97980①

SANDIA REPORT

M97051307

SAND97-8202 • UC-1409
Unlimited Release
Printed October 1996

CURRENT - A Computer Code for Modeling Two-Dimensional, Chemically Reacting, Low Mach Number Flows

W. S. Winters, G. H. Evans, C. D. Moen

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94551
for the United States Department of Energy
under Contract DE-AC04-94AL85000

Approved for public release; distribution is unlimited.

This report has been reproduced 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 (615) 576-8401, FTS 626-8401.

Available to the public from:

National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.
Springfield, VA 22161

SAND97-8202
Unlimited Release
Printed October 1996

CURRENT - A Computer Code for Modeling Two-Dimensional, Chemically Reacting, Low Mach Number Flows

W. S. Winters, G. H. Evans, and C. D. Moen
Thermal and Plasma Processes Department
Sandia National Laboratories/California

Abstract

This report documents CURRENT, a computer code for modeling two-dimensional, chemically reacting, low Mach number flows including the effects of surface chemistry. CURRENT is a finite volume code based on the SIMPLER algorithm. Additional convergence acceleration for low Peclet number flows is provided using improved boundary condition coupling and preconditioned gradient methods. Gas-phase and surface chemistry is modeled using the CHEMKIN software libraries. The CURRENT user-interface has been designed to be compatible with the Sandia-developed mesh generator and post processor ANTIPASTO and the post processor TECPLOT. This report describes the theory behind the code and also serves as a user's manual.

DISCLAIMER

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

Contents

List of Figures

1 Nomenclature

D_{km} = multicomponent diffusion coefficient defined in Section 3 by (8);
mixture average diffusion coefficient defined in Section 3 by (9)

$\mathcal{D}_{ik}$ = binary diffusion coefficient

D_k^T = multicomponent thermal diffusion coefficient

M = molecular weight of mixture

M_k = molecular weight of species k

P = pressure

R = universal gas constant

T = temperature

U = contravariant component of velocity normal to surface of constant ξ

V = contravariant component of velocity normal to surface of constant η

X_k = mole fraction of species k

Y_k = mass fraction of species k

c_p = constant pressure specific heat per unit mass of the mixture

c_{p_k} = constant pressure specific heat per unit mass of species k

f = circumferential component of velocity divided by r ($f \equiv w/r$)

$\sqrt{g}$ = jacobian of coordinate transformation defined in Section 3 by (31)

$\vec{g}$ = gravitational vector

h = enthalpy per unit mass of the mixture

h_k = enthalpy per unit mass of species k

$\vec{j}_k^c$ = diffusion mass flux vector of species k due to concentration gradients

$\vec{j}_k^T$ = diffusion mass flux vector of species k due to temperature gradients

k = mixture thermal conductivity

$\dot{\vec{m}}_k''$ = mass flux of species k

$\dot{\vec{m}}_s''$ = mixture mass flux at surface

$\vec{n}$ = outward normal vector from surface

$\vec{n}_{\text{cv}}$ = outward normal vector from control volume surface

p_m = motion pressure (in momentum equation)

$\vec{q}$ = diffusive energy flux vector

$\vec{q}_{\text{cnd}}$ = conduction energy flux vector defined in Section 3 by (7)

$\vec{q}_{\text{intdif}}$ = species interdiffusion energy flux vector defined in Section 3 by (7)

r = radial coordinate

r_k = net mass production rate of species k due to homogeneous chemical reaction

t = time

u = x component of velocity

v = r (or y) component of velocity

$\vec{v}$ = mass average velocity of mixture

$\vec{v}_k$ = velocity of species k

w = θ component of velocity

x = axial coordinate

y = lateral coordinate

Greek symbols

ρ = mixture density

μ = mixture viscosity

$\nu_{k\alpha}$ = net stoichiometric coefficient for species k in surface reaction α defined in Section 3 by (36)

Φ = dependent variable

ξ = curvilinear coordinate

η = curvilinear coordinate

θ = circumferential coordinate

$\vec{\vec{\tau}}$ = stress tensor

2 Introduction

Sandia has a long history in modeling steady and unsteady, multidimensional, low Mach number fluid flow and heat transfer, see *e.g.* reference [1]. This modeling has been expanded to include multicomponent gas-phase and surface chemistry with an eye toward analyzing flows in thermal-chemical reactors [2]-[3]. Such reactors are widely used by the microelectronics industry for chemical vapor deposition (CVD), etching, cleaning and other processes. Sandia's more recent experience in modeling thermal-chemical processes has led to the development of the computer code CURRENT which is used to model two-dimensional, transient and steady-state, low Mach number flow and heat transfer, including multi-species transport, gas-phase chemistry, and surface chemistry using the CHEMKIN software libraries.

The computational "engine" for CURRENT is based on the well-known SIMPLER algorithm [4]. This algorithm provides a foundation for solving the low Mach number fluid flow conservation equations using a control volume formulation (finite volume technique) on a single logically rectangular grid or domain. In CURRENT, this method is extended to include multiple logically rectangular, but not necessarily orthogonal, domains making it possible to model flows in any two-dimensional planar or axisymmetric geometry. A more detailed discussion of this modeling technique is presented in Section 3.

While the SIMPLER algorithm has proven to be a reliable solution method, it sometimes leads to unacceptably slow convergence rates, particularly for low Peclet number problems frequently encountered in low pressure CVD modeling. Convergence rates can also be slowed due to complex boundary conditions (*e.g.* strong diffusion at in-flow boundaries, solid boundaries with surface chemistry, etc.). In order to overcome these difficulties, CURRENT utilizes the General Minimal Residual algorithm (GMRES) in conjuction with improved boundary condition coupling to accelerate the convergence of SIMPLER iterations. This acceleration method and its implementation into CURRENT are discussed in Section 4.

Section 5 contains a detailed discussion on how to use CURRENT. Included is an overview of the CURRENT "modeling philosophy." An example problem, designed to demonstrate most of the code's documented capabilities, is presented and the reader is taken through the step-by-step procedure of pre-processing (mesh generation), code execution, and post processing. CURRENT's user interface has been designed to work with the Sandia-developed finite element pre- and post-processor ANTIPASTO [5]. A brief discussion of how CURRENT interfaces with ANTIPASTO (AP) is also presented. A complete list of CURRENT input commands and their usage can be found in Appendix A.

The source code for CURRENT, CHEMKIN, SURFACE CHEMKIN and a set of makefiles and run scripts is available in the form of a single "tar" file. CURRENT will run on most UNIX workstations and mainframes. Section 6 contains a description of how to set up a computing environment from the tar file and size the code for a particular problem. Proper code installation may be verified by running the above mentioned "example problem." Input files for the example problem are distributed as part of the CURRENT tar file.

As of this writing, all thermodynamic properties, transport properties, and gas-phase and surface chemical kinetics information required by CURRENT are obtained from subroutine calls to CHEMKIN [6], SURFACE CHEMKIN [7], and the CHEMKIN gas phase transport software package [8]. Since the CHEMKIN family of codes is well documented, this report will not attempt to provide detailed discussions on the theory or use of CHEMKIN, SURFACE CHEMKIN or the gas phase transport package.

3 Theoretical Background

This section describes the mathematical model (equations and boundary conditions) that is solved in CURRENT. The numerical implementation and solution strategy are also discussed.

3.1 General Code Description

The CURRENT computer code solves the two-dimensional, transient or steady, low Mach number, variable property Navier-Stokes, energy, and species conservation equations including rotating surfaces in general non-orthogonal curvilinear coordinates. The base geometry may be either planar or cylindrical. Three components of momentum are solved for axisymmetric problems with rotation and two components of momentum for planar problems. Complex geometry is handled using multiple non-overlapping mesh regions (cf. Figures 3, 16). The code is coupled to the CHEMKIN [6] software libraries, providing generality for treating chemically reacting mixtures of gases including multicomponent diffusion and thermal diffusion [8]. Surface chemistry is coupled into the code via SURFACE CHEMKIN [7].

The numerical method consists of a control volume formulation [4] with staggered control volumes (cf. Figure 2) used for the discretization of the conservation equations of momentum, mass, energy, and chemical species. The differential equations are first transformed from either a cylindrical or a planar coordinate system to a general non-orthogonal curvilinear coordinate system [9],[10] (cf. Figure 1); the equations are then integrated over the control volumes. The remaining spatial derivatives are discretized with either first or second-order schemes (user selectable). The time derivatives are discretized with a first-order backward Euler difference.

The solution of the coupled nonlinear equations is iterative and requires a modest amount of memory; it consists of a reliable implicit line relaxation method involving multiple sweeps across mesh lines, mesh directions, and mesh blocks, solving for each variable in a sequential fashion. For some problems, convergence is accelerated by using this solution scheme as a preconditioner to a gradient algorithm. This acceleration technique is discussed in Section 4. Additional features of CURRENT, which will not be discussed further since they are not yet supported in the user interface, include (1) source terms for mass, momentum, and energy (allowing CVD problems with makeup injection of reactants and two phase flow problems such as fiber drying and gas/particle flows to be solved), (2) surface energy balances (enabling the solution of the conjugate problem of chemically reacting gas flow coupled with conduction in solid materials), and (3) user specified material types (i.e., other than CHEMKIN) and properties. Applications of CURRENT have been in chemical vapor deposition (CVD) transport processes ([2], [3],[11]), mixed convection flows and buoyant instabilities ([1],[12]), fiber drying [13] and fiber CVD [14], and two phase gas-particle flows [15].

3.2 Transport Equations

The governing equations (in vector form) for the transport of a variable property, low Mach number mixture of chemical species are:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0 \tag{1}$$

$$\frac{\partial (\rho \vec{v})}{\partial t} + \nabla \cdot \left(\rho \vec{v} \vec{v} + \vec{\vec{\tau}} \right) = -\nabla p_m + \rho \vec{g} \tag{2}$$

$$\frac{\partial \left(\rho Y_k\right)}{\partial t} + \nabla \cdot \left(\rho \vec{v} Y_k + \vec{j}_k^{\,c} + \vec{j}_k^{\,T} \right) = r_k \ , \quad k = 1, ..., K_g - 1 \tag{3}$$

$$\frac{\partial \left(\rho h\right)}{\partial t} + \nabla \cdot \left(\rho \vec{v} h + \vec{q} \right) = 0 \tag{4}$$

In the above equations, gravity is assumed to be the only external force acting on the chemical species. In the species conservation equations (3), r_k is the net mass production rate per unit volume of species k by homogeneous chemical reactions; $\vec{j}_k^{\,c}$ is the diffusive mass flux of species k caused by concentration gradients (Fickian diffusion):

$$\vec{j}_k^{\,c} = -\frac{\rho D_{km}}{M} \nabla (Y_k M) + D_{km} Y_k M \sum_{\substack{i=1 \\ i \neq k}}^{K_g} \frac{\vec{j}_i^{\,c}}{M_i \mathcal{D}_{ik}}, \tag{5}$$

$\vec{j}_k^{\,T}$ is the diffusive mass flux of species k caused by temperature gradients (thermal diffusion):

$$\vec{j}_k^{\,T} = -D_k^T \nabla (\ln T). \tag{6}$$

Mass transport by pressure diffusion is neglected. The energy equation (4) is given in terms of enthalpy where $h = \sum_{k=1}^{K_g} h_k Y_k$. Viscous dissipation and the diffusion-thermo energy flux (the Dufour effect) are neglected; the diffusive energy flux vector, $\vec{q}$, is given by:

$$\vec{q} = \vec{q}_{\text{cnd}} + \ \vec{q}_{\text{intdif}} = -k \nabla T + \sum_{k=1}^{K_g} h_k \left(\vec{j}_k^{\,c} + \vec{j}_k^{\,T} \right). \tag{7}$$

The summation term in the above equation is the energy flux by species interdiffusion in a multicomponent mixture. The multicomponent diffusion coefficients in (5) are given by:

$$D_{km} = \left[\sum_{\substack{i=1 \\ i \neq k}}^{K_g} X_i / \mathcal{D}_{ik} \right]^{-1}, \tag{8}$$

where $\mathcal{D}_{ik}$ are the binary diffusion coefficients.

If the mixture averaged approximation to diffusion is made (an option in the code), then

$$D_{km} = \frac{1 - Y_k}{\sum_{\substack{i=1 \\ i \neq k}}^{K_g} X_i / \mathcal{D}_{ik}} \tag{9}$$

and the summation term in (5) is dropped. The thermal diffusion coefficients D_k^T in (6), the binary diffusion coefficients $\mathcal{D}_{ik}$ in (8), and the viscosities and thermal conductivities of the individual gas species k are obtained from kinetic theory (see [8] and [16]) using the Lennard-Jones (6-12) and Stockmayer potentials.

The energy equation in the form of Equation (4) does not explicitly contain the energy release rates due to homogeneous reactions and may be convenient to use for some problems (phase change); however, since the problems typically solved with CURRENT involve the specification of boundary conditions in terms of temperature, the energy equation is rewritten as:

$$\frac{\partial\ (\rho T)}{\partial t} + \nabla \cdot (\rho \vec{v} T) = -\frac{1}{c_p} \nabla \cdot \vec{q}_{\text{cnd}} - \frac{1}{c_p} \sum_{k=1}^{K_g} h_k \left[r_k - \nabla \cdot \left(\vec{j}_k^c + \vec{j}_k^T \right) \right]. \tag{10}$$

The density ρ of the mixture is determined from the ideal gas equation of state:

$$\rho = \frac{PM}{RT}. \tag{11}$$

The viscosity μ and thermal conductivity k of the mixture are determined from mixture averaging rules (see [8]). The specific heat of the mixture c_p is determined from

$$c_p = \sum_{k=1}^{K_g} c_{p_k} Y_k \tag{12}$$

with the specific heats of the individual species c_{p_k} determined from polynomial fits over specified temperature ranges in CHEMKIN. The stress tensor, $\vec{\vec{\tau}}$, in Equation (2) is for a Newtonian fluid with the bulk viscosity set equal to zero [17]. Note that the pressure gradient, ∇p_m, in Equation (2) involves the variable p_m which is different from the pressure, P, in the equation of state (11); this approximation to the equations of motion in which P is constant or at most a function of time only and p_m varies spatially to satisfy the continuity equation (1) everywhere has been shown [18] to be valid for low Mach number flow.

Dimensional equations have been shown in this section to simplify the presentation. However, the equations are implemented in CURRENT in a dimensionless form [2], allowing the user to solve either a dimensional or a dimensionless problem. Since solving a dimensionless problem requires knowledge of the scales used in the nondimensionalization (which are not presented here), the default of solving dimensional equations is assumed throughout this document. For more information on this topic, contact the second author.

3.3 General Curvilinear Coordinate Transformation

Complex geometry can be an important consideration in transport processes. In the case of chemical vapor deposition (CVD), gases flow from small diameter piping into larger reactor chambers (usually through

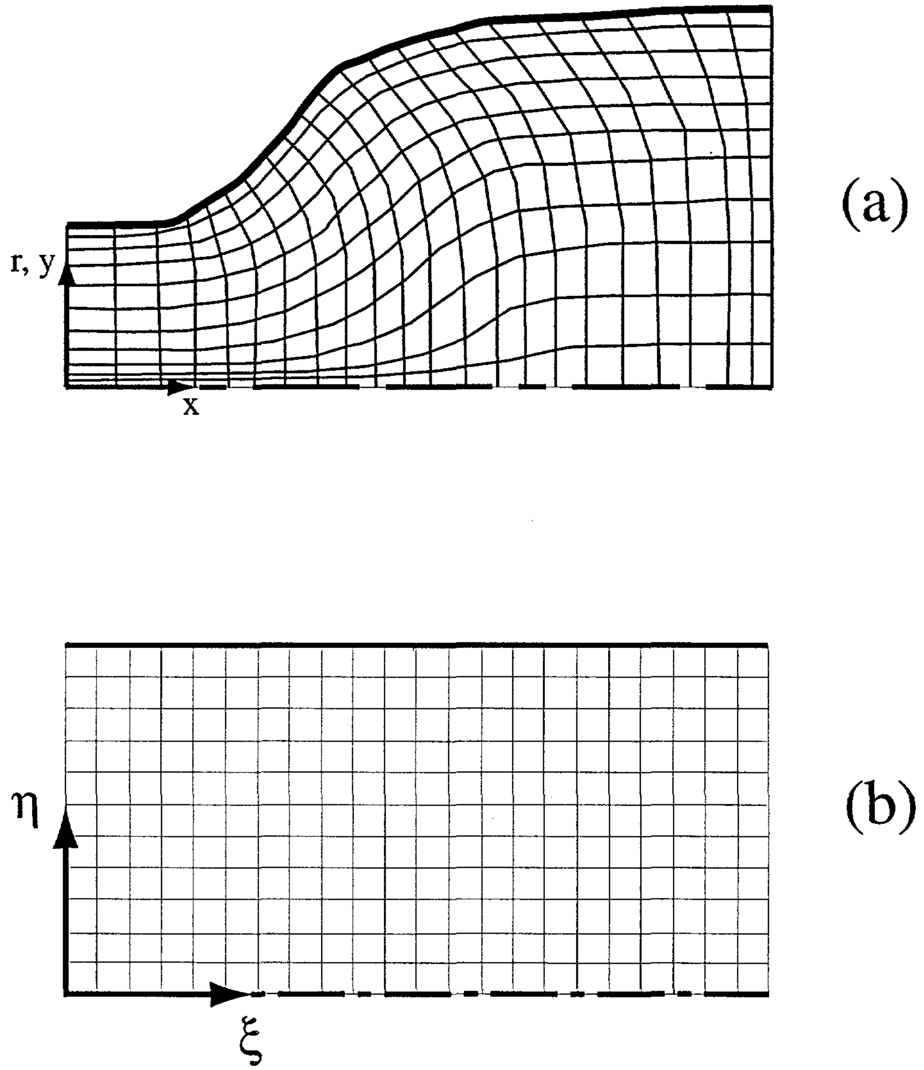


Figure 1: General curvilinear coordinate transformation: (a) physical and (b) computational planes.

various types of manifolds); it is desirable to avoid recirculation regions in the reactor. Reactor side walls are sometimes angled to allow transitions from one flow region to another. It is sometimes important to accelerate the flow as it passes over the deposition zone; this is accomplished by tilting the susceptor or angling the opposite wall of the reactor.

In CURRENT, the conservation equations (1-3), (10) are transformed to a general curvilinear coordinate system to enable accurate representation of geometrically complex flow domains. Starting from cylindrical coordinates the equations are first transformed (cf. Figure 1) from (x, r, θ, t) to (ξ, η, θ, t) and then integrated over control volumes:

Mixture continuity equation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial\left(\rho u\right)}{\partial x} + \frac{1}{r}\frac{\partial\left(r\rho v\right)}{\partial r} = 0 \tag{13}$$

transformed:

$$\frac{\partial\rho}{\partial t} + \frac{1}{\sqrt{g}}\left\{\frac{\partial}{\partial\xi}(\rho U_g) + \frac{\partial}{\partial\eta}(\rho V_g)\right\} = 0 \tag{14}$$

integrated:

$$\sqrt{g}\,\frac{\Delta\rho}{\Delta t} + \rho U_g\Big|_w^e + \rho V_g\Big|_s^n = 0 \tag{15}$$

Axial component of momentum equation:

$$\begin{aligned}\frac{\partial(\rho u)}{\partial t} &+ \frac{\partial}{\partial x}\left(\rho uu - \mu\frac{\partial u}{\partial x}\right) + \frac{1}{r}\frac{\partial}{\partial r}\left(r\rho vu - r\mu\frac{\partial u}{\partial r}\right) = -\frac{\partial p_m}{\partial x} + g_x(\rho-\rho_{\text{ref}}) + \frac{1}{r}\frac{\partial}{\partial r}\left(r\mu\frac{\partial v}{\partial x}\right)\\ &+ \frac{\partial}{\partial x}\left\{\mu\frac{\partial u}{\partial x} - \frac{2\mu}{3}\left[\frac{1}{r}\frac{\partial(rv)}{\partial r} + \frac{\partial u}{\partial x}\right]\right\}\end{aligned} \tag{16}$$

transformed:

$$\begin{aligned}\frac{\partial\,(\rho u)}{\partial t} &+ \frac{1}{\sqrt{g}}\frac{\partial}{\partial\xi}\left(\rho U_g u - \frac{r^2 q_1}{\sqrt{g}}\mu u_\xi\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial\eta}\left(\rho V_g u - \frac{r^2 q_3}{\sqrt{g}}\mu u_\eta\right) = -\frac{r r_\eta}{\sqrt{g}}\frac{\partial p_m}{\partial\xi} + \frac{r r_\xi}{\sqrt{g}}\frac{\partial p_m}{\partial\eta}\\ &+ g_x(\rho-\rho_{\text{ref}}) - \frac{1}{\sqrt{g}}\frac{\partial}{\partial\xi}\left(\frac{r^2 q_2}{\sqrt{g}}\mu u_\eta\right) - \frac{1}{\sqrt{g}}\frac{\partial}{\partial\eta}\left(\frac{r^2 q_2}{\sqrt{g}}\mu u_\xi\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial\xi}\left\{\frac{r^2 r_\eta}{\sqrt{g}}\frac{\mu}{3}\Big(r_\eta u_\xi - r_\xi u_\eta\Big)\right.\\ &\left. - \frac{r r_\eta}{\sqrt{g}}\frac{2\mu}{3}\Big[x_\xi(rv)_\eta - x_\eta(rv)_\xi\Big] - \frac{r^2 x_\eta}{\sqrt{g}}\mu\Big(r_\eta v_\xi - r_\xi v_\eta\Big)\right\} + \frac{1}{\sqrt{g}}\frac{\partial}{\partial\eta}\left\{-\frac{r^2 r_\xi}{\sqrt{g}}\frac{\mu}{3}\Big(r_\eta u_\xi - r_\xi u_\eta\Big)\right.\\ &\left. + \frac{r r_\xi}{\sqrt{g}}\frac{2\mu}{3}\Big[x_\xi(rv)_\eta - x_\eta(rv)_\xi\Big] + \frac{r^2 x_\xi}{\sqrt{g}}\mu\Big(r_\eta v_\xi - r_\xi v_\eta\Big)\right\}\end{aligned} \tag{17}$$

integrated:

$$\begin{aligned}\sqrt{g}\,\frac{\Delta(\rho u)}{\Delta t} &+ \left(\rho U_g u - \frac{r^2 q_1}{\sqrt{g}}\mu u_\xi\right)\Bigg|_w^e + \left(\rho V_g u - \frac{r^2 q_3}{\sqrt{g}}\mu u_\eta\right)\Bigg|_s^n = -\{r r_\eta\}_{\text{ucv}}\,p_m\Big|_w^e + \{r r_\xi\}_{\text{ucv}}\,p_m\Big|_s^n\\ &+ \left[\sqrt{g}\,g_x(\rho-\rho_{\text{ref}})\right]_{\text{ucv}} - \left(\frac{r^2 q_2}{\sqrt{g}}\mu u_\eta\right)\Bigg|_w^e - \left(\frac{r^2 q_2}{\sqrt{g}}\mu u_\xi\right)\Bigg|_s^n + \left\{\frac{r^2 r_\eta}{\sqrt{g}}\frac{\mu}{3}\Big(r_\eta u_\xi - r_\xi u_\eta\Big)\right.\\ &\left. - \frac{r r_\eta}{\sqrt{g}}\frac{2\mu}{3}\Big[x_\xi(rv)_\eta - x_\eta(rv)_\xi\Big] - \frac{r^2 x_\eta}{\sqrt{g}}\mu\Big(r_\eta v_\xi - r_\xi v_\eta\Big)\right\}\Bigg|_w^e + \left\{-\frac{r^2 r_\xi}{\sqrt{g}}\frac{\mu}{3}\Big(r_\eta u_\xi - r_\xi u_\eta\Big)\right.\\ &\left. + \frac{r r_\xi}{\sqrt{g}}\frac{2\mu}{3}\Big[x_\xi(rv)_\eta - x_\eta(rv)_\xi\Big] + \frac{r^2 x_\xi}{\sqrt{g}}\mu\Big(r_\eta v_\xi - r_\xi v_\eta\Big)\right\}\Bigg|_s^n\end{aligned} \tag{18}$$

Radial component of momentum equation:

$$\begin{aligned}\frac{\partial\,(\rho v)}{\partial t} &+ \frac{\partial}{\partial x}\left(\rho uv - \mu\frac{\partial v}{\partial x}\right) + \frac{1}{r}\frac{\partial}{\partial r}\left(r\rho vv - r\mu\frac{\partial v}{\partial r}\right) = -\frac{\partial p_m}{\partial r} + g_r(\rho-\rho_{\text{ref}}) + r\rho f^2\\ &+ \frac{1}{r}\frac{\partial}{\partial r}\left\{r\mu\left[\frac{\partial v}{\partial r} - \frac{2}{3}\left(\frac{1}{r}\frac{\partial(rv)}{\partial r} + \frac{\partial u}{\partial x}\right)\right]\right\} - \frac{\mu}{r}\left\{\frac{2v}{r} - \frac{2}{3}\left[\frac{1}{r}\frac{\partial(rv)}{\partial r} + \frac{\partial u}{\partial x}\right]\right\} + \frac{\partial}{\partial x}\left(\mu\frac{\partial u}{\partial r}\right)\end{aligned} \tag{19}$$

transformed:

$$
\begin{aligned}
\frac{\partial\,(\rho v)}{\partial t} \;&+\; \frac{1}{\sqrt{g}}\frac{\partial}{\partial \xi}\Big(\rho U_g v - \frac{r^2 q_1}{\sqrt{g}}\mu v_\xi\Big) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial \eta}\Big(\rho V_g v - \frac{r^2 q_3}{\sqrt{g}}\mu v_\eta\Big) = -\frac{r x_\xi}{\sqrt{g}}\frac{\partial p_m}{\partial \eta} + \frac{r x_\eta}{\sqrt{g}}\frac{\partial p_m}{\partial \xi} + r\rho f^2 - \frac{2\mu v}{r^2} \\
&+\; g_r(\rho - \rho_{\rm ref}) - \frac{1}{\sqrt{g}}\frac{\partial}{\partial \xi}\Big(\frac{r^2 q_2}{\sqrt{g}}\mu v_\eta\Big) - \frac{1}{\sqrt{g}}\frac{\partial}{\partial \eta}\Big(\frac{r^2 q_2}{\sqrt{g}}\mu v_\xi\Big) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial \xi}\bigg\{\frac{r^2 x_\eta}{\sqrt{g}}\frac{2\mu}{3}\Big(r_\eta u_\xi - r_\xi u_\eta\Big) \\
&+\; \frac{r x_\eta}{\sqrt{g}}\frac{2\mu}{3}\Big[x_\xi (rv)_\eta - x_\eta (rv)_\xi\Big] - \frac{r^2 x_\eta}{\sqrt{g}}\mu\Big(x_\xi v_\eta - x_\eta v_\xi\Big) + \frac{r^2 r_\eta}{\sqrt{g}}\mu\Big(x_\xi u_\eta - x_\eta u_\xi\Big)\bigg\} \\
&+\; \frac{1}{\sqrt{g}}\frac{\partial}{\partial \eta}\bigg\{-\frac{r^2 x_\xi}{\sqrt{g}}\frac{2\mu}{3}\Big(r_\eta u_\xi - r_\xi u_\eta\Big) - \frac{r x_\xi}{\sqrt{g}}\frac{2\mu}{3}\Big[x_\xi (rv)_\eta - x_\eta (rv)_\xi\Big] + \frac{r^2 x_\xi}{\sqrt{g}}\mu\Big(x_\xi v_\eta - x_\eta v_\xi\Big) \\
&-\; \frac{r^2 r_\xi}{\sqrt{g}}\mu\Big(x_\xi u_\eta - x_\eta u_\xi\Big)\bigg\} + \frac{1}{\sqrt{g}}\frac{2\mu}{3}\bigg\{(r_\eta u_\xi - r_\xi u_\eta) + \frac{1}{r}\Big[x_\xi (rv)_\eta - x_\eta (rv)_\xi\Big]\bigg\}
\end{aligned}
\tag{20}
$$

integrated:

$$
\begin{aligned}
\sqrt{g}\frac{\Delta(\rho v)}{\Delta t} \;&+\; \Big(\rho U_g v - \frac{r^2 q_1}{\sqrt{g}}\mu v_\xi\Big)\bigg|_w^e + \Big(\rho V_g v - \frac{r^2 q_3}{\sqrt{g}}\mu v_\eta\Big)\bigg|_s^n = -\{r x_\xi\}_{\rm vcv} p_m\Big|_s^n + \{r x_\eta\}_{\rm vcv} p_m\Big|_w^e - \Big(\frac{r^2 q_2}{\sqrt{g}}\mu v_\eta\Big)\bigg|_w^e \\
&-\; \Big(\frac{r^2 q_2}{\sqrt{g}}\mu v_\xi\Big)\bigg|_s^n + \bigg\{\sqrt{g}\Big[r\rho f^2 - \frac{2\mu v}{r^2} + g_r(\rho - \rho_{\rm ref})\Big]\bigg\}_{\rm vcv} + \bigg\{\frac{r^2 x_\eta}{\sqrt{g}}\frac{2\mu}{3}\Big(r_\eta u_\xi - r_\xi u_\eta\Big) \\
&+\; \frac{r x_\eta}{\sqrt{g}}\frac{2\mu}{3}\Big[x_\xi (rv)_\eta - x_\eta (rv)_\xi\Big] - \frac{r^2 x_\eta}{\sqrt{g}}\mu\Big(x_\xi v_\eta - x_\eta v_\xi\Big) + \frac{r^2 r_\eta}{\sqrt{g}}\mu\Big(x_\xi u_\eta - x_\eta u_\xi\Big)\bigg\}\bigg|_w^e \\
&+\; \bigg\{-\frac{r^2 x_\xi}{\sqrt{g}}\frac{2\mu}{3}\Big(r_\eta u_\xi - r_\xi u_\eta\Big) - \frac{r x_\xi}{\sqrt{g}}\frac{2\mu}{3}\Big[x_\xi (rv)_\eta - x_\eta (rv)_\xi\Big] + \frac{r^2 x_\xi}{\sqrt{g}}\mu\Big(x_\xi v_\eta - x_\eta v_\xi\Big) \\
&-\; \frac{r^2 r_\xi}{\sqrt{g}}\mu\Big(x_\xi u_\eta - x_\eta u_\xi\Big)\bigg\}\bigg|_s^n + \bigg\{\frac{2\mu}{3}\bigg\{(r_\eta u_\xi - r_\xi u_\eta) + \frac{1}{r}\Big[x_\xi (rv)_\eta - x_\eta (rv)_\xi)\Big]\bigg\}\bigg\}_{\rm vcv}
\end{aligned}
\tag{21}
$$

