104.82
  .      .        .            .     .      .
  .      .        .            .     .      .
  .      .        .            .     .      .
28.78  26.65  8213.49       148.87 55.17  105.37
```

```
c
c23456789012345678901234567890123456789012345678901234567890123456789012
c
      program gdtaxi
c
c     Revision 19990420
c
c *** Gamma-densitometry tomography axisymmetric reconstruction
c     using even radial polynomials.
c
      implicit double precision (a-h,o-z)
c
      parameter (nsm=1000)
      parameter (nfcnm=10)
      dimension spos(nsm), full(nsm), empt(nsm), flow(nsm)
      dimension snrm(nsm), void(nsm)
      dimension nexp(nfcnm), amat(nfcnm,nfcnm), bvec(nfcnm)
      dimension cmat(nfcnm,nfcnm)
      dimension ravf(nsm), axvf(nsm)
      dimension cravf(nfcnm), caxvf(nfcnm)
      dimension cramf(nfcnm), caxmf(nfcnm)
      dimension ccoeff(nfcnm,nfcnm), dcoeff(nfcnm,nfcnm)
c
 1001 format (1x,d18.12)
 1002 format (1x,i4)
 1003 format (1x,i4,5(1x,d11.5))
 1004 format (6(1x,d11.5))
 2000 format (1x,a)
 2001 format (1x,a12,d18.12)
 2002 format (1x,a12,i4)
c
c *** Read in geometric and fitting information.
c
```

```
      write (6,2000) 'Reading input parameters from gdtaxi_inp.dat'
      open (unit=24, status='old', file='gdtaxi_inp.dat')
      read (24,*) rinner
      read (24,*) scentr
      read (24,*) dxedge
      read (24,*) iclip
      read (24,*) cliplo
      read (24,*) cliphi
      read (24,*) tau
      read (24,*) nexpm
      write (6,2001) '   rinner = ', rinner
      write (6,2001) '   scentr = ', scentr
      write (6,2001) '   dxedge = ', dxedge
      write (6,2002) '   iclip  = ', iclip
      write (6,2001) '   cliplo = ', cliplo
      write (6,2001) '   cliphi = ', cliphi
      write (6,2001) '   tau    = ', tau
      write (6,2002) '   nexpm  = ', nexpm
      close (unit=24)
c
c *** Read in full, empty, flow scans.
c *** Do some error checking for consistent files.
c
      write (6,2000) 'Reading experimental data from '
      write (6,2000) '   gdtaxi_ful.dat gdtaxi_emp.dat gdtaxi_flo.dat'
      open (unit=21, status='old', file='gdtaxi_ful.dat')
      open (unit=22, status='old', file='gdtaxi_emp.dat')
      open (unit=23, status='old', file='gdtaxi_flo.dat')
      read (21,*) x1f, dxf, xnf, y1f, dyf, ynf
      read (22,*) x1e, dxe, xne, y1e, dye, yne
      read (23,*) x1b, dxb, xnb, y1b, dyb, ynb
      nxf = nint(xnf)
      nyf = nint(ynf)
      nxe = nint(xne)
      nye = nint(yne)
      nxb = nint(xnb)
      nyb = nint(ynb)
      if ((nyf.ne.0).or.(nye.ne.0).or.(nyb.ne.0)) then
         close (unit=21)
         close (unit=22)
         close (unit=23)
         write (6,*) '*** SCAN IS IN Y-DIRECTION ***'
         write (6,*) 'ful ', nyf, ' emp ', nye, ' flo ', nyb
         go to 998
         end if
      if ((nxf.ne.nxe).or.(nxf.ne.nxb)) then
         close (unit=21)
         close (unit=22)
         close (unit=23)
         write (6,*) '*** SCANS HAVE DIFFERENT NUMBERS OF POINTS ***'
         write (6,*) 'ful ', nxf, ' emp ', nxe, ' flo ', nxb
         go to 998
         end if
      tol = 0.05 * rinner
      if ((abs(x1f-x1e).gt.tol).or.(abs(x1f-x1b).gt.tol)) then
c         close (unit=21)
c         close (unit=22)
c         close (unit=23)
         write (6,*) '*** SCANS HAVE DIFFERENT ORIGINS ***'
         write (6,*) 'ful ', x1f, ' emp ', x1e, ' flo ', x1b
c         go to 998
         end if
      if ((abs(dxf-dxe).gt.tol).or.(abs(dxf-dxb).gt.tol)) then
```

```
         close (unit=21)
         close (unit=22)
         close (unit=23)
         write (6,*) '*** SCANS HAVE DIFFERENT STEP SIZES ***'
         write (6,*) 'ful ', dxf, ' emp ', dxe, ' flo ', dxb
         go to 998
         end if
      if (nxf+1.gt.nsm) then
         close (unit=21)
         close (unit=22)
         close (unit=23)
         write (6,*) '*** NOT ENOUGH POINTS AVAILABLE ***'
         write (6,*) nxf, nsm
         go to 998
         end if
c
c *** Skip header information
c
      read (21,*) tmlive
      read (22,*) tmlive
      read (23,*) tmlive
c
c *** Read in data and compute the nominal rates.
c
      ns = nxf + 1
      do 100 is = 1, ns, 1
         read (21,*,end=998,err=998) sposx, sposy, frate
         read (22,*,end=998,err=998) sposx, sposy, erate
         read (23,*,end=998,err=998) sposx, sposy, brate
         spos(is) = sposx
         full(is) = frate
         empt(is) = erate
         flow(is) = brate
  100    continue
c
c *** Close files.
c
      close (unit=21)
      close (unit=22)
      close (unit=23)
c
c *** Compute best fit to ray averages.
c
      write (6,2000) 'Computing results'
      nfcn = nexpm / 2 + 1
      if (nfcn.gt.nfcnm) then
         close (unit=24)
         write (6,2000) '*** NOT ENOUGH FITTING FUNCTIONS AVAILABLE ***'
         write (6,2000) nfcn, nfcnm
         go to 998
         end if
      do 120 ifcn = 1, nfcn, 1
         nexp(ifcn) = 2 * ( ifcn - 1 )
  120    continue
c
c *** Correct the rates for nonlinear detector response.
c
      do 140 is = 1, ns, 1
         full(is) = full(is) / ( 1. - full(is) * tau )
         empt(is) = empt(is) / ( 1. - empt(is) * tau )
         flow(is) = flow(is) / ( 1. - flow(is) * tau )
  140    continue
c
```

```
c *** Take logarithms to get absorptions.
c
      do 150 is = 1, ns, 1
         full(is) = - log( full(is) )
         empt(is) = - log( empt(is) )
         flow(is) = - log( flow(is) )
  150    continue
c
c *** Compute left and right array bounds for inner diameter.
c
      s1 = scentr - rinner + dxedge
      s2 = scentr + rinner - dxedge
      ns1 = 1
      ns2 = 0
      do 160 is = 1, ns, 1
         if (spos(is).le.s1) ns1 = ns1 + 1
         if (spos(is).lt.s2) ns2 = ns2 + 1
  160    continue
c
c *** Compute the normalized position and the average void fraction on rays.
c
      do 180 is = 1, ns, 1
         snrm(is) = ( spos(is) - scentr ) / rinner
         void(is) = 0.
         if ((is.ge.ns1).and.(is.le.ns2)) then
            void(is) = ( full(is) - flow(is) ) / ( full(is) - empt(is) )
            if (iclip.ne.0) void(is) = max(min(cliphi,void(is)),cliplo)
            end if
  180    continue
c
c *** Compute matrix and vector for least-squares fit of data.
c
      do 200 ifcn1 = 1, nfcn, 1
         bvec(ifcn1) = 0.
      do 200 ifcn2 = 1, nfcn, 1
         amat(ifcn1,ifcn2) = 0.
  200    continue
c
      do 260 is = ns1, ns2, 1
      do 240 ifcn1 = 1, nfcn, 1
         fcn1 = 1.
         if (nexp(ifcn1).ne.0) fcn1 = snrm(is)**nexp(ifcn1)
         bvec(ifcn1) = bvec(ifcn1) + void(is) * fcn1
      do 220 ifcn2 = 1, nfcn, 1
         fcn2 = 1.
         if (nexp(ifcn2).ne.0) fcn2 = snrm(is)**nexp(ifcn2)
         amat(ifcn1,ifcn2) = amat(ifcn1,ifcn2) + fcn1 * fcn2
  220    continue
  240    continue
  260    continue
c
c *** Solve the linear system.
c
      do 280 ifcn1 = 1, nfcn, 1
         cravf(ifcn1) = bvec(ifcn1)
         do 270 ifcn2 = 1, nfcn, 1
            cmat(ifcn1,ifcn2) = amat(ifcn1,ifcn2)
  270       continue
  280    continue
c
      call gauss1(cmat,nfcn,nfcnm,cravf)
c
c *** Compute the c and d coefficients needed to transform.
```

```
c
      do 300 ifcn1 = 1, nfcn, 1
      do 300 ifcn2 = 1, nfcn, 1
         ccoeff(ifcn1,ifcn2) = cfcn(ifcn1-1,ifcn2-1)
         dcoeff(ifcn1,ifcn2) = dfcn(ifcn1-1,ifcn2-1)
  300    continue
c
c *** Convert the ray averaged void fraction coefficients
c *** into radial void fraction coefficients.
c
      do 340 ifcn1 = 1, nfcn, 1
         caxvf(ifcn1) = 0.
      do 320 ifcn2 = 1, nfcn, 1
         caxvf(ifcn1) = caxvf(ifcn1) + ccoeff(ifcn1,ifcn2)*cravf(ifcn2)
  320    continue
  340    continue
c
c *** Calculate the ray averaged void fraction fit
c *** and the radial void fraction fit.
c
      do 380 is = 1, ns, 1
         ravf(is) = 0.
         axvf(is) = 0.
         if ((is.ge.ns1).and.(is.le.ns2)) then
            do 360 ifcn = 1, nfcn, 1
               fcn = snrm(is) ** nexp(ifcn)
               ravf(is) = ravf(is) + cravf(ifcn) * fcn
               axvf(is) = axvf(is) + caxvf(ifcn) * fcn
  360          continue
            end if
  380    continue
c
c *** Compute area-averaged void fraction and 1 - void fraction.
c
      avggas = 0.
      do 390 ifcn = 1, nfcn, 1
         avggas = avggas + caxvf(ifcn) * 2. / (2. + dfloat(nexp(ifcn)))
  390    continue
      avgliq = 1. - avggas
      write (6,2001) '   avggas = ', avggas
      write (6,2001) '   avgliq = ', avgliq
c
c *** Compute 1 - void fraction fit.
c
      do 400 ifcn = 1, nfcn, 1
         caxmf(ifcn) = - caxvf(ifcn)
         cramf(ifcn) = - cravf(ifcn)
         if (ifcn.eq.1) then
            caxmf(ifcn) = caxmf(ifcn) + 1.
            cramf(ifcn) = cramf(ifcn) + 1.
            end if
  400    continue
      do 410 ifcn = 1, nfcn, 1
         write (6,2001) '   caxvfi = ', caxvf(ifcn)
  410    continue
      do 420 ifcn = 1, nfcn, 1
         write  (6,1003) nexp(ifcn), caxvf(ifcn), cravf(ifcn),
     1                   caxmf(ifcn), cramf(ifcn)
  420    continue
c
c *** Write results to output files.
c
      write (6,2000) 'Writing output parameters to gdtaxi_out.dat'
      open (unit=30, status='unknown', file='gdtaxi_out.dat')
      write (30,1001) rinner
      write (30,1001) scentr
      write (30,1001) dxedge
      write (30,1002) iclip
      write (30,1001) cliplo
      write (30,1001) cliphi
      write (30,1001) tau
      write (30,1002) nexpm
      write (30,1001) avggas
      write (30,1001) avgliq
      do 430 ifcn = 1, nfcn, 1
         write (30,1001) caxvf(ifcn)
  430    continue
      do 440 ifcn = 1, nfcn, 1
         write (30,1003) nexp(ifcn), caxvf(ifcn), cravf(ifcn),
     1                   caxmf(ifcn), cramf(ifcn)
  440    continue
      close (unit=30)
c
      write (6,2000) 'Writing profiles to gdtaxi_gas.dat gdtaxi_liq.dat'
      open (unit=28, status='unknown', file='gdtaxi_gas.dat')
      open (unit=29, status='unknown', file='gdtaxi_liq.dat')
      do 450 is = 1, ns, 1
         if ((is.ge.ns1).and.(is.le.ns2)) then
            sc = spos(is) - scentr
            vmix = 1. - void(is)
            ramf = 1. - ravf(is)
            axmf = 1. - axvf(is)
            write (28,1004) snrm(is), void(is), ravf(is), axvf(is),
     1                      spos(is), sc
            write (29,1004) snrm(is), vmix,     ramf,     axmf,
     1                      spos(is), sc
            end if
  450    continue
      close (unit=28)
      close (unit=29)
c
      write (6,*) ' '
      write (6,1004) (caxvf(i),i=1,nfcn,1)
      write (6,*) ' '
c
c *** completed, stop.