Circumferential component of momentum equation:

$$
\frac{\partial(\rho f)}{\partial t} \;+\; \frac{\partial}{\partial x}\Big(\rho u f - \mu\frac{\partial f}{\partial x}\Big) + \frac{1}{r}\frac{\partial}{\partial r}\Big(r\rho v f - r\mu\frac{\partial f}{\partial r}\Big) = -\frac{2\rho v f}{r} + \frac{2\mu}{r}\frac{\partial f}{\partial r}
\tag{22}
$$

transformed:

$$
\begin{aligned}
\frac{\partial\,(\rho f)}{\partial t} \;&+\; \frac{1}{\sqrt{g}}\frac{\partial}{\partial \xi}\Big(\rho U_g f - \frac{r^2 q_1}{\sqrt{g}}\mu f_\xi\Big) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial \eta}\Big(\rho V_g f - \frac{r^2 q_3}{\sqrt{g}}\mu f_\eta\Big) = -\frac{2\rho v f}{r} - \frac{1}{\sqrt{g}}\frac{\partial}{\partial \xi}\Big(\frac{r^2 q_2}{\sqrt{g}}\mu f_\eta\Big) \\
&-\; \frac{1}{\sqrt{g}}\frac{\partial}{\partial \eta}\Big(\frac{r^2 q_2}{\sqrt{g}}\mu f_\xi\Big) + \frac{2\mu}{\sqrt{g}}\Big(x_\xi f_\eta - x_\eta f_\xi\Big)
\end{aligned}
\tag{23}
$$

integrated:

$$\sqrt{g}\,\frac{\Delta(\rho f)}{\Delta t} + \left(\rho U_g f - \frac{r^2 q_1}{\sqrt{g}}\mu f_\xi\right)\Bigg|_w^e + \left(\rho V_g f - \frac{r^2 q_3}{\sqrt{g}}\mu f_\eta\right)\Bigg|_s^n = -\left\{\sqrt{g}\frac{2\rho v f}{r}\right\}_{\text{scv}} - \left(\frac{r^2 q_2}{\sqrt{g}}\mu f_\eta\right)\Bigg|_w^e$$

$$- \left(\frac{r^2 q_2}{\sqrt{g}}\mu f_\xi\right)\Bigg|_s^n + \left\{2\mu(x_\xi f_\eta - x_\eta f_\xi)\right\}_{\text{scv}} \qquad (24)$$

species k $(k = 1, ..., K_g - 1)$ mass conservation equation:

$$\frac{\partial\,(\rho Y_k)}{\partial t} + \frac{\partial}{\partial x}\left(\rho u Y_k - \rho D_{km}\frac{\partial Y_k}{\partial x}\right) + \frac{1}{r}\frac{\partial}{\partial r}\left(r\rho v Y_k - r\rho D_{km}\frac{\partial Y_k}{\partial r}\right) = r_k$$

$$+ \frac{\partial}{\partial x}\left[Y_k D_{km}\left(\frac{\rho}{M}\frac{\partial M}{\partial x} - M\sum_{\substack{i=1\\ i\neq k}}^{K_g}\frac{j_{ix}{}^c}{M_i\boldsymbol{\mathcal{D}}_{ik}}\right) + \frac{D_k{}^T}{T}\frac{\partial T}{\partial x}\right]$$

$$+ \frac{1}{r}\frac{\partial}{\partial r}\left[rY_k D_{km}\left(\frac{\rho}{M}\frac{\partial M}{\partial r} - M\sum_{\substack{i=1\\ i\neq k}}^{K_g}\frac{j_{ir}{}^c}{M_i\boldsymbol{\mathcal{D}}_{ik}}\right) + \frac{rD_k{}^T}{T}\frac{\partial T}{\partial r}\right] \qquad (25)$$

transformed:

$$\frac{\partial\,(\rho Y_k)}{\partial t} + \frac{1}{\sqrt{g}}\frac{\partial}{\partial \xi}\left(\rho U_g Y_k - \frac{r^2 q_1}{\sqrt{g}}\rho D_{km} Y_{k\xi}\right) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial \eta}\left(\rho V_g Y_k - \frac{r^2 q_3}{\sqrt{g}}\rho D_{km} Y_{k\eta}\right) = r_k$$

$$- \frac{1}{\sqrt{g}}\frac{\partial}{\partial \xi}\left(\frac{r^2 q_2}{\sqrt{g}}\rho D_{km} Y_{k\eta}\right) - \frac{1}{\sqrt{g}}\frac{\partial}{\partial \eta}\left(\frac{r^2 q_2}{\sqrt{g}}\rho D_{km} Y_{k\xi}\right)$$

$$+ \frac{1}{\sqrt{g}}\frac{\partial}{\partial \xi}\left\{\frac{r^2}{\sqrt{g}}\frac{\rho D_{km} Y_k}{M}(q_1 M_\xi - q_2 M_\eta) - rY_k D_{km} M\left[r_\eta\sum_{\substack{i=1\\ i\neq k}}^{K_g}\frac{j_{ix}{}^c}{M_i\boldsymbol{\mathcal{D}}_{ik}} - x_\eta\sum_{\substack{i=1\\ i\neq k}}^{K_g}\frac{j_{ir}{}^c}{M_i\boldsymbol{\mathcal{D}}_{ik}}\right]\right.$$

$$\left. + \frac{r^2}{\sqrt{g}}\frac{D_k{}^T}{T}(q_1 T_\xi - q_2 T_\eta)\right\} + \frac{1}{\sqrt{g}}\frac{\partial}{\partial \eta}\left\{\frac{r^2}{\sqrt{g}}\frac{\rho D_{km} Y_k}{M}(q_3 M_\eta - q_2 M_\xi)\right.$$

$$\left. - rY_k D_{km} M\left[x_\xi\sum_{\substack{i=1\\ i\neq k}}^{K_g}\frac{j_{ir}{}^c}{M_i\boldsymbol{\mathcal{D}}_{ik}} - r_\xi\sum_{\substack{i=1\\ i\neq k}}^{K_g}\frac{j_{ix}{}^c}{M_i\boldsymbol{\mathcal{D}}_{ik}}\right] + \frac{r^2}{\sqrt{g}}\frac{D_k{}^T}{T}(q_3 T_\eta - q_2 T_\xi)\right\} \qquad (26)$$

integrated:

$$\sqrt{g}\,\frac{\Delta(\rho Y_k)}{\Delta t} + \left(\rho U_g Y_k - \frac{r^2 q_1}{\sqrt{g}}\rho D_{km} Y_{k\xi}\right)\Bigg|_w^e + \left(\rho V_g Y_k - \frac{r^2 q_3}{\sqrt{g}}\rho D_{km} Y_{k\eta}\right)\Bigg|_s^n = \sqrt{g}\,r_k - \left(\frac{r^2 q_2}{\sqrt{g}}\rho D_{km} Y_{k\eta}\right)\Bigg|_w^e$$

$$- \left(\frac{r^2 q_2}{\sqrt{g}}\rho D_{km} Y_{k\xi}\right)\Bigg|_s^n + \left\{\frac{r^2}{\sqrt{g}}\frac{\rho D_{km} Y_k}{M}(q_1 M_\xi - q_2 M_\eta) - rY_k D_{km} M\left[r_\eta\sum_{\substack{i=1\\ i\neq k}}^{K_g}\frac{j_{ix}{}^c}{M_i\boldsymbol{\mathcal{D}}_{ik}}\right.\right.$$

$$- \; x_\eta \sum_{\substack{i=1\\ i\neq k}}^{K_g} \frac{{j_{i_r}}^c}{M_i \mathcal{D}_{ik}} \Bigg] + \frac{r^2}{\sqrt{g}} \frac{{D_k}^T}{T} \left(q_1 T_\xi - q_2 T_\eta \right) \Bigg\} \Bigg|_w^e + \Bigg\{ \frac{r^2}{\sqrt{g}} \frac{\rho D_{km} Y_k}{M} \left(q_3 M_\eta - q_2 M_\xi \right)$$

$$- \; r Y_k D_{km} M \Bigg[x_\xi \sum_{\substack{i=1\\ i\neq k}}^{K_g} \frac{{j_{i_r}}^c}{M_i \mathcal{D}_{ik}} - r_\xi \sum_{\substack{i=1\\ i\neq k}}^{K_g} \frac{{j_{i_x}}^c}{M_i \mathcal{D}_{ik}} \Bigg] + \frac{r^2}{\sqrt{g}} \frac{{D_k}^T}{T} \left(q_3 T_\eta - q_2 T_\xi \right) \Bigg\} \Bigg|_s^n \qquad (27)$$

energy equation:

$$\frac{\partial\,(\rho T)}{\partial t} \; + \; \frac{\partial}{\partial x}\left(\rho u T - \frac{k}{c_p}\frac{\partial T}{\partial x} \right) + \frac{1}{r}\frac{\partial}{\partial r}\left(r \rho v T - r \frac{k}{c_p}\frac{\partial T}{\partial r} \right) = \frac{k}{{c_p}^2}\left(\frac{\partial T}{\partial x}\frac{\partial c_p}{\partial x} + \frac{\partial T}{\partial r}\frac{\partial c_p}{\partial r} \right)$$

$$- \; \frac{1}{c_p} \sum_1^{K_g} h_k r_k - \frac{1}{c_p} \sum_1^{K_g} c_{p_k} \left[\left({j_{k_x}}^c + {j_{k_x}}^T \right) \frac{\partial T}{\partial x} + \left({j_{k_r}}^c + {j_{k_r}}^T \right) \frac{\partial T}{\partial r} \right] \qquad (28)$$

transformed:

$$\frac{\partial\,(\rho T)}{\partial t} \; + \; \frac{1}{\sqrt{g}}\frac{\partial}{\partial \xi}\Big(\rho U_g T - \frac{r^2 q_1}{\sqrt{g}}\frac{k}{c_p} T_\xi \Big) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial \eta}\Big(\rho V_g T - \frac{r^2 q_3}{\sqrt{g}}\frac{k}{c_p} T_\eta \Big) = -\frac{1}{\sqrt{g}}\frac{\partial}{\partial \xi}\Big(\frac{r^2 q_2}{\sqrt{g}}\frac{k}{c_p} T_\eta \Big)$$

$$- \; \frac{1}{\sqrt{g}}\frac{\partial}{\partial \eta}\Big(\frac{r^2 q_2}{\sqrt{g}}\frac{k}{c_p} T_\xi \Big) - \frac{1}{c_p}\sum_1^{K_g} h_k r_k + \frac{r^2}{\sqrt{g}}\frac{k}{{c_p}^2}\left(q_1 T_\xi c_{p_\xi} - q_2 T_\eta c_{p_\xi} - q_2 T_\xi c_{p_\eta} + q_3 T_\eta c_{p_\eta} \right)$$

$$- \; \frac{r}{\sqrt{g}}\frac{1}{c_p}\sum_1^{K_g} c_{p_k} \Big[\left({j_{k_x}}^c + {j_{k_x}}^T \right)\left(r_\eta T_\xi - r_\xi T_\eta \right) + \left({j_{k_r}}^c + {j_{k_r}}^T \right)\left(x_\xi T_\eta - x_\eta T_\xi \right) \Big] \qquad (29)$$

integrated:

$$\sqrt{g}\,\frac{\Delta(\rho T)}{\Delta t} \; + \; \left(\rho U_g T - \frac{r^2 q_1}{\sqrt{g}}\frac{k}{c_p} T_\xi \right)\Bigg|_w^e + \left(\rho V_g T - \frac{r^2 q_3}{\sqrt{g}}\frac{k}{c_p} T_\eta \right)\Bigg|_s^n = \sqrt{g} r_k - \left(\frac{r^2 q_2}{\sqrt{g}}\frac{k}{c_p} T_\eta \right)\Bigg|_w^e$$

$$- \; \left(\frac{r^2 q_2}{\sqrt{g}}\frac{k}{c_p} T_\xi \right)\Bigg|_s^n - \sqrt{g}\frac{1}{c_p}\sum_1^{K_g} h_k r_k + \left[r^2 \frac{k}{{c_p}^2}\left(q_1 T_\xi c_{p_\xi} - q_2 T_\eta c_{p_\xi} - q_2 T_\xi c_{p_\eta} + q_3 T_\eta c_{p_\eta} \right) \right]_{\text{scv}}$$

$$- \; \left\{ r \frac{1}{c_p}\sum_1^{K_g} c_{p_k} \Big[\left({j_{k_x}}^c + {j_{k_x}}^T \right)\left(r_\eta T_\xi - r_\xi T_\eta \right) + \left({j_{k_r}}^c + {j_{k_r}}^T \right)\left(x_\xi T_\eta - x_\eta T_\xi \right) \Big] \right\}_{\text{scv}} \qquad (30)$$

In the integrated equations (15), (18), (21), (24), (27), and (30), the dimensions of the control volumes in the transformed coordinate system (ξ, η) are taken to be unity $(\Delta\xi = \Delta\eta = 1)$, and the equations are divided by 2π. The letters n, s, e, and w designate the north, south, east, and west faces of the control volume, respectively (cf. Figure 2). Subscripts ξ and η indicate derivatives with respect to those variables, respectively; subscripts scv, ucv, and vcv refer to scalar, x component of momentum, and r or y component of momentum control volumes, respectively. The Jacobian of the transformation is given by:

$$\sqrt{g} = r(x_\xi r_\eta - x_\eta r_\xi). \tag{31}$$

Other transformation relations are:

$$\begin{aligned}
q_1 &= {x_\eta}^2 + {r_\eta}^2 \\
q_2 &= x_\xi x_\eta + r_\xi r_\eta \\
q_3 &= {x_\xi}^2 + {r_\xi}^2 \\
U_g &= r(ur_\eta - vx_\eta) = \sqrt{g}\, U \\
V_g &= r(vx_\xi - ur_\xi) = \sqrt{g}\, V
\end{aligned} \tag{32}$$

where U and V are contravariant components of velocity (velocity components normal to surfaces of constant ξ and η coordinate values, respectively, [9]). Note that Equations (18) and (21), even though transformed, are still equations for the x (axial) and r (radial) components of momentum, respectively; thus the x and r components of velocity (u and v, respectively), shown at the faces of the scalar control volumes in Figures 2a and 2b, are not normal to those surfaces. In the circumferential component of momentum equation (24), the circumferential component of velocity has been scaled by the radial coordinate ($f \equiv w/r$). These equations reduce to the two-dimensional planar coordinate equations by setting r=1, replacing derivatives r_ξ and r_η with y_ξ and y_η, eliminating the terms (with the exception of the gravitational term) on the right hand side of Equation (21) that are evaluated over the control volume (those terms that are enclosed in brackets with the vcv subscript), and deleting the circumferential momentum equation (solving a 2D planar problem is an option in the code).

The integrated equations (18), (21), (24), (27), and (30) are solved in CURRENT. The mixture continuity equation (15) is not solved directly as written; instead the SIMPLER method [4] is used where Equations (18) and (21) are recast into expressions for u and v, which are substituted into (15) resulting in a Poisson equation for the pressure, p_m, which is then solved. A velocity correction is also part of the SIMPLER method; it consists of solving a Poisson equation for pressure correction and using the pressure corrections to update the velocity components to satisfy Equation (15). The procedure used to obtain the pressure correction equation is very similar to the one outlined here for pressure (see [4]).

A scalar control volume (scv) is shown in Figure 2a with a representative dependent variable, Φ (pressure p_m, temperature T, or mass fraction Y_k), located at its center. The velocity components are located at the midpoints of the sides of the scv; the coordinates of these dependent variables (the computer code variables for the coordinates are shown in Figure 2a) are obtained by linear interpolation of the coordinates of the corners of the scv which are determined in the mesh generation code AP (see Section 5). The control volumes for the axial and radial components of momentum (ucv and vcv) are staggered from the scalar control volume (scv) as shown in Figure 2b to avoid oscillations in the pressure field [4].

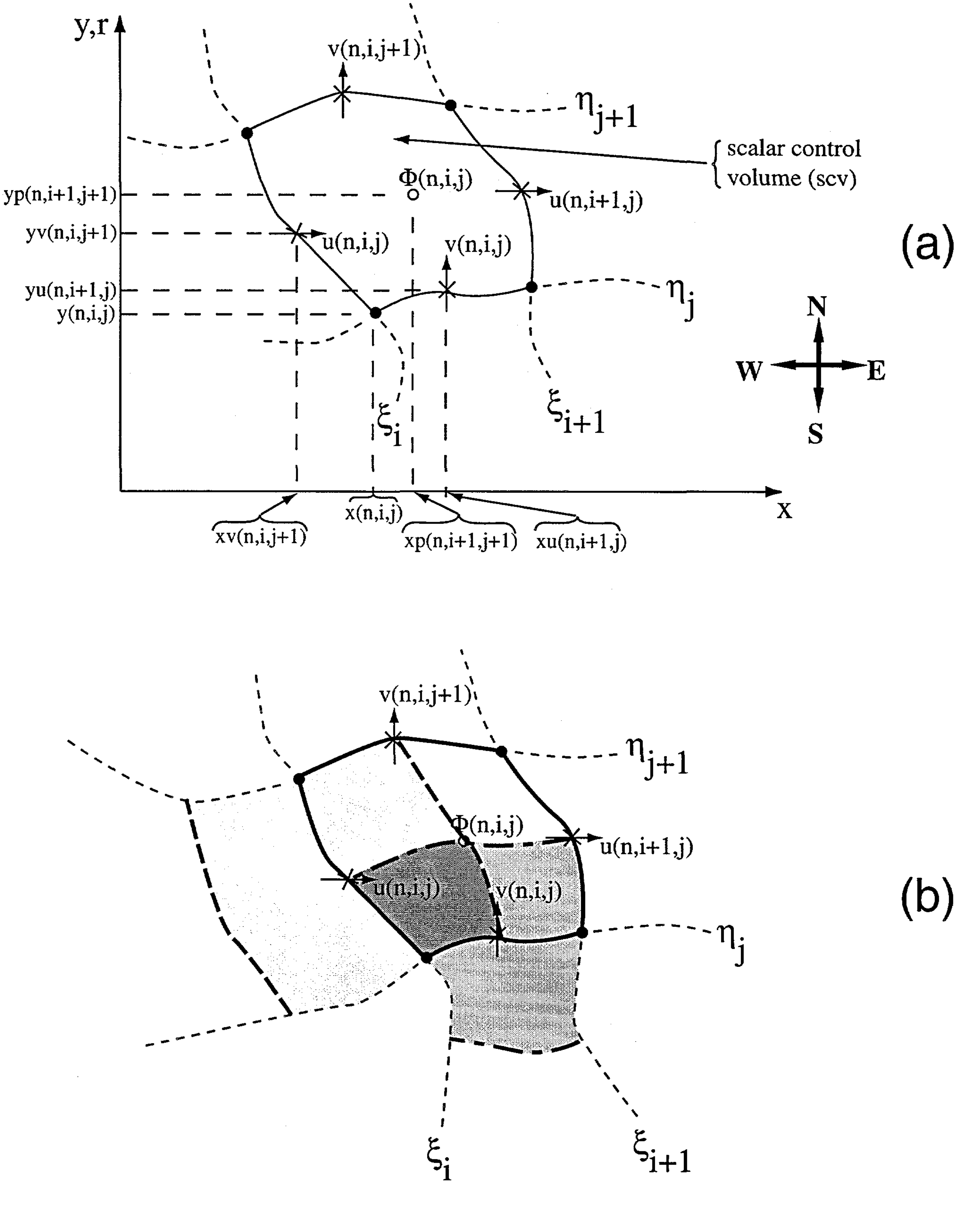


Figure 2: Control volumes for dependent variables: (a) scalar control volume coordinates; (b) staggered control volumes for scalar, x component of momentum, and r (or y) component of momentum variables.

Centered finite-differences are applied to discretize the remaining derivatives on the right hand sides of Equations (18), (21), (24), (27), and (30). For the convective-diffusive terms on the left hand sides of the equations, either the hybrid scheme [4], central differences, or the QUICK scheme [19] is used (user selectable). For cell Reynolds numbers ($\mathrm{Re}_{\mathrm{cell}}$) with absolute values greater than two, the hybrid scheme results in a first-order upwind difference of the convective term and the diffusive term is deleted; for $|\mathrm{Re}_{\mathrm{cell}}| < 2$, the hybrid scheme results in central differences for both the convective and diffusive terms.

3.4 Domain Decomposition

The CURRENT code is designed to solve problems with complex geometry and complex boundary conditions. This is accomplished by decomposing the problem into contiguous and non-overlapping regions as shown in the example problem (cf. Figure 16) and by applying the generalized coordinate transformation. The rules that apply to regions are summarized here (cf. Figure 3).

The n,s,e,w compass applies to regions as well as to individual control volumes. Only south faces of regions can be designated as symmetry boundaries (e.g., S_1 in Figure 3). Interpolation of variables between regions is not required. There must be a one-to-one correspondence between control volumes that are in adjacent regions and that lie along the common side of the two regions. For example, in Figure 3 the east side of region 2 is connected to the west side of region 3; the number of control volumes along the east side of region 2 matches the number on the west side of region 3, and the nodes of the east faces of control volumes in region 2 that lie along the line segment P_4P_8 connecting regions 2 and 3 match the corresponding nodes of the west faces of control volumes in region 3 (line segment P_4P_8 is called an internal boundary). For any side of a region, there is only one type of boundary condition allowed (e.g., S_2 of region 1 is a surface where deposition occurs, and S_8 is an inflow boundary).

Each region has a row of ghost-point control volumes surrounding it for the purpose of applying boundary conditions. The coordinates of the ghost-point control volumes that lie outside the computational domain are obtained by extrapolation from the control volume nodal values assigned in the mesh generation program AP. The dependent variable values at the ghost-point control volumes that are outside the computational domain are determined by linear extrapolation of the dependent variable value at the boundary and at the interior control volume adjacent to the boundary (*cf.* Figure 4). The application of boundary conditions is discussed in the next section.

The relationships between dependent variables at internal boundaries of regions is shown in Figure 3 for the case of a north-south boundary for scalar variables, Φ, and the y (or r) component of velocity, v. The three indices (n, i, j) of variables Φ and v are the region number n and the i and j grid values. Variable Φ(1, i, jmax) in region 1 has a ghost-point value, Φ(1, i, jmax+1), which is the variable Φ(2, i, 2) in region 2. Similarly, Φ(1, i, jmax) is the ghost-point value for variable Φ(2, i, 2) in region 2, where Φ(2, i, 1)=Φ(1, i, jmax). The y-component of velocity is located on the interface: in region 1, it is designated v(1, i, jmax+1) and in region 2 it is v(2, i, 2). The coordinate system, $(x, r$ or $y)$, is global with the x axis

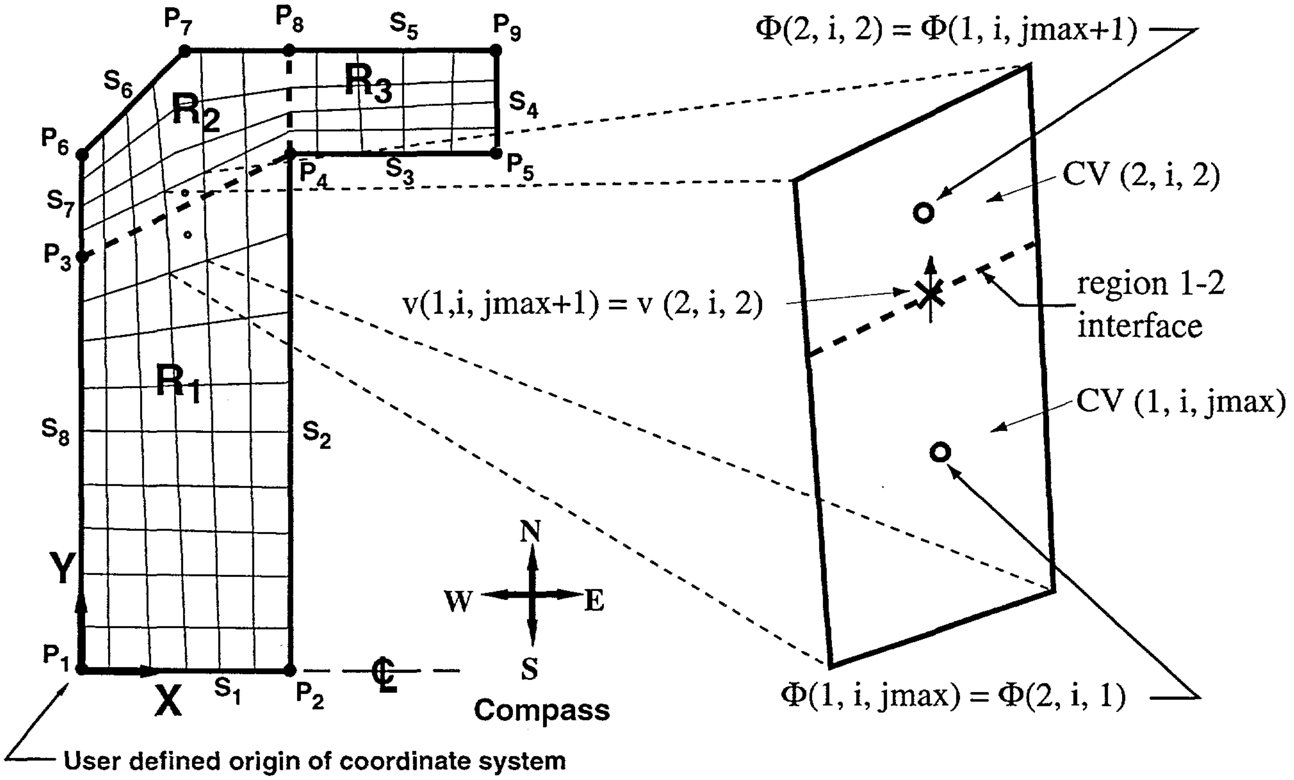

P7
P8
S5
P9
Φ(2, i, 2) = Φ(1, i, jmax+1)
S6
R2
R3
S4
P6
P5
P4
S3
CV (2, i, 2)
S7
P3
region 1-2
interface
v(1,i, jmax+1) = v (2, i, 2)
R1
CV (1, i, jmax)
S8
S2
N
Y
W
E
P1
S
X
S1
P2
℄
Compass
Φ(1, i, jmax) = Φ(2, i, 1)
User defined origin of coordinate system

Figure 3: Dependent variable relationships at interfaces between regions in CURRENT.

along the e-w compass directions, and the r or y axis along the n-s directions.

3.5 Boundary Conditions

As noted previously, for applying the boundary conditions a single layer of ghost-point control volumes surrounds each of the regions that make up a problem description in CURRENT. Along any one side of a region only one type of boundary condition specification is allowed. The following boundary condition types are supported in CURRENT: (1) inflow, (2) outflow, (3) symmetry, (4) surface, and (5) internal. There are two types of inflow boundary condition: (a) convective and (b) net mass flux. Outflow boundary conditions are problem dependent; zero gradient conditions are typically specified. A symmetry boundary condition can be applied on a region's south side only. There are several types of surface boundary conditions: (a) rotating or nonrotating, (b) reacting or nonreacting, and (c) specified temperature or heat flux.

If the mass transfer Peclet number ($\mathrm{Pec}_m = \mathrm{Re}\ \mathrm{Sc}_{km}$, where Re is the Reynolds number and Sc_{km} is the Schmidt number for species k in mixture m) is small (typically at low pressure), then diffusion can be significant at the inflow boundary of a chemically reacting flow problem. In this situation the net mass flux option:

$$\dot{\vec{m}}_k'' = \rho Y_k \vec{v}_k = \rho Y_k \vec{v} + \vec{j}_k \ , \quad k = 1, ..., K_g \tag{33}$$

which maintains the specified mass flow rate of each species by determining the relative contributions of convection and diffusion to $\dot{\vec{m}}_k''$ across the inflow boundary, is recommended. Using this option, the species mass fractions at the ghost-point control volumes upstream of the inflow boundary and the mass average velocity at the faces of the control volumes that form the inflow boundary are determined to satisfy Equations (33).

A proper outflow boundary condition is one that does not alter the transport processes within the computational domain. The effect that a particular outflow boundary condition has on a numerical solution is usually determined by either comparing the solution with other solutions (benchmark numerical solutions, analytical solutions, etc.) or moving the outflow boundary and running the modified problem to see the effect on the solution in the region of interest. A recommendation is to place the outflow boundary where there will be no recirculation regions anticipated and where the flow can develop into a one-dimensional or fully developed flow. Then the zero gradient or fully developed flow conditions:

$$\nabla\Phi \cdot \vec{n} = 0 \tag{34}$$

can be applied at the outflow boundary (by determining the value of the dependent variable Φ at the ghost-point control volume outside the boundary that satisfies (34)) with little or no effect on the transport in the domain of interest.

The symmetry condition is applied by determining the value of the dependent variable Φ at the ghost-point control volume outside the symmetry axis that satisfies (34).

Surface boundary conditions for velocity and temperature are applied by setting the ghost-point control volume value equal to a linear extrapolation through the value of the variable at the control volume adjacent to the boundary and the specified boundary condition (cf. Figure 4 where a part of the north side of region n is shown). For zero slip boundary conditions at nonreacting surfaces, $u_s = v_s = 0$; if the surface is not rotating, $f_s = 0$. For a rotating surface, the rotation rate, f_s is specified. For a north adiabatic surface, T(n,i,jmax+1) $=$ T(n,i,jmax).

Boundary conditions for the pressure and pressure correction equations are determined as follows: first the link to the ghost-point control volume outside the computational domain is broken by setting the appropriate coefficient in the discretised equation (44) equal to zero. This is equivalent to setting a zero gradient condition for the dependent variable at the boundary. For pressure correction the desired result of no correction of the velocity boundary condition is obtained. For the pressure equation (which is really the mixture continuity equation (15)), the mass flow rate that enters the control volume through the control volume face adjacent to the boundary is added into the source term, s_u, on the right hand side of the discretized pressure equation (44).