c
      go to 999
  998 write (6,2000) '*** ABNORMAL STOP ***'
  999 stop 'gdtaxi'
      end
c
c23456789012345678901234567890123456789012345678901234567890123456789012
c
      function cfcn(m,n)
c
c *** Computes backward transformation coefficients.
c
      implicit double precision (a-h,o-z)
c
      cfcn = 0.
      if (m.le.n) then
         m2 = 2 * m
         n2 = 2 * n
         n2m2= n2 - m2
         nm = n - m
         fact = - dfloat(2 * m + 1) / dfloat(2**n2 * ( n2m2 - 1 ) )
         bin1 = bico(n2m2,nm)
         bin2 = bico(m2,m)
         cfcn = fact * bin1 * bin2
         end if
c
      return
      end
c
c2345678901234567890123456789012345678901234567890123456789012345678901012
c
      function dfcn(n,m)
c
c *** Computes forward transformation coefficients.
c
      implicit double precision (a-h,o-z)
c
      dfcn = 0.
      if (n.le.m) then
         m2 = 2 * m
         n2 = 2 * n
         m2n2= m2 - n2
         mn = m - n
         fact = dfloat(2**n2) / dfloat(2 * m + 1)
         bin1 = bico(m2n2,mn)
         bin2 = bico(m2,m)
         dfcn = fact * bin1 / bin2
         end if
c
      return
      end
c
c2345678901234567890123456789012345678901234567890123456789012345678901012
c
      subroutine gauss1(a,n,np,b)
c
c *** Gauss-Jordan elimination with full pivoting (Numerical Recipes).
c
      implicit double precision (a-h,o-z)
c
      parameter (nmax=10)
      dimension a(np,np), b(np)
      dimension ipiv(nmax), indxr(nmax), indxc(nmax)
c
      do 0100 j = 1, n, 1
         ipiv(j) = 0
 0100    continue
c
      do 0700 i = 1, n, 1
         big = 0.
         do 0250 j = 1, n, 1
            if (ipiv(j).ne.1) then
               do 0200 k = 1, n, 1
                  if (ipiv(k).eq.0) then
                     if (abs(a(j,k)).ge.big) then
                        big = abs(a(j,k))
                        irow = j
                        icol = k
                        end if
                  else if (ipiv(k).gt.1) then
                     pause 'Singular matrix'
                     end if
 0200             continue
```

```
               end if
 0250       continue
         ipiv(icol) = ipiv(icol) + 1
c
         if (irow.ne.icol) then
            do 0300 l = 1, n, 1
               dum = a(irow,l)
               a(irow,l) = a(icol,l)
               a(icol,l) = dum
 0300          continue
            dum = b(irow)
            b(irow) = b(icol)
            b(icol) = dum
            end if
         indxr(i) = irow
         indxc(i) = icol
         if (a(icol,icol).eq.0.) pause 'Singular matrix'
         pivinv = 1. / a(icol,icol)
         a(icol,icol) = 1.
         do 0400 l = 1, n, 1
            a(icol,l) = a(icol,l) * pivinv
 0400       continue
         b(icol) = b(icol) * pivinv
         do 0600 ll = 1, n, 1
            if (ll.ne.icol) then
               dum = a(ll,icol)
               a(ll,icol) = 0.
               do 0500 l = 1, n, 1
                  a(ll,l) = a(ll,l) - a(icol,l) * dum
 0500             continue
               b(ll) = b(ll) - b(icol) * dum
               end if
 0600       continue
 0700    continue
c
      do 0900 l = n, 1, -1
         if (indxr(l).ne.indxc(l)) then
            do 0800 k = 1, n, 1
               dum = a(k,indxr(l))
               a(k,indxr(l)) = a(k,indxc(l))
               a(k,indxc(l)) = dum
 0800          continue
            end if
 0900    continue
c
      return
      end
c
c23456789012345678901234567890123456789012345678901234567890123456789012
c
      function bico(n,k)
c
c *** Binomial coefficient.
c
      implicit double precision (a-h,o-z)
c
      bico = anint(exp(factln(n)-factln(k)-factln(n-k)))
c
      return
      end
c
c23456789012345678901234567890123456789012345678901234567890123456789012
c
```

```
      function factln(n)
c
c *** Logarithm of factorial.
c
      implicit double precision (a-h,o-z)
c
      sum = 0.
      do 0100 i = 1, n, 1
         sum = sum + log(dfloat(i))
 0100    continue
c
      factln = sum
c
      return
      end
c
c234567890123456789012345678901234567890123456789012345678901234567890l2
c
```

Appendix J
Three-Phase Reconstruction Code `GDTEIT.F`

This Fortran 77 program by J. R. Torczynski uses results from the reconstruction codes `GDTAXI` and `EITAXI` to determine radial distributions in three-phase, solid-gas-liquid vertical column flows. The algorithm is discussed fully in Section 5.2. An example of the single input file follows.

`gdteit_inp.dat:`

9.525	column inner radius, R_{col} (cm)	
0.0001	gas phase gamma ray attenuation coefficient (1/cm)	
0.0856	liquid phase gamma ray attenuation coefficient (1/cm)	
0.0866	solid phase gamma ray attenuation coefficient (1/cm)	
0.6	Maxwell-Hewitt coefficient, α (unitless)	
1.4	$\}C_0$	
0.	$\}C_1$	Coefficients of conductivity profile from `EITAXI` (Eq. 3.8)
0.	$\}C_2$	
4	degree n of polynomial fit in `GDTAXI` reconstruction	
0.2	$\}C_0$	
0.0	$\}C_2$	Coefficients of phase profile from `GDTAXI`
0.0	$\}C_4$	

The polynomial phase profile from the code `GDTAXI` takes the form $\varepsilon_L(r) = 1 - \sum_{\substack{i=0 \\ i\,\text{even}}}^{n} C_i \left(\frac{r}{R_{col}}\right)^i$.

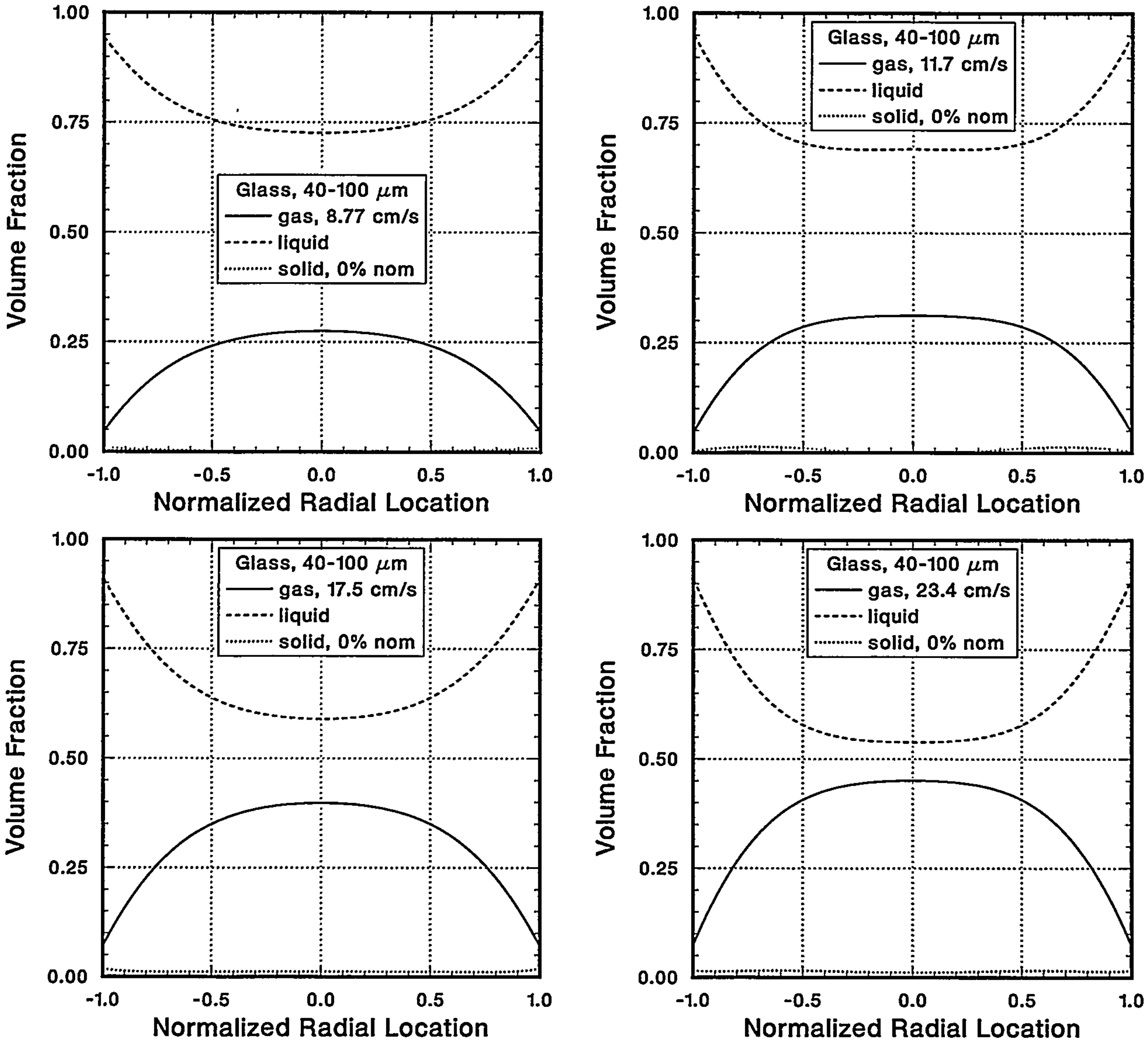
```
c
c234567890123456789012345678901234567890123456789012345678901234567890l2
c
      program gdteit
c
c     Revision 19990420
c
c *** Uses results from gdtaxi and eitaxi
c     to determine three-phase material distribution.
c
      implicit double precision (a-h,o-z)
```

```
c
      parameter (ngdtm=10)
      parameter (neitm=2)
      parameter (nr=50)
c
      dimension cgdt(0:ngdtm)
      dimension ceit(0:neitm)
      dimension eg(0:nr), el(0:nr), es(0:nr)
      dimension radnrm(0:nr)
      dimension raddim(0:nr)
      dimension cndnrm(0:nr)
      dimension gamnrm(0:nr)
c
 1001 format (1x,d18.12)
 1002 format (1x,i4)
 1003 format (1x,i4,1x,d18.12)
 1004 format (7(1x,d11.5))
 2000 format (1x,a)
 2001 format (1x,a12,d18.12)
 2002 format (1x,a12,i4)
c
c *** Read physical parameters and profile coefficients.
c
      write (6,2000) 'Reading input parameters from gdteit_inp.dat'
      open (unit=21, status='old', file='gdteit_inp.dat')
      read (21,*) radius
      read (21,*) gamgas
      read (21,*) gamliq
      read (21,*) gamsol
      read (21,*) heweta
      neit = neitm
      do 0100 ieit = 0, neit, 1
         read (21,*) ceit(ieit)
 0100    continue
      read (21,*) ngdt2
      ngdt = ngdt2 / 2
      do 0150 igdt = 0, ngdt, 1
         read (21,*) cgdt(igdt)
 0150    continue
      close (unit=21)
      write (6,2001) '   radius = ', radius
      write (6,2001) '   gamgas = ', gamgas
      write (6,2001) '   gamliq = ', gamliq
      write (6,2001) '   gamsol = ', gamsol
      write (6,2001) '   heweta = ', heweta
      do 0200 ieit = 0, neit, 1
         write (6,2001) '   ceit(i)= ',  ceit(ieit)
 0200    continue
      write (6,2002) '   ngdt2  = ', ngdt2
      do 0250 igdt = 0, ngdt, 1
         write (6,2001) '   cgdt(i)= ',  cgdt(igdt)
 0250    continue
c
c *** Determine the gas, liquid, and solid volume fraction profiles.
c
      write (6,2000) 'Computing results'
      gampar = ((gamsol - gamgas) / (gamliq - gamgas)) - 1.