At a reacting surface, the boundary conditions for species and velocity are coupled as shown in Figure 5b. For situations where the mass transfer at the surface is due to surface chemistry only, the mass flux of each gas-phase species k at the surface is related to its production/destruction rate by surface chemical reactions (cf. Figure 5a):

$$(\rho Y_k \vec{v}_k \cdot \vec{n})_s = \sum_{\alpha=1}^{I} \nu_{k\alpha} r_\alpha M_k \tag{35}$$

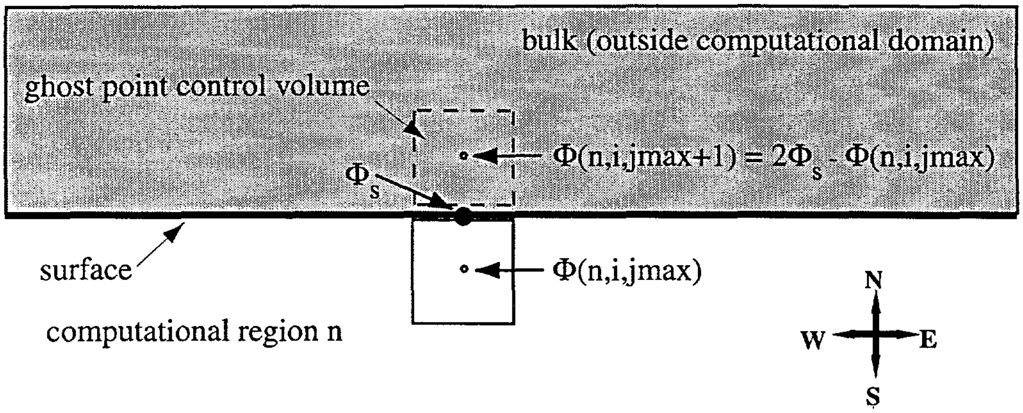

bulk (outside computational domain)
ghost point control volume
Φ_s
Φ(n,i,jmax+1) = 2Φ_s - Φ(n,i,jmax)
surface
Φ(n,i,jmax)
computational region n
N
W
E
S

Figure 4: Ghost-point control volume outside the computational domain showing linear extrapolation of dependent variable through the specified surface boundary condition.

where n is the outward pointing normal from the deposition surface, I is the number of surface reactions involving species k, and $\nu_{k\alpha}$ is the net stoichiometric coefficient for species k in surface reaction α:

$$\nu_{k\alpha} = \nu_{k\alpha}'' - \nu_{k\alpha}' \tag{36}$$

and the reactions are given in the form:

$$\sum_{k=1}^{K} \nu_{k\alpha}'\chi_k \rightleftharpoons \sum_{k=1}^{K} \nu_{k\alpha}''\chi_k \tag{37}$$

where χ_k is the k^{th} chemical species and the summation is over all species K (gas, surface, and bulk). Both the gas-phase molar production rates, r_α, in Equation (35), due to surface chemical reactions and the surface chemistry mechanism are implemented in CURRENT using the formalism of SURFACE CHEMKIN [7]. The surface species are assumed to be in local equilibrium (i.e., the net production rate of surface species is zero for each surface species k); we note, however, that finite rate surface chemical kinetics are accounted for in CURRENT via the specified surface chemical mechanism. Surface species distributions are determined by solving a set of coupled algebraic equations for the surface species site fractions for each segment of the surface that has a gas-phase control volume adjacent to it [20]. The composition of the gas mixture at the surface s, which is used in the surface chemistry rate expressions, is determined by linear extrapolation through two interior points:

$$\Phi_s = [3\Phi(\mathrm{n,i,jmax}) - \Phi(\mathrm{n,i,jmax}-1)]/2 \tag{38}$$

where Φ represents either gas-phase mass fraction Y_k or mole fraction X_k. The surface boundary condition for the conservation equation (27) of gas-phase species k involves replacing the mass flux (consisting of both convection and diffusion) of gas-phase species k across the gas-phase control volume surface that is adjacent

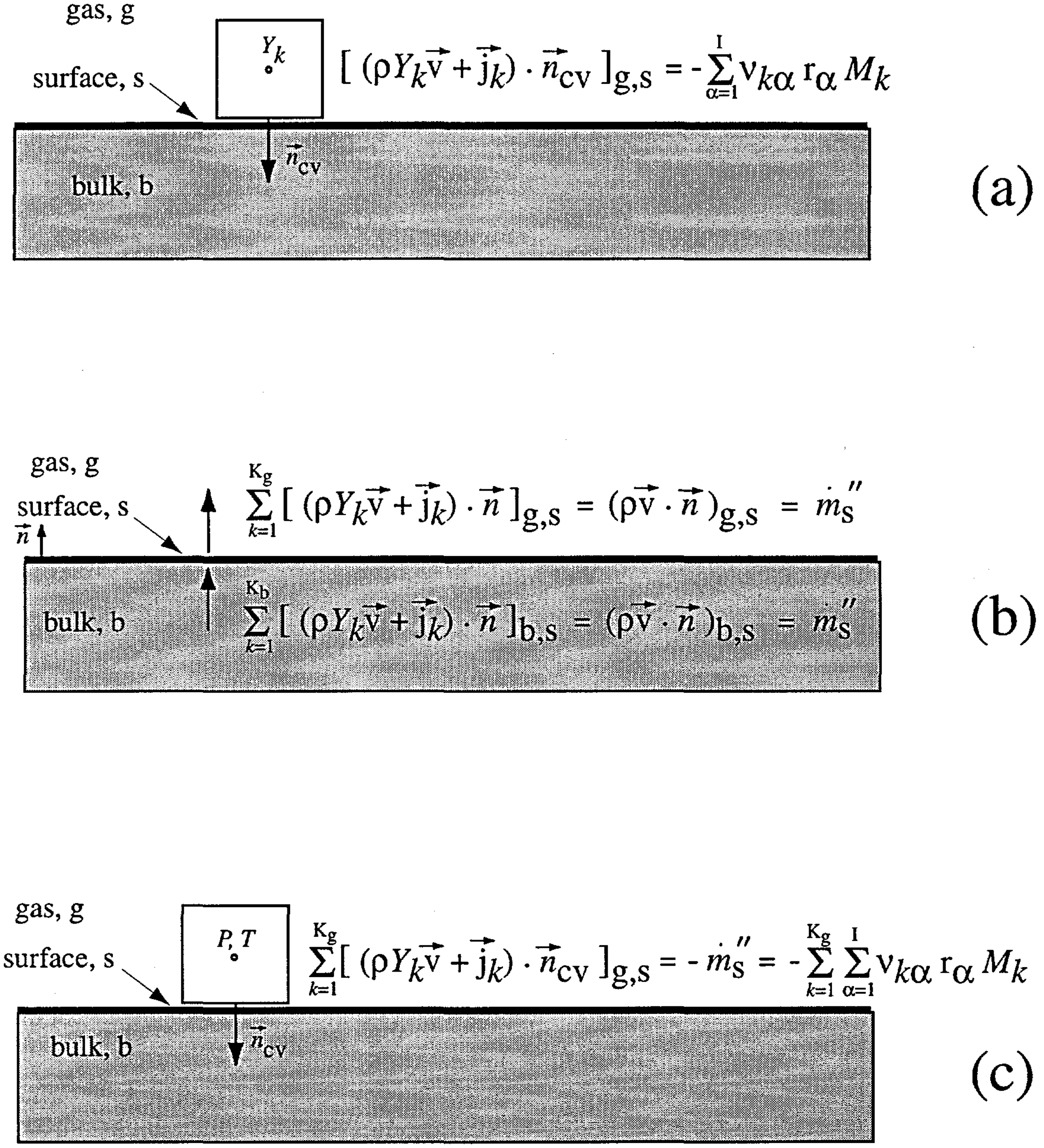

gas, g
surface, s
Y_k
$[\,(\rho Y_k \vec{v} + \vec{j}_k)\cdot \vec{n}_{cv}\,]_{g,s} = -\sum_{\alpha=1}^{I} \nu_{k\alpha}\, r_\alpha\, M_k$
$\vec{n}_{cv}$
bulk, b
(a)
gas, g
surface, s
$\vec{n}$
$\sum_{k=1}^{K_g} [\,(\rho Y_k \vec{v} + \vec{j}_k)\cdot \vec{n}\,]_{g,s} = (\rho \vec{v}\cdot \vec{n})_{g,s} = \dot{m}_s''$
bulk, b
$\sum_{k=1}^{K_b} [\,(\rho Y_k \vec{v} + \vec{j}_k)\cdot \vec{n}\,]_{b,s} = (\rho \vec{v}\cdot \vec{n})_{b,s} = \dot{m}_s''$
(b)
gas, g
surface, s
P, T
$\sum_{k=1}^{K_g} [\,(\rho Y_k \vec{v} + \vec{j}_k)\cdot \vec{n}_{cv}\,]_{g,s} = -\dot{m}_s'' = -\sum_{k=1}^{K_g}\sum_{\alpha=1}^{I} \nu_{k\alpha}\, r_\alpha\, M_k$
bulk, b
$\vec{n}_{cv}$
(c)

Figure 5: Mass transfer at a reacting surface: (a) mass flux of gas-phase species k across a control volume surface adjacent to a reacting surface; (b) overall steady-state mass balance at a surface; (c) mass flux of gas mixture across a control volume surface adjacent to a reacting surface.

to the deposition surface:

$$(\rho Y_k \vec{v}_k \cdot \vec{n}_{cv})_s = (\rho Y_k \vec{v} \cdot \vec{n}_{cv})_s + [(\vec{j}_k^c + \vec{j}_k^T) \cdot \vec{n}_{cv}]_s \tag{39}$$

with the right hand side of Equation (35). Summing Equation (35) over all species in the gas-phase (cf. Figures 5b,c) gives the mass average velocity component normal to a surface, $(\vec{v} \cdot \vec{n})_s$, where surface chemistry is the only mass transfer mechanism:

$$\sum_{k=1}^{K_g} (\rho Y_k \vec{v}_k \cdot \vec{n})_s = (\rho \vec{v} \cdot \vec{n})_s = \sum_{k=1}^{K_g} \sum_{\alpha=1}^{I} \nu_{k\alpha} r_\alpha M_k. \tag{40}$$

Due to the coordinate transformation, the component of the mass average velocity normal to the surface, $(\vec{v} \cdot \vec{n})_s$, is the contravariant velocity component U or V, depending on whether the deposition surface corresponds to a constant value of ξ or η, respectively.

The species boundary condition that is applied to the gas-phase control volume adjacent to a surface where there is no mass transfer (deposition, evaporation, ablation, transpiration, etc.) is:

$$[(\vec{j}_k^c + \vec{j}_k^T) \cdot \vec{n}_{cv}]_s = 0 \tag{41}$$

since in this case the velocity component $(\vec{v} \cdot \vec{n})_s$ is zero.

An energy balance on a surface element is shown in Figure 6a where radiation transport from the surface and the Dufour effect in the gas have been neglected. The energy release rates due to surface chemical reactions do not appear explicitly in the energy balance when cast in terms of enthalpy:

$$\{-k\nabla T \cdot \vec{n} + \sum_{k=1}^{K_g} [(\rho Y_k \vec{v} + \vec{j}_k^c + \vec{j}_k^T) \cdot \vec{n} h_k]\}_{s,g} = \{-k\nabla T \cdot \vec{n} + \sum_{k=1}^{K_b} [(\rho Y_k \vec{v} + \vec{j}_k^c + \vec{j}_k^T) \cdot \vec{n} h_k]\}_{s,b}. \tag{42}$$

The surface energy balance (42) for the case of surface chemistry can be written as:

$$(-k\nabla T \cdot \vec{n})_{s,g} + \sum_{k=1}^{K_g} \sum_{\alpha=1}^{I} \nu_{k\alpha} r_\alpha M_k h_k = (-k\nabla T \cdot \vec{n})_{s,b} + \sum_{k=1}^{K_b} \sum_{\alpha=1}^{I} \nu_{k\alpha} r_\alpha M_k h_k. \tag{43}$$

Equation (43) can be solved for the surface temperature, T_s, if the heat flux in the bulk, $(-k\nabla T \cdot \vec{n})_{s,b}$, is given, or it can be solved for the bulk heat flux if the surface temperature is specified. Although CURRENT is designed to solve surface energy balances and the conjugate problem of conduction in the bulk, the user interface does not support this option now. For a chemically reacting surface with a specified surface temperature, the surface boundary condition for the gas-phase energy control volume is determined by evaluating the left hand side of Equation (42) with properties at the surface obtained by linear extrapolation from the interior control volumes.

The surface energy balance in terms of fluxes from the surface of the gas-phase control volume adjacent to the reacting surface and in terms of energy release rates due to the surface chemical reactions and conduction in the bulk is shown in Figure 6b. If the conduction heat flux in the bulk were given (not the typical situation), the fluxes from the surface of the gas-phase control volume shown on the left hand side of the equation in Figure 6b (and the left hand side of Equation (42)) could be replaced by the right hand side of the equations (in Figure 6b or in Equation (43)).

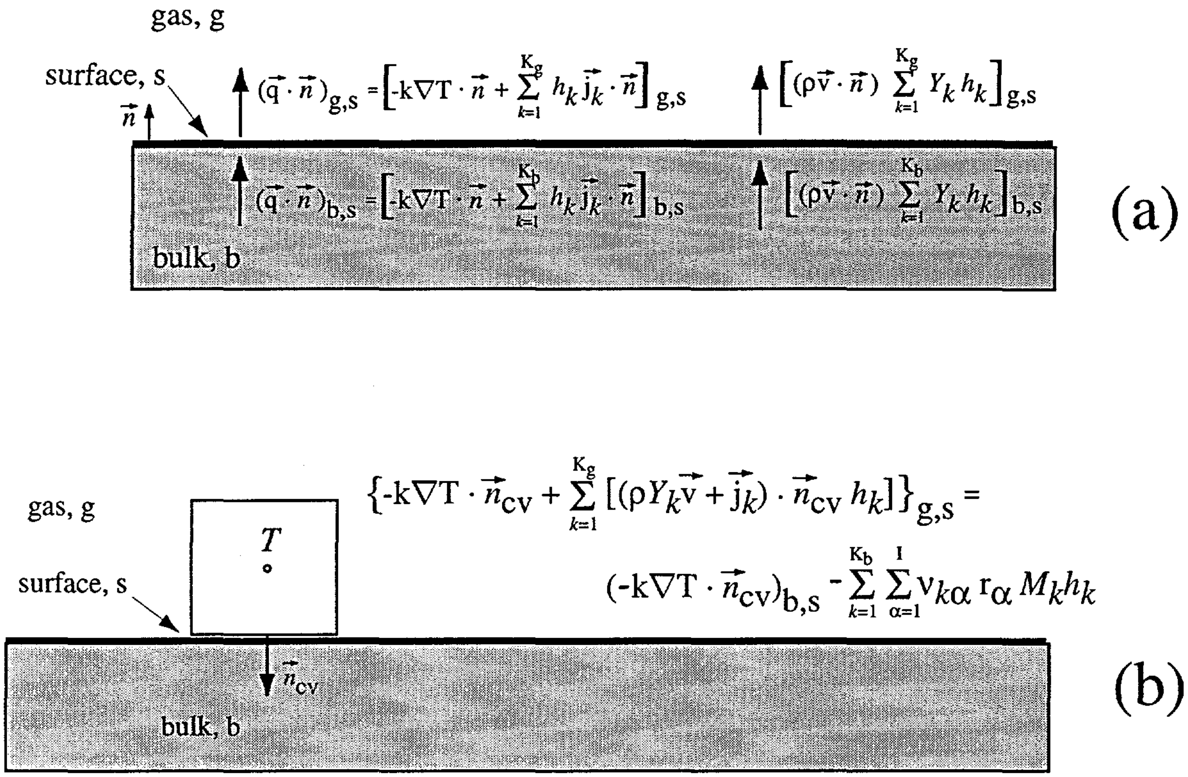

gas, g
surface, s
$\vec{n}$
$(\vec{q}\cdot\vec{n})_{g,s} = \left[-k\nabla T\cdot\vec{n} + \sum_{k=1}^{K_g} h_k \vec{j}_k\cdot\vec{n}\right]_{g,s}$
$\left[(\rho\vec{v}\cdot\vec{n}) \sum_{k=1}^{K_g} Y_k h_k\right]_{g,s}$
$(\vec{q}\cdot\vec{n})_{b,s} = \left[-k\nabla T\cdot\vec{n} + \sum_{k=1}^{K_b} h_k \vec{j}_k\cdot\vec{n}\right]_{b,s}$
$\left[(\rho\vec{v}\cdot\vec{n}) \sum_{k=1}^{K_b} Y_k h_k\right]_{b,s}$
bulk, b
(a)
gas, g
surface, s
T
$\{-k\nabla T\cdot\vec{n}_{cv} + \sum_{k=1}^{K_g} [(\rho Y_k\vec{v} + \vec{j}_k)\cdot\vec{n}_{cv}\, h_k]\}_{g,s} =$
$(-k\nabla T\cdot\vec{n}_{cv})_{b,s} - \sum_{k=1}^{K_b}\sum_{\alpha=1}^{I} \nu_{k\alpha}\, r_\alpha M_k h_k$
$\vec{n}_{cv}$
bulk, b
(b)

Figure 6: Energy transfer at a surface: (a) surface energy balance (neglecting radiation and the Dufour effect); (b) the energy balance in terms of energy fluxes from the surface of a gas-phase control volume adjacent to a reacting surface (left hand side of equation) and the energy generation rate due to surface reactions and conduction in the bulk (right hand side of equation).

3.6 Numerical Solution

Equations (15), (18), (21), (24), (27), and (30) can be written in a single form [4]:

$$a_P\Phi_P = a_n\Phi_N + a_s\Phi_S + a_e\Phi_E + a_w\Phi_W + s_u \tag{44}$$

where $a_P = \sum_{i=n,s,e,w} a_i - s_P$, and Φ_P are the dependent variables: p_m, u, v, f, Y_k, and T at control volume P. For example, consider the conservation equation (27) for gas-phase species Y_k, and for simplicity assume that the central differencing scheme is used. Figure 7 shows the control volume P and its four neighbors in the computational plane. The coefficients for Equation (44) consist of a diffusive component and a convective component:

$$\begin{aligned} a_n &= dn - cn/2, \\ a_s &= ds + cs/2, \\ a_e &= de - ce/2, \\ a_w &= dw + cw/2. \end{aligned}$$

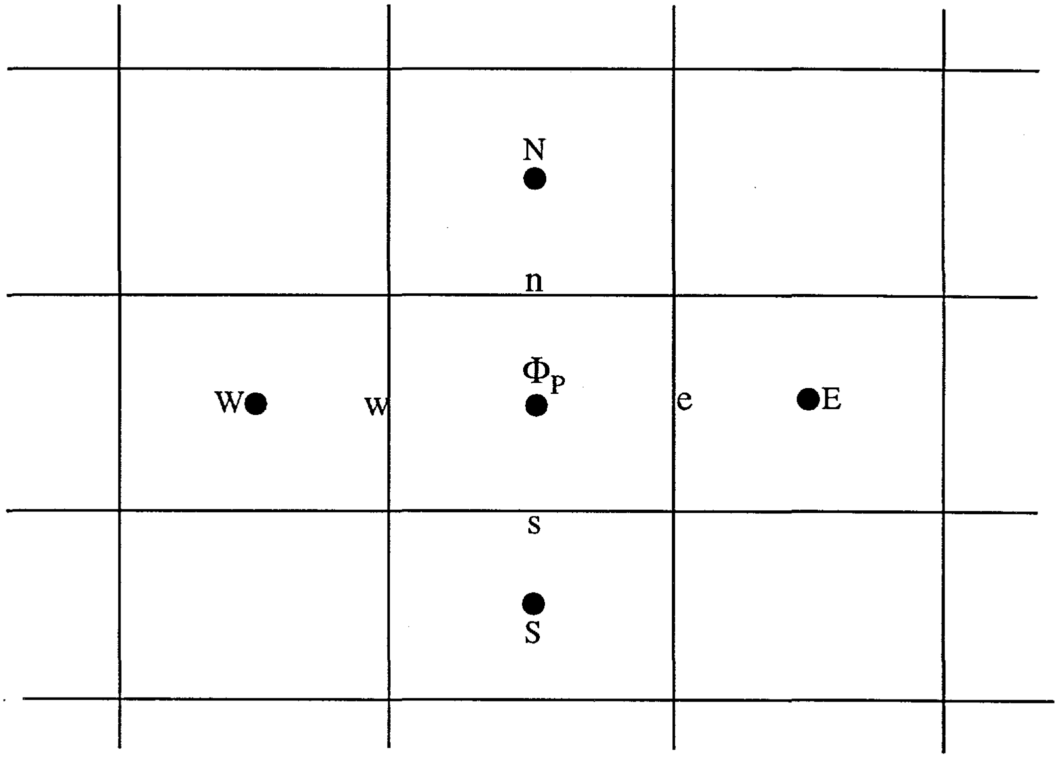


Figure 7: Control volume P for dependent variable Φ_P in the computational domain.

Relative to Equation (27), the diffusive transport contributions are:

$$\begin{aligned}
dn &= (r^2 q_3 \rho D_{km}/\sqrt{g})_n, \\
ds &= (r^2 q_3 \rho D_{km}/\sqrt{g})_s, \\
de &= (r^2 q_1 \rho D_{km}/\sqrt{g})_e, \\
dw &= (r^2 q_1 \rho D_{km}/\sqrt{g})_w,
\end{aligned}$$

and the convective transport contributions are:

$$\begin{aligned}
cn &= (\rho V_g)_n, \\
cs &= (\rho V_g)_s, \\
ce &= (\rho U_g)_e, \\
cw &= (\rho U_g)_w.
\end{aligned}$$

and the source terms are:

$$\begin{aligned}
s_u &= -(cn - cs + ce - cw) Y_{k_{\mathrm{P}}} + \sqrt{g} (\rho Y_k)_{\mathrm{P}}{}^{n-1}/\Delta t + \mathrm{RHS}|_{\mathrm{eqn.(27)}}, \\
s_{\mathrm{P}} &= -\sqrt{g} \rho_{\mathrm{P}}{}^{n}/\Delta t.
\end{aligned}$$

The $(n-1)$ superscript on the one s_u term refers to the previous time level; all other terms are evaluated at time level n. The coefficients for hybrid differencing are given in [4] and those for the QUICK scheme are in [19]. The coefficients for the other conservation equations are defined analogously, where, as mentioned earlier, the SIMPLER scheme is used to replace (15) with an equation for p_m. The species conservation equations (27) are solved for $K_g - 1$ species, and the remaining species, Y_{K_g}, is obtained from $1 - \sum_{k=1}^{K_g-1} Y_k$. The species not solved for is the one with the largest mass fraction.

Under-relaxation is applied to the Equations (44) in the manner described in [4]. Typical under-relaxation factors are 0.5 with no under-relaxation of the pressure or pressure-correction equations. The linearized difference equations (44) are solved along mesh lines in the computational domain using the tridiagonal matrix algorithm (TDMA). The solution proceeds sequentially through the variables, each variable being solved over all the computational regions before proceeding to the next variable. Within each region, the algorithm performs alternating sweeps of the TDMA solver. One global iteration is complete when all the variables are solved in this manner, and the boundary conditions have been updated.

Iterations are continued until convergence is obtained. Convergence criteria for stopping the iterations within a time step are based on relative and absolute changes in the dependent variables from one iteration to the next. All variables at all control volumes are checked. The relative and absolute convergence criteria can be specified in the input file through the REAL array. However, the default values in CURRENT are sufficiently small that these criteria are typically not reached for a steady-state problem, and the user controls the number of iterations taken and checks for convergence in the manner described in Section 5 (by comparing profiles and fields of the dependent variables at selected iteration intervals). The restart feature of the code is useful to continue the iterations to convergence. In a steady-state problem, the iterations are performed within a single, very large time step that is set to 10^{20} seconds.

The main steps in the solution sequence are as follows:

```
Start
    1.  Set default values.
    2.  Read problem definition and grid.
    3.  Initialize CHEMKIN, the transport properties software, and SURFACE CHEMKIN.
    4.  Set reference properties and scale factors.
    5.  Interpolate grid and define metric coefficients.
If not a restart, then
    6.  Initialize all field variables and set boundary conditions.
else
    6.  Read all field variables from restart file and set boundary conditions.
end if
A. Begin time step loop.
  B. Begin iteration loop within a time step.
    7.  Solve modified mixture continuity equation ((15) using the SIMPLER technique [4])
        for the pressure field (variable $p_m$).
    8.  Update scalar boundary conditions.
    9.  Solve $\theta$ component of momentum equation ((24), variable $f$).
    10.  Solve energy equation (30) for the temperature field (variable $T$).
    11.  Solve species equations (27) for gas-phase mass fraction distributions (variable $Y_k$,  $k = 1, ..., K_g -$
         and determine surface species site fraction distributions on chemically reacting surfaces.
    12.  Determine $Y_{K_g}$ from $1 - \sum_{k=1}^{K_g - 1} Y_k$.
    13.  Update fluid mixture properties.
    14.  Determine species diffusive mass fluxes by solving the system of equations ((5), $k = 1, ..., K_g$)
         that are equivalent to the Stefan-Maxwell equations.
    15.  Update scalar variable and property boundary conditions.
    16.  Solve $x$ and $r$ or $y$ components of momentum equations ((18), (21), variables $u, v$).
    17.  Update contravariant velocity components (32).
    18.  Update velocity boundary conditions.
    19.  Correct velocity components using solution of pressure correction equation (part of SIMPLER sche
    20.  Check global mass and energy balances and implement constraint on outflow velocity.
    21.  Check for convergence and print diagnostics, output files, and restart file if requested.
If not converged, then
  If iterations have reached maximum number, then
    22.  stop.
  else
    22.  Loop back to B (iterate).
  end if
else
    22.  Perform time step control.
    23.  Check global mass and energy balances.
    24.  Print diagnostics, output files, and restart file if requested.
    25.  Increment time.
  If time has reached maximum time, then
    26.  stop.
  else
    26.  Loop back to A (next time step).
  end if
end if
```

4 Convergence Acceleration

The SIMPLER algorithm, described in the previous sections, converges slowly at low pressures where the Peclet number for mass diffusion is low. The root of the low-Peclet number convergence problem is the interaction between the solution procedure and the unfavorable scaling within the chemical species mass-flux-preserving inflow boundary condition. Modifications to the SIMPLER algorithm can improve the solution times of difficult problems by several orders of magnitude [21], and usually provides modest improvements for all problems. The algorithm enhancements described in this section are not standard features of the CURRENT code and can be activated only through one of solution method "switches".

Chemical reaction rates, which usually cause convergence problems in reacting flow computations, play a lesser role in the convergence rate degradation. The reactions are relatively slow at low pressure and not very energetic so there is not a strong effect on the temperature. Surface chemistry is more problematic than gas-phase chemistry since it provides highly nonlinear sources and sinks for gas-phase species at physical boundaries.

For highly diffusive problems, convergence is degraded by poor numerical propagation of information between physical boundaries. Difficulties arise at an inflow boundary where the net mass flow condition (33) is set and convection must balance diffusion to preserve specified species mass fluxes. Information propagates very slowly from such a boundary for a particular class of segregated solution algorithms with explicit updating of boundary conditions, such as SIMPLER. Boundary condition equations should be implicitly coupled to interior equations with complete solution of the resulting linearized system.

4.1 Peclet Number Scaling and Inflow Boundary Conditions

A scaling problem between convection and diffusion is artificially introduced by using a flux-inflow boundary condition (such as the "net mass flux" condition, Section 3.5) near a strong species composition gradient. For low Peclet numbers, diffusive transport dominates convective transport. Yet, for the limiting case of uniform inflow composition and no chemical reactions, the convective transport term determines the species distribution within the flow domain. The Peclet number scaling makes it difficult to enforce the convective part of the boundary condition. The difference between the strength of convection and diffusion makes it difficult to propagate information in the solution procedure when the boundary condition is enforced explicitly.

The scaling is demonstrated using the inflow boundary condition, Equation (33), where the diffusion flux is simplified by the assumption of Fickian diffusion:

$$\frac{\dot{m}_k}{A} = \rho u Y_k - \rho D \frac{\partial Y_k}{\partial x}. \tag{45}$$

Dividing by the total mass flux gives:

$$\overline{Y}_k = Y_k - \frac{D}{u}\frac{\partial Y_k}{\partial x}, \tag{46}$$

where $\overline{Y}_k = \dot{m}_k/\dot{m}$ represents the reference inflow mass fraction for species k. The cell-Peclet number, $\mathrm{Pe}_{\Delta x}$, is defined by the length scale of the finite volume, Δx, the convective velocity, and a mass diffusivity, D:

$$\mathrm{Pe}_{\Delta x} = \frac{u\Delta x}{D}. \tag{47}$$

The boundary condition is discretized about the inflow cell face using centered differences and the explicit formula for updating the boundary point is:

$$Y_{k,1} = \frac{1 - \frac{1}{2}\mathrm{Pe}_{\Delta x}}{1 + \frac{1}{2}\mathrm{Pe}_{\Delta x}} Y_{k,2} + \frac{\mathrm{Pe}_{\Delta x}}{1 + \frac{1}{2}\mathrm{Pe}_{\Delta x}} \overline{Y}_k. \tag{48}$$

The ghost-cell mass fraction is $Y_{k,1}$ and the first interior cell value is $Y_{k,2}$. For small limiting values of the cell-Peclet number, the ghost-cell mass fraction is more dependent on the interior point than the reference value $\overline{Y}_k$ during the explicit update. Yet, the mass fractions must approach the reference value when there are no chemical reactions or sources. In the limit of small Peclet number, Equation (48) becomes:

$$Y_{k,1} = Y_{k,2} + \mathrm{Pe}_{\Delta x}\overline{Y}_k. \tag{49}$$

Bad initial guesses for the mass fractions result in very slow convergence to the actual solution. The interior points are strongly dependent on the boundary points because of the elliptic nature of the partial differential equations. The scaling argument indicates that any implicit scheme used to update interior points without implicitly coupling the boundary points is ineffective.

The solution procedure can be improved by implicitly coupling the physical boundary conditions in the line-relaxation scheme. The boundary information is propagated implicitly in the direction of the line-solve, but still "explicit-like" in the sweep direction. The full set of physical boundaries, which drive the interior equations, are not implicitly coupled. The explicit propagation of information is degraded by the presence of internal block boundaries. In order to maximize the performance of the solution procedure, all boundary conditions must be implicitly coupled; each boundary point should simultaneously see every other boundary point.

4.2 Enhancements to the Navier-Stokes Code

The solution algorithm enhancements to the SIMPLER algorithm in the CURRENT code involve only the species transport equations and the pressure-correction equation. The modifications to the species equations are such that boundary condition equations are treated implicitly and the linear problem, Equation (44), is solved to completion using a matrix-free, preconditioned gradient algorithm.

Matrix-free, preconditioned gradient algorithms provide an efficient solution to the boundary communication problem. Gradient algorithms are a generalization of the conjugate gradient method, used to iteratively invert matrices. Preconditioning improves the convergence characteristics of gradient algorithms. In CURRENT, the generalized minimal residual (GMRES) gradient algorithm [22] is used to invert the linear system with the block-by-block line-relaxation implicit scheme acting as a preconditioner. The GMRES scheme enhances the implicitness of the line-relaxation scheme. The TDMA line-relaxation algorithm in CURRENT, described in Section 3.6, is replaced with a more general line-relaxation algorithm, capable of handling the implicit boundary condition equations.

A new problem arises when the linearized species equations are fully converged at each step; the nonlinear solution process becomes unstable. In the baseline SIMPLER algorithm, neither continuity nor the linearized species equations are satisfied exactly at each nonlinear iteration. The errors tend to offset each other and there are no large instabilities. Conversely, the enhanced solution algorithm does such a good job of satisfying the transport equations that they become very sensitive to mass errors. Errors in continuity cause artificial sources and sinks in the species equations. Stability can be increased by satisfying the continuity equation more rigorously. Specific modifications are described in the following sections.

4.2.1 Species Transport Equation

The linear matrix increases in size because all the ghost cell values, which are used to help set boundary conditions, become equations implicitly coupled to the interior transport equations. The equations for ghost cells at grid block interfaces equate the ghost cells of one block to the equivalent interior cells of the adjacent block. The relations for explicitly updating physical boundary conditions are cast as implicit equations.

The under-relaxation used in the SIMPLER algorithm (see Section 3.6) must be reformulated for use with the enhanced solution strategy. The under-relaxation adds diagonal dominance to the linearized equations, but sharply degrades performance when there are Peclet number scaling problems. The under-relaxation acts as an artificial time term, slowing down the propagation of information even more. A more favorable implementation of under-relaxation is to apply the damping parameter to the solution increment which results from the solution of the linearized equations.

4.2.2 Continuity Equation

The solution to the problem of converging the continuity equation is twofold. First, it is recognized that continuity errors during iteration are due to the use of zero-gradient boundary conditions for the pressure-correction on all boundaries and incomplete convergence of the linear pressure-correction equation. The zero-gradient boundary condition on the pressure-correction term does not allow outflow boundary velocities to change. The outflow boundary condition is reformulated so that the outflow velocity can be corrected in a manner consistent with continuity. Secondly, the matrix-free gradient algorithm is added to the solution algorithm for the pressure-correction equation to accelerate convergence.

The pressure-correction procedure is reformulated in a manner consistent with other projection methods for low Mach number flows [23, 24, 25]. The gradient of the pressure-correction corresponds to a velocity increment and the pressure-correction variable itself is related to a pressure increment. In the modified algorithm, the pressure-correction is set to zero at the outflow boundary ghost points. A non-zero velocity correction can then be calculated at the outflow boundary and the velocity updated in a way that is consistent with continuity. Continuity is satisfied in each finite-volume at each nonlinear step. The zero-condition also corresponds to fixing outflow pressure since the pressure increment is zero. This does not mean the outflow pressure is fixed for the entire calculation because the pressure itself is still updated according to the SIMPLER algorithm.

The actual implementation requires matrix coefficients at the outflow face. If the outflow is on an east face, then a value for A_e would be required in Equation (44) for the interior volume adjacent to the outflow boundary. In the baseline SIMPLER algorithm, A_e is set to zero. In the modified algorithm, the matrix coefficient is extrapolated from the interior.

5 Using CURRENT

This section describes the process of using CURRENT to solve problems. Included is a discussion of problem formulation, mesh creation, input file preparation, code execution and post-processing. New users will be exposed to these steps through the presentation of a comprehensive example problem designed to exploit most currently supported features of the code.

The example problem is a steady-state, multi-species Navier-Stokes flow with thermal transport (*i.e.* energy equation is solved) and both gas-phase and surface chemistry. All transport and gas properties are obtained from CURRENT's subroutine calls to CHEMKIN software libraries. The details of the CHEMKIN input file preparation will not be presented (see References [6, 7, 8]) but a discussion of meshing and post-processing using ANTIPASTO (AP) [5] will be provided.

5.1 Overview

The procedure for CURRENT pre-processing (mesh generation and input file preparation), execution (running CURRENT with CHEMKIN), and post-processing (examining the output from CURRENT) involves a number of different steps and files, both ASCII and binary. The purpose of this section is to outline the basic steps as they are represented schematically in Figure 8. It may be informative to refer frequently to this figure during the discussion.

Several files and file types are associated with running CURRENT with CHEMKIN. The following three letter extensions are used to indicate file types:

```
.inp - An ascii input file for CURRENT/CHEMKIN
.dat - An ascii data file for CURRENT/CHEMKIN
.bin - A binary file associated with CURRENT/CHEMKIN
.out - An ascii file associated with CURRENT/CHEMKIN
```

The input files *mesh.inp, problem.inp, chem.inp* and *surf.inp* are unique to each problem. Data files *parser.dat, therm.dat* and *tran.dat* are often used for many different problems and seldom require modification by the user. CHEMKIN data files *therm.dat* and *tran.dat* require modification if modeled species are not in the database. The files *restart_inp.bin, restart_out.bin* and *ap.bin* are binary input and output files which are automatically generated during CURRENT execution. The ascii output files *parser.out, resids.out, monitor.out, current.out, chem.out, tran.out, surf.out* and *chemkin.out* are diagnostic message files which may or may not be generated during execution depending on options set in *problem.inp*. The *.out* files are primarily used in code development for examining code performance and should be of little interest to the casual user. One exception is the file *tecplot.out* which is an ascii output file that can be used in post-processing with TECPLOT [26].

The basic steps in setting up and running CURRENT with CHEMKIN may be listed as follows:

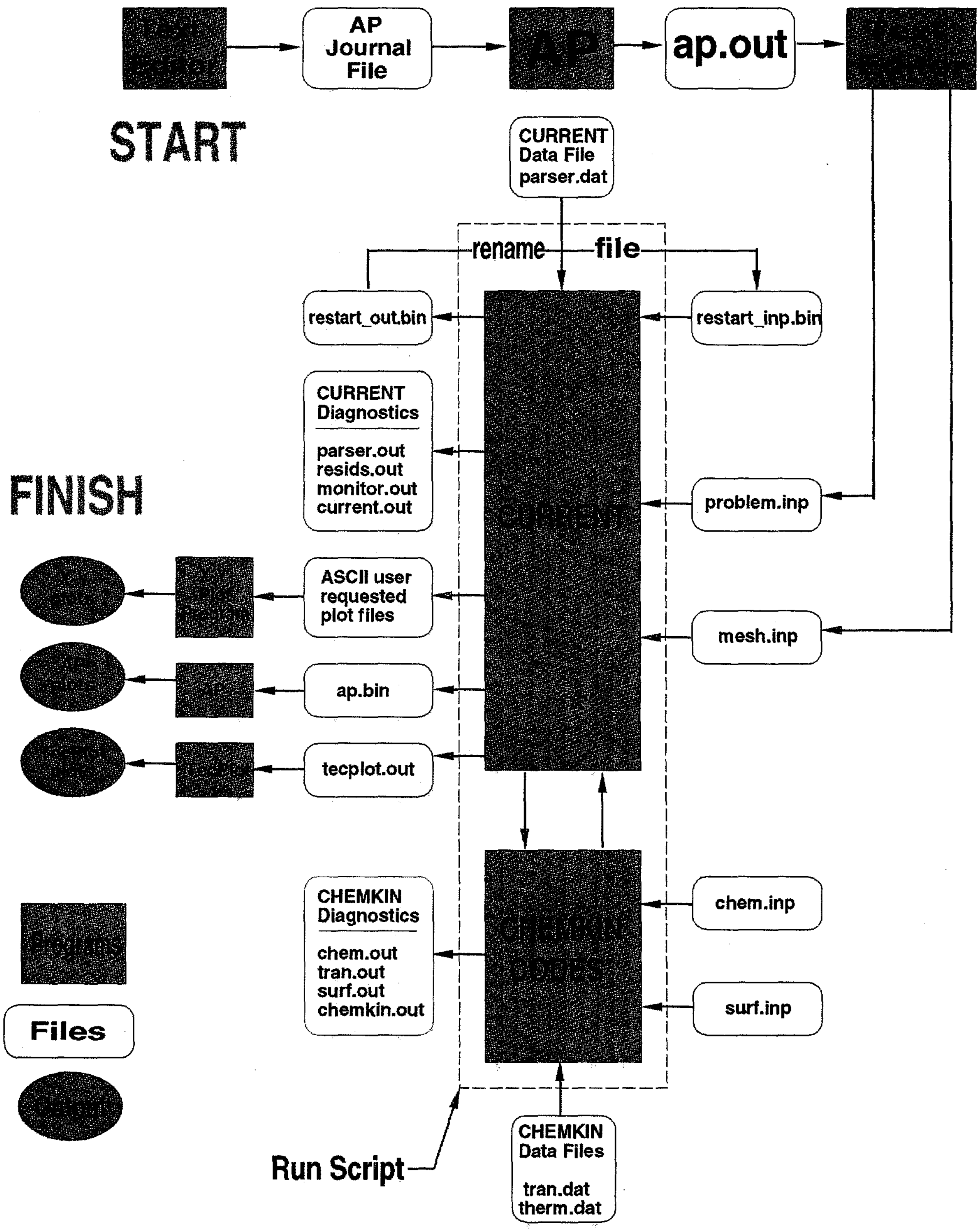


Figure 8: A schematic roadmap for running CURRENT with CHEMKIN.

1. Formulate the Problem.

   A. Obtain all geometry information, dimensions etc. (cgs units).

   B. Determine which equations need to be solved.

   C. Determine the boundary conditions.

   D. Gather up the required chemistry input and data files (cgs units). These are:

      chem.inp - The input file for gas-phase CHEMKIN
      surf.inp - The input file for surface CHEMKIN
      therm.dat - CHEMKIN thermodynamic data for gases present
      tran.dat - CHEMKIN transport data for gases present

2. Generate a meshing journal file which can be read by AP for the purpose of generating the AP output file called "ap.out". The meshing journal file may have any name and must contain AP legal commands for generating a mesh. Most users also include a lengthy set of "pass-through" commands which eventually become the CURRENT input file "problem.inp". These pass-through commands are passed directly through to the ap.out file without being interpreted as meshing commands. The advantage of this is that the AP meshing journal file will contain a complete problem description including instructions for AP mesh generation and the CURRENT problem formulation.

3. Create a mesh. Run AP and have it read the meshing journal file and write the AP output file which will be called "ap.out".

4. Make the CURRENT input files. Use your favorite text editor or a supplied UNIX script to split ap.out into the two required CURRENT input files. These files are:

   mesh.inp - This is an extremely long ASCII file which contains element and nodal connectivity information. Users never need to look at this file.

   problem.inp - This file contains the complete CURRENT problem description and is very "readable" both by the CURRENT input parsing software and the user. This very important file can be created directly from a text editor but as mentioned above, it is better to generate it via "pass-through" commands in the meshing journal file.

5. Run CURRENT/CHEMKIN. Using the input files mesh.inp, problem.inp, chem.inp, and surf.inp together with the data files parser.dat (contains rules for parsing CURRENT input data), therm.dat (CHEMKIN thermal data file), and tran.dat (CHEMKIN transport data file), execute CURRENT and CHEMKIN and produce the desired output plot files. Execution "restarts" will require an additional file called restart_inp.bin. On UNIX workstations code execution is accomplished using a single run script (start.s or restart.s).

6. Post process. Use AP to post-process the CURRENT output file ap.bin. Use your favorite XY plotting software to plot any ASCII data produced directly by CURRENT (e.g. heat flux profiles, etc.) If you wish, you may also process the file tecplot.out using TECPLOT. TECPLOT post-processing will not be discussed here.

5.2 Example Problem Formulation

The geometry for the example problem under consideration is shown in Figure 9. The example problem is representative of a typical CVD process for growing silicon dioxide. Flow enters an axisymmetric reactor through a hole at the top (diameter $2A$). The reactor has a maximum diameter of $2(A+B+D)$ as shown. A spinning CVD growth surface is located a distance $(a+b)$ from the inlet. Flow exits the reactor through an annulus having a length d and inner and outer diameters of $2(A+B)$ and $2(A+B+D)$ respectively. The side wall of the reactor has a slanted section near the top. A slanted wall was used in the example problem to illustrate that grids need not be orthogonal. In fact, the grid boundaries may even be curvilinear. All metrics required to treat non-orthogonal grids are accounted for when solving the equations.

Representation of the real geometry with a computational mesh involves dividing the computational space up into logically (but not necessarily orthogonal) rectangular regions. One can think of each of these regions as having north, south, east and west sides. In this case, the term "logically rectangular" means that the north and south sides have the same number of computational control volumes spanning across them, similarly for the east and west sides. Other requirements for meshing the geometry will become apparent as we proceed.

The reactor pressure for this problem is 1 torr. The temperature of the gas mixture entering through the top is 300 K. The growth substrate is maintained at 1000 K. The top surface (dimension B in Figure 9) is 300 K. The side wall temperature is prescribed as a function of X according to Figure 10 where X is the

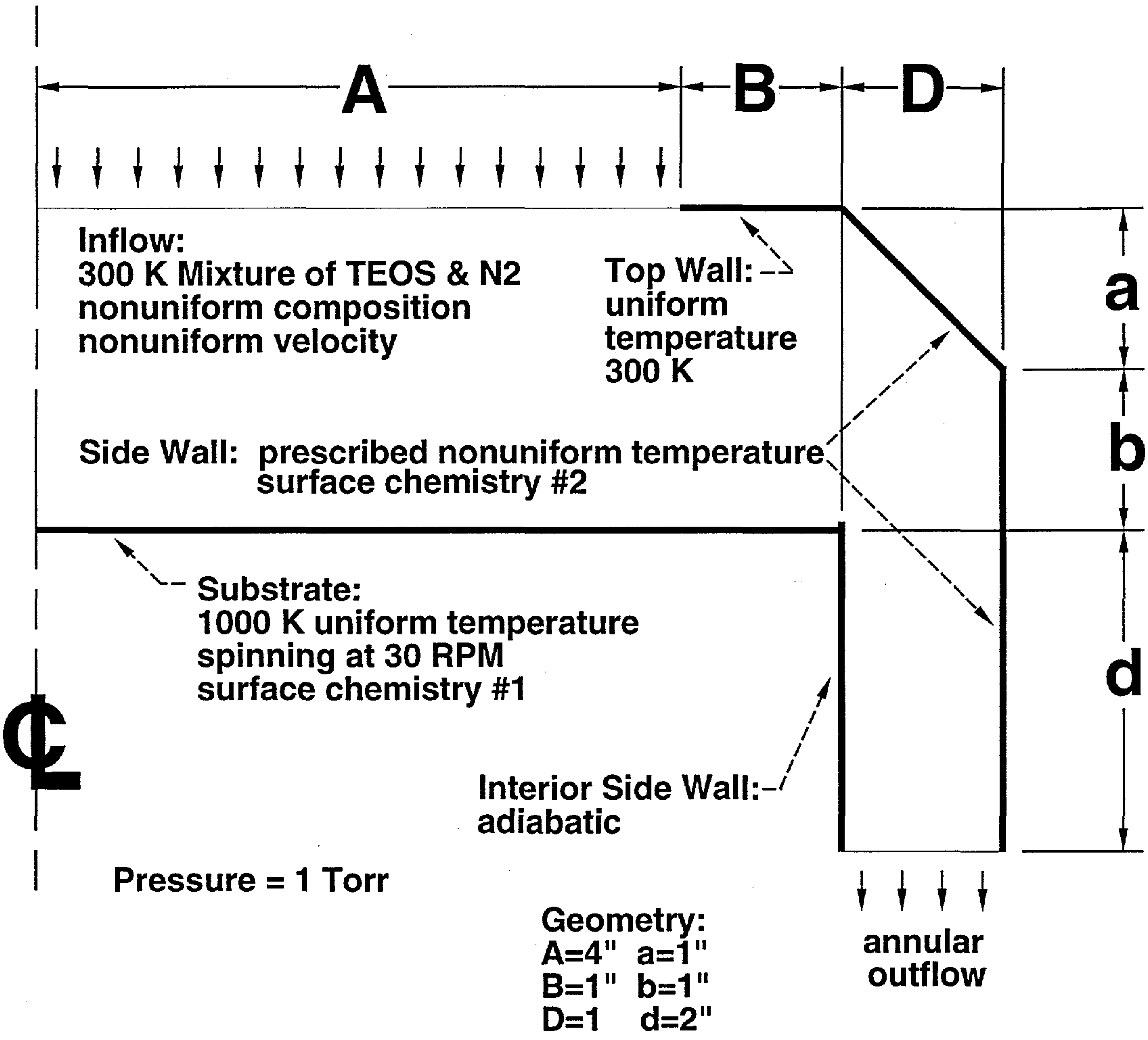


Figure 9: Reactor geometry for the example problem.

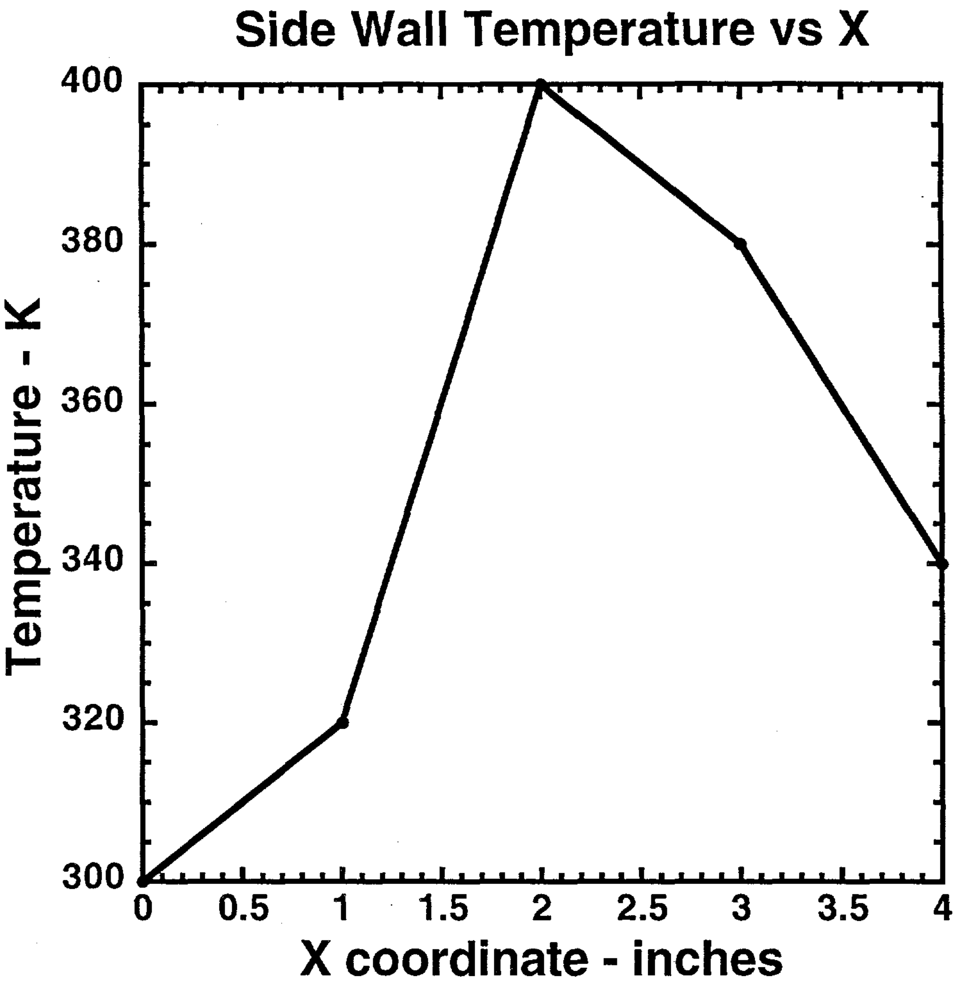


Figure 10: Prescribed side wall temperature profile for example problem.

axis of symmetry for the problem and $X = 0$ corresponds to the inlet. Note that the prescribed side wall temperature is applied to the slanted side wall (dimension a) as well as the straight sections (dimensions b and d). The inner surface of the annular exit channel is assumed to be perfectly insulated (adiabatic). The inlet flow varies in molar composition and velocity according to Figures 11 and 12 respectively. A total of six gas-phase species are present in the problem. Two of these enter as a mixture through the top. The remaining four are formed as a result of gas-phase and surface chemistry.

For the purpose of supplying input to both CURRENT and gas-phase CHEMKIN, we need to designate the species by number. The correspondence between CURRENT gas-phase species numbering and the actual name of the gas-phase species is determined by the order in which these names appear in the *chem.inp* file (See Figure 13).

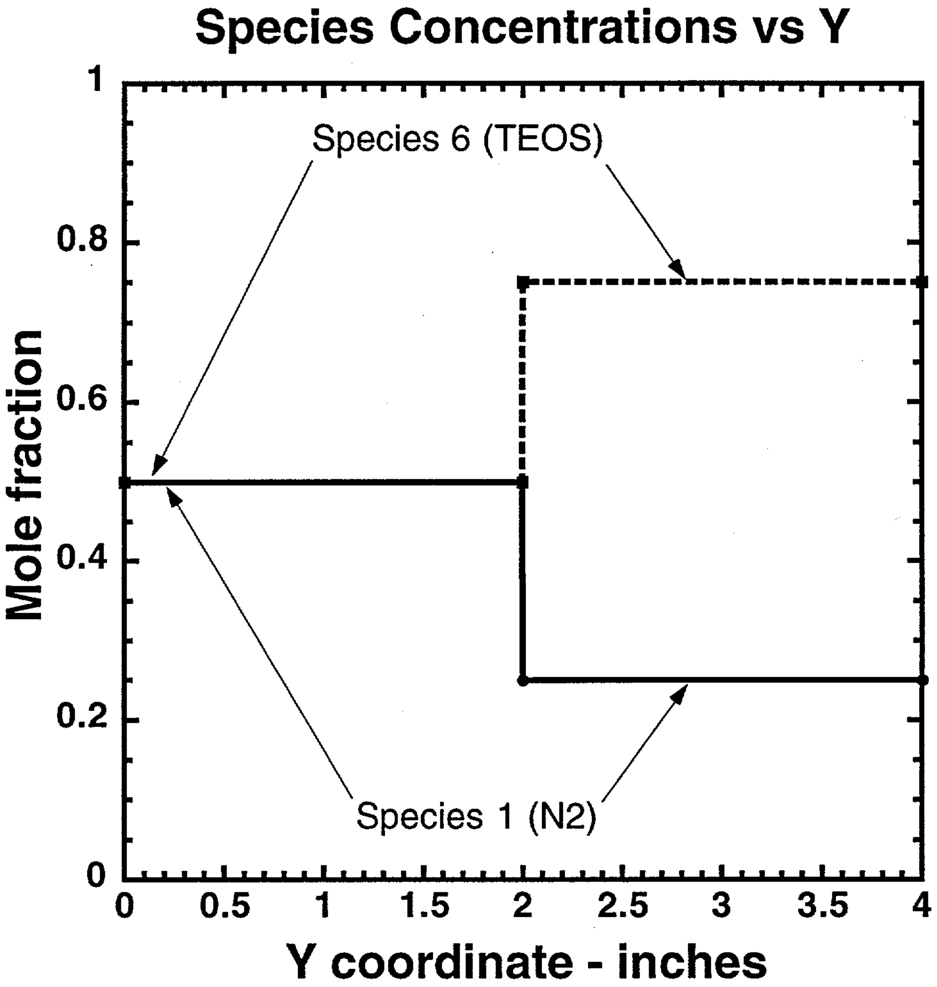


Figure 11: Prescribed inlet composition profile for example problem.

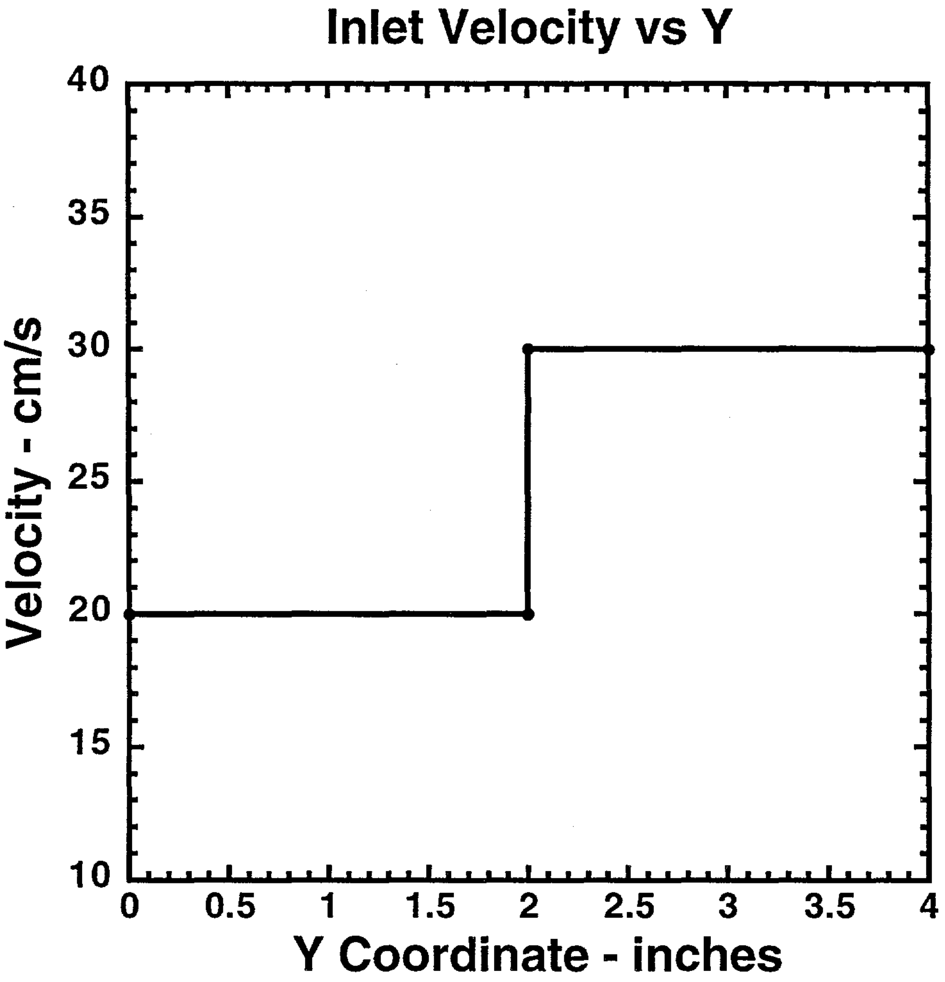


Figure 12: Prescribed inlet velocity profile for example problem.

We chose the following numbering system:

Species No.	Chemical Name
1	N2
2	SI(OH)(OC2H5)3 (we'll call it "intermediate")
3	C2H5OH
4	C2H4
5	H2O
6	SI(OC2H5)4 (Tetraethoxysilane or TEOS)

The numbering of species is arbitrary but it is highly recommended that the last species (in this case No. 6) be a predominant (*i.e.* largest mass fraction) species rather than a trace species. Following this rule will insure more rapid convergence and a more accurate solution.

In our example problem, only species 1 and 6 (Nitrogen and TEOS) enter the reactor. Their mole fractions and the gas mixture velocity vary according to the profiles previously presented in Figures 11 and 12. Note that all mole fractions sum to 1.0 regardless of location. Furthermore, the profiles from Figures 11 and 12 divide the flow into a simple core flow/shroud flow arrangement. The inflow boundary conditions within the core $(0 \leq Y \leq 2)$ and shroud $(2 \leq Y \leq 4)$ are uniform.

Two separate surface chemistry mechanisms are included in the example problem. The first, which consists of 5 species, acts over the top surface of the spinning disk. As with gas-phase chemistry, CURRENT identifies individual surface species by number. The correspondence between CURRENT surface species numbering and the actual name of the surface species is determined by the order in which these names appear in the *surf.inp*. We chose the following order when describing the first surface chemistry mechanism (see "MATERIAL 1" in Figure 14):

Species No	Surface Species Name
1	SIG3(OH)
2	SIG3E
3	SIG(OH)2E
4	SIG(OH)E2
5	SIGE3

The second surface chemistry mechanism, which consists of two surface species, acts along the reactor side wall. We chose the following order when describing the first surface chemistry mechanism (see "MATERIAL 2" in Figure 14):

Species No	Surface Species Name
1	SIG3(OH)
2	SIGE3

The gas-phase and surface chemistry reaction mechanisms were determined from the literature and from independent chemical modeling. The required CHEMKIN input files, *chem.inp* and *surf.inp*, are shown without comment in Figures 13 and 14 respectively.

Occasionally, the chemist will modify the "generic" thermal and transport data files *therm.dat* and *tran.dat* in order to provide the required thermal and transport data for a particular species. We shall assume that this has been done and that all required chemistry input and data files (*chem.inp, surf.inp, therm.dat*, and *tran.dat*) are now available.

We have specified all the boundary conditions, geometry, and have obtained the required chemistry input and data files for the problem. We are now in a position to generate a computational mesh and to prepare the required input files that CURRENT will use to compute the converged steady state solution (gas temperature, velocity, concentration, and surface site fractions).

5.3 Creating the AP Meshing Journal File

A few words on input command syntax are appropriate here. CURRENT input files follow the same syntax rules as AP, or any other code which uses the INTERP [27] and COMPRO [28] software developed at Sandia. Input commands are keyword driven; that is, an input command line contains one or more keywords followed by a delimiter (space, comma, or equal sign), then one or more pieces of data. The keywords are case-insensitive and order-independent on the command line. Abbreviations for most keywords exist to minimize typing. The ordering of all command lines in the *problem.inp* and *mesh.inp* files is completely arbitrary. Select the ordering which makes the most sense to you. The ordering in the example input file presented here is only a suggestion. CURRENT will perform error checking on both command syntax and on the "legality" of the problem formulation.

Figure 15 contains a listing of an AP meshing journal file used to create the computational mesh for the CURRENT example problem. The file contains three types of commands, (1) comments (lines beginning with "!") which are not interpreted in any way by AP, (2) pass through commands (lines beginning with ">") which are also not interpreted by AP but merely "passed through" and printed in the AP output file *ap.out*, and (3) actual AP commands (all lines not beginning with either "!" or ">"). The pass through commands will eventually become the part of the CURRENT input commands appearing in the input file *problem.inp*. We will discuss the precise meaning of the CURRENT input commands (shown in Figure 17 as pass through commands) in Section 5.4.