      do 0400 ir = 0, nr, 1
         rn = dfloat(ir) / dfloat(nr)
         radnrm(ir) = rn
         raddim(ir) = rn * radius
         cn = 1.
         cn = cn + ceit(1) * (2. * rn ** 2 - 1.)
         cn = cn + ceit(2) * (1. - 6. * rn ** 2 + 6. * rn ** 4)
         cn = cn / ceit(0)
         cndnrm(ir) = cn
         gn = cgdt(0)
         do 0300 igdt = 1, ngdt, 1
            gn = gn + cgdt(igdt) * rn ** (2 * igdt)
 0300       continue
         gamnrm(ir) = gn
         call phase3(heweta,cn,gampar,gn,eg0,el0,es0)
         eg(ir) = eg0
         el(ir) = el0
         es(ir) = es0
 0400    continue
c
c *** Determine the average volume fractions.
c
      avgeg  = radnrm(0) * eg(0) + radnrm(nr) * eg(nr)
      avgel  = radnrm(0) * el(0) + radnrm(nr) * el(nr)
      avges  = radnrm(0) * es(0) + radnrm(nr) * es(nr)
      avgcnd = radnrm(0) * cndnrm(0) + radnrm(nr) * cndnrm(nr)
      avggam = radnrm(0) * gamnrm(0) + radnrm(nr) * gamnrm(nr)
      do 0500 ir = 1, nr-1, 2
         avgeg  = avgeg  + 4. * radnrm(ir) * eg(ir)
         avgel  = avgel  + 4. * radnrm(ir) * el(ir)
         avges  = avges  + 4. * radnrm(ir) * es(ir)
         avgcnd = avgcnd + 4. * radnrm(ir) * cndnrm(ir)
         avggam = avggam + 4. * radnrm(ir) * gamnrm(ir)
 0500    continue
      do 0550 ir = 2, nr-2, 2
         avgeg  = avgeg  + 2. * radnrm(ir) * eg(ir)
         avgel  = avgel  + 2. * radnrm(ir) * el(ir)
         avges  = avges  + 2. * radnrm(ir) * es(ir)
         avgcnd = avgcnd + 2. * radnrm(ir) * cndnrm(ir)
         avggam = avggam + 2. * radnrm(ir) * gamnrm(ir)
 0550    continue
      avgeg  = avgeg  * 2. / ( 3. * dfloat(nr) )
      avgel  = avgel  * 2. / ( 3. * dfloat(nr) )
      avges  = avges  * 2. / ( 3. * dfloat(nr) )
      avgcnd = avgcnd * 2. / ( 3. * dfloat(nr) )
      avggam = avggam * 2. / ( 3. * dfloat(nr) )
      write (6,2001) '   avgeg  = ', avgeg
      write (6,2001) '   avgel  = ', avgel
      write (6,2001) '   avges  = ', avges
      write (6,2001) '   avgcnd = ', avgcnd
      write (6,2001) '   avggam = ', avggam
c
c *** Write profile file.
c
      write (6,2000) 'Writing profiles to gdteit_gls.dat'
      open (unit=28, status='unknown', file='gdteit_gls.dat')
      do 0700 ir = nr, 1, -1
         write (28,1004) -radnrm(ir), -raddim(ir),
     1      eg(ir), el(ir), es(ir), cndnrm(ir), gamnrm(ir)
 0700    continue
      do 0750 ir = 0, nr, 1
         write (28,1004) radnrm(ir), raddim(ir),
     1      eg(ir), el(ir), es(ir), cndnrm(ir), gamnrm(ir)
 0750    continue
c
c *** Write output file.
c
      write (6,2000) 'Writing output parameters to gdteit_out.dat'
      open (unit=29, status='unknown', file='gdteit_out.dat')
      write (29,1001) radius
      write (29,1001) gamgas
      write (29,1001) gamliq
      write (29,1001) gamsol
      write (29,1001) heweta
      do 0800 ieit = 0, neit, 1
         write (29,1001) ceit(ieit)
 0800    continue
      write (29,1002) ngdt2
      do 0850 igdt = 0, ngdt, 1
         write (29,1001) cgdt(igdt)
 0850    continue
      write (29,1001) avgeg
      write (29,1001) avgel
      write (29,1001) avges
      write (29,1001) avgcnd
      write (29,1001) avggam
      close (unit=29)
c
c *** Stop, end.
c
      stop 'gdteit'
      end
c
c23456789012345678901234567890123456789012345678901234567890123456789012
c
      subroutine phase3(heweta,cn,gampar,gn,eg0,el0,es0)
c
c *** Solves three simultaneous equations
c     (phases sum to unity, gdt, eit) analytically
c     for gas, liquid, solid volume fractions.
c     Uses modified Maxwell-Hewitt relation: (1-x)/(1+a*x)
c     Uses this relation recursively for solid-liquid, gas-solliqmix.
c
      implicit double precision (a-h,o-z)
c
      a = heweta
      b = gn
      c = gampar
      d = cn
c
      aa = 1. + c - a * a * d + a * c * d
      bb = a * a * b * d + c * d - (1. + b + 2. * c + a * d + a * c * d)
      cc = b + c + a * b * d - c * d
c
      eg0 = - ( bb + sqrt(bb * bb - 4. * aa * cc) ) / ( 2. * aa )
      tmp = ( 1. - eg0 ) / ( 1. + a * eg0 )
      tmp = d / tmp
      tmp = ( 1. - tmp ) / ( 1. + a * tmp )
      es0 = ( 1. - eg0 ) * tmp
      el0 = 1. - eg0 - es0
c
      return
      end
c
c23456789012345678901234567890123456789012345678901234567890123456789012
c
```

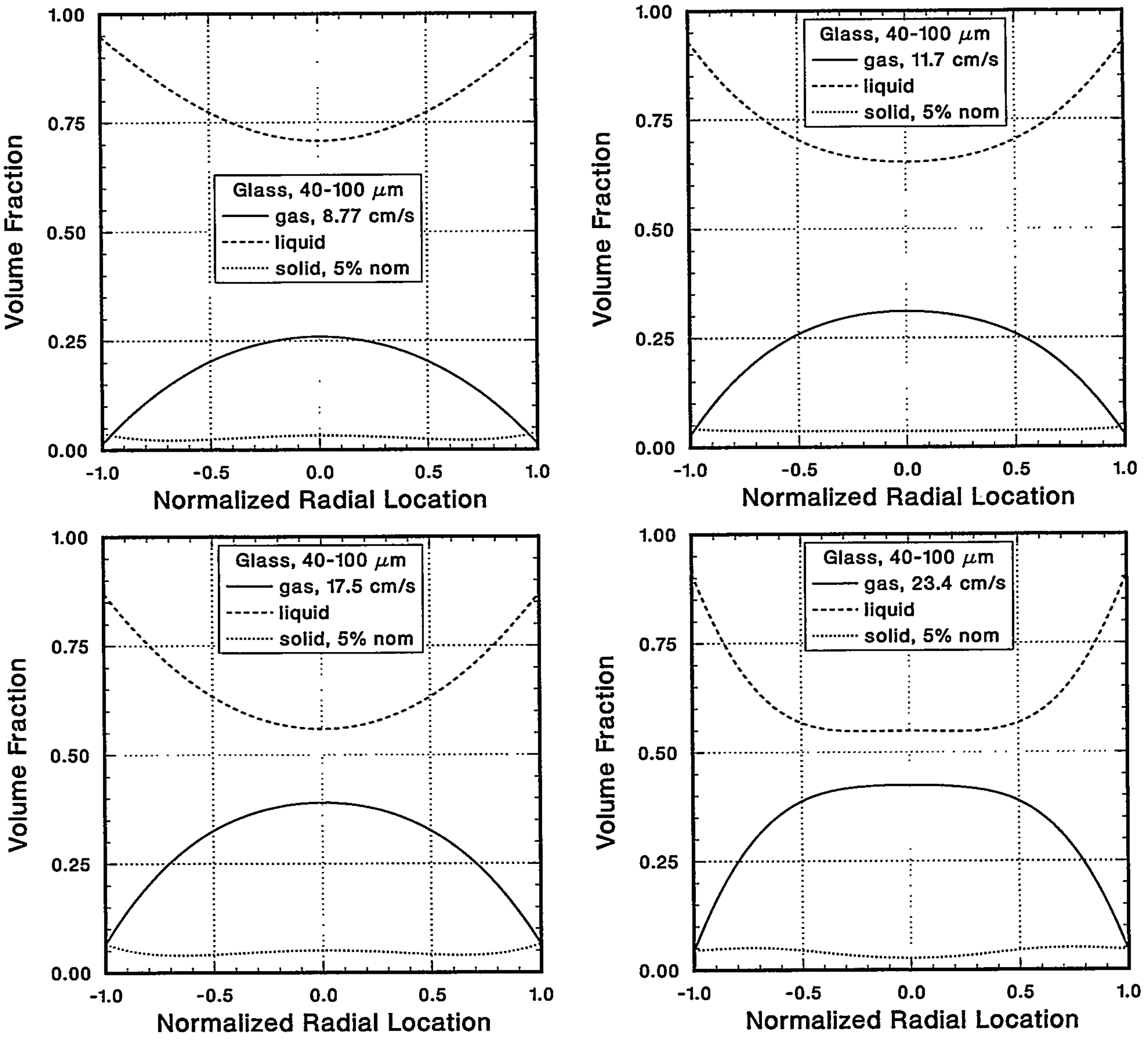
Appendix K
Phase Volume Fraction Profiles

This appendix contains graphs of the radial phase volume fraction profiles for all three-phase flow conditions discussed in Chapter 5. Table 5.2 lists all combinations of gas flow rates, particle types and solid volume fractions tested. The conditions that produced each graph are listed in its legend. Note that the flow conditions marked “solid, 0% nominal” include a surfactant particle coating dissolved in the liquid phase, but no solid particles. The flow conditions marked “deionized water” contain no surfactant and no solid phase. The presence of the surfactant changes the gas phase distribution through a reduction in surface tension, as comparison of the graphs shows.