In the listing shown in Figure 15, upper and lower case AP and CURRENT commands are used to illustrate the minimum number of keyword characters required to uniquely specify a keyword. For example, line 1 could just as easily be replaced by:

```
s ec on
```

and the pass through command spread over lines 41-43 could be replaced by the considerably shorter two lines:

```
ELEMENTS
  SI O H C N
END
SPECIES
  N2
  SI(OH)(OC2H5)3
  C2H5OH
  C2H4
  H2O
   SI(OC2H5)4
END
THERMO
SI(OC2H5)4        40894H  20C   8O   4SI  1G   300.000  3000.000 1000.00    0 1
 0.32309456E+02 0.40201847E-01-0.33278952E-05-0.40219956E-08 0.88291446E-12    2
-0.17431191E+06-0.12745566E+03 0.50599384E+01 0.85404783E-01 0.27717997E-05    3
-0.42832159E-07 0.15639958E-10-0.16494597E+06 0.21772026E+02                   4
SI(OH)(OC2H5)3    40894H  16C   6O   4SI  1G   300.000  3000.000 1000.00    0 1
 0.30722271E+02 0.30770605E-01-0.25774402E-05-0.30498899E-08 0.66850084E-12    2
-0.17340573E+06-0.11836710E+03 0.89422655E+01 0.70058122E-01-0.53697897E-06    3
-0.37873331E-07 0.15943972E-10-0.16629886E+06-0.43492889E+00                   4
SI(OH)2(OC2H5)2   40894H  12C   4O   4SI  1G   300.000  3000.000 1000.00    0 1
 0.26000654E+02 0.22271410E-01-0.18192239E-05-0.21698796E-08 0.47313976E-12    2
-0.17143084E+06-0.98501183E+02 0.76863847E+01 0.56552853E-01-0.14430059E-05    3
-0.32509597E-07 0.14469464E-10-0.16559706E+06 0.15393066E+00                   4
SI(OH)3(OC2H5)    40894H   8C   2O   4SI  1G   300.000  3000.000 1000.00    0 1
 0.19800476E+02 0.14579202E-01-0.11543475E-05-0.13929080E-08 0.29806677E-12    2
-0.16889798E+06-0.68544312E+02 0.75322161E+01 0.39030805E-01-0.22118543E-05    3
-0.23559673E-07 0.11338996E-10-0.16516952E+06-0.30923195E+01                   4
SI(OH)4           40894H   4O   4Si 1   0G   300.000  3000.000 1000.00    0 1
 0.14569509E+02 0.49252058E-02-0.34521807E-06-0.40573092E-09 0.83636453E-13    2
-0.16601905E+06-0.42529125E+02 0.85398741E+01 0.21113478E-01-0.45791253E-05    3
-0.16384208E-07 0.10137381E-10-0.16468839E+06-0.12136473E+02                   4
O=SI(OC2H5)2      40894H  10C   4O   3SI  1G   300.000  3000.000 1000.00    0 1
 0.20602577E+02 0.20795811E-01-0.18450947E-05-0.20959716E-08 0.46654770E-12    2
-0.10564674E+06-0.71234840E+02 0.43081031E+01 0.49828056E-01-0.14389319E-06    3
-0.27417245E-07 0.11348846E-10-0.10028161E+06 0.17160416E+02                   4
END
REACTIONS
  SI(OC2H5)4 = SI(OH)(OC2H5)3 + C2H4             4.9E13   0.0   61460.
END
```

Figure 13: chem.inp - CHEMKIN input file.

```
MATERIAL 1

SITE/SIO2/      SDEN/0.750E-9/
  SIG3(OH)    ! Hf = -191.89
  SIG3E       ! Hf = -191.41
  SIG(OH)2E   ! Hf = -241.73
  SIG(OH)E2   ! Hf = -241.25
  SIGE3       ! Hf = -240.77
END
BULK/Glass/
  SIO2(D)/2.19/
END
BULK/Fictitious/
    O(D)/2.19/      ! Hf = -97.48 kcal mole-1
END
THERMO
 300.   600.   10000.
SIG3(OH)        121591O  2SI 1H  1   I  300.00  3000.00 1000.00    1
 0.66466584E+01 0.33231564E-02-0.29541198E-06-0.31399386E-09 0.69825405E-13  2
-0.98982922E+05-0.33869411E+02 0.26748490E+01 0.12014943E-01-0.13939117E-05  3
-0.83051193E-08 0.44394740E-11-0.97866992E+05-0.13004364E+02                 4
SIG3E           121591C  2H  5O  2SI 1I  300.00  3000.00 1000.00    1
 0.12659941E+02 0.11178698E-01-0.96102099E-06-0.11064012E-08 0.24469427E-12  2
-0.10153567E+06-0.64059296E+02 0.34100885E+01 0.28247360E-01-0.56295153E-06  3
-0.16109125E-07 0.70305402E-11-0.98559328E+05-0.14126812E+02                 4
SIG2(OH)2       121591O  3SI 1H  2   I  300.00  3000.00 1000.00    1
 0.87919779E+01 0.53339764E-02-0.41594888E-06-0.49502752E-09 0.10635092E-12  2
-0.13692208E+06-0.44418247E+02 0.47610464E+01 0.13908056E-01-0.12391115E-05  3
-0.84472296E-08 0.43385655E-11-0.13576220E+06-0.23143377E+02                 4
SIG2(OH)E       121591C  2H  6O  3SI 1I  300.00  3000.00 1000.00    1
 0.14530663E+02 0.13423699E-01-0.11037475E-05-0.13165601E-08 0.28805297E-12  2
-0.13945747E+06-0.73503326E+02 0.44301777E+01 0.32143697E-01-0.64655001E-06  3
-0.17989029E-07 0.78838108E-11-0.13622044E+06-0.19021687E+02                 4
SIG2E2          121591C  4H  10O  3SI 1I  300.00  3000.00 1000.00    1
 0.20543951E+02 0.21279231E-01-0.17693552E-05-0.21089661E-08 0.46292158E-12  2
-0.14201020E+06-0.10438638E+03 0.51654549E+01 0.48375994E-01 0.18445488E-06  3
-0.25792907E-07 0.10474785E-10-0.13691275E+06-0.20837475E+02                 4
SIG(OH)2E       62692C  2H  7O  3SI 1I  300.00  3000.00 1000.00    1
 0.13577039E+02 0.14641373E-01-0.11149998E-05-0.14189627E-08 0.30468458E-12  2
-0.12723145E+06-0.61076580E+02 0.57280946E+01 0.27964467E-01 0.50749497E-06  3
-0.13237452E-07 0.50265321E-11-0.12457464E+06-0.18233299E+02                 4
SIG(OH)E2       62692C  4H  11O  3SI 1I  300.00  3000.00 1000.00    1
 0.19473230E+02 0.22570338E-01-0.17838936E-05-0.22180249E-08 0.48075975E-12  2
-0.12976159E+06-0.90081039E+02 0.66617913E+01 0.43213133E-01 0.17241006E-05  3
-0.19916659E-07 0.68400398E-11-0.12528808E+06-0.19669796E+02                 4
SIGE3           62692C  6H  15O  3SI 1I  300.00  3000.00 1000.00    1
 0.25609777E+02 0.30346680E-01-0.24455890E-05-0.30030116E-08 0.65423480E-12  2
-0.13233747E+06-0.11858306E+03 0.72320976E+01 0.60353167E-01 0.21851304E-05  3
-0.28773197E-07 0.10164894E-10-0.12596620E+06-0.17743988E+02                 4
SIO2(D)         72391SI 1O  2       S  298.00  2000.00 1000.00    1
 0.48925619E+01 0.41191629E-02-0.94570083E-07-0.80073115E-09 0.25433412E-12  2
-0.11005530E+06-0.23469570E+02 0.22325585E+01 0.12478522E-01-0.28715690E-05  3
-0.96847970E-08 0.62160411E-11-0.10962063E+06-0.10594849E+02                 4
O(D)            121591O  1          S  300.00  3000.00 1000.00    1
 0.23625109E+01 0.57858101E-03-0.82644831E-07-0.58489401E-10 0.15022128E-13  2
-0.49901734E+05-0.13022270E+02 0.31750876E+00 0.54401071E-02-0.10133198E-05  3
-0.46017092E-08 0.26723929E-11-0.49372941E+05-0.24420185E+01                 4
END
REACTIONS
 SI(OC2H5)4 + SIG3(OH) => SIO2(D) + SIGE3 + C2H5OH        2.5E4   0.   44600.
                        STICK
SIG3E    <=> SIG3(OH) + C2H4           &
        1.2E+12   0.   47000.
SIG(OH)E2 <=> SIG(OH)2E + C2H4          &
        2.4E+12  0.   47000.
SIGE3    <=> SIG(OH)E2 + C2H4           &
        3.6E+12  0.   47000.
SIG(OH)2E <=> SIG3(OH) + C2H5OH          &
        1.4e+12 0. 44000.
SIG(OH)E2 <=> SIG3E + C2H5OH            &
        1.4e+12 0. 44000.
SIG(OH)2E <=> SIG3E+H2O                   &
        140e+10 0. 45000.
SI(OH)(OC2H5)3 + SIG3(OH) = SIO2(D) + SIGE3 + H2O       7.0   0.   12000.
                            STICK
END

MATERIAL 2

SITE/SIO2/      SDEN/0.750E-9/
  SIG3(OH)    ! Hf = -191.89
  SIGE3       ! Hf = -240.77
END
BULK/Glass/
  SIO2(D)/2.19/
END
THERMO
 300.   600.   10000.
SIG3(OH)        121591O  2SI 1H  1   I  300.00  3000.00 1000.00    1
 0.66466584E+01 0.33231564E-02-0.29541198E-06-0.31399386E-09 0.69825405E-13  2
-0.98982922E+05-0.33869411E+02 0.26748490E+01 0.12014943E-01-0.13939117E-05  3
-0.83051193E-08 0.44394740E-11-0.97866992E+05-0.13004364E+02                 4
SIGE3           62692C  6H  15O  3SI 1I  300.00  3000.00 1000.00    1
 0.25609777E+02 0.30346680E-01-0.24455890E-05-0.30030116E-08 0.65423480E-12  2
-0.13233747E+06-0.11858306E+03 0.72320976E+01 0.60353167E-01 0.21851304E-05  3
-0.28773197E-07 0.10164894E-10-0.12596620E+06-0.17743988E+02                 4
SIO2(D)         72391SI 1O  2       S  298.00  2000.00 1000.00    1
 0.48925619E+01 0.41191629E-02-0.94570083E-07-0.80073115E-09 0.25433412E-12  2
-0.11005530E+06-0.23469570E+02 0.22325585E+01 0.12478522E-01-0.28715690E-05  3
-0.96847970E-08 0.62160411E-11-0.10962063E+06-0.10594849E+02                 4
END
REACTIONS
 SI(OC2H5)4 + SIG3(OH) => SIO2(D) + SIGE3 + C2H5OH        2.5E4   0.   44600.
                        STICK
END
```

Figure 14: surf.inp - SURFACE CHEMKIN input file.

```
1  Set ECho ON
2
3  ! AP, CURRENT, and CHEMKIN parsing software all ignore any characters typed
4  ! after an explanation point.
5  ! Hence these lines are comment lines. The above AP command will echo
6  ! every line following it until "Set ECho OFF" is encountered.
7  ! The lines below which start with ">" are "pass through" commands. They
8  ! are ignored by AP but typed directly into the AP output text file which
9  ! for this example will be named "ap.out"
10
11 >TITle "CURRENT Example Problem"
12 >Dump medium
13 >SPECies Gas 6 Surface 5, 2
14 >LAST_iteration 100
15 >PRINT_frequency=100
16 >
17 >SWITCH OFF Restart
18 >SWITCH ON  MASS_adjustment
19 >SWITCH ON  Hybrid
20 >SWITCH ON  Chemistry
21 >SWITCH ON  U_velocity V_velocity W_velocity P P_Correction
22 >SWITCH ON  Temperature Species
23 >SWITCH OFF THERMAL_diffusion
24 >SWITCH ON  MULTicomponent_diffusion
25 >SWITCH OFF PLANar_geometry
26 >SWITCH ON NET_flow
27 >SWITCH OFF Dump
28 >
29 >SWITCH ON GMRES_P_Correction
30 >SWITCH ON GMRES_Species
31 >SWITCH OFF GMRES_Temperature
32 >
33 >INITial U_velocity=0.0 V_velocity=0.0 W_velocity=0.0  P=1333. !&
34 >      Temperature=300. Species=.75,.0,.0,.0,.0,.25
35 >
36 >RELaxation U_velocity=0.5 V_velocity=0.5 W_velocity=0.5 !&
37 >        P=1.0 P_Correction=1.0 Temperature=0.5  Species=.5,.5,.5,.5,.5,.5
38 >
39 >ANGLE -1.5708
40 >
41 >SURface Name=substrate ISOthermal=on Temperature=1000.  ROTation=on !&
42 >      Omega=3.142 REAction=ON  MATerial=1 !&
43 >      Site_fractions=.96541152 2.1666e-2,7.8848e-4,3.4648e-3, 8.6692e-3
44 >
45 >SURface Name=top_wall  ISOthermal=ON Temperature=300.
46 >
47 >SURface Name=side_wall ISOthermal=ON Temperature=1.0 TFunction=side_wall !&
48 >      REAction=OFF MATerial=2 Site_fractions=0.9,0.1
49 >
50 >SURFace Name=in_side_wall ISOthermal=OFF
51 >
52 >INflow Name=screen U_velocity=1.0 UFunction=screen_u !&
53 >     V_velocity=0.0 W_velocity=0.0 Temperature=300. !&
54 >     Species=1.,0.,0.,0.,0.,1. SFunction=screen_s1,f0,f0,f0,f0,screen_s6
55 >
56 >MATerial Region 1  Name CHEMKIN
57 >MATerial Region 2  Name CHEMKIN
58 >MATerial Region 3  Name CHEMKIN
59 >
60 >CONNections Region 1  North  2  South  0  East  0  West  0
61 >CONNections Region 2  North  0  South  1  East  3  West  0
62 >CONNections Region 3  North  0  South  0  East  0  West  2
63 >
64 >CONDitions Region 1  South=SYMmetry East=SURface substrate West=INFlow screen
65 >CONDitions Region 2  North=SURface side_wall West=SURface top_wall
66 >CONDitions Region 3  North=SURface side_wall South=SURface in_side_wall !&
67 >              East=OUTflow
68 >
69 >SEGment Number 1  Region 1  Side South  Origin West
70 >SEGment Number 2  Region 1  Side East   Origin South
71 >SEGment Number 3  Region 3  Side South  Origin West
72 >SEGment Number 4  Region 3  Side East   Origin South
73 >SEGment Number 5  Region 3  Side North  Origin East
74 >SEGment Number 6  Region 2  Side North  Origin East
75 >SEGment Number 7  Region 2  Side West   Origin North
76 >SEGment Number 8  Region 1  Side West   Origin North
77 >
78 >STREAMline Pathname outside
79 >
80 >PATH Name outside Segments 1:8
81 >PATH Name substrate Segment 2
82 >
83 >PLOT PATH substrate Variable h_flux
84 >PLOT PATH substrate Variable m_flux
85 >PLOT PATH substrate Variable s2_flux
86 >PLOT PATH substrate Variable s6_flux
87 >PLOT PATH substrate Variable s1_site
88 >PLOT PATH substrate Variable s2_site
89 >PLOT PATH substrate Variable s3_site
90 >PLOT PATH substrate Variable s4_site
91 >PLOT PATH substrate Variable s5_site
92 >
93 >Function Name side_wall XYpair 0.0       300.
94 >Function Name side_wall XYpair 2.54      320
95 >Function Name side_wall XYpair 5.08      400.0
96 >Function Name side_wall XYpair 7.62      380
97 >Function Name side_wall XYpair 10.16      340.
98 >
99 >Function Name screen_s1  XYpair  0.0      .5
100 >Function Name screen_s1  XYpair  5.07999  .5
101 >Function Name screen_s1  XYpair  5.08001  .25
102 >Function Name screen_s1  XYpair  10.16    .25
103 >
104 >Function Name screen_s6  XYpair  0.0      .5
105 >Function Name screen_s6  XYpair  5.07999  .5
106 >Function Name screen_s6  XYpair  5.08001  .75
107 >Function Name screen_s6  XYpair  10.16    .75
108 >
109 >Function Name f0  XYpair  0.0  0.0
110 >Function Name f0  XYpair  1.0  0.0
111 >
112 >Function Name screen_u  XYpair  0.0      20.
113 >Function Name screen_u  XYpair  5.07999  20.
114 >Function Name screen_u  XYpair  5.08001  30.
115 >Function Name screen_u  XYpair  10.16    30.
116
117 ! Actual ap mesh generation commands now follow.
118
119 MAcro Define Name "scf"  Definition "2.54"  ! convert from inch to cm
120
121 MAcro Define Name "xa"   Definition "1.0"
122 MAcro Define Name "xb"   Definition "1.0"
123 MAcro Define Name "xd"   Definition "2.0"
124
125 MAcro Define Name "ya"   Definition "4.0"
126 MAcro Define Name "yb"   Definition "1.0"
127 MAcro Define Name "yd"   Definition "1.0"
128
129 POint Cartesian  1  X 0.0       Y 0.0
130 POint Cartesian  2  X {xa+xb}   Y 0.0
131 POint Cartesian  3  X 0.0       Y ya
132 POint Cartesian  4  X {xa+xb}   Y {ya+yb}
133 POint Cartesian  5  X {xa+xb+xd}  Y {ya+yb}
134 POint Cartesian  6  X 0.0       Y {ya+yb}
135 POint Cartesian  7  X xa        Y {ya+yb+yd}
136 POint Cartesian  8  X {xa+xb}   Y {ya+yb+yd}
137 POint Cartesian  9  X {xa+xb+xd}  Y {ya+yb+yd}
138
139 Curve Straight   1  Points 1,2
140 Curve Kink       2  Points 3,4,5
141 Curve Kink       3  Points 6,7,8,9
142 Curve Straight   4  Points 1,6
143 Curve Straight   5  Points 2,8
144 Curve Straight   6  Points 5,9
145
146 GRAphics New
147
148 Region 1  CURves 1,5,2,4
149 Region 2  CURves 2,5,3,4
150 Region 3  CURves 2,6,3,5
151
152 GRAphics New
153
154 Set DEVice 0
155
156 Mesh Quadrilateral 1  D1 20 D2 40  R1 .5  R3 2  R2 .5 R4 2
157 Mesh Quadrilateral 2  D1 20 D2 15  R1 .5  R3 2
158 Mesh Quadrilateral 3  D1 20 D2 15  R1  2  R3 .5
159
160 GRAphics New
161
162 MERge Regions all
163
164 RENumber Nodes COMpress
165 RENumber Nodes Profile
166
167 Zoom Find Nodes
168
169 Set DEVice 15
170
171 Group Elements Regions Name "1"  Region 1 Append
172 Group Elements Regions Name "2"  Region 2 Append
173 Group Elements Regions Name "3"  Region 3 Append
174
175 Write Sfel Scalefactor scf
176
177 GRAphics New
178
179 Plot Boundary all
180
181 Plot Elements all
182
183 ! These comment lines conclude the meshing journal file.  Execution
184 ! of the journal file will now stop and an ap prompt will be issued.
185 ! The user may now type additional ap commands interactively to display
186 ! more detail regarding the mesh.
```

Figure 15: AP journal file for the example problem.

```
sur n=substrate iso=on t=1000 r=on o=3.142 rea=on &
s=.96541152,2.1666e-2,7.8848e-4,3.4648e-3,8.6692e-3
```

In AP and CURRENT input files, the "&" character can be used at the end of a line to indicate to the command parsing software that the command is continued on the next line. Hence when lines 41-43 are passed through by AP to *ap.out*, they will be converted to one very long line which exceeds 80 characters. Since the maximum line length that the parsing software can accommodate is 80 characters, the long line will have to be split up into shorter lines using the "&" character before it is placed in the *problem.inp* file. An alternative method for dealing with long lines is to make use of the comment character "!" in the journal file as we did for the example (see Figure 15). Once the pass through commands are processed by AP, we need only replace all occurrences of the string "!&" with "&".

Having made the above observations, we are now in a position to describe the step by step process of specifying the AP meshing commands necessary to generate the computational mesh for the example problem. Throughout this discussion, some knowledge of AP meshing commands and techniques will be assumed. Users unfamiliar with AP may consult Reference [5] for details.

We begin by preparing a simple mesh generation diagram similar to the one shown in Figure 16. Note that the orientation of the reactor has been changed so that the axis of symmetry now lies along the X axis. This is how the example problem mesh will be displayed on the screen during AP pre- and post-processing. Note the location of the $X-Y$ origin. The origin is selected by the user and is completely arbitrary except in the case of axisymmetric problems where $Y = 0$ must correspond to the axis of symmetry. Notice the orientation of the "compass" in Figure 16. The compass is used to refer to various sides of computational regions and must be oriented with respect to the $X-Y$ coordinate system in the manner shown in Figure 16.

The building of a mesh is really quite simple and can be summarized by the following steps:

1. Define a set of points (P1,P2,...,P9).

2. Define a set of curves connecting the points (C1,C2,...,C6).

3. Define rectangular regions, each of which is bounded by four different curves (R1,R2,R3).

4. Mesh each rectangular region into elements so that the number of elements along the north side of the region is equal to the number along the south side. Similarly the number of element along the east side of the region must be equal to the number along the west side. In the finite difference world we refer to such meshing as being logically rectangular.

5. Ensure that when regions share the same internal boundary that the corners of the regions as well as the corners of all elements along the internal boundary align exactly. In the finite element world we would say that all the nodes along the sides of the two regions sharing the same internal flow boundary coincide exactly.

6. Merge all regions and eliminate duplicate node numbering along internal boundaries.

7. Renumber nodes to minimize the nodal connectivity bandwidth.

8. Place all the elements within a particular region in a grouped list and give that list a name which is identical to the region number.

9. Instruct AP to write the ap.out file which contains pass through commands and an ASCII description of the mesh which can be directly read by CURRENT.

The actual AP meshing commands for the example problem begin on line 119 of Figure 15. Lines 119 through 127 define "macros" for important key dimensions and for the conversion factor for used to convert inches to centimeters when writing the *ap.out* file. CURRENT expects all units in the input files to be CGS (centimeters-grams-seconds).

Lines 129 through 137 define the nine points in the mesh diagram (Figure 15). Note that numerous macro definitions have been used to specify the $X - Y$ coordinates of the points. This makes changing key dimensions in the reactor model quite simple since only one command (the macro command) need be changed to change a dimension.

Lines 139 through 144 define the six curves we will need to enclose our three regions. Note that curves need not be straight lines. Curves 2 and 3 are kinked curves which are actually composed of multiple straight lines joined together. AP also provides for arcs, semicircles, etc. for curve definitions [5].

Line 146 erases the screen.

Lines 148 through lines 150 define the three computational regions. To define a region we merely give it a unique number and designate the four curves which form its bounding sides. The ordering of the curves is important! CURRENT expects the curves to be ordered as follows:

```
south side, east side, north side, west side.
```

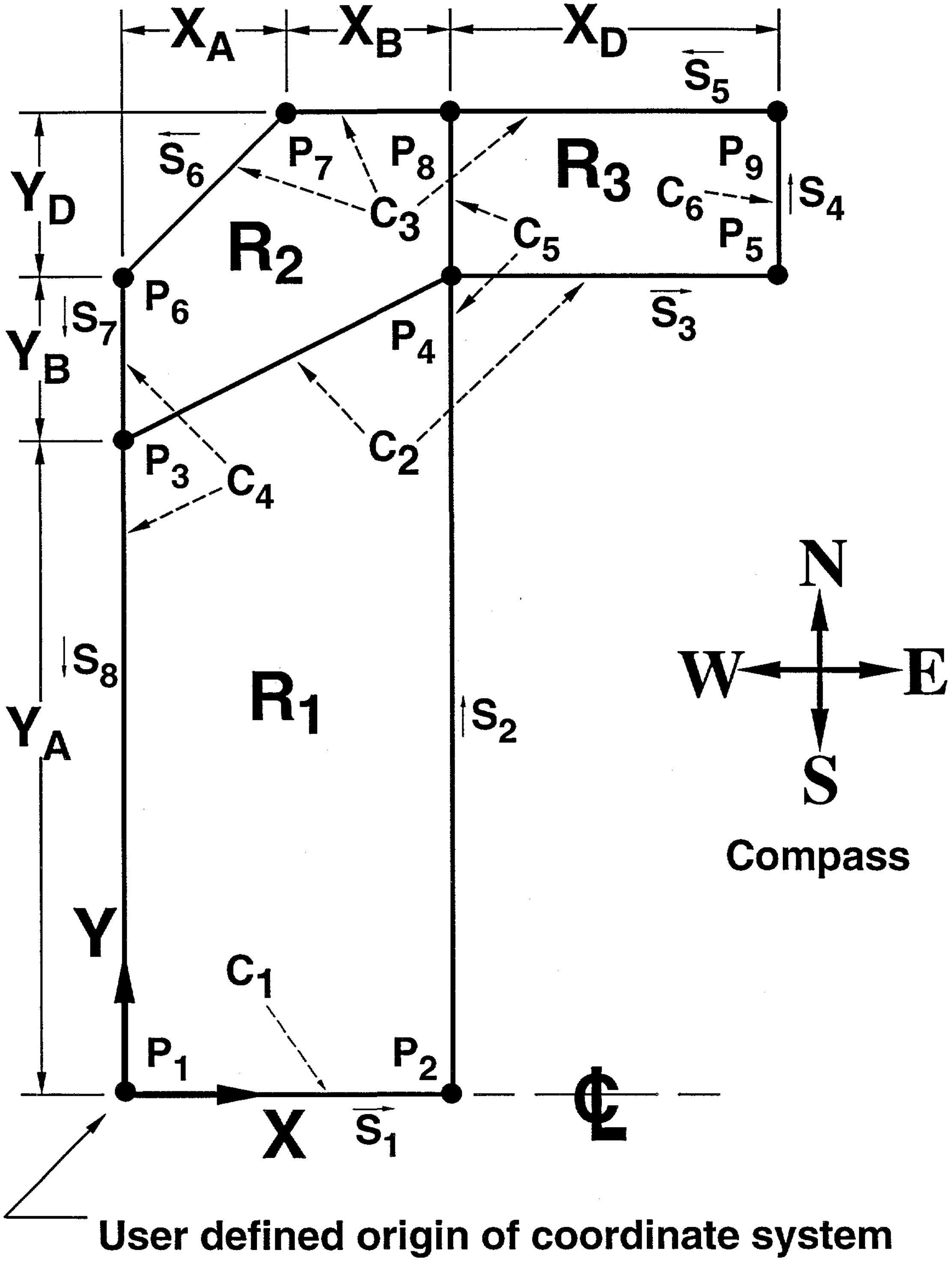


Figure 16: Mesh generation diagram for the example problem.

Note that the regions, while being logically rectangular, do not necessarily need to have parallel sides, orthogonal corners, or even straight sides. This is an extremely powerful feature which enables CURRENT to model virtually any shaped flow region. Note, for example, that the south side of region 2 is not orthogonal to the coordinate system or to the east and west sides. Note also that the north side of region 2 isn't even a straight line. Instead it is composed of line segments connecting points 6,7, and 8 (a portion of curve 3).

Line 152 clears the screen again and line 154 sets the display to the null device. These lines can be left out but leaving them in will speed execution since we won't have to wait for AP to draw each and every element to the screen when we mesh each region.

Lines 156 to 158 are used to mesh the three regions using AP's mesh quadrilateral command. The mesh quadrilateral command is the only acceptable method for meshing a region for CURRENT. To resolve the thin boundary layer near the substrate we have chosen to use the ratio parameters in the mesh quadrilateral command. (See Reference [5] for details.) Note also that the discretization specified by lines 156 to 158 insures that the requirements listed in steps 4 and 5 above are adhered to.

Line 160 erases the screen again.

Line 162 merges the three regions together while lines 164 and 165 compress the nodes (replaces two coincident nodes with a single node), renumber the nodes and minimize the bandwidth. These three lines must appear in all meshing journal files.

Line 175 resizes the display to encompass the entire mesh.

Line 169 sets the display device (in this case to DIGLIB [29] display device 15 which is an X window device) so that we can now see the mesh we have created.

Lines 171 through 173 group all the elements in each region into groups having names "1", "2" , and "3" (step 8 above).

Line 167 writes the AP output file *ap.out* which we will use next to create the two CURRENT input files *mesh.inp* and *problem.inp.* Note that when this operation is performed AP will send a message to the screen warning you that some regions have not been assigned a "material number". You can ignore this message.

Line 179 will display all unmerged boundaries. You should see no lines drawn inside the flow domain. If you do see lines you have violated the requirements listed in steps 4 and/or 5 above.

Line 181 will display all the elements in the mesh. Each element corresponds to a single control volume.

Since AP is an interactive mesh generator/post-processor, we could type each of the commands listed in Figure 15 one by one after each prompt. However, because of the large number of pass-through and meshing commands involved, it would be wiser to create a journal file with a text editor and invoke AP to read it [5].

The meshing process is now complete. If you wish, you can remain in AP and continue to type interactive commands to "zoom in" and examine the mesh more closely.

5.4 Preparing the CURRENT Input Files

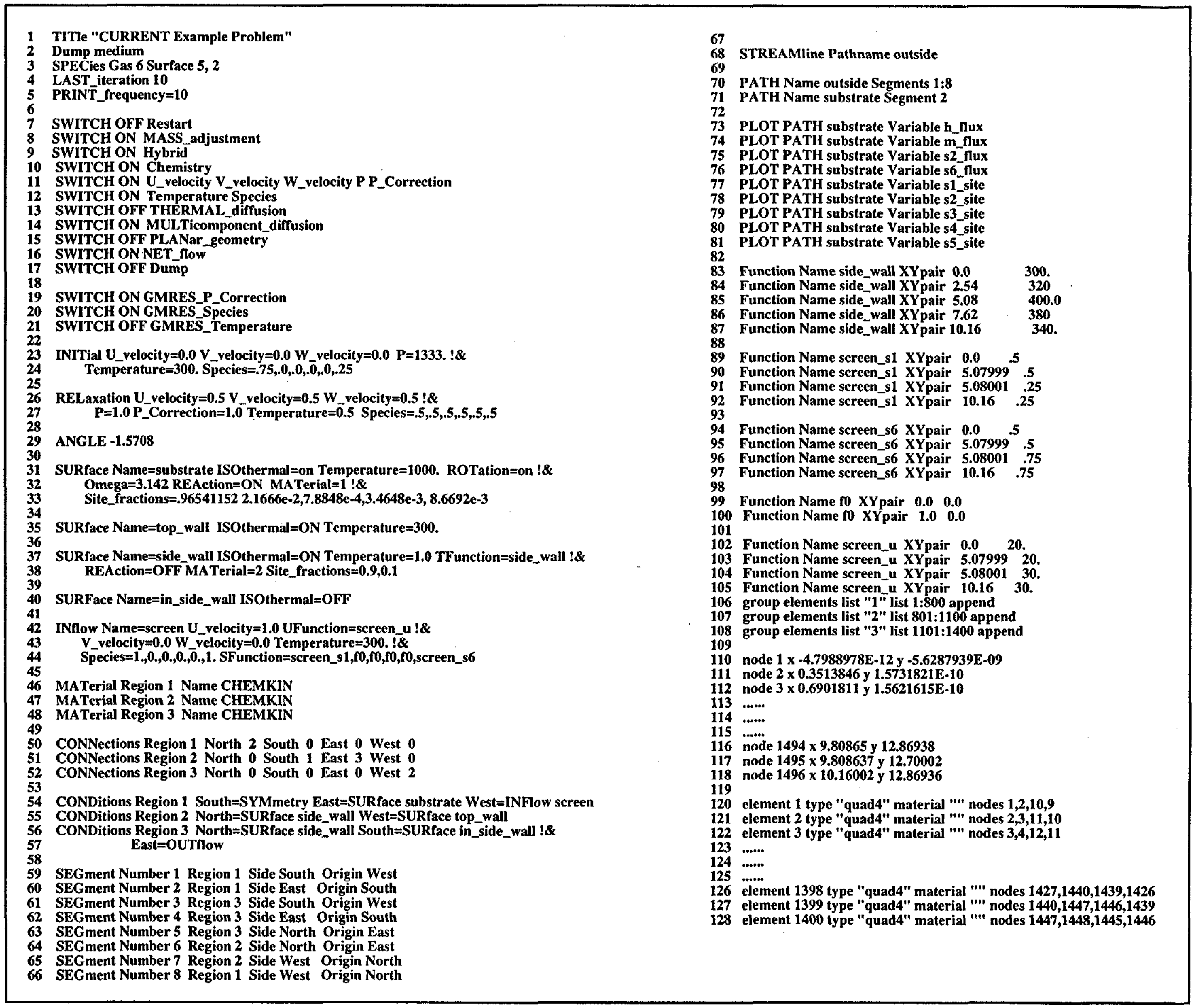
In this section, we discuss the preparation of two important CURRENT input files, namely the *mesh.inp* file and the *problem.inp* file. As previously mentioned, these two input files together with the chemistry input files *chem.inp* and *surf.inp* and the data files *therm.dat, tran.dat* and *parser.dat* are all that is required to run CURRENT with CHEMKIN; the file *restart_inp.bin* is also required for restarts. Actually we have already prepared *mesh.inp* and *problem.inp* in the form of the AP-generated output file *ap.out.* The only step remaining is to divide the *ap.out* file into *mesh.inp* and *problem.inp* using whatever means necessary (a text editor, a UNIX script, etc.).