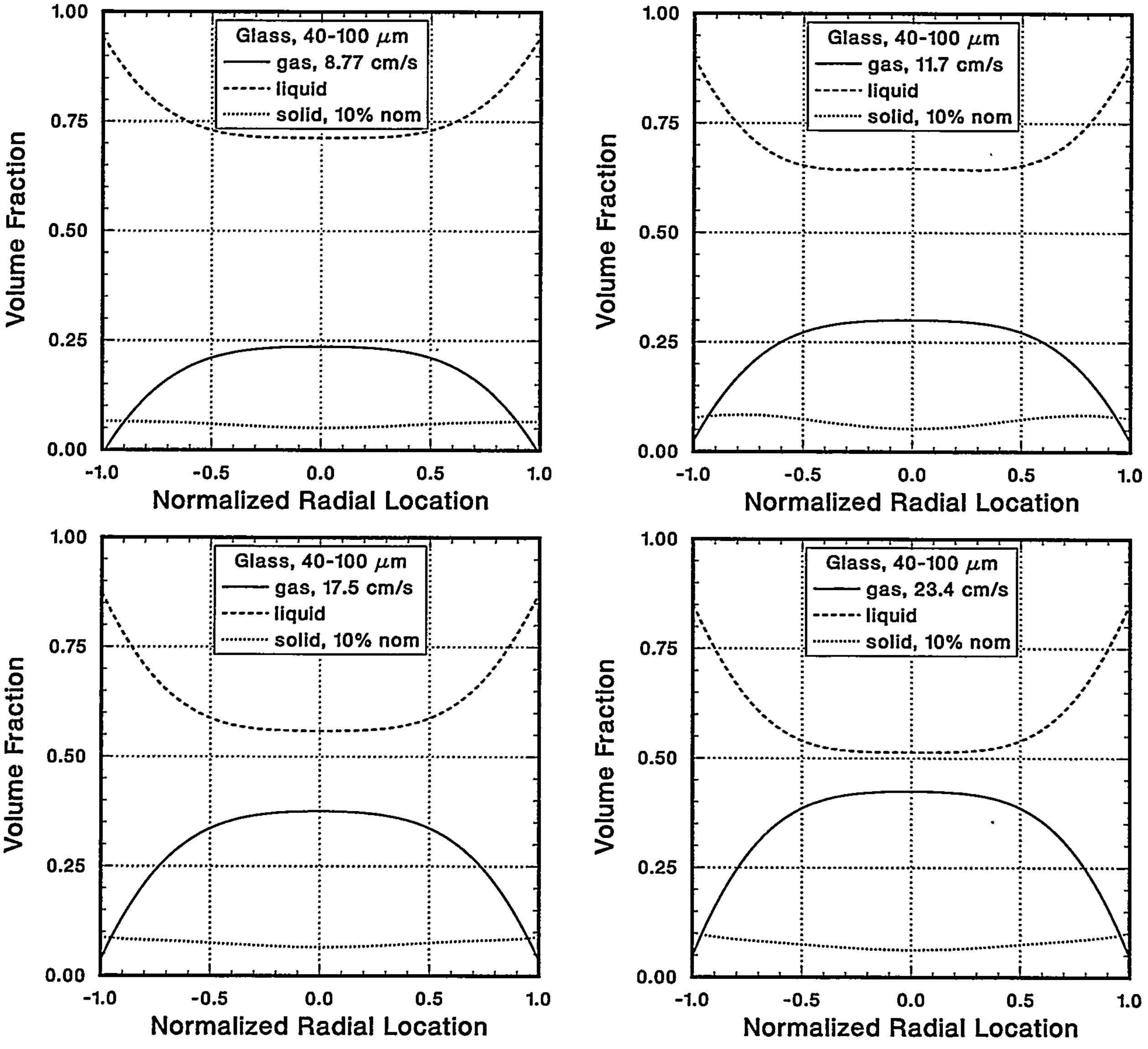
Glass: 40-100 µm, 0% by volume



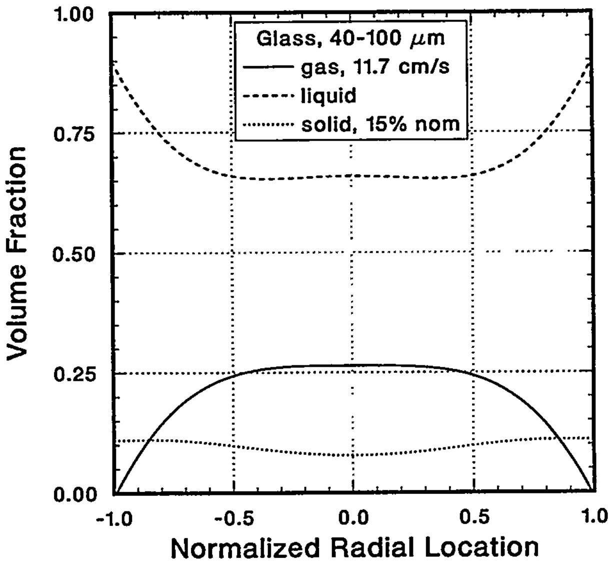
Glass: 40-100 µm, 5% by volume

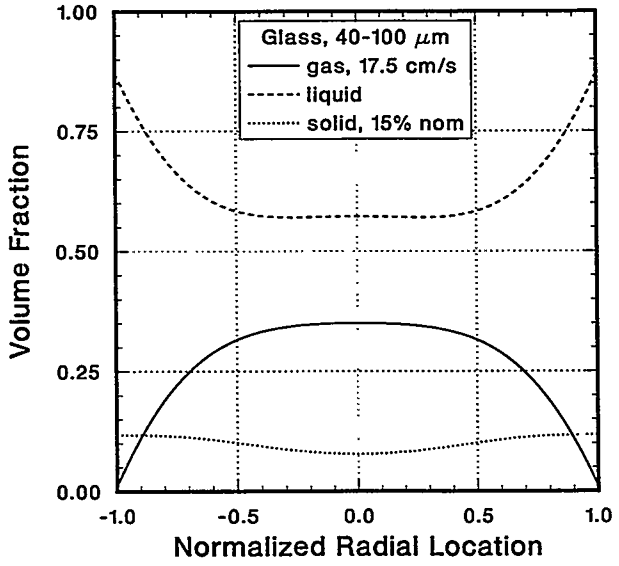


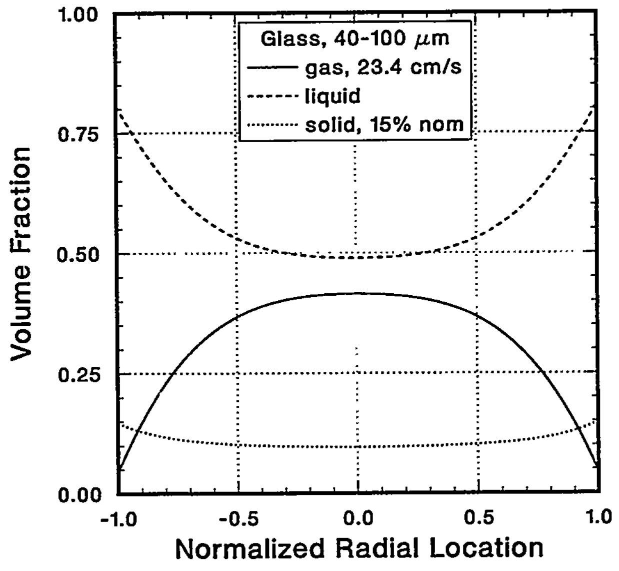
Glass: 40-100 μm, 10% by volume



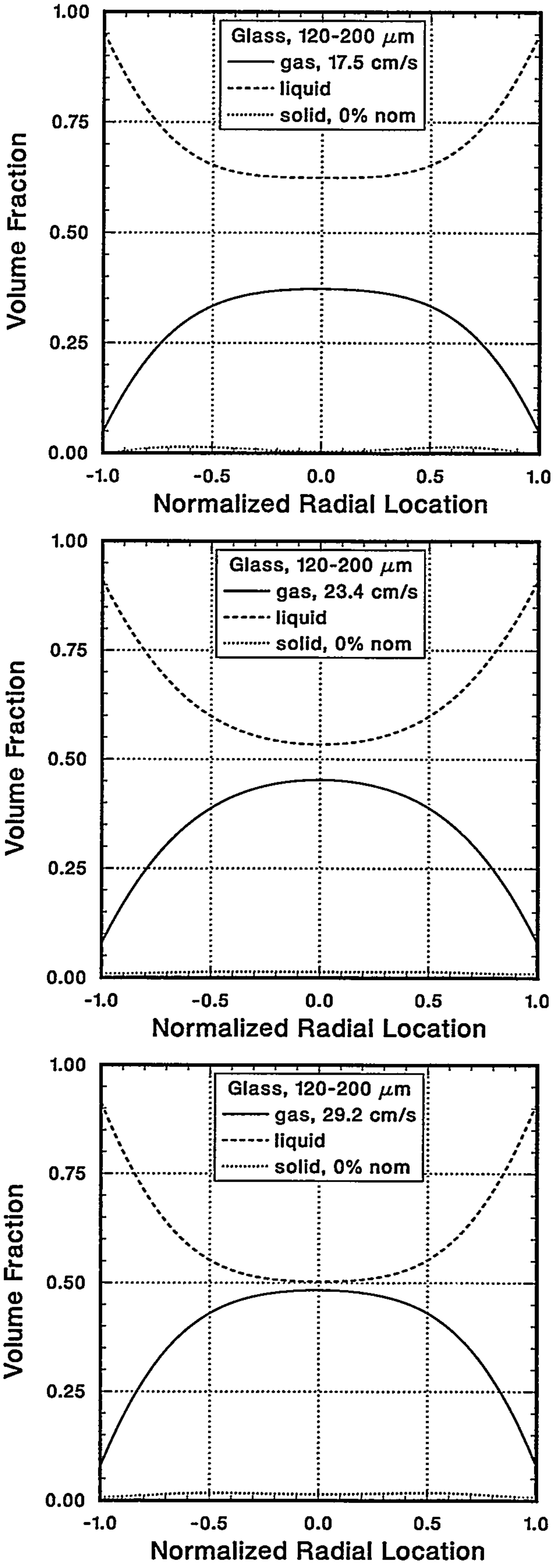
Glass: 40-100 µm, 15% by volume



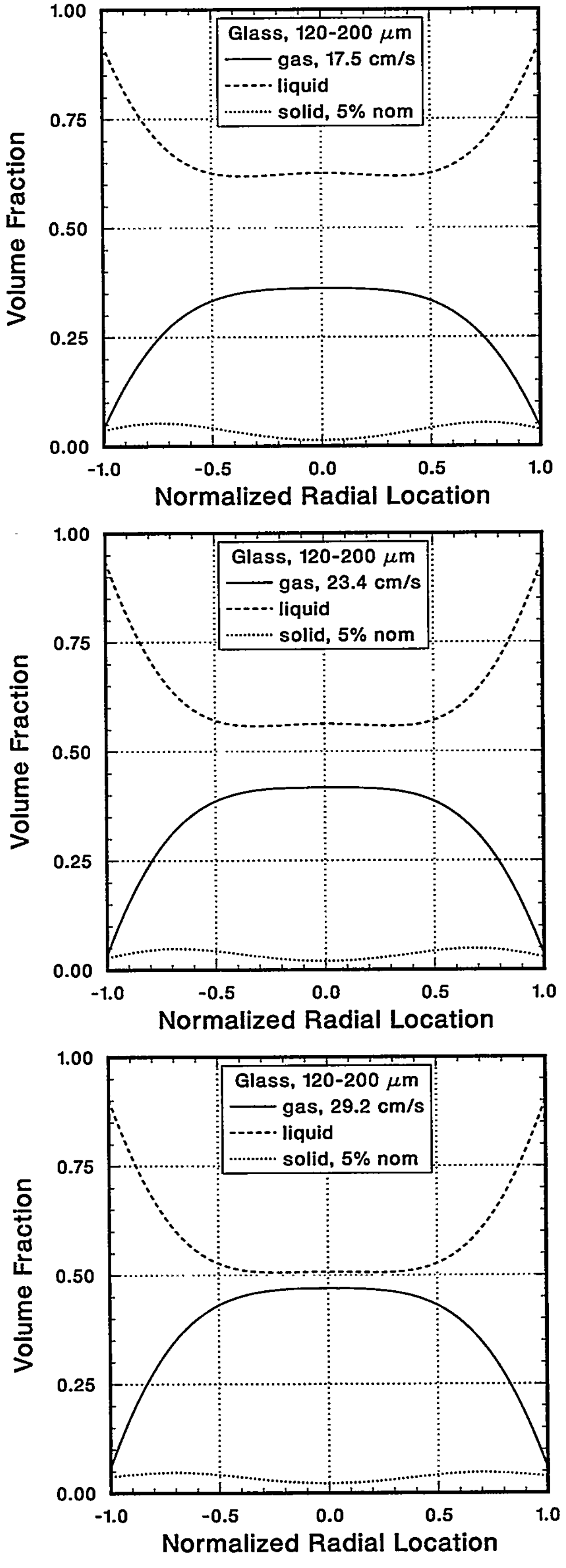




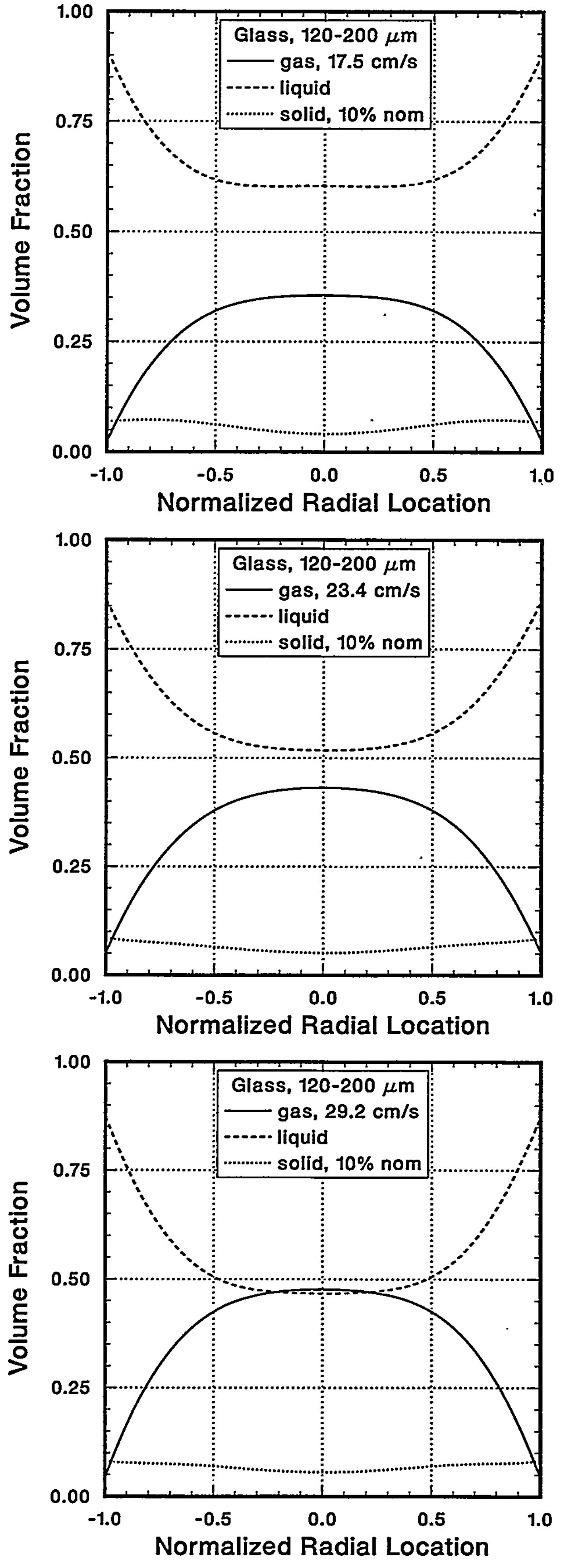
Glass: 120-200 μm, 0% by volume



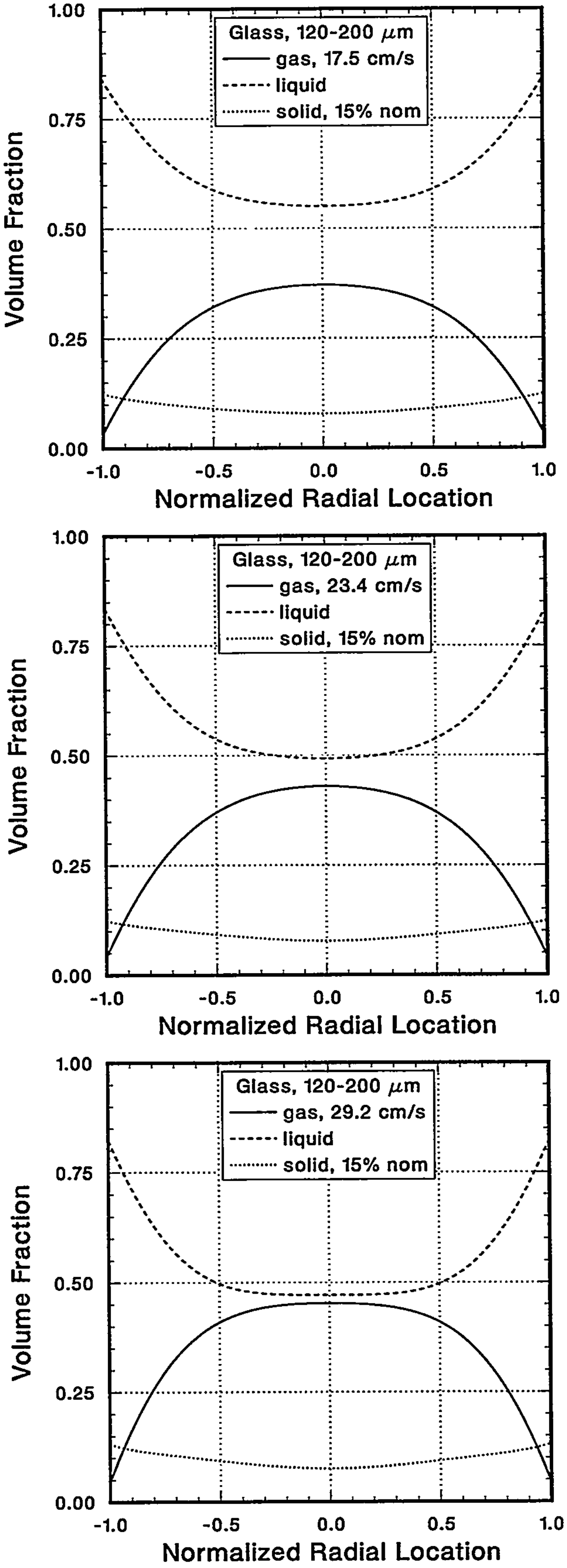
Glass: 120-200 μm, 5% by volume



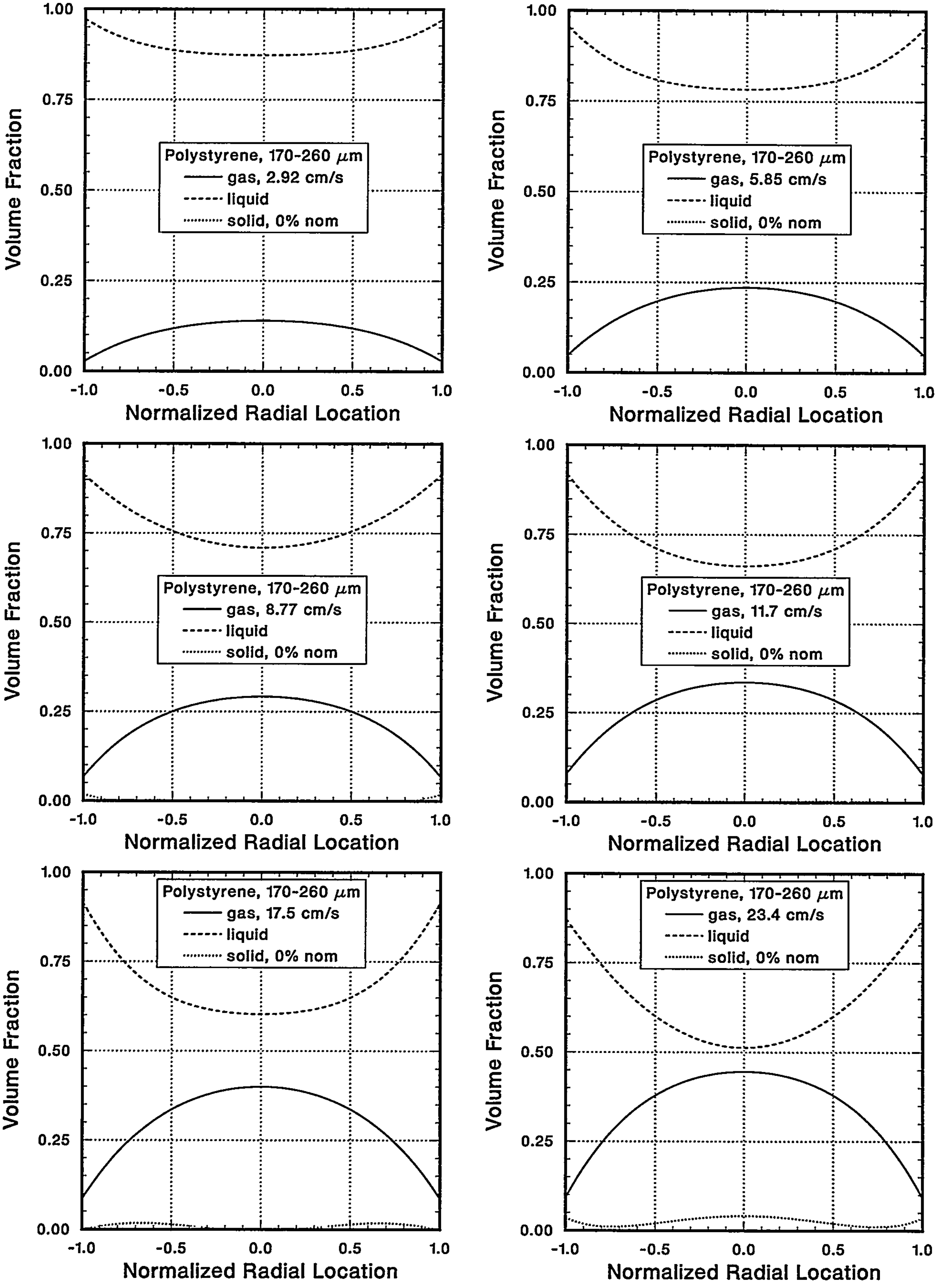
Glass: 120-200 μm, 10% by volume



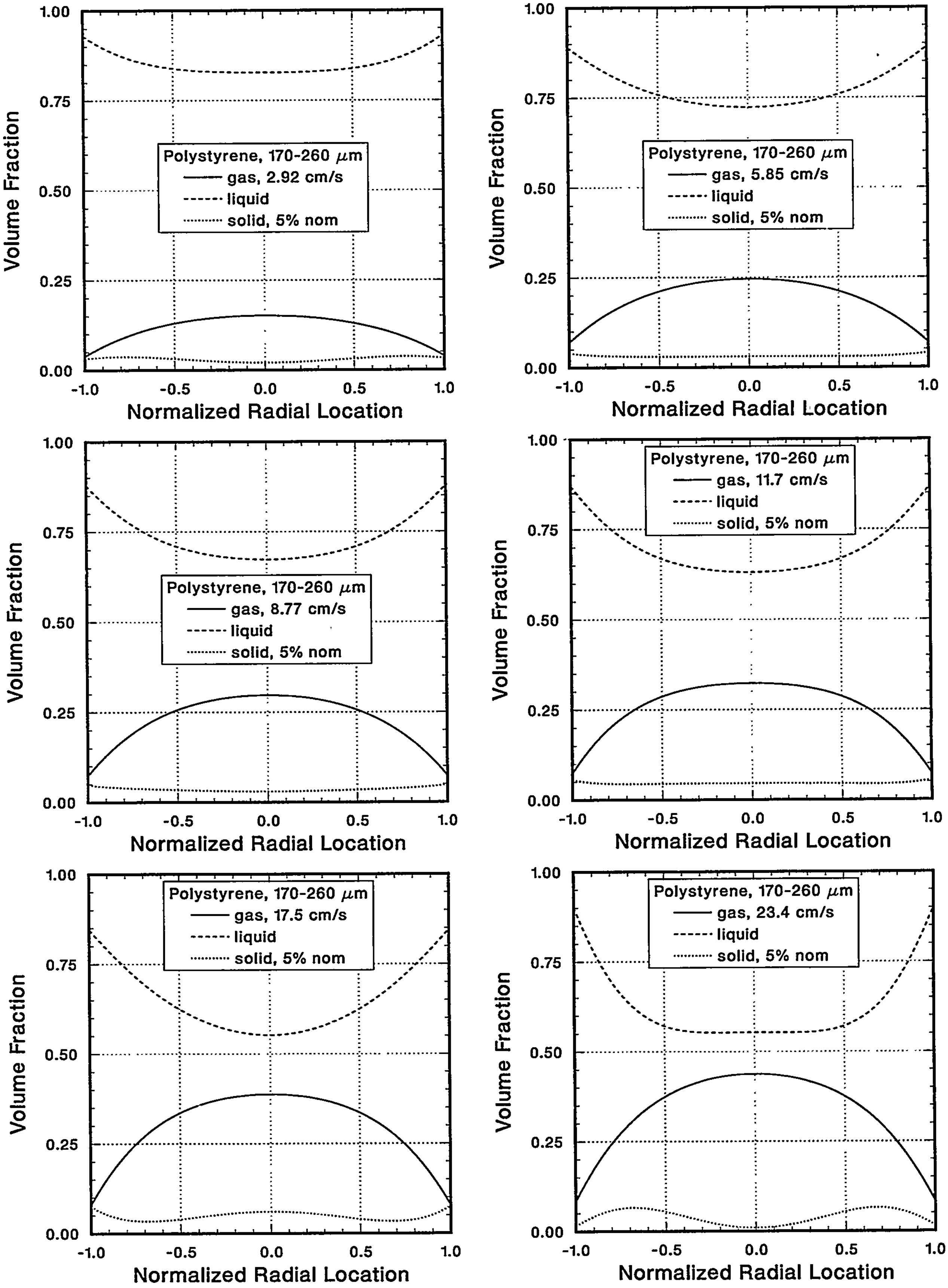
Glass: 120-200 μm, 15% by volume



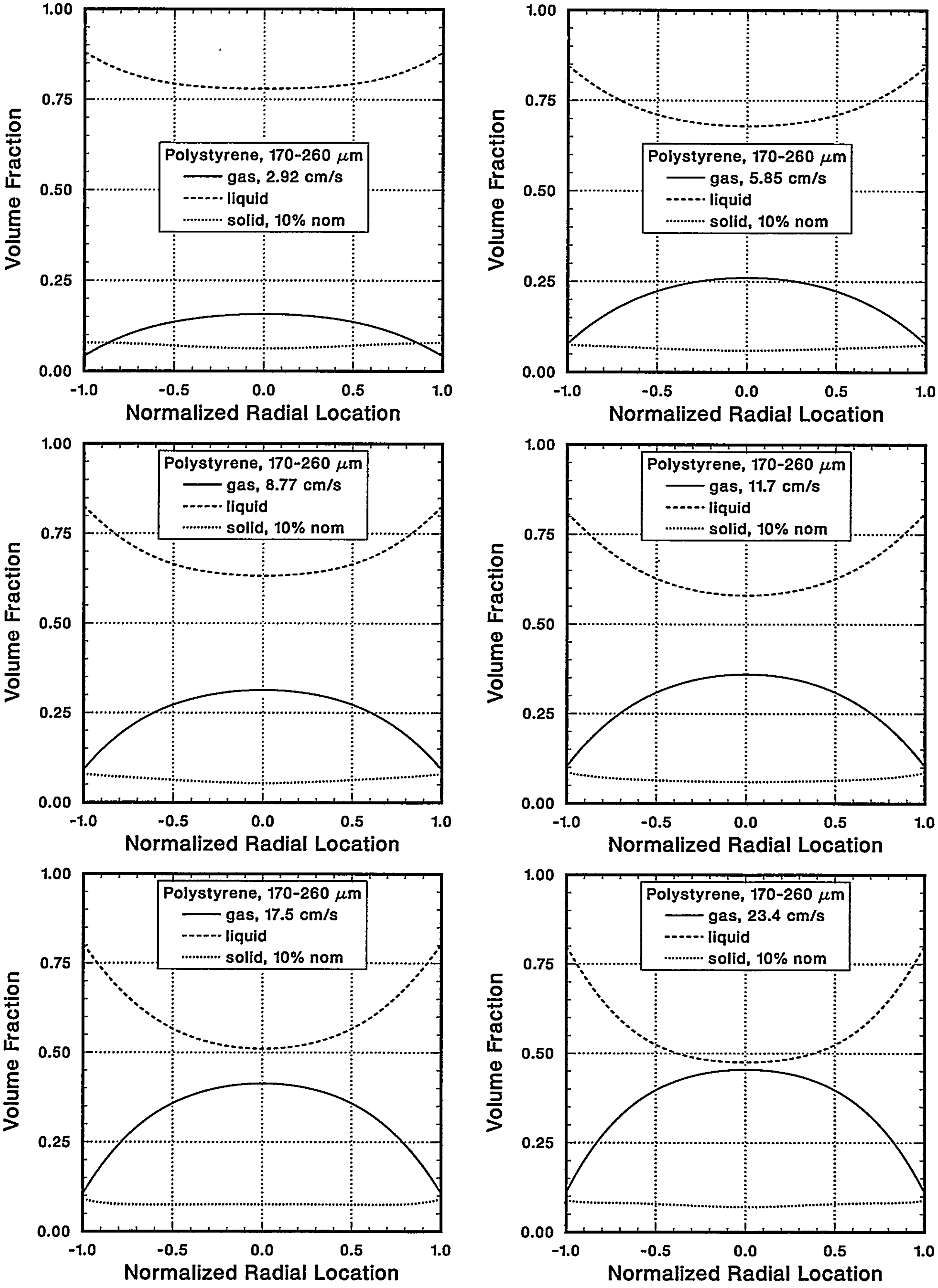
Polystyrene: 170-260 μm, 0% by volume



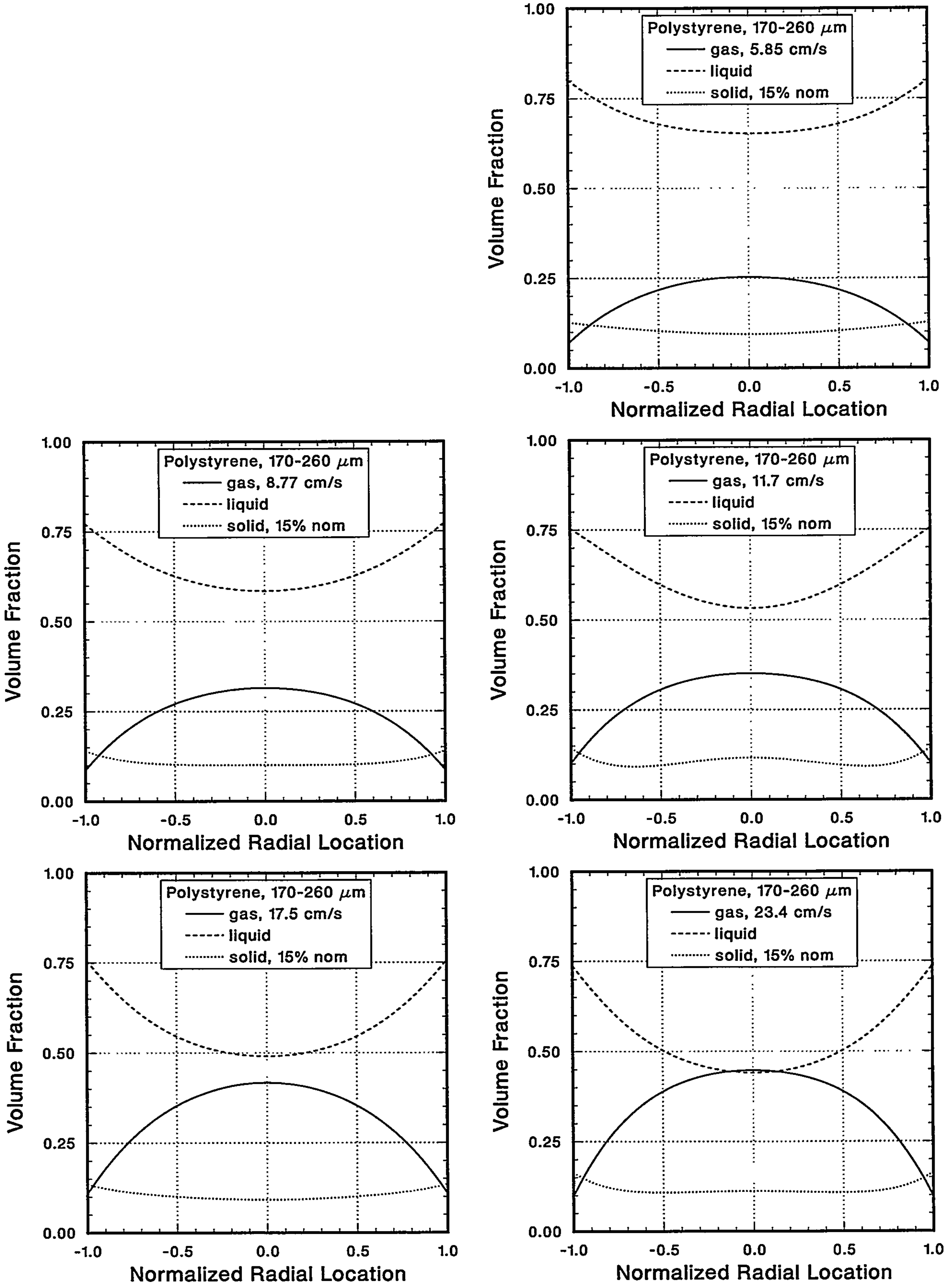
Polystyrene: 170-260 µm, 5% by volume



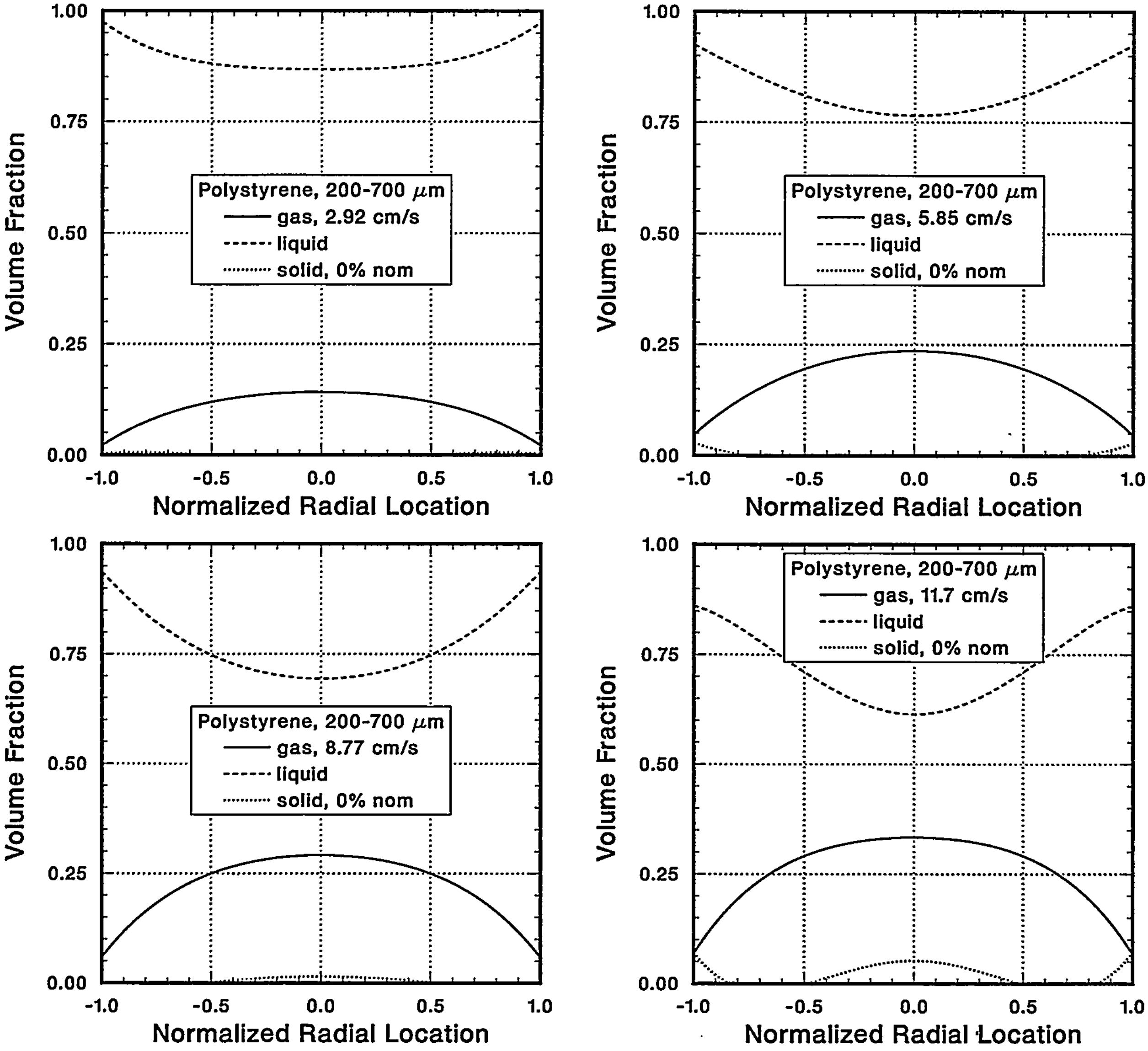
Polystyrene: 170-260 μm, 10% by volume



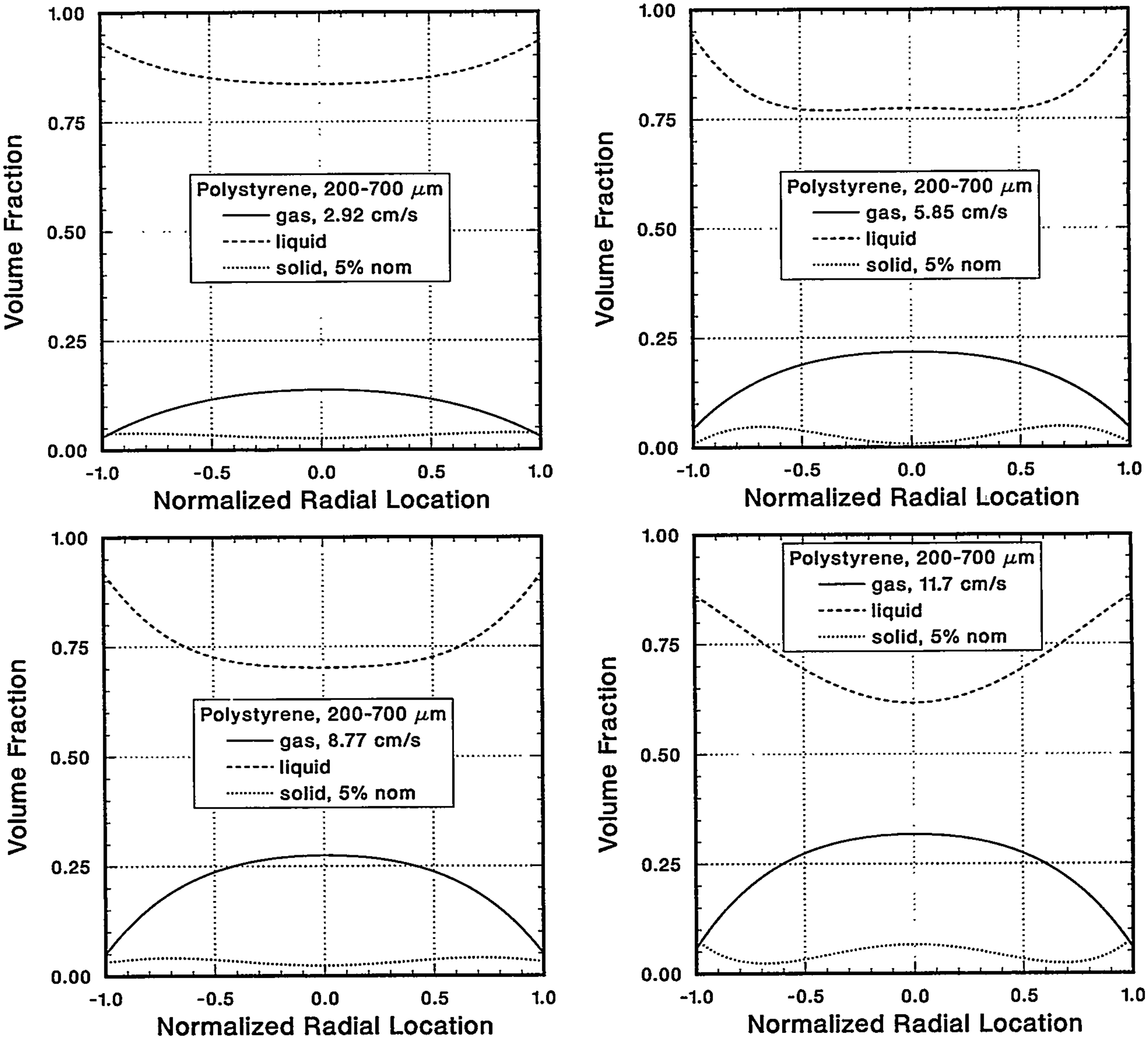
Polystyrene: 170-260 μm, 15% by volume



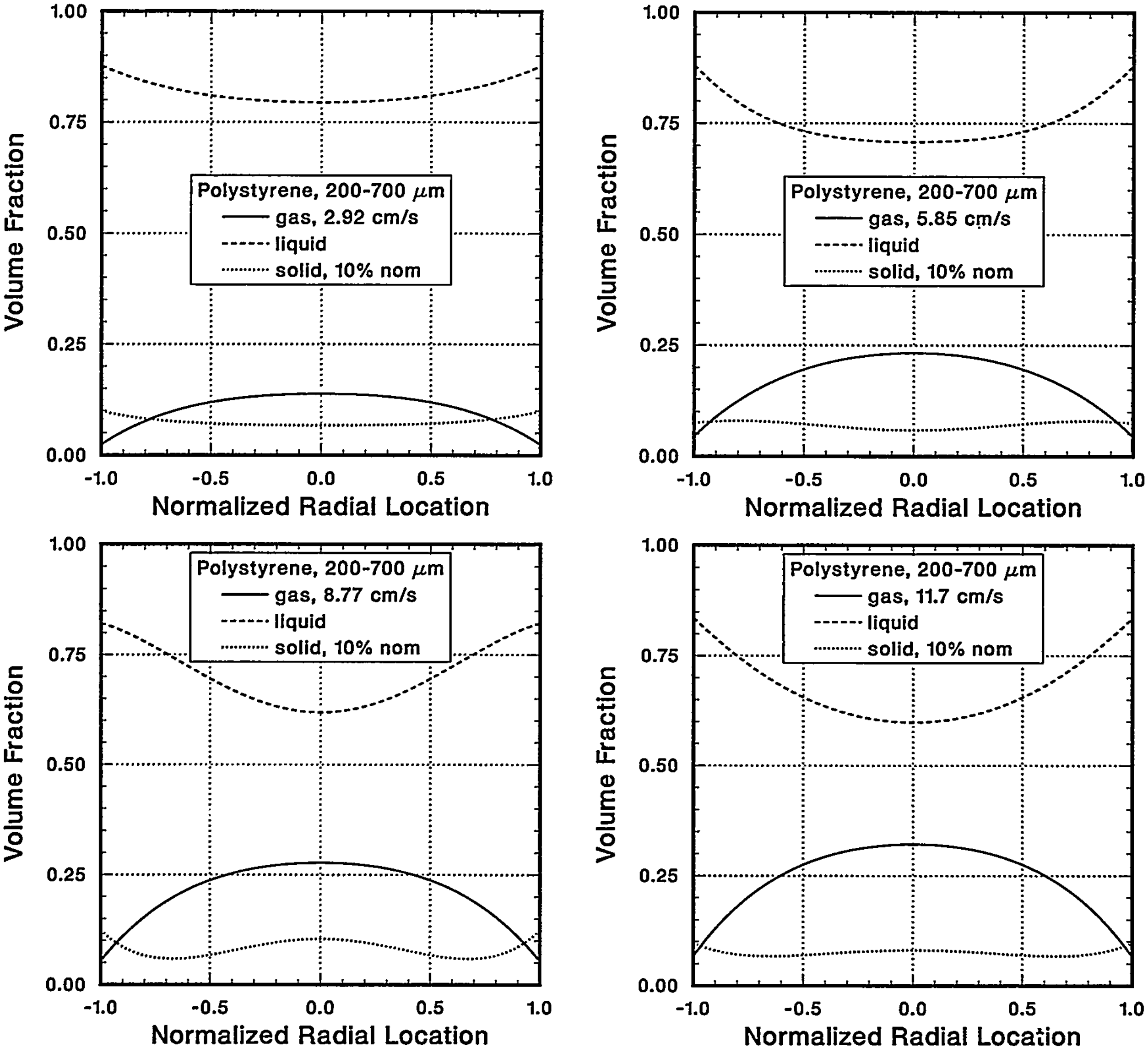
Polystyrene: 200-700 µm, 0% by volume



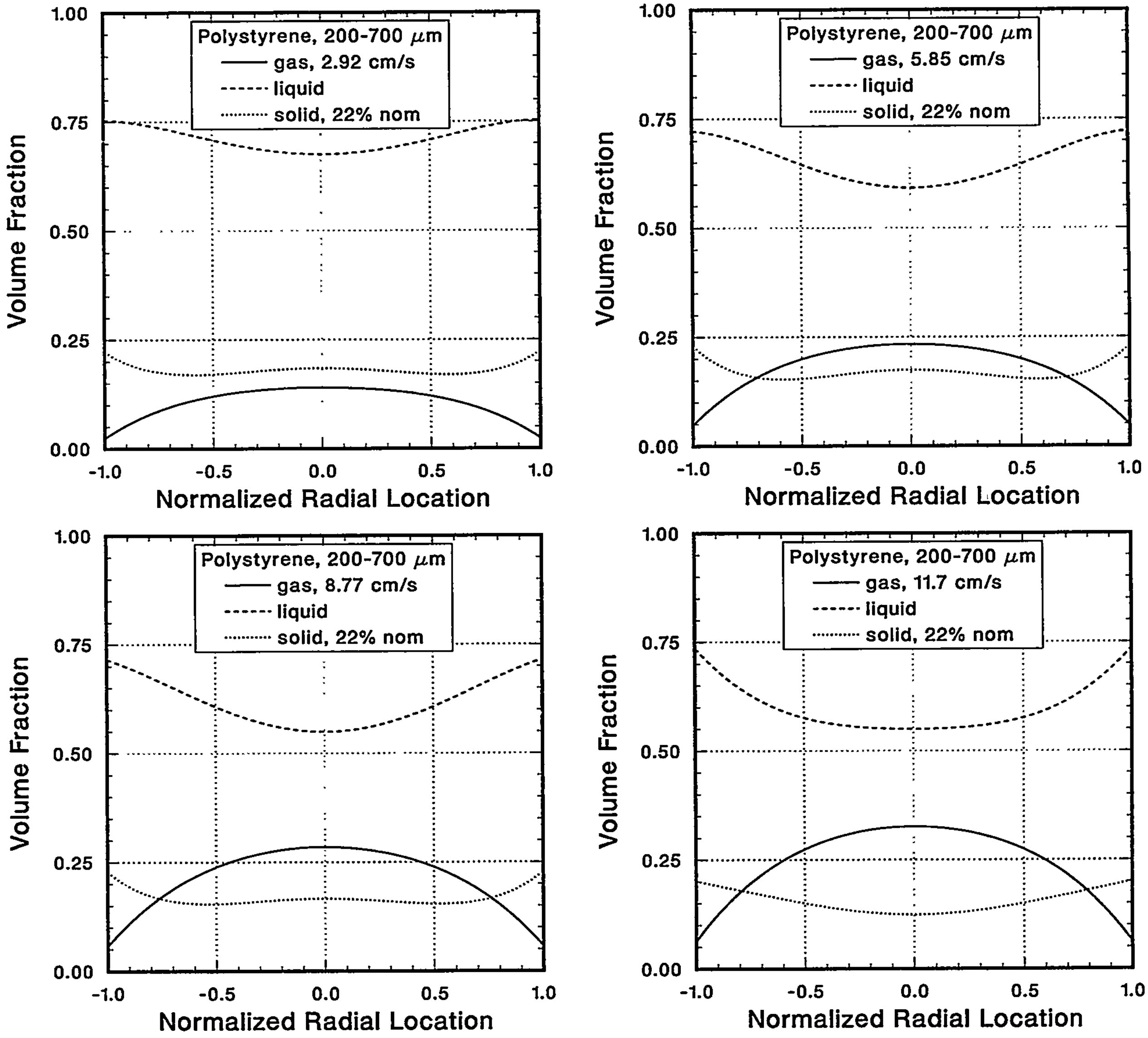
Polystyrene: 200-700 µm, 5% by volume



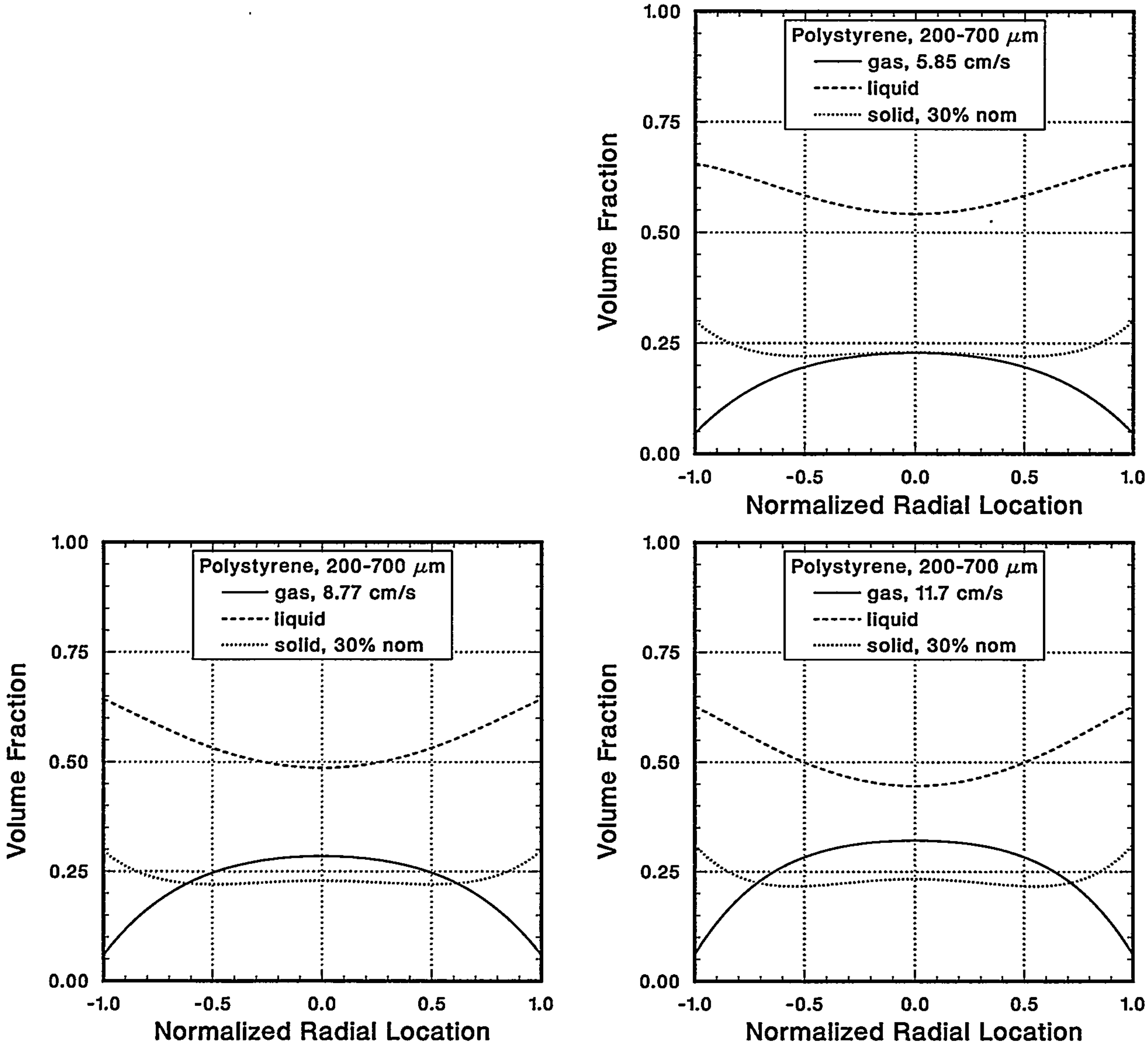
Polystyrene: 200-700 µm, 10% by volume



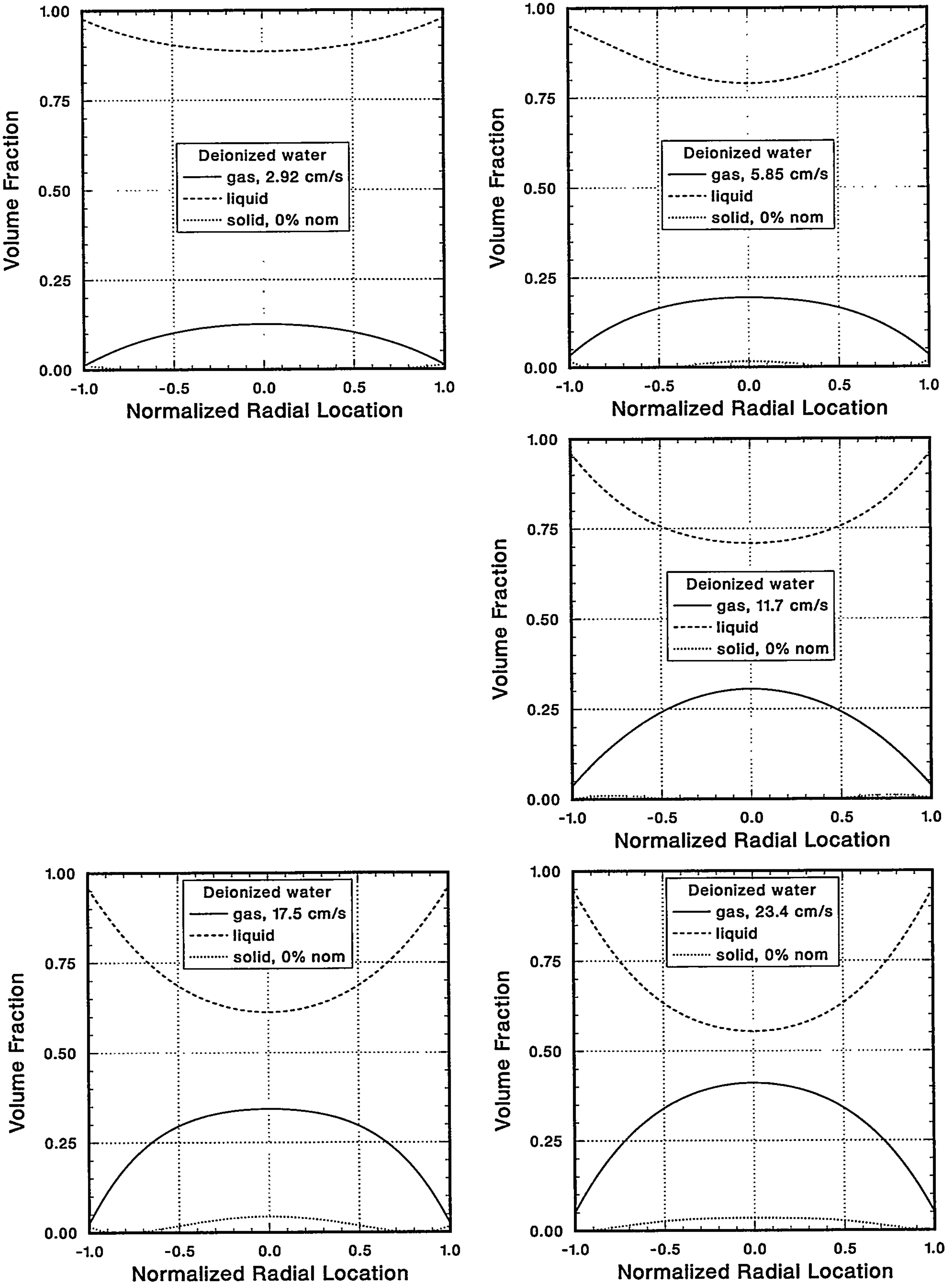
Polystyrene: 200-700 μm, 22% by volume



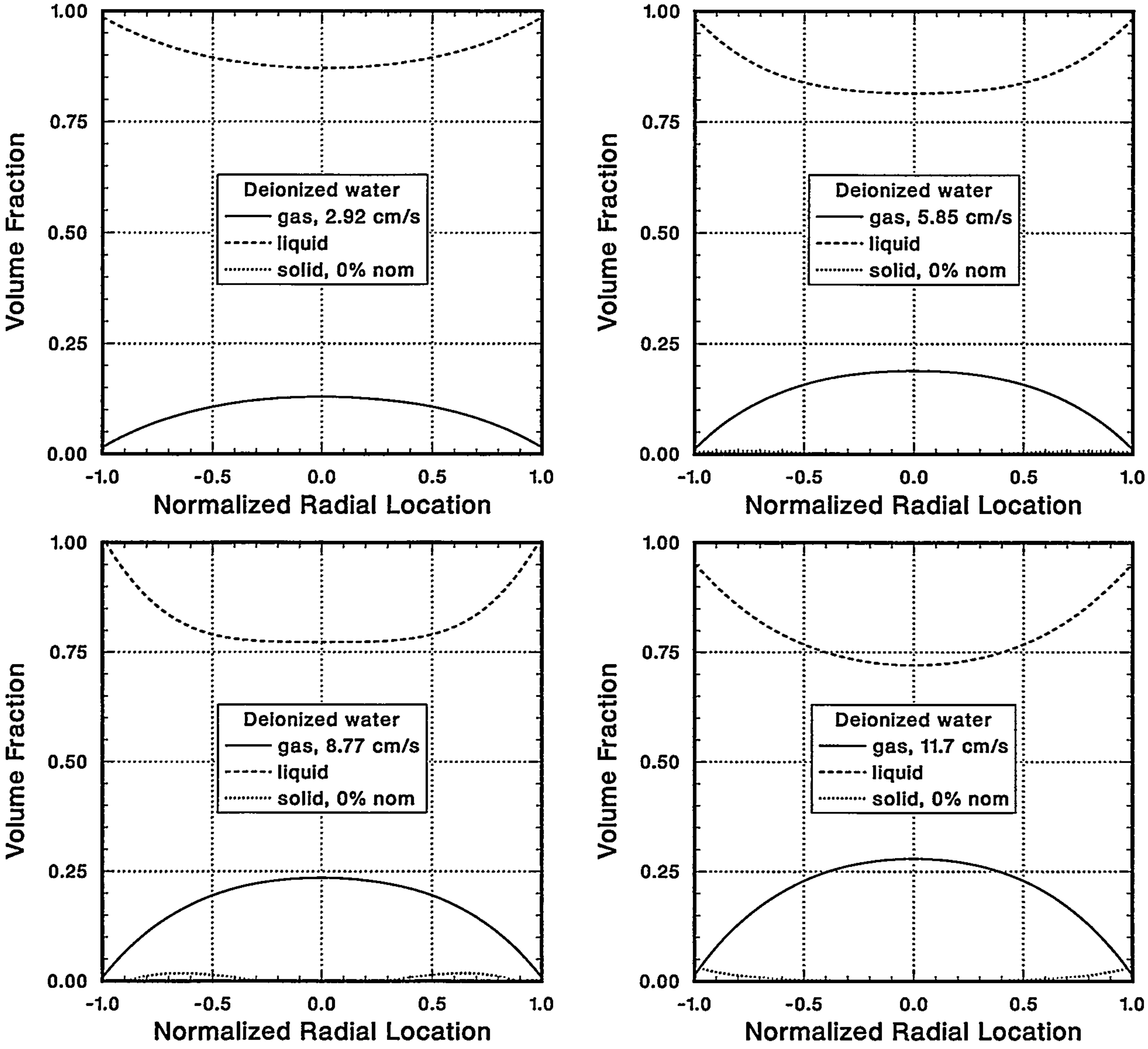
Polystyrene: 200-700 µm, 30% by volume



Deionized water



Deionized water



Distribution

1 Prof. Muthanna H. Al-Dahhan
Washington University
Department of Chemical Engineering