A listing of the *ap.out* file is shown in Figure 17. Lines 1 through 108 contain all the commands necessary for the *problem.inp* file, and the remaining lines contain the lines required for the *mesh.inp* file. Except for lines 106 through 108, all lines in the *problem.inp* file originated as "pass-through" commands. These lines could have just as easily been generated separately from any text editor capable of producing an ASCII text file. As mentioned previously, we chose to embed the lines into the meshing journal file so that one file (the journal file) would hold a complete description of the problem. Lines 106 through 108 (one line for each region in the example problem) were generated by AP when it wrote the *ap.out* file. These lines list all the elements (control volumes) in each of the regions. This information is unimportant to the CURRENT user but very important to the CURRENT program; together with the *problem.inp* file, it completely describes how the finite element data generated by AP is to be converted into the finite volume data base required by CURRENT. Hence all *problem.inp* files must contain one of these "group elements" commands for each computational region in the problem. Fortunately the user can rely on AP to generate these lines automatically. (See lines 171 through 173 in Figure 15.)

In order to save space, many lines from what will become the *mesh.inp* file have been removed from the *ap.out* listing shown in Figure 17. There are in fact 2,896 lines that have to be embedded in the *mesh.inp* file: 1,496 node commands and 1,400 element commands. These commands all have syntax like that shown in lines 110 through 128. Typically the *mesh.inp* files for CURRENT are enormous compared to other input files. Once generated, the user never need be concerned about the *mesh.inp* file unless the problem is remeshed (physical dimensions changed or alternate discretizations used). Just keep the file around during execution of CURRENT and never look at it again!

The remaining discussion in this section will focus on the CURRENT input commands which appear in the *problem.inp* file. These commands (lines 1 through 108) completely describe the setup of the CURRENT example problem. A more complete "reference" for CURRENT input commands is presented in Appendix A.

The example problem presented here was designed to demonstrate most of the features available in CURRENT as of this writing. It is hoped that knowledge gained by studying this simple example can be extrapolated to larger and more comprehensive problems. Most problems can be initiated using the example journal file as a "template."



```
TITle "CURRENT Example Problem"
Dump medium
SPECies Gas 6 Surface 5, 2
LAST_iteration 10
PRINT_frequency=10

SWITCH OFF Restart
SWITCH ON  MASS_adjustment
SWITCH ON  Hybrid
SWITCH ON  Chemistry
SWITCH ON  U_velocity V_velocity W_velocity P P_Correction
SWITCH ON  Temperature Species
SWITCH OFF THERMAL_diffusion
SWITCH ON  MULTIcomponent_diffusion
SWITCH OFF PLANar_geometry
SWITCH ON NET_flow
SWITCH OFF Dump

SWITCH ON GMRES_P_Correction
SWITCH ON GMRES_Species
SWITCH OFF GMRES_Temperature

INITial U_velocity=0.0 V_velocity=0.0 W_velocity=0.0  P=1333. !&
      Temperature=300. Species=.75,.0,.0,.0,.0,.25

RELaxation U_velocity=0.5 V_velocity=0.5 W_velocity=0.5 !&
        P=1.0 P_Correction=1.0 Temperature=0.5  Species=.5,.5,.5,.5,.5,.5

ANGLE -1.5708

SURface Name=substrate ISOthermal=on Temperature=1000.  ROTation=on !&
      Omega=3.142 REAction=ON  MATerial=1 !&
      Site_fractions=.96541152 2.1666e-2,7.8848e-4,3.4648e-3, 8.6692e-3

SURface Name=top_wall  ISOthermal=ON Temperature=300.

SURface Name=side_wall ISOthermal=ON Temperature=1.0 TFunction=side_wall !&
      REAction=OFF MATerial=2 Site_fractions=0.9,0.1

SURFace Name=in_side_wall ISOthermal=OFF

INflow Name=screen U_velocity=1.0 UFunction=screen_u !&
     V_velocity=0.0 W_velocity=0.0 Temperature=300. !&
     Species=1.,0.,0.,0.,0.,1. SFunction=screen_s1,f0,f0,f0,f0,screen_s6

MATerial Region 1  Name CHEMKIN
MATerial Region 2  Name CHEMKIN
MATerial Region 3  Name CHEMKIN

CONNections Region 1  North  2  South  0  East  0  West  0
CONNections Region 2  North  0  South  1  East  3  West  0
CONNections Region 3  North  0  South  0  East  0  West  2

CONDitions Region 1  South=SYMmetry East=SURface substrate West=INFlow screen
CONDitions Region 2  North=SURface side_wall West=SURface top_wall
CONDitions Region 3  North=SURface side_wall South=SURface in_side_wall !&
            East=OUTflow

SEGment Number 1  Region 1  Side South  Origin West
SEGment Number 2  Region 1  Side East   Origin South
SEGment Number 3  Region 3  Side South  Origin West
SEGment Number 4  Region 3  Side East   Origin South
SEGment Number 5  Region 3  Side North  Origin East
SEGment Number 6  Region 2  Side North  Origin East
SEGment Number 7  Region 2  Side West   Origin North
SEGment Number 8  Region 1  Side West   Origin North

STREAMline Pathname outside

PATH Name outside Segments 1:8
PATH Name substrate Segment 2

PLOT PATH substrate Variable h_flux
PLOT PATH substrate Variable m_flux
PLOT PATH substrate Variable s2_flux
PLOT PATH substrate Variable s6_flux
PLOT PATH substrate Variable s1_site
PLOT PATH substrate Variable s2_site
PLOT PATH substrate Variable s3_site
PLOT PATH substrate Variable s4_site
PLOT PATH substrate Variable s5_site

Function Name side_wall XYpair  0.0          300.
Function Name side_wall XYpair  2.54          320
Function Name side_wall XYpair  5.08          400.0
Function Name side_wall XYpair  7.62          380
Function Name side_wall XYpair 10.16           340.

Function Name screen_s1  XYpair   0.0       .5
Function Name screen_s1  XYpair   5.07999   .5
Function Name screen_s1  XYpair   5.08001   .25
Function Name screen_s1  XYpair   10.16     .25

Function Name screen_s6  XYpair   0.0       .5
Function Name screen_s6  XYpair   5.07999   .5
Function Name screen_s6  XYpair   5.08001   .75
Function Name screen_s6  XYpair   10.16     .75

Function Name f0  XYpair   0.0   0.0
Function Name f0  XYpair   1.0   0.0

Function Name screen_u  XYpair   0.0       20.
Function Name screen_u  XYpair   5.07999   20.
Function Name screen_u  XYpair   5.08001   30.
Function Name screen_u  XYpair   10.16     30.
group elements list "1" list 1:800 append
group elements list "2" list 801:1100 append
group elements list "3" list 1101:1400 append

node 1 x -4.7988978E-12 y -5.6287939E-09
node 2 x 0.3513846 y 1.5731821E-10
node 3 x 0.6901811 y 1.5621615E-10
......
......
......
node 1494 x 9.80865 y 12.86938
node 1495 x 9.808637 y 12.70002
node 1496 x 10.16002 y 12.86936

element 1 type "quad4" material "" nodes 1,2,10,9
element 2 type "quad4" material "" nodes 2,3,11,10
element 3 type "quad4" material "" nodes 3,4,12,11
......
......
......
element 1398 type "quad4" material "" nodes 1427,1440,1439,1426
element 1399 type "quad4" material "" nodes 1440,1447,1446,1439
element 1400 type "quad4" material "" nodes 1447,1448,1445,1446
```

Figure 17: ap.out file listing for the example problem.

Line 1 allows the user to supply a title for the problem. This title will appear in the AP post-processing data base. This command is optional.

Line 2 determines the size of the *parser.out* file, a diagnostic dump file that is generated when the "dump" switch is turned "on". In our example problem, line 17 indicates that the dump switch is turned off. This command is optional.

Line 3 specifies the number of gas-phase and surface species. Note that in this problem we have two surface chemistry mechanisms. The first (substrate) has 5 surface species and the second (side wall) has 2 surface species. This command is required.

Line 4 indicates the iteration number at which calculations will stop. This command is required.

Line 5 specifies the frequency in iterations that output files (problem states) are written. Included in these output files is the restart file *restart_out.bin* which can be used to resume calculations by simply renaming it to *restart_in.bin.* Note that whenever calculations are restarted the "last_iteration" parameter must be increased. The print_frequency command is required.

Line 7 is the "restart switch". Turn it on for restarts. The default is off *i.e.*, if this command does not appear in the input file, it is assumed that restart is switched off. When restart is switched off, iterations will start from iteration number 1 using the initial conditions determined from line 23.

Lines 8 through 16 are the recommended switch positions for most Navier- Stokes calculations with the energy equation and gas-phase chemistry. Except for the net_flow switch, the default positions for these switches is off. The net_flow switch set in the on position ensures that the net flow at an inflow boundary resulting from combined diffusion and convection will be equal to the flow rate specified by inlet velocity, state, and composition for that boundary. When net_flow switch is in the off position, only the convective portion of the mass flow is fixed by the inlet velocity, state, and composition specified for that boundary. If convection dominates diffusion, the net and convective flow will be nearly identical and it won't matter how the switch is set.

One switch which does not appear in this example file is the "steady switch." Its default position is "on" to indicate that the calculation is a steady state calculation. Place the command "switch off steady" in the *problem.inp* file when transient calculations are to be performed. See Appendix A for more information on performing transient calculations.

Line 17 switches off the dump option. This is the default. When dump is switched on, CURRENT will print out some rather large ASCII diagnostic dump files named *parser.out, resids.out, monitor.out*, and *current.out.* CHEMKIN produces four additional diagnostic dump files *chem.out, tran.out, surf.out*, and *chemkin.out.* The user cannot prevent the CHEMKIN files from being written.

Lines 19 through 21 show recommended switch positions for GMRES acceleration. Providing such acceleration to this example problem leads to a converged solution in approximately 500 iterations. Without such acceleration (*i.e.* standard SIMPLER method) convergence takes several thousand iterations. Not all problems benefit from GMRES acceleration. See Section 4 for a more complete discussion of this topic.

Line 23 (continued on line 24) specifies the initial conditions for all field variables in the problem. Don't forget to replace the string "!&" with "&" to enable a proper line continuation. (Recall that the symbol "!" appearing anywhere in an AP or CURRENT input file causes the parsing software to treat all succeeding characters on the line as "comments.") The initial conditions are imposed only for non-restart calculations. On restarts, the starting values for the field variables are obtained from the restart file, *restart_in.bin*, and not from the initial command. The only exception to this is the pressure level (in this case 1333 dynes/cm^2) which can be altered using the initial command even on restarts. The initial command is required.

Line 26 (continued on 27) specifies relaxation parameters (fractions between 0 and 1) for all field variables. These parameters may be altered on restarts. This command is required.

Line 29 specifies the inclination angle of the CURRENT X-axis with respect to the horizon. For our example reactor shown in Figure 9, set the angle to -1.5708 radians ($-90°$). If we wanted to turn the reactor upside down we would set the angle to +1.5708 radians ($+90°$). This command is required.

Lines 31 through 40 contain commands that assign certain attributes to user-defined and named surfaces. Line 31 (continued on 32 and 33) defines a surface called "substrate" as being isothermal (as opposed to adiabatic) with a uniform temperature of 1000 K. The substrate rotates with an angular velocity of 3.142 radians/ sec (30 RPM). Surface chemistry is turned on permitting the surface chemistry reaction defined in *surf.inp* to take place. Whenever we turn surface chemistry on at a surface, we must supply initial guesses for the surface species site fractions. These fractions follow the site_fractions keyword and have values between 0 and 1. (The order is important.) Site fraction guesses are only used at start-up and are ignored after the first iteration or on subsequent restarts. Solution of the nonlinear equations for the surface site fractions is taken care of "automatically" using the FACE software developed by J.F. Grcar [20].

Line 35 defines a less complicated surface called "top_wall." It is a non-rotating, non-reacting surface having a uniform temperature of 300 K.

Line 37 (continued on 38) defines a surface called "side_wall" to be a non-rotating, non-reacting surface with the prescribed temperature distribution shown in Figure 10. In order to accomplish this we must define a piecewise linear function which, in this case, is also called "side_wall." The $X - Y$ pairs which make up this function are defined elsewhere in the file (lines 83-87). The total temperature at any X coordinate on the surface is determined from the product of the temperature keyword value (in this case 1.0) and the interpolated factor from the piecewise linear function "side_wall." Functional variations for inlet flow compositions and inlet velocities are defined using a similar technique. Note that a surface chemistry mechanism (mechanism number 2 with two surface species) has been specified for the side wall but in this particular surface description it is "turned off." We might use such a strategy in order to reach a steady state solution more quickly or to avoid instabilities. We could, for example, reach a stable steady state solution for the flow problem and then turn on the surface chemistry for a restart.

Line 40 defines an adiabatic (isothermal switch is turned "OFF") surface called "in_side_wall." The surface is non-rotating and non-reacting.

Line 42 (continued on 43 and 44) defines an inlet flow boundary named "screen" which has a nonuniform axial (u) velocity component and no radial (v) or circumferential (w) velocity components. The velocity variation results from the product of the user defined function "screen_u" which has $X-Y$ pairs given in lines 102-105 and the constant following the U_velocity keyword, which, in this case, is 1.0. This velocity distribution is identical to that shown in Figure 12. The inlet composition variations shown in Figure 11 are defined in a similar fashion. Since we have six total gas-phase species we must supply six individual mole fractions. At any location along the inlet, the mole fractions must sum to 1.0. For uniform molar composition across an inlet we would only need to supply values for the species keyword (one value for each gas-phase species). In the present problem, however, we wish to specify a spatial variation in composition. This requires the specification of six named piecewise linear curve functions. We used a dummy null function which we named "f0" for four of the six species and the functions named "screen_s1" and "screen_s6" for the remaining two species (recall that species 1 is nitrogen and species 6 is TEOS, the only gases entering the reactor).

A few words about meshing strategy are in order here. Note that we could have just as easily implemented the variations in composition and velocity from Figure 11 and 12 without using piecewise linear curves. We could have simply used two regions each of which had a uniform composition and velocity. This is not the preferred method, however, since the addition of an extra region in the problem will most certainly affect the convergence rate for the solver. This is true even if we split region 1 into two regions and keep the same total number of elements (control volumes). It is always better from a computational standpoint to use piecewise linear curves and the fewest possible regions.

Lines 46 through 48 are used to tell CURRENT which material properties to use within a computational region. At this writing, only one material model is available through the user interface and that model is CHEMKIN (gas and surface phases). With additional coding, alternate material properties can be used for various regions. One could, for example, solve a conjugate heat transfer problem involving a flowing gas and a solid body with finite heat capacity and thermal conductivity. For the present time, however, the user interface only supports CHEMKIN material models. Hence all *problem.inp* files must have one CHEMKIN material command for each region.

Lines 50 through 52 describe how the regions are linked to each other. A "0" region number is used to indicate that the particular region side is not shared by any other region. One connection command is required for each region.

Lines 54 through 56 (continued on 57) list all the boundary conditions on each region. The compass directions are used to differentiate between the various sides. If a boundary is an internal boundary, *i.e.* within the flow field, the word internal can be used at the boundary type. But since internal is the default boundary type, we can leave these off entirely as was done in lines 54 through 56. That leaves only four more boundary types to specify, namely "surface, inflow, symmetry", and "outflow." No additional information is required in specifying symmetry or outflow boundaries, but surface and inflow boundaries require that the name of the surface or inflow be specified along with the boundary type. In the present example, the surface names are "substrate", "top_wall", "side_wall", and "in_side_wall." The only inflow for the problem has the name "screen." One condition command is required for each region.

Lines 59 through 66 contain definitions of eight line segments which are labeled 1 through 8. Line segments are sides of regions that have a unique orientation. For example, line 66 defines segment 8 as being the west side of region 1. The segment has an origin which begins on the north side of the region. We could easily define a segment 9 which is also on the west side of region 1 but has its origin on the south side of the region. Segments 8 and 9 would actually occupy the same space but their origins would be different making them unique segments.

User defined segments are used to build up paths taking into account the origin of the segment. Line 70 defines a path which has the name "outside." It is composed of segments 1 through 8. The field 1:8 means the same thing as 1,2,3,4,5,6,7,8 or 1,2,3,4:8 or 1:2,3:4,5,6:8, etc. Paths must be continuous, *i.e.*, no missing segments, and the origin of the each segment must be taken into account when linking segments together. In other words, the origin of the current segment in the path must coincide with the end of the previous segment.

The path named "outside" is a very special path since it consists of all of the segments required to continuously outline the boundaries of the flow field. If the user wishes to display streamlines as part of the output this special path must be defined and specified in the "streamline" command. Line 68 gives an example of the streamline command. Providing for streamlines as part of the output is optional; hence lines 59 through 70 need not be present.

Line 71 defines the simplest of all paths, *i.e.*, a single segment. This path lies along the substrate so it is logical that it has been given the name "substrate." Having defined this path, we are now in a position to output the distribution of certain variables along it. If the path is a surface we may output heat flux. In the case of a surface with chemistry, distributions of net mass flux, the flux of individual gas-phase species, and individual surface site fractions for each of the surface species may also be placed in the output files.

Lines 73 through 81 contain commands to output substrate distributions for heat flux, net mass flux, the flux of gas-phase species 2 (intermediate), the flux of gas-phase species 6 (TEOS) and the site fractions for all five surface species.

5.5 A Few Words On Post-processing

Distributions for each of the variables in lines 73 through 81 of Figure 17 will be written to separate ascii files at the print frequency (line 5). The output file names take the form *pathname.variablename.* Hence lines 73 through 81 will produce the following files:

```
substrate.h_flux
substrate.m_flux
substrate.s2_flux
substrate.s6_flux
substrate.s1_site
substrate.s2_site
substrate.s3_site
substrate.s4_site
substrate.s5_site
```

Figure 18 contains a listing of the file *substrate.h_flux* printed after 100, 200, 300, 400, 500 and 600 iterations. Some lines have intentionally been omitted from the listing to save space. The first line printed in each cycle of printing is a comment line indicating which variable is being printed. Column 1 of the file is the distance from the origin of the path. In the case of the path "substrate" this column is the Y-axis and $Y = 0$ corresponds to the location of the axis of symmetry. Column 2 of the file contains the heat flux in CGS units (ergs/cm^2-s). The remaining two columns in the *substrate.h_flux* file are the X and Y coordinates of the path "substrate" from its beginning to its end. Similar displays of data are given in the remaining mass flux and site fraction files.

Plotting column 2 versus column 1 will result in a plot of the substrate heat flux distribution. Such a plot is shown in Figure 19 for all the first three printing cycles and the converged solution. It is quite clear that the solution at 100 iterations is far from converged. Plots like the one shown in Figure 19 serve as a graphical representation of how well the calculation is converging in the area above the substrate. Since heat flux is a quantity computed from differentiating the temperature field (a primary dependent variable), it is extremely sensitive. Hence plots which show successive heat flux profiles collapsing to a single curve are a strong indication of convergence.

The above mentioned plot files represent a small part of the available output from CURRENT. Two additional output files named *tecplot.out* and *ap.bin* will be generated at the print frequency given in line 5. These files represent the bulk of the available output. The file *tecplot.out* is readable by the display program TECPLOT. Users interested in displaying CURRENT output using TECPLOT should consult reference [26]. The file *ap.bin* is a SEACO (a Sandia-developed data structure) data base file directly readable by AP. Hence AP not only serves as a pre-processor (mesh generator) for CURRENT, it is also a post-processor.

After loading the *ap.bin* file into AP, users may display line contour plots, color-filled contour plots, plots of variables plotted against time (transient analysis), plots of variables plotted along lines defined interactively in AP, plots of vector fields such as velocity, plots of the mesh, plots overlaid with other plots such as filled temperature or concentration profiles overlaid by streamlines, and much more. The variables available for plotting include pressure, temperature, species mole fractions, species mass fractions, velocity components and streamlines (providing the streamline command is present in the *problem.inp* file). Instructions for producing these plots are available in the AP documentation [5]. Several examples of AP output for the converged example problem are shown in Figures 20 through 22. Also shown are insets

```
c Start of heat flux profile plot
 0.219850E+00 -0.130721E+07  0.508001E+01  0.219850E+00
 0.655679E+00 -0.130878E+07  0.508001E+01  0.655679E+00
 0.108383E+01 -0.131175E+07  0.508001E+01  0.108383E+01
......
......
......
 0.121425E+02 -0.238335E+07  0.508001E+01  0.121425E+02
 0.123683E+02 -0.302815E+07  0.508001E+01  0.123683E+02
 0.125901E+02 -0.681206E+07  0.508001E+01  0.125901E+02
c Start of heat flux profile plot
 0.219850E+00 -0.956382E+06  0.508001E+01  0.219850E+00
 0.655679E+00 -0.957452E+06  0.508001E+01  0.655679E+00
 0.108383E+01 -0.959603E+06  0.508001E+01  0.108383E+01
......
......
......
 0.121425E+02 -0.191311E+07  0.508001E+01  0.121425E+02
 0.123683E+02 -0.248253E+07  0.508001E+01  0.123683E+02
 0.125901E+02 -0.565751E+07  0.508001E+01  0.125901E+02
c Start of heat flux profile plot
 0.219850E+00 -0.850656E+06  0.508001E+01  0.219850E+00
 0.655679E+00 -0.851590E+06  0.508001E+01  0.655679E+00
 0.108383E+01 -0.853511E+06  0.508001E+01  0.108383E+01
......
......
......
 0.121425E+02 -0.171435E+07  0.508001E+01  0.121425E+02
 0.123683E+02 -0.223938E+07  0.508001E+01  0.123683E+02
 0.125901E+02 -0.511304E+07  0.508001E+01  0.125901E+02
c Start of heat flux profile plot
 0.219850E+00 -0.814260E+06  0.508001E+01  0.219850E+00
 0.655679E+00 -0.815147E+06  0.508001E+01  0.655679E+00
 0.108383E+01 -0.816985E+06  0.508001E+01  0.108383E+01
......
......
......
 0.121425E+02 -0.161993E+07  0.508001E+01  0.121425E+02
 0.123683E+02 -0.211994E+07  0.508001E+01  0.123683E+02
 0.125901E+02 -0.483521E+07  0.508001E+01  0.125901E+02
c Start of heat flux profile plot
 0.219850E+00 -0.801425E+06  0.508001E+01  0.219850E+00
 0.655679E+00 -0.802294E+06  0.508001E+01  0.655679E+00
 0.108383E+01 -0.804104E+06  0.508001E+01  0.108383E+01
......
......
......
 0.121425E+02 -0.157675E+07  0.508001E+01  0.121425E+02
 0.123683E+02 -0.206372E+07  0.508001E+01  0.123683E+02
 0.125901E+02 -0.469870E+07  0.508001E+01  0.125901E+02
c Start of heat flux profile plot
 0.219850E+00 -0.796876E+06  0.508001E+01  0.219850E+00
 0.655679E+00 -0.797740E+06  0.508001E+01  0.655679E+00
 0.108383E+01 -0.799540E+06  0.508001E+01  0.108383E+01
......
......
......
 0.121425E+02 -0.155806E+07  0.508001E+01  0.121425E+02
 0.123683E+02 -0.203875E+07  0.508001E+01  0.123683E+02
 0.125901E+02 -0.463533E+07  0.508001E+01  0.125901E+02
```

Figure 18: substrate.h_flux output file for the example problem.

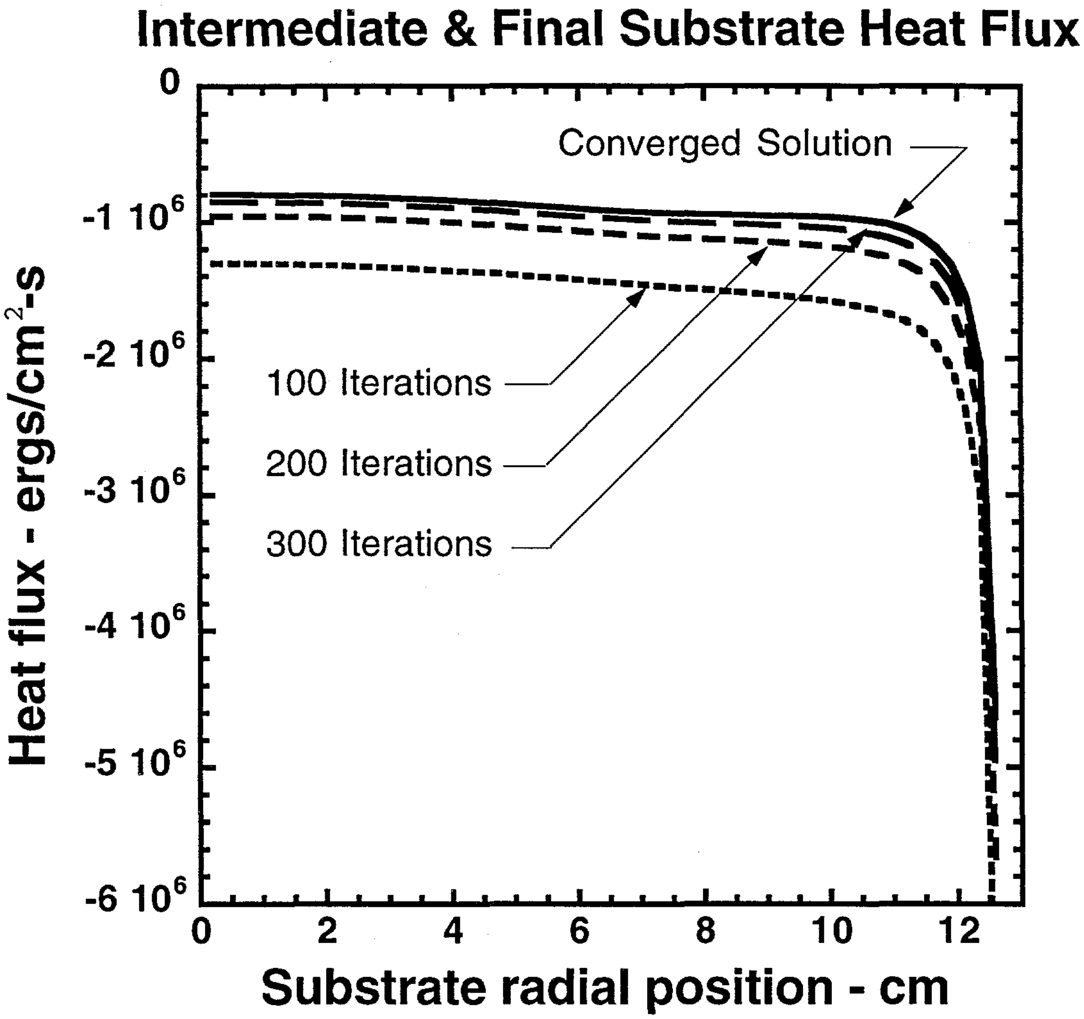
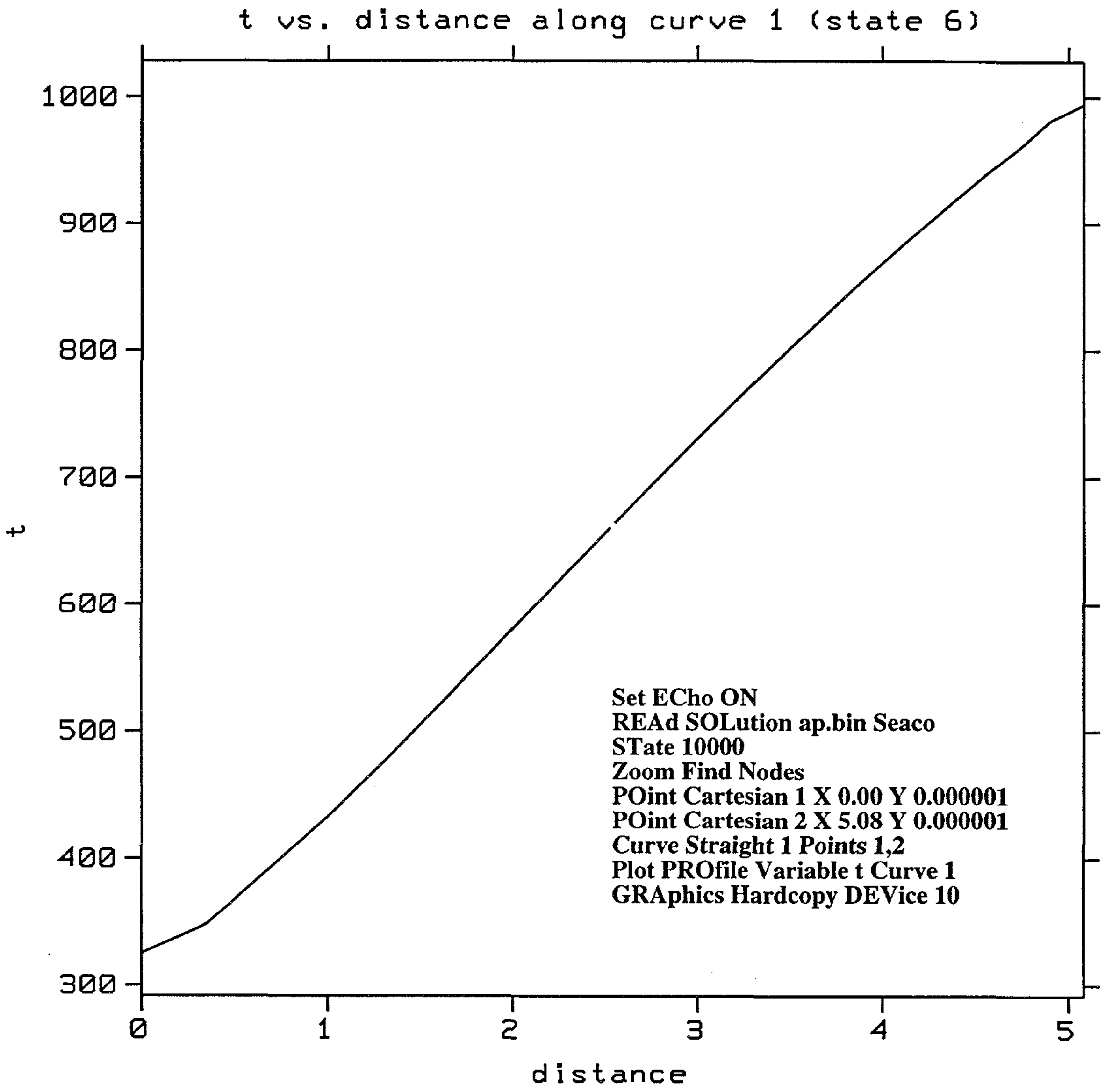


Figure 19: Heat flux profiles plotted from file substrate.h_flux.

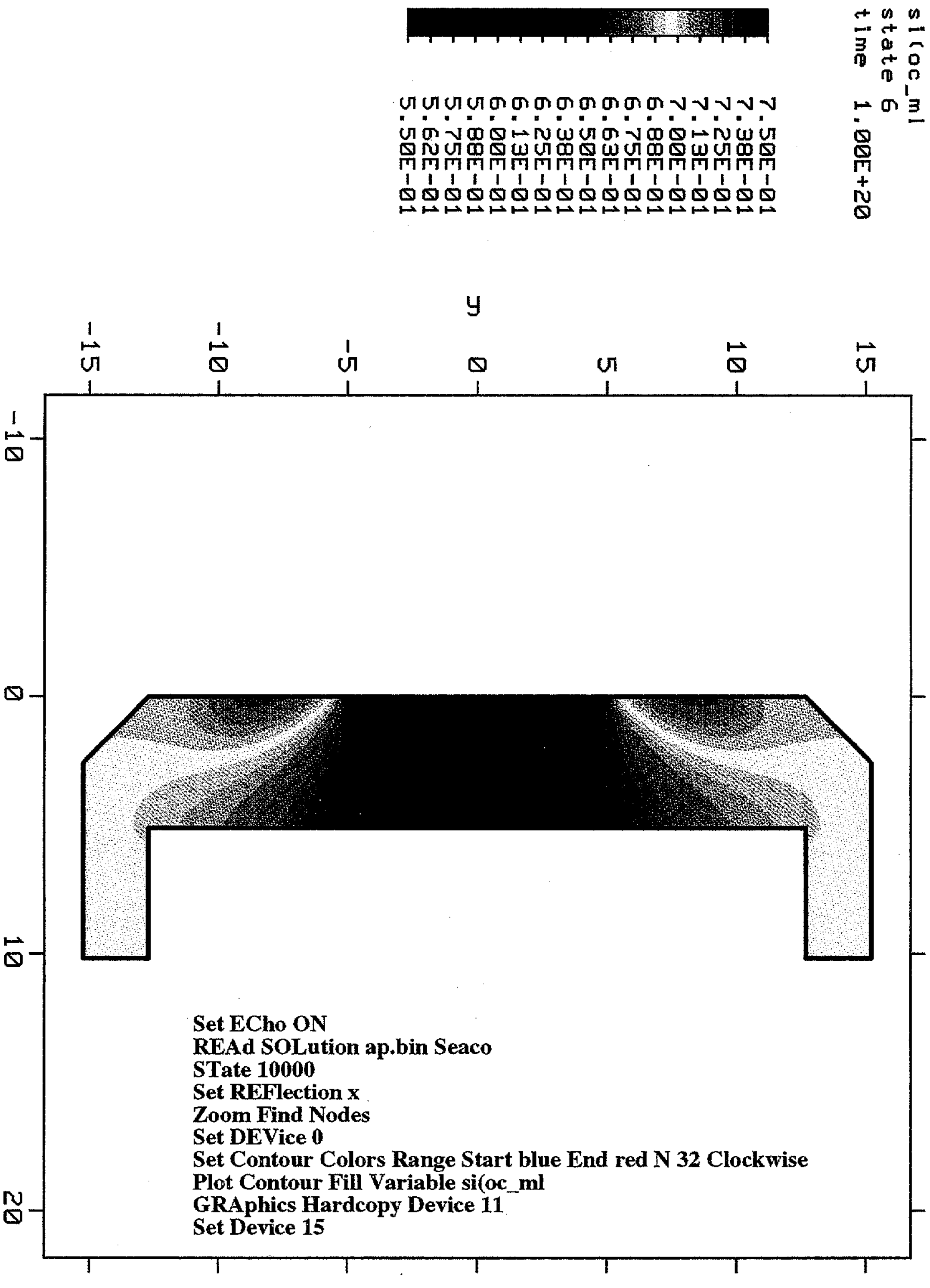
containing listings of the AP post-processing commands used to generate the plots. Upper case characters were used as before to indicate the shortest form of each command.

Figure 20 shows an AP-generated plot of the temperature distribution along the centerline of the reactor. Figure 21 shows a cross-section of the reactor with contours of TEOS concentration. Finally in Figure 22, we see a reactor cross-section showing color-filled temperature contours with superimposed streamlines.



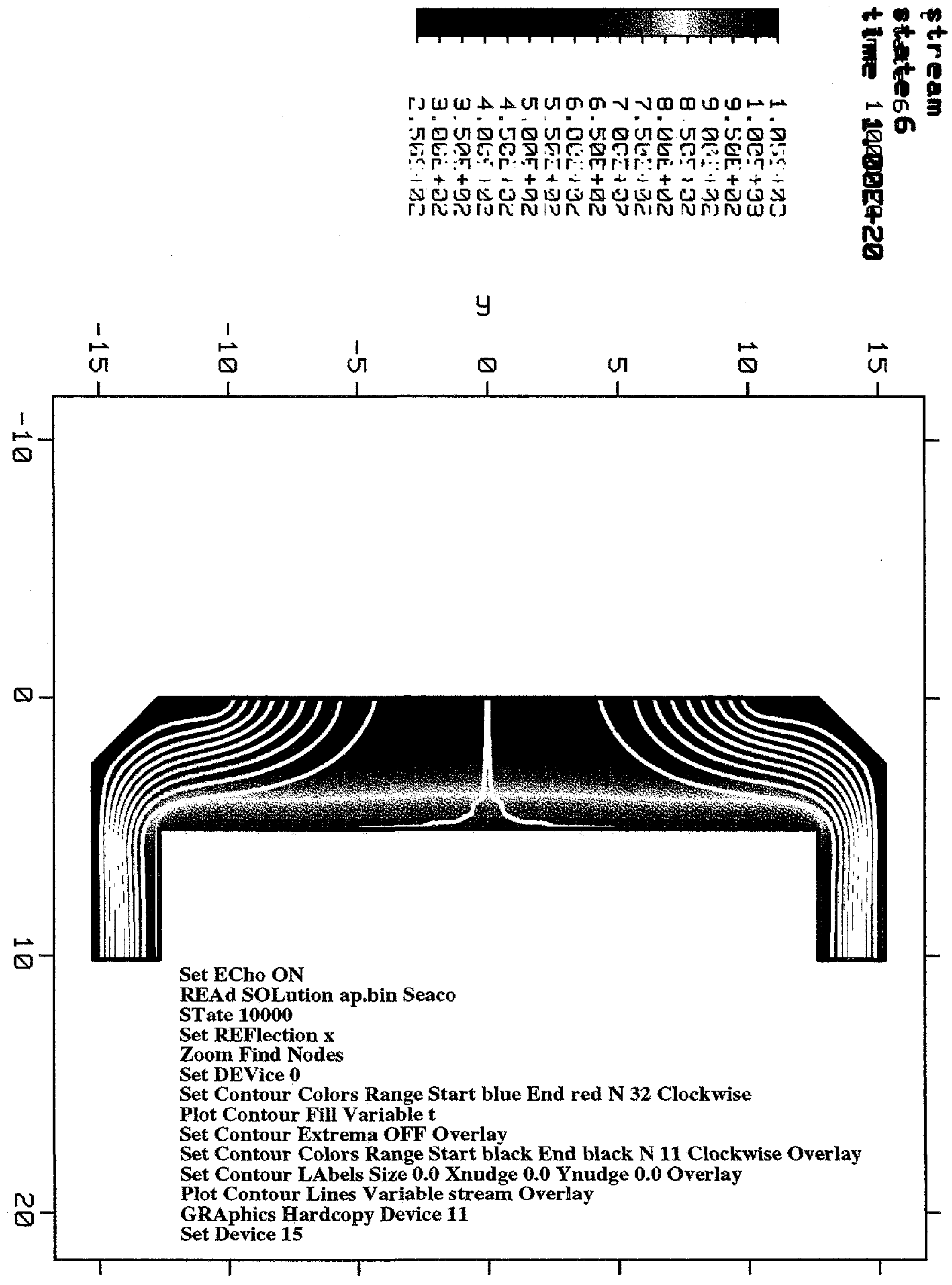
```
Set ECho ON
REAd SOLution ap.bin Seaco
STate 10000
Zoom Find Nodes
POint Cartesian 1 X 0.00 Y 0.000001
POint Cartesian 2 X 5.08 Y 0.000001
Curve Straight 1 Points 1,2
Plot PROfile Variable t Curve 1
GRAphics Hardcopy DEVice 10
```

Figure 20: Reactor centerline temperature distribution plotted with AP.



```
Set ECho ON
REAd SOLution ap.bin Seaco
STate 10000
Set REFlection x
Zoom Find Nodes
Set DEVice 0
Set Contour Colors Range Start blue End red N 32 Clockwise
Plot Contour Fill Variable si(oc_ml
GRAphics Hardcopy Device 11
Set Device 15
```

Figure 21: Contours of $Si(OC_2H_5)_4$ (TEOS) concentration plotted with AP.



```
Set ECho ON
REAd SOLution ap.bin Seaco
STate 10000
Set REFlection x
Zoom Find Nodes
Set DEVice 0
Set Contour Colors Range Start blue End red N 32 Clockwise
Plot Contour Fill Variable t
Set Contour Extrema OFF Overlay
Set Contour Colors Range Start black End black N 11 Clockwise Overlay
Set Contour LAbels Size 0.0 Xnudge 0.0 Ynudge 0.0 Overlay
Plot Contour Lines Variable stream Overlay
GRAphics Hardcopy Device 11
Set Device 15
```

Figure 22: Contours of temperature with superimposed streamlines.

6 Installing and Running CURRENT

The CURRENT software is distributed as a single compressed tar file called *current.tar.Z.* CURRENT can be installed on nearly any UNIX based workstation or mainframe. The procedure for setting up a CURRENT computing environment and executing the example problem of Section 5 may be summarized as follows:

1. Move the "current.tar.Z" file to the UNIX directory where you wish to set up A CURRENT computing environment.

2. Type the following UNIX command:

   ```
   zcat current.tar.Z | tar xf -
   ```

   This will produce the CURRENT computing environment which consists of a top level directory "current" and four subdirectories beneath it called "bin", "tools", "example", and "data."

3. Change directories to the "bin" directory and if necessary edit the "code_size.h" file to reflect the size of your problem. The parameters in the "code_size.h" file are sufficient to run the example problem. If your problem is larger you MUST modify this file.

4. Edit the file "makefile" to reflect the UNIX platform type (SGI, Cray, Sun, etc.). This involves "commenting" and "uncommenting" appropriate lines in the file.

5. Type the following UNIX command:

   ```
   make
   ```

   This will result in a compilation of all source code required to run CURRENT with CHEMKIN and SURFACE CHEMKIN. Executable images "current.exe, tran.exe, chem.exe," and "surf.exe" will be produced. This step must be performed each time the code is installed or whenever the code is "resized" for a new problem.

6. Change directories to the "example" directory.

7. Type the following UNIX commands:

```
chmod a+x start.s
chmod a+x restart.s
```

This will make the CURRENT execution scripts "start.s" and restart.s executable.

8. Execute the start.s script by typing:

```
start.s
```

If your installation is correct, this will cause CURRENT to perform the first 10 iterations of the example problem.

Except for Step 5, the above steps take very little computer time. Step 5 could take 3 to 30 minutes depending on the platform type and level of compiler optimization.

6.1 The *bin* Directory

The CURRENT computing environment that results from Step 2 is shown schematically in Figure 23. Of the four subdirectories, the *bin* directory contains the largest number files. Within the *bin* directory, most users need only be concerned with the *code_size.h* and the *makefile* files since these are the files which MUST be edited by the user to customize the computing environment to the size of the problem and the type of computing platform.

Figure 24 shows a listing of the *code_size.h* file as it appears after Step 2. If one examines the lines 31 through 36, it becomes apparent that the sizing parameters specified are the minimums required for the example problem, namely 20 maximum control volumes in the X-direction for any one region, 40 maximum control volumes in the Y-direction for any one region, a maximum of three regions, six gas-phase species maximum, five surface species maximum, and a maximum of two different surface reaction types. Users may wish to edit *code_size.h* and specify a set of "default" parameters which are larger than any problem the user expects to encounter. The only limitation on problem size is the available computer memory. The advantage of this strategy is that Step 5 above only needs to be performed once and not each and every time the user sets up a new problem. The disadvantage is that when a user shares a UNIX platform with other users, the CURRENT job could be assigned a lower execution priority because of its excessively large size. In any event, the CURRENT parsing software will perform bounds checking to insure that sufficient memory exists to perform a calculation. If the parameters specified in *code_size.h* are not large enough, warning messages will be issued and code execution will terminate.

Figure 25 shows a listing of the first 96 lines of the *current/bin/makefile* file as it appears after Step 2.

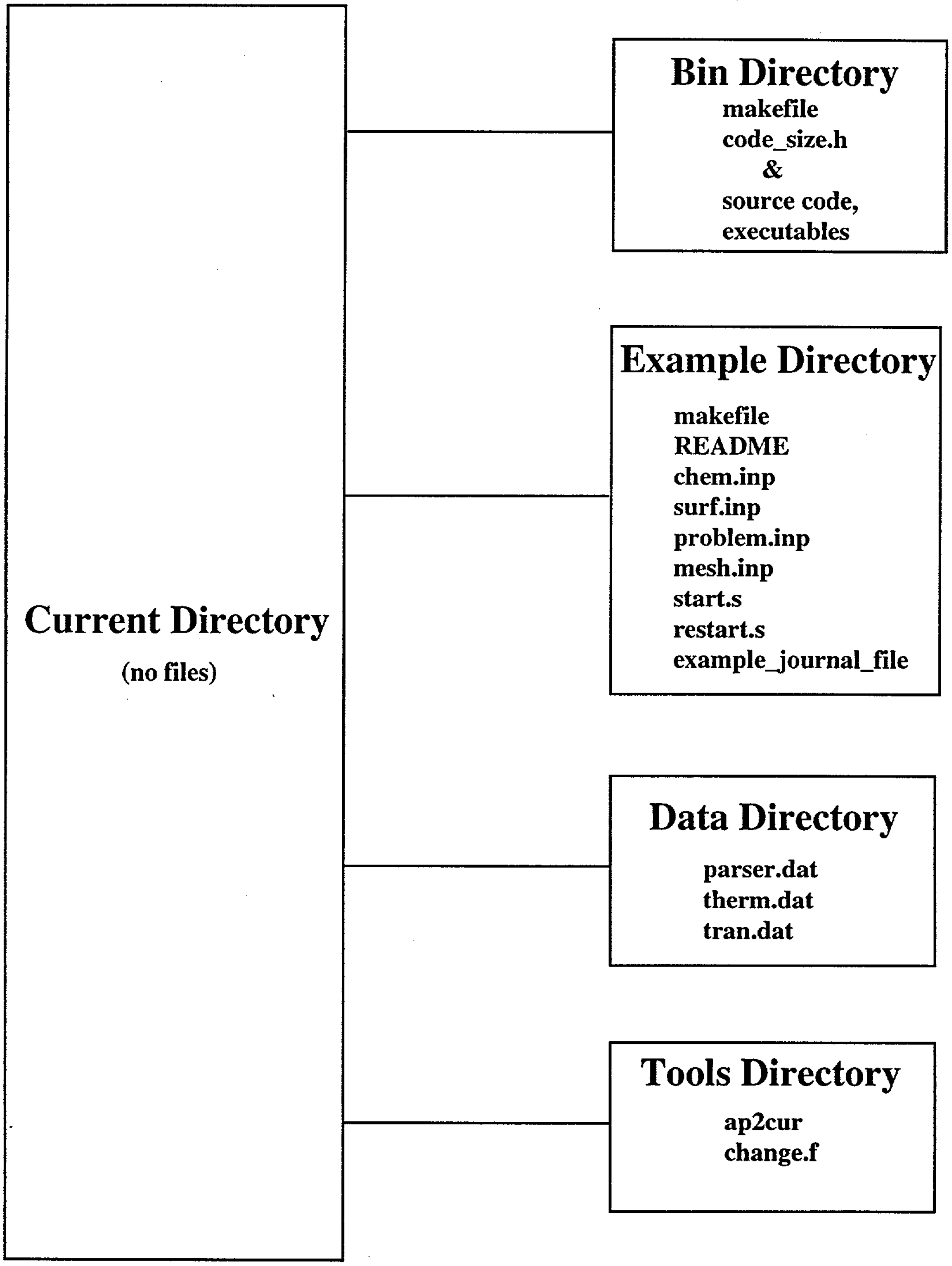
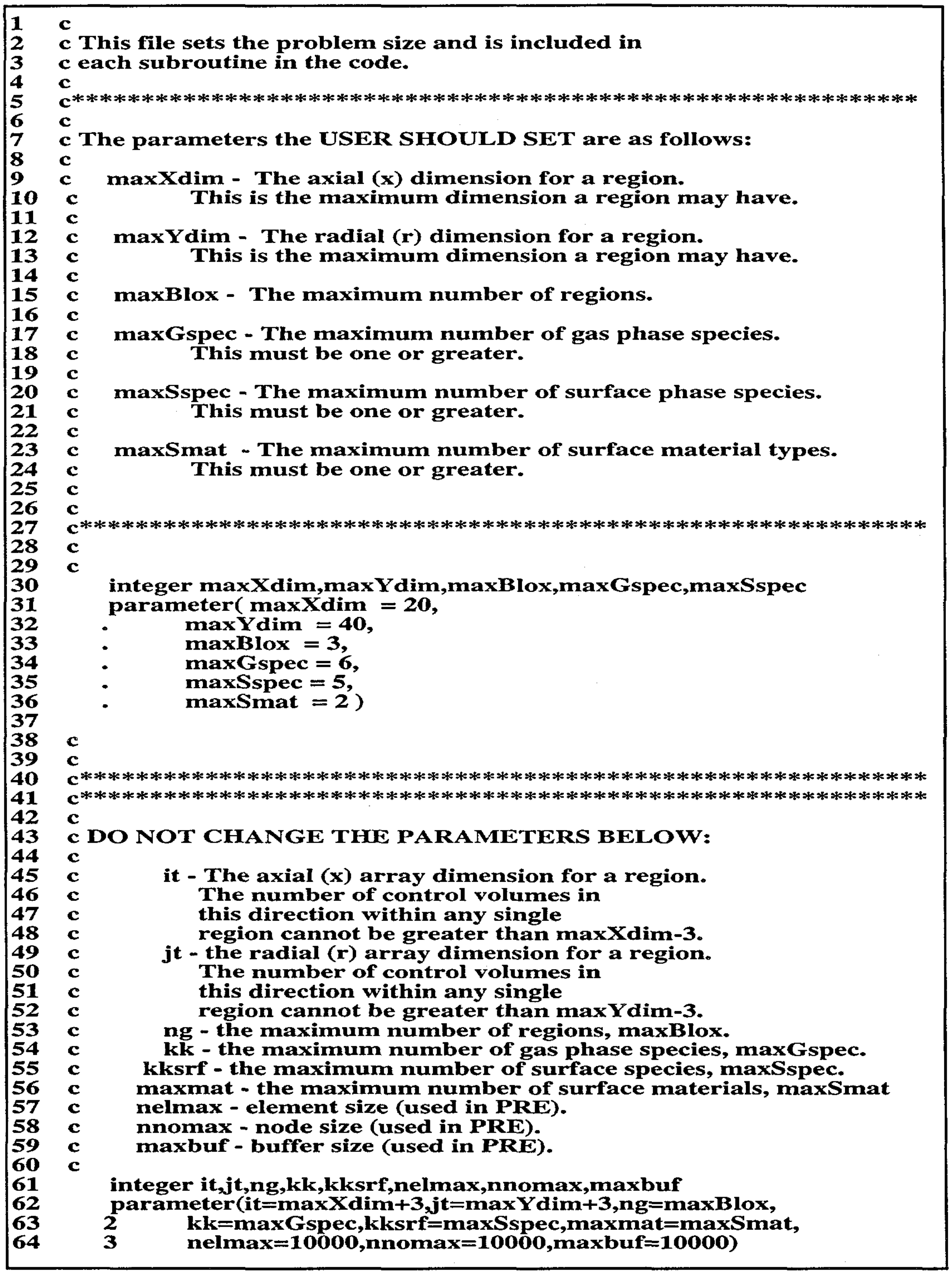


Figure 23: Directory structure of the CURRENT computing environment.



```
c
c This file sets the problem size and is included in
c each subroutine in the code.
c
c*************************************************************
c
c The parameters the USER SHOULD SET are as follows:
c
c    maxXdim -  The axial (x) dimension for a region.
c              This is the maximum dimension a region may have.
c
c    maxYdim -  The radial (r) dimension for a region.
c              This is the maximum dimension a region may have.
c
c    maxBlox -  The maximum number of regions.
c
c    maxGspec - The maximum number of gas phase species.
c              This must be one or greater.
c
c    maxSspec - The maximum number of surface phase species.
c              This must be one or greater.
c
c    maxSmat  - The maximum number of surface material types.
c              This must be one or greater.
c
c
c*************************************************************
c
c
      integer maxXdim,maxYdim,maxBlox,maxGspec,maxSspec
      parameter( maxXdim  = 20,
     .         maxYdim  = 40,
     .         maxBlox  = 3,
     .         maxGspec = 6,
     .         maxSspec = 5,
     .         maxSmat  = 2 )

c
c
c*************************************************************
c*************************************************************
c
c DO NOT CHANGE THE PARAMETERS BELOW:
c
c         it - The axial (x) array dimension for a region.
c              The number of control volumes in
c              this direction within any single
c              region cannot be greater than maxXdim-3.
c         jt - the radial (r) array dimension for a region.
c              The number of control volumes in
c              this direction within any single
c              region cannot be greater than maxYdim-3.
c         ng - the maximum number of regions, maxBlox.
c         kk - the maximum number of gas phase species, maxGspec.
c      kksrf - the maximum number of surface species, maxSspec.
c     maxmat - the maximum number of surface materials, maxSmat
c     nelmax - element size (used in PRE).
c     nnomax - node size (used in PRE).
c     maxbuf - buffer size (used in PRE).
c
      integer it,jt,ng,kk,kksrf,nelmax,nnomax,maxbuf
      parameter(it=maxXdim+3,jt=maxYdim+3,ng=maxBlox,
     2        kk=maxGspec,kksrf=maxSspec,maxmat=maxSmat,
     3        nelmax=10000,nnomax=10000,maxbuf=10000)
```

Figure 24: Listing of the *code_size.h* file.

The makefile is configured to build a CURRENT/CHEMKIN executable on the HP 735. Executables for other platform types can be built by first "commenting" lines 29 through 34 with "#" characters and "uncommenting" the corresponding lines for the appropriate platform. With some knowledge of the Fortran compiler/linker options, platforms not shown in Figure 25 can easily be added.

6.2 The *example* Directory

The *example* directory contains all the input files, the journal file, the run scripts and a *README* file for the example problem. To run the example problem, move to this directory and type the name of the appropriate run script (see discussion below). Execution will proceed and all output files will be written/overwritten to the *example* directory.

The files listed in the *example* directory can serve as templates for new problems. To create an environment for a new problem the user must duplicate the *example* directory at the same level (one level below the *current* directory and in parallel with the *example* directory). The process is most easily explained using the following UNIX command sequence:

```
cd current
mkdir current/new_problem
cp example/* new_problem
cd new_problem
```

The file *new_problem/example_journal_file* must be edited to build a journal file for the "new problem". Using the procedure described in Section 5, the input files *problem.inp, mesh.inp, chem.inp,* and *surf.inp* can be created and placed in the *new_problem* directory. Solutions to the new problem are obtained by executing the run scripts in the *new_problem* directory and all output files will be written to that directory.

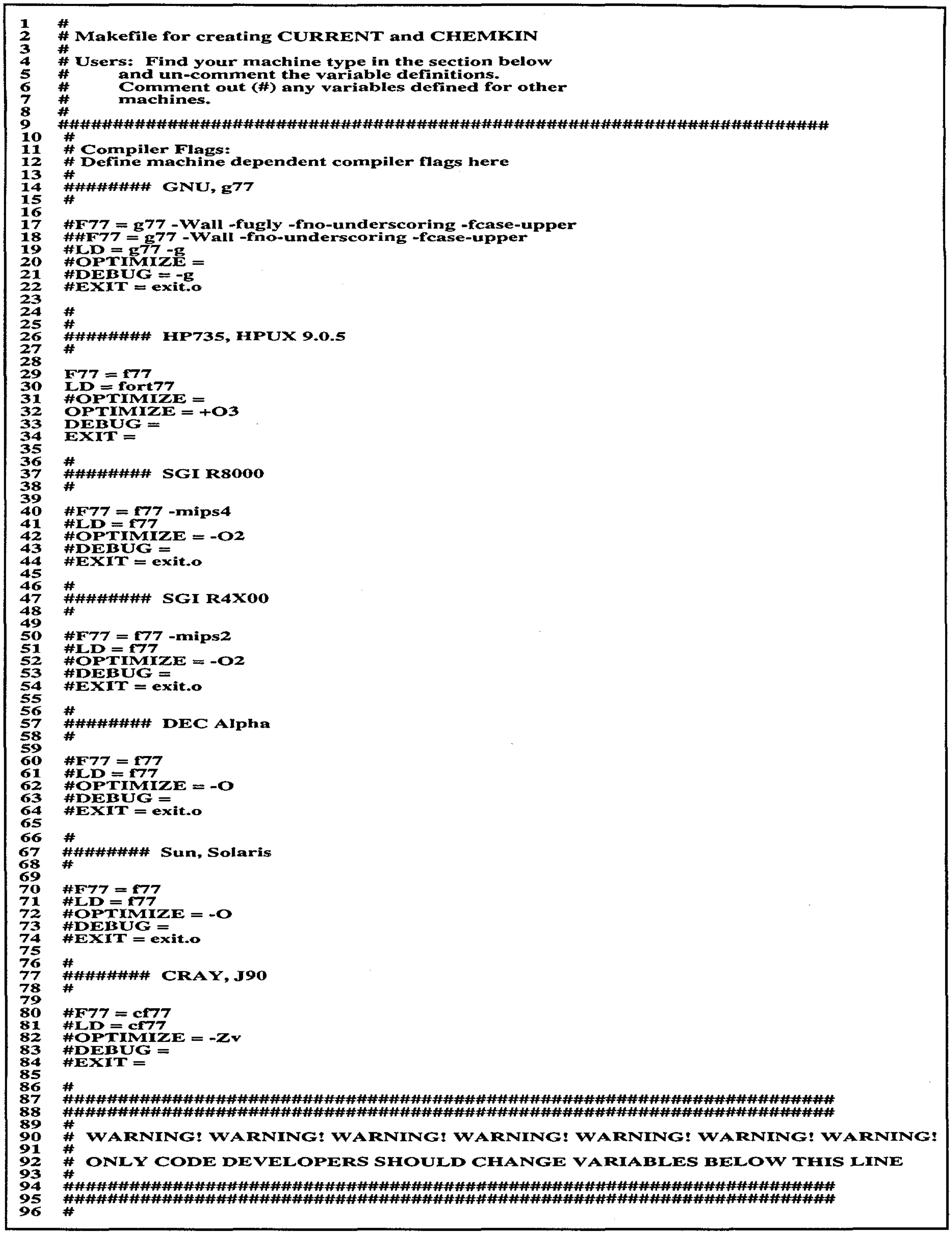
6.3 The *data* Directory

The *data* directory contains the three *.dat* files CURRENT needs to run the example problem and all other problems. The file *parser.dat* will never need to be altered. The remaining two CHEMKIN data files will only need editing if the user wishes to work with species not yet documented in the these two files. See Refererence [6], [7], and [8] for details.

6.4 The *tools* Directory

As the name implies, the *tools* directory contain tools the user may find useful in file preparation. These tools are summarized as follows:

```
ap2cur   - A UNIX script that can be used to produce "problem.inp" and "mesh.inp" from
           "ap.out"
change.f - Fortran source code (must be compiled and linked) for the change utility.
```



```
#
# Makefile for creating CURRENT and CHEMKIN
#
# Users:  Find your machine type in the section below
#         and un-comment the variable definitions.
#         Comment out (#) any variables defined for other
#         machines.
#
################################################################################
#
# Compiler Flags:
# Define machine dependent compiler flags here
#
########  GNU, g77
#

#F77 = g77 -Wall -fugly -fno-underscoring -fcase-upper
##F77 = g77 -Wall -fno-underscoring -fcase-upper
#LD = g77 -g
#OPTIMIZE =
#DEBUG = -g
#EXIT = exit.o

#
#
########  HP735, HPUX 9.0.5
#

F77 = f77
LD = fort77
#OPTIMIZE =
OPTIMIZE = +O3
DEBUG =
EXIT =

#
########  SGI R8000
#

#F77 = f77 -mips4
#LD = f77
#OPTIMIZE = -O2
#DEBUG =
#EXIT = exit.o

#
########  SGI R4X00
#

#F77 = f77 -mips2
#LD = f77
#OPTIMIZE = -O2
#DEBUG =
#EXIT = exit.o

#
########  DEC Alpha
#

#F77 = f77
#LD = f77
#OPTIMIZE = -O
#DEBUG =
#EXIT = exit.o

#
########  Sun, Solaris
#

#F77 = f77
#LD = f77
#OPTIMIZE = -O
#DEBUG =
#EXIT = exit.o

#
########  CRAY, J90
#

#F77 = cf77
#LD = cf77
#OPTIMIZE = -Zv
#DEBUG =
#EXIT =

#
################################################################################
################################################################################
#
#  WARNING! WARNING! WARNING! WARNING! WARNING! WARNING! WARNING!
#
#  ONLY CODE DEVELOPERS SHOULD CHANGE VARIABLES BELOW THIS LINE
#
################################################################################
################################################################################
#
```

Figure 25: Listing of the top portion of file *current/bin/makefile.*

The change utility is Sandia-developed software [30] for converting source code from single to double precision (and vice-versa).