Campus Box 1198
One Brookings Drive
St. Louis, MO 63130-4899

1 Dr. Sayavur I. Bakhtiyarov
Space Power Institute
231 Leach Center
Auburn University
Auburn, AL 36849-5320

1 Dr. Bharat L. Bhatt
Air Products and Chemicals, Inc.
7201 Hamilton Boulevard
Allentown, PA 18195-1501

3 Prof. Steven L. Ceccio
University of Michigan
MEAM
2011 Lay Auto Lab
Ann Arbor, MI 48109-2121

1 Dr. Ray A. Cocco
The Dow Chemical Company
1776 Building
Midland, MI 48674-1776

1 Mr. Daniel J. Driscoll
U.S. Department of Energy
National Energy Technology Laboratory
P.O. Box 880
3610 Collins Ferry Road
Morgantown, WV 26507-0880

1 Prof. Milorad P. Dudukovic
Washington University
Department of Chemical Engineering
Campus Box 1198
One Brookings Drive
St. Louis, MO 63130-4899

1 Prof. Liang-Shih Fan
Ohio State University
Department of Chemical Engineering
140 West 19th Avenue
Columbus, OH 43210-1180

1 Prof. Rodney O. Fox
Department of Chemical Engineering
Iowa State University
3157 Sweeney Hall
Ames, IA 50011-2230

1 Dr. B. A. Kashiwa
Los Alamos National Laboratory
Theoretical Division
Fluid Dynamics Group
MS B216
Los Alamos, NM 87545

1 Mr. John Shen
U.S. Department of Energy
MS FE-24 GTN
Germantown, MD 20874-1290

1 Prof. Sankaran Sundaresan
Princeton University
Chemical Engineering Dept.
A315 Engineering Quadrangle
Princeton, NJ 08544

1 Dr. Tyler Thompson
The Dow Chemical Company
1776 Building
Midland, MI 48674-1776

1 Mr. Bradley J. Tomer
U.S. Department of Energy
National Energy Technology Laboratory
P.O. Box 880
3610 Collins Ferry Road
Morgantown, WV 26507-0880

1 Dr. Bernard A. Toseland
Air Products and Chemicals, Inc.
7201 Hamilton Boulevard
Allentown, PA 18195-1501

1 Dr. W. Brian VanderHeyden
Los Alamos National Laboratory
Theoretical Division
Fluid Dynamics Group
MS B216
Los Alamos, NM 87545

1	MS 0139	D. L.	Crawford, 9100 (acting)
1	0511	J. C.	Cummings, 1010
1	0710	A. P.	Sylwester, 6245
1	0724	R. J.	Eagan, 6000
1	0826	W. L.	Hermina, 9113
3	0826	J. R.	Torczynski, 9113
1	0828	T. C.	Bickel, 9101
3	0834	A. C.	Ratzel, 9112
3	0834	D. L.	George, 9112
1	0834	W. C.	Ginn, 9112
1	0834	T. W.	Grasser, 9112
3	0834	T. J.	O'Hern, 9112
3	0834	K. A.	Shollenberger, 9112
1	0834	S. M.	Trujillo, 9112
1	0834	J. E.	Johannes, 9114
5	1349	N. B.	Jackson, 1841
1	9018		Central Technical Files, 8940-2
2	0899		Technical Library, 4916
1	0612		Review and Approval Desk, 4912, for DOE/OSTI