6.5 Running CURRENT

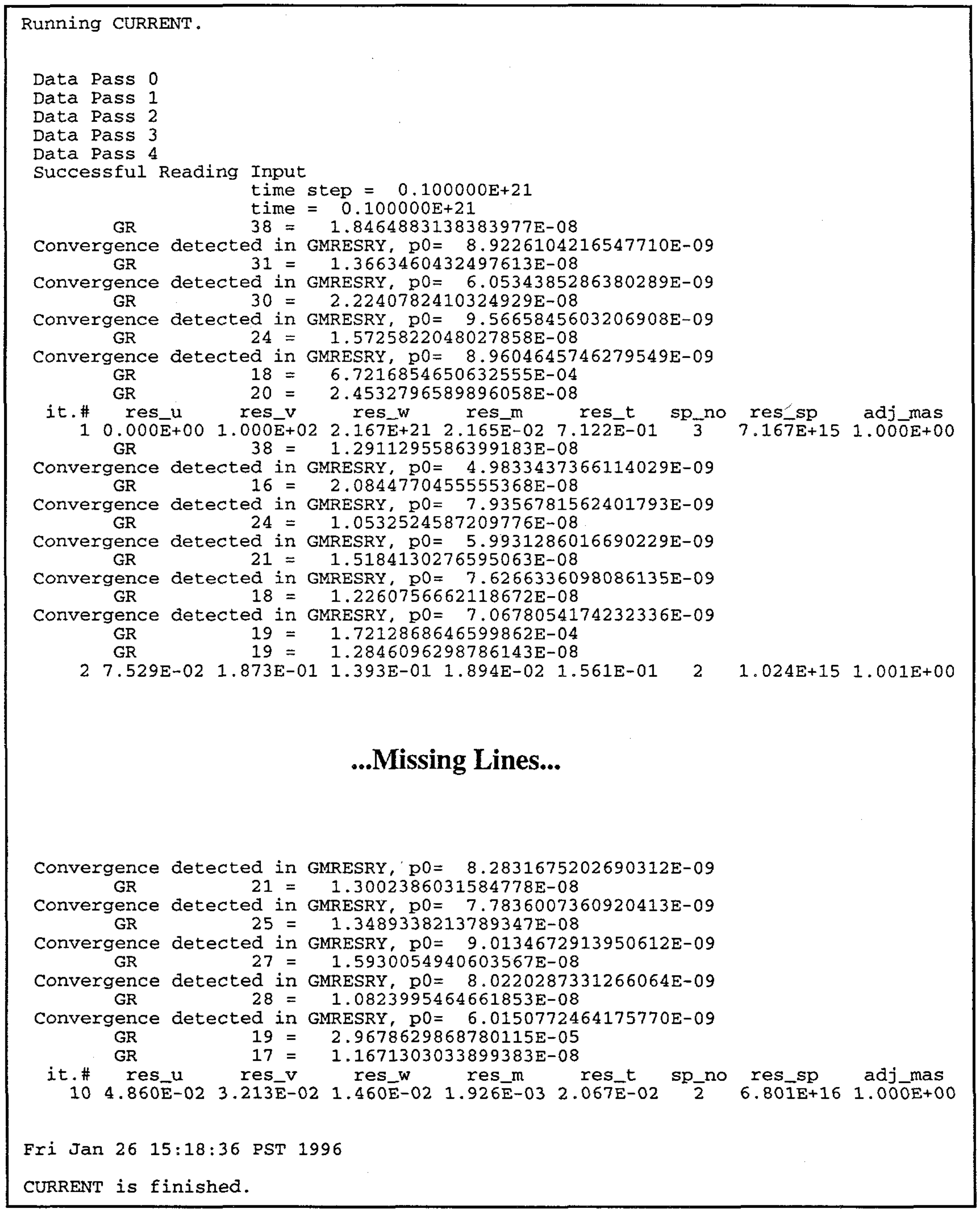
Figure 26 shows some of the messages printed to the screen during execution of the first 10 iterations of the example problem. The first message "Running CURRENT" is an indication that script execution is under way. That message is followed by a series of messages from the CURRENT parsing software as it reads *problem.inp* and *mesh.inp*. The message "Successful Reading Input" indicates that all input data has been read, no syntax errors were found, and the problem setup seems to be "legitimate." CURRENT then prints out the time step and the time, signaling the beginning of iterations. Since the example problem is a steady state calculation, CURRENT assigns a single exceedingly large time step for the problem. In steady state calculations, no further time stepping will occur. When GMRES acceleration is used, a series of diagnostics are printed out prior to each SIMPLER iteration. These are the lines beginning with "GR 38 = 1.84...." and "Convergence detected...". As each search direction is added in the gradient algorithm, the residual of the preconditioned linear system, GR, is printed out. The number of these messages tends to be higher at the beginning of a problem because more search directions are required to invert the matrix. As convergence of the SIMPLER method proceeds, these messages become fewer and may disappear entirely leaving only the single line message for each SIMPLER iteration.

The single-line SIMPLER iteration message includes the SIMPLER iteration number, a listing of the residuals for velocities u, v, and w, mass m, temperature t, the species number having the highest residual, its residual value and finally the size of the mass adjustment factor. The mass adjustment factor "adj_mas" is the factor applied to the outflow velocity to insure that inflow equals the outflow at the beginning of the next iteration. Switching on the optional mass adjustment in the input file almost always speeds convergence of the SIMPLER method. Generally, mass adjustment factors near 1.0 indicate a well behaved problem.

You may wish to restart the example problem calculation and compute a converged solution. With GMRES acceleration this will take about 600 iterations. Before restarting the calculation you must edit the *problem.inp* file and make the following changes:

```
1. Increase the last_iteration perameter to 600.
2. Change the print_frequency to something more managable, say 100.
3. "Switch On" the restart flag.
```

Restarts of CURRENT can be accomplished using the script *restart.s* which is identical to *start.s* except that prior to CURRENT execution, the file *restart_out.bin* is copied to *restart_in.bin*.



```
Running CURRENT.


 Data Pass 0
 Data Pass 1
 Data Pass 2
 Data Pass 3
 Data Pass 4
 Successful Reading Input
                    time step =  0.100000E+21
                    time =  0.100000E+21
        GR          38 =   1.8464883138383977E-08
 Convergence detected in GMRESRY, p0=  8.9226104216547710E-09
        GR          31 =   1.3663460432497613E-08
 Convergence detected in GMRESRY, p0=  6.0534385286380289E-09
        GR          30 =   2.2240782410324929E-08
 Convergence detected in GMRESRY, p0=  9.5665845603206908E-09
        GR          24 =   1.5725822048027858E-08
 Convergence detected in GMRESRY, p0=  8.9604645746279549E-09
        GR          18 =   6.7216854650632555E-04
        GR          20 =   2.4532796589896058E-08
  it.#   res_u     res_v     res_w     res_m     res_t    sp_no  res_sp    adj_mas
     1 0.000E+00 1.000E+02 2.167E+21 2.165E-02 7.122E-01   3   7.167E+15 1.000E+00
        GR          38 =   1.2911295586399183E-08
 Convergence detected in GMRESRY, p0=  4.9833437366114029E-09
        GR          16 =   2.0844770455555368E-08
 Convergence detected in GMRESRY, p0=  7.9356781562401793E-09
        GR          24 =   1.0532524587209776E-08
 Convergence detected in GMRESRY, p0=  5.9931286016690229E-09
        GR          21 =   1.5184130276595063E-08
 Convergence detected in GMRESRY, p0=  7.6266336098086135E-09
        GR          18 =   1.2260756662118672E-08
 Convergence detected in GMRESRY, p0=  7.0678054174232336E-09
        GR          19 =   1.7212868646599862E-04
        GR          19 =   1.2846096298786143E-08
     2 7.529E-02 1.873E-01 1.393E-01 1.894E-02 1.561E-01   2   1.024E+15 1.001E+00
```

...Missing Lines...

```
 Convergence detected in GMRESRY, p0=  8.2831675202690312E-09
        GR          21 =   1.3002386031584778E-08
 Convergence detected in GMRESRY, p0=  7.7836007360920413E-09
        GR          25 =   1.3489338213789347E-08
 Convergence detected in GMRESRY, p0=  9.0134672913950612E-09
        GR          27 =   1.5930054940603567E-08
 Convergence detected in GMRESRY, p0=  8.0220287331266064E-09
        GR          28 =   1.0823995464661853E-08
 Convergence detected in GMRESRY, p0=  6.0150772464175770E-09
        GR          19 =   2.9678629868780115E-05
        GR          17 =   1.1671303033899383E-08
  it.#   res_u     res_v     res_w     res_m     res_t    sp_no  res_sp    adj_mas
    10 4.860E-02 3.213E-02 1.460E-02 1.926E-03 2.067E-02   2   6.801E+16 1.000E+00


Fri Jan 26 15:18:36 PST 1996

CURRENT is finished.
```

Figure 26: CURRENT screen messages.

References

[1] G.H. Evans and R. Greif. "A Study of Traveling Wave Instabilities in a Horizontal Channel Flow with Applications to Chemical Vapor Deposition". *International Journal of Heat and Mass Transfer*, **32**(5):895–911, 1989.

[2] G.H. Evans and R. Greif. "A Two-Dimensional Model of the Chemical Vapor Deposition of Silicon Nitride in a Low-Pressure Hot-Wall Reactor including Multicomponent Diffusion". *International Journal of Heat and Mass Transfer*, **37**(10):1535–1543, 1994.

[3] W.S. Winters, G.H. Evans, and R. Greif. "A Two-Dimensional Numerical Model of Gas Mixing and Deposition in a Rotating Disk CVD Reactor". eds. T.M. Besmann, M.D. Allendorf, McD. Robinson, and R.K. Ulrich, editors, *CVD XIII, Proceedings of the Thirteenth International Conference on Chemical Vapor Deposition*, pp. 89–94, The Electrochemical Society, Pennington, NJ. 1996.

[4] S.V. Patankar. *Numerical Heat Transfer and Fluid Flow.* McGraw-Hill, New York, NY, 1980.

[5] P.E. Nielan, K.J. Perano, and W.E. Mason. "ANTIPASTO: An Interactive Mesh Generator and Preprocessor for Two-Dimensional Analysis Programs". Technical Report SAND90–8203, Sandia National Laboratories, Livermore, CA, May 1990.

[6] R.J. Kee, F.M. Rupley, E. Meeks, and J.A. Miller. "Chemkin-III: A Fortran Chemical Kinetics Package for the Analysis of Gas Phase Chemical and Plasma Kinetics". Technical Report SAND96–8216, Sandia National Laboratories, Livermore, CA, May 1996.

[7] M.E. Coltrin, R.J. Kee, F.M. Rupley, and Ellen Meeks. "SURFACE CHEMKIN-III: A Fortran Package for Analyzing Heterogeneous Chemical Kinetics at a Solid-Surface-Gas-Phase Interface". Technical Report SAND96–8217, Sandia National Laboratories, Livermore, CA, May 1996.

[8] R.J. Kee, G. Dixon Lewis, J. Warnatz, M.E. Coltrin, and J.A. Miller. "A Fortran Computer Code Package for the Evaluation of Gas-Phase Multicomponent Transport Properties". Technical Report SAND86–8246, Sandia National Laboratories, Livermore, CA, September 1991.

[9] J.F. Thompson, Z.U.A. Warsi, and C. Wayne Mastin. *Numerical Grid Generation.* Elsevier Science Publishing Co., New York, 1985.

[10] W. Shyy, S. S. Tong, and S. M. Correa. "Numerical Recirculating Flow Calculation using a Body-Fitted Coordinate System". *Numerical Heat Transfer.*

[11] G. Evans and R. Greif. "A Numerical Model of the Flow and Heat Transfer in a Rotating Disk Chemical Vapor Deposition Reactor". *Journal of Heat Transfer*, **109**:928–935, November 1987.

[12] G. Evans and R. Greif. "Unsteady Convection in Channel Flows with Applications to Chemical Vapor Deposition". *Proceedings of the 9th International Heat Transfer Conference*, Vol. **2**, pp. 471–476, Hemisphere Publishing Corp., New York. 1990.

[13] fiber drying tower modeling studies, 3M CRADA, unpublished; also unpublished Sandia Report on fiber drying by R.S. Larson.

[14] S. Joh and G.H. Evans. "Chemical Vapor Deposition of Boron Nitride Coatings on Moving Fibers". eds.

T.M. Besmann, M.D. Allendorf, McD. Robinson, and R.K. Ulrich, editors, *CVD XIII, Proceedings of the Thirteenth International Conference on Chemical Vapor Deposition*, pp. 588–593, The Electrochemical Society, Pennington, NJ. 1996.

[15] G. Evans, W. Houf, R. Greif, and C. Crowe. "Gas-Particle Flow within a High Temperature Solar Cavity Receiver including Radiation Heat Transfer". *Journal of Solar Energy Engineering*, **109**:134–142, May 1987.

[16] G. Dixon-Lewis. "Flame structure and flame reaction kinetics II. Transport phenomena in multicomponent systems". *Proceedings of the Royal Society, A.*, **307**:111–135, 1968.

[17] R.B. Bird, W.E. Stewart, and E.N. Lightfoot. *Transport Phenomena.* John Wiley, New York, 1960.

[18] S. Paolucci. "On the filtering of sound from the Navier-Stokes equations". Technical Report SAND82–8257, Sandia National Laboratories, Livermore, CA, 1982.

[19] T. Hayase, J.A.C. Humphrey, and R. Greif. "A Consistently Formulated QUICK Scheme for Fast and Stable Convergence Using Finite-Volume Iterative Calculation Procedures". *Journal of Computational Physics*, **98**:108–118, 1992.

[20] J.R. Grcar. "The FACE Computer Code". Sandia National Laboratories, Livermore, CA, Private Communication, 1995.

[21] C. D. Moen. "Improving Convergence Rates for Low Pressure Material Processing Calculations". *Proceedings of the ASME Fluids Engineering Division Summer Meeting*, pp. 181–188. ASME-FED, Vol 238, July 1996.

[22] Y. Saad and M.H. Schultz. "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetrical Linear Systems". *SIAM Journal of Scientific and Statistical Computing*, **7**(3):856–869, July 1986.

[23] A.J. Chorin. "Numerical Solution of the Navier-Stokes Equations". *Mathematics of Computation*, **22**(1):745–762, August 1968.

[24] H. Dwyer. "Calculation of Low Mach Number Reacting Flows". *AIAA Journal*, **28**(1):98–105, January 1991.

[25] H. Dwyer. "Navier-Stokes Calculations of Multi-Dimensional Flows with Complex Chemical Kinetics". *Computing Systems in Engineering*, **5**(2):105–116, 1994.

[26] Amtec Engineering. "Tecplot- Interactive Data Visualization for Scientists and Engineers, Version 6 User's Manual". Technical report, Amtec Engineering, Inc., P.O. Box 3633, Bellevue, WA 98009-3633, 1993.

[27] K.J. Perano and V.N. Kaliakin. "INTERP- A Fortran Callable Free Format Data Interpretation Subroutine System". Technical Report SAND87–8244, Sandia National Laboratories, Livermore, CA, March 1989.

[28] K.J. Perano. "COMPRO - A Subroutine System for Syntactical Analysis". Technical Report SAND89-8441, Sandia National Laboratories, Livermore, CA, 1989.

[29] W.E. Mason. "SNLL DIGLIB". Sandia National Laboratories, Livermore, CA (The documentation and software are based on the original version of DIGLIB written by H.R. Brand of Lawrence Livermore National Laboratory)., 1989.

[30] J.F. Grcar. "The Change Tool for Changing Programs and Scripts". Technical Report SAND92–8225, Sandia National Laboratories, Livermore, CA, September 1992.

A APPENDIX - CURRENT Commands

In this Appendix, all user-accessible CURRENT commands are listed alphabetically with their optional and required keyword parameters. In keeping with the convention of Section 5, the part of the command or keyword given in uppercase represents the most abbreviated form. Examples of the longest and shortest form of each command are given. Intermediate forms are permitted providing they are spelled correctly. Command lines ending in"&" are continued on the next line. The command line syntax used by CURRENT is identical to that described in Reference [5] for ANTIPASTO. Commands may consist of command names, keywords and data. Any number of *blanks, tabs, commas,* and/or *equals signs* may be used as delimiters. Commands and keywords are case insensitive and may have abbreviated forms to minimize typing.

ANGLE

```
Purpose: Specifies the inclination angle (radians) of the X-axis with
         respect to the horizontal.

Example: ANGLE = -1.5708
         angle -1.5808

Remarks: The most common orientation for the gravity vector is -1.5708 radians
         or -90 degrees (\it{e.g.,} a vertical CVD reactor with an upward facing
         substrate).
```

CONNections

```
Purpose: List I.D. numbers of the regions bordering a particular region.

Example: CONNections Region 6 North 0 South 3 East 1 West 4
         conn r 6 w 4 s 3 e 1

Remarks: Every region in the problem must have a CONNections command.
         If a North, South, East or West keyword and data are missing, a
         value of 0 is assumed meaning the region is not connected on
         that side. In the example region 6 is connected to region 3 on
         the south, 1 on the east and 4 on the west. No regions are connected
         on the north, i.e. the north must be a boundary.
```

CONDitions

Purpose: Specifies the boundary types and associated data for all four sides of a region.

```
Example 1: CONDitions Region 2 East INTernal West OUTflow South SYMmetry &
                                North INTernal
           cond r 2    w=out, s=sym
```

```
Example 2: CONDitions Region 1 North=SURface reactor_wall South=INTernal &
                                East=INFlow reactor_inlet West=INTernal
           cond r=1 n=sur reactor_wall e=inf reactor_inlet
```

Remarks: Every region in the problem must have a CONDitions command. Boundary condition types are: internal, symmetry, outflow, inflow and surface. Inflow and surface boundaries have an additional keyword associated with them, the "name" of the inflow or surface boundary. These names may not be abbreviated. Missing boundaries are assumed to be internal boundaries. In example 1, region 2 has an internal boundary on east, an outflow boundary on the west, a symmetry boundary on the south, and another internal boundary on the north. In example 2, region 1 has internal boundaries on its south and west sides, a surface boundary on the north having the attributes of the surface named "reactor_wall", and an inflow boundary on the east having the attributes of the inflow named "reactor_inlet". The attributes for "reactor_wall" and "reactor_inlet" would have to be specified elsewhere in the CURRENT input file using the SURface and INflow commands.

Dump

Purpose: Determines the size of the diagnostic output file "parser.out"

Example: Dump Small

```
d s
```

Remarks: The user may specify Small, Medium, or Large (abreviated s, m, l respectively) parser.dat dump files. Medium or large dump files are of little use to most users. The file will only be printed if the dump switch is turned on (see switch command). This statement is optional. If it does not appear "Dump=Medium" is assumed.

Element

Purpose: Specifies nodal numbering for an element.

Remarks: Element commands are generated automatically by the mesh generator AP. They are a required part of the "mesh.inp" file. There should be one Element command for each control volume in the problem.

Function

Purpose: Assigns an X-Y pair to a user defined piecewise linear function.

Example: Function Name screen_u XYpair 1.0 10.0

```
f n velocity_distribution xy 1 10
```

Remarks: At least two of these commands are necessary to complete the definition of a piecewise linear function. Naming of piecewise linear functions (in this case, "screen_u") is arbitrary.

GRoup

Purpose: AP mesh generation command that directs AP to make a list of all elements that make up a region.

```
Example: Group Elements Region Name "1" Append
         g e r n "1" a
```

Remarks: One of these mesh generation commands must be placed in the meshing journal file for each region in the problem. The example shows the form of the command for region number 1. For any region "n" one need only replace "1" with "n". AP will produce its own set of group commands containing the actual list of elements for each region. These AP generated lists must appear in the problem.inp for each region.

INflow

Purpose: Indentifies the attributes of an inflow boundary.

```
Example: INflow Name tube_inlet U_velocity=1.0 UFunction=screen_u &
               V_velocity=0.0, W_velocity=0.0 Temperature=300. &
               Species=1.0,0.0,1.0 SFunction=screen_s1,dummy,screen_s3


         in n tube_inlet u 1 uf=screen_u v=0 w=0 t=300 &
            s=1,0,1 sf=screen_s1,dummy_s2,screen_s3
```

Remarks: One INflow command must be present for each set of inflow attributes. There may be multiple inflow boundaries.

The example defines an inflow boundary named "tube_inlet" having a u-velocity distribution defined by the user specified piecewise linear function "screen_u". v and w-velocities are 0.0, the temperature is 300 K and species 1 and 3 have mole fraction distributions defined by the user specified piecewise linear functions "screen_s1" and "screen_s3." Note that in this 3 species example, three functions were specified at the boundary even though one of the species, species 2, is not present. Species 2 was assigned a dummy function named "dummy_s2" which must be defined elswhere in the input file. The functions "screen_s1" or "screen_s2" could have been substituted for "dummy_s2." Note also that we also have the flexibility to assign functions for V_velocity, W_velocity, and Temperature using the keyword "Vfunction", WFunction", and TFunction" respectively. Whenever a function is specified for a boundary quantity, the constant following the name of the boundary quantity (1.0 in the above case for U_velocity) serves a multiplier for the value interpolated from the piecewise linear function.

INITial

Purpose: Assign flow field initial conditions at problem startup.

Example:
```
INITial U_velocity=0 V_velocity=0 W_velocity=0 P=1333. &
        Temperature=300. Species=.1,.2,.7
INIT u=0,v=0,w=0,p=1333,t=300,s=.1,.2,.7
```

Remarks: One INITial command must be present for all problems. Initial conditions for all solved variables must be provided. All initial conditions are applied uniformly over the flow field at problem startup. On restarts the INITial command must be present but it has no effect except to set the pressure level. Other field variables take on the values specified in the "restart_in.bin" file.

LAST_iteration

Purpose: Specifies the last SIMPLER iteration to be taken before advancing the time step.

Example:
```
LAST_iteration 1000
LAST=1000
```

Remarks: One of these commands must be present for all problems. In a steady state problem, iterations will stop when the iteration count reaches the LAST_iteration number or when all residual values fall below the convergence parameter set internally in the code. In a transient problem, reaching the LAST_iteration count causes the time step to be advanced and the iteration count to be reset to zero. Time step advancement and counter reset will also occur if convergence is achieve during a time step.

MATerial

Purpose: Identifies the material model for a particular region.

Example:
```
MATerial Region 3 Name chemkin
mat r=3 n=chemkin
```

Remarks: One of these statements must be present for each region in the problem. The only user-supported material model at this writing is the CHEMKIN model.

Node

Purpose: Specifies nodal coordinates for a node.

Remarks: These commands are generated automatically by the mesh generator AP. They are a required part of the "mesh.inp" file. There should be one Node command for each control volume in the problem.

PATH

Purpose: Lists the segments which make up a user defined path.

Example:
```
PATH Name part_surface Segments 1,2,3,7,8,10,11,45,49
path n=part_surface s=1:3,7,8:11,45,49
```

Remarks: One PATH command must be present for each user-defined path. User-defined paths are useful in printing out flow streamline information over the flow domain and heat flux, species flux, and site fractions along surfaces. The naming of paths ("part_surface" in the example) is arbitrary. Paths are made up of user-defined segments (see SEGment command). Paths must be continuous with no missing segments, and the orgin of each segment must be taken into account when linking the segments in the list. In other words, the orgin of any segment in the list must coincide with the end of the previous segment in the list.

PRINT_frequency

Purpose: Specifies how often to print output and restart files.

Example:

```
PRINT_frequency 100
print=100
```

Remarks: One PRINT_frequency command must be present for all steady state problems. If the command is missing in a transient calculation printing will default to the completion of each time step. More frequent printing can occur if the command is present and providing the PRINT_frequency parameter is less than the LAST_iteration parameter.

RELaxation

Purpose: Specify SIMPLER relaxation parameters for all variables.

Example:

```
RELaxation U_velocity=0.5 V_velocity=0.5 W_velocity=0.5 &
     P=1.0 P_Correction=1.0 Temperature=0.3 Species=.4,.4
REL u=.5 v=.5 w=.5 p=1 p_c=1 t=.3 s=.4,.4
```

Remarks: One RELaxation command must be present for all problems. Relaxation parameters between 0.0 and 1.0 must be specified for each "switched on" variable in the problem. For no relaxation, use 1.0.

SURface

Purpose: Indentifies the attributes of a surface.

Example 1:

```
SURface Name disk ISOthermal=ON Temperature=1.0 TFunction=tdisk &
      REAction=ON Material=2 Site_fractions=.9,.06,.04 &
      ROTation=ON Omega=3.142
```

```
          sur n iso=on disk t=1.0 tf=tdisk rea=on m=2 s=.9,.06,.04 rot=on o=3.142

Example 2: SURface Name wall ISOthermal=OFF REAction=OFF ROTation=OFF
           sur n wall
```

Remarks: One SURface command must be present for each surface type. Example 1 describes a surface named "disk" as having a temperature distribution defined by the user specified piecewise linear function "tdisk." The temperature at any point along "disk" is the product of the constant following the Temperature keyword (in this case 1.0) and the value interpolated from the piecewise linear function "tdisk." A simple isothermal surface is easily specified by turning the ISOthermal flag "ON", specifying the isothermal temperature after the Temperature keyword, and not supplying a TFunction command. An adiabatic surface is specified by turning the ISOthermal keyword "OFF." If the surface is declared isothermal the Temperature and TFunction keywords are ignored. Example 1 also indicates that "disk" is a reacting surface having surface chemistry mechanism number 2 (material 2 in the surf.inp file). The best guess for the three surface species site fractions are .9,.06 and .04. One site fraction guess is required for each surface species in the surface chemistry mechanism. Finally in Example 1, the "disk" surface is described as rotating with a speed of 3.142 radians per second.

Example 2 describes a simple adiabatic surface named "wall." Note that the short forms of Examples 1 and 2 make use of the default settings for ISOthermal, REAction, and ROTation. The default settings for these keywords is "OFF", i.e. not including the keywords implies that they are turned "OFF."

An additional keyword "Accom" (surface accomodation factor) is available for specifying slip flow boundary conditions at surfaces. Users interested in this feature should contact the authors.

SWITCH

Purpose: "Switches" a code feature "on" or "off."

Example 1: SWITCH ON Restart
switch on r

Example 2: SWITCH OFF Restart

Remarks: The following simple table lists all the "switchable" features of the code:

FEATURE KEYWORD	DEFAULT	DESCRIPTION
Restart	OFF	Designates a run as a restart
MASS_adjustment	OFF	Provides mass adjustment at outflows
U_velocity	OFF	Activates "u" momentum equation
V_velocity	OFF	Activates "v" momentum equation
W_velocity	OFF	Activates "w" momentum equation
P	OFF	Activates pressure equation
P_Correction	OFF	Activates pressure correction equation
T_emperature	OFF	Activates energy equation
Species	OFF	Activates species equations
Chemistry	OFF	Activates gas phase chemistry
MULTicomponent_diffusion	OFF	Provides for multicomponent diffusion
THERMAL_diffusion	OFF	Provides for thermal diffusion
PLANar_geometry	OFF	Planar (not axisymmetric) geometry
NET_flow	ON	Activates "net flow" option at inlet
Dump	OFF	Enables printing of ".out" files
Hybrid_differencing	OFF	Provides for hybrid spacial differencing
Quick_differencing	OFF	Provides for quick spacial differencing
GMRES_Pressure	OFF	Enables GMRES acceleration of Pressure
GMRES_P_Correction	OFF	Enables GMRES acceleration of P_C
GMRES_Species	OFF	Enables GMRES acceleration of Species eqns.
GMRES_Temperature	OFF	Enables GMRES acceleration of Energy eqn.

Switch commands that place switches in their default positions need not be included in the input file.

SEGment

Purpose: Defines a line segment.

Example: SEGment Number 12 Region 4 Side North Orgin East

```
seg n 12 r 4 s n o e
```

Remarks: A segment is a straight line that coincides with the side of a region. In the example, segment 12 is defined as a line which runs along the north side of region 4. The orgin of the segment is "east", that is the segment is said to run from "east" to "west" along the "north" side of region 4. The key word qualifiers which follow the Side and Orgin keywords are the compass directions North, South, East and West. They may be abreviated using n,s,e and w respectively.
Segments are joined together to form "paths" (see description of the path command) which can then be used to output the variation of a quantity like heat flux, mass flux, site fractions along a surface.

STREAMline

Purpose: Specifies which paths will be used for the streamline calculation.

Example 1: STREAMline PATHnames perimeter

```
stream path perimeter
```

Example 2: STREAMline PATHnames outside part_1_surface part_2_surface

```
stream path outside part_1_surface part_2_surface
```

Remarks: This command is required if streamlines are to be displayed as part of the output. The software that computes streamlines must know the path surrounding the flow field. In Example 1, a path called "perimeter" is just such a path. (Also see the PATH and SEGment commands.) The path may be defined to circle the flow field in either a clockwise or counterclockwise fashion. Multiple pathnames may be listed for those situations in which internal solid obstructions are imbeded in the flow field. This is illustrated in Example 2, where the flowfield is surrounded by a path named "outside" and contains two solid objects surrounded by paths named "part_1_surface" and "part_2_surface."

SPECies

Purpose: Defines the maximum number of gas phase and surface species.

Example 1:
```
SPECies Gas 6 Surface 5
spec g=6 s=5
```

Example 2:
```
SPECies Gas 4 Surface 8,3,7
spec g=5 s=8,3,7
```

Remarks: One SPECies command is required for each problem. In Example 1, there are six gas phase species and one surface chemistry mechanism having five surface species. In Example 2, there are four gas phase species and three different surface mechanisms having eight, three, and seven surface species respectively.

TITle

Purpose: Assigns a title to the database produced in the "ap.bin" file.

Example:
```
TITle "CURRENT Example Problem"
tit "CURRENT Example Problem"
```

Remarks: This command is optional. The title phrase must be enclosed in quotes if spaces are present.

TRANSient

Purpose: Declares a problem to be transient (not steady state) and provides start time, stop time and timestep parameters.

Example:
```
TRANSient START_time 0.0 STOP_time 10.0 STEP_time 0.1
trans start 0 stop 10 step .1
```

Remarks: This command is required for all transient problems.

Distribution

EXTERNAL DISTRIBUTION:

N. S. Alvi
SEMATECH
2706 Montopolis Drive
Austin, TX 78741-6499

S. Ashby
Lawrence Livermore National Laboratory
M/S L-316
P. O. Box 808
Livermore, CA 94551-0808

R. Barker
AFMA
1150 Seventeenth St., NW, Suite 310
Washington, DC 20036

Prof. T. Cale
ERC 175
Chemical/Bio & Materials Eng.
Arizona State University
Tempe, AZ 85287-6006

M. R. Chandrachood
Applied Materials
3100 Bowers Ave.
M/S 0225
Santa Clara, CA 95054

Prof. J. Y. Chen
Dept. of Mech. Eng.
University of California
Berkeley, CA 94720

Prof. C. Crowe
School of Mechanical & Materials Eng.
P.O. Box 642920
Washington State University
Pullman, WA 99164-2920

Prof. D. S. Dandy
Colorado State University
Dept. Agriculture and Chem Eng.
Fort Collins, 80523

Prof. H. A. Dwyer
Dept of Aero. and Mech. Eng.
University of California
Davis, CA 95616

Prof. R. W. Dutton
CIS Extension, Room 333
Integrated Circuits Laboratory
Stanford University
Stanford, CA 94305-4075

E. Egan
Motorola Semiconductor Products Sector
2200 West Broadway Rd., MS350
Mesa, AZ 85202

Prof. A. F. Emery
Department of Mech. Eng.
Box 352600
University of Washington
Seattle, WA 98195-2600

Prof. D. L. Flamm
Dept. EECS
187M Cory Hall
University of California
Berkeley,CA 94596.

T. Francis
Applied Materials, MS 0225
3100 Bowers Ave.
Santa Clara, CA 95054

P. Gadgil
Genus, Inc.
1139 Karlstad Drive
Sunnyvale, CA 94089

R. Geels
SDL, Inc.
80 Rose Orchard Way
San Jose, CA 95134-1365

F. Geyling
SEMATECH
2706 Montopolis Drive
Austin, TX 78741-6499

Prof. D. Goodwin
Mail Code 104-44
Mech. Eng. & Applied Physics
California Institute of Technology
Pasadena, CA 91125

S. A. Gokoglu
Lewis Research Center, MS 106-1
NASA
Cleveland, OH 44135

Prof. R. Greif
Dept. of Mech. Eng.
University of California
Berkeley, CA 94720

M. J. Hartig
Motorola
3501 Ed Bluestein Blvd., MD: K10
Austin, TX 78721

Prof. K. F. Jensen
Dept. Chem. Eng. MIT 66-566
Massachusetts Institute of Technology
Cambridge, Mass 02139-4307

S. Joh
Novellus Systems, Inc.
3970 N. 1st St.
M/S 251
San Jose, CA 95134

Prof. R. J. Kee
Engineering Division
Colorado School of Mines
Golden, CO 80401-1887

Prof. C. R. Kleijn
Delft University of Technology
Kramers Laboratorium voor Fysische Technologie
Prins Bernhardlaan 6, 2628 BW Delft
The Netherlands

Karson Knutson
AG Associates
4425 Fortran Drive
San Jose, CA 95134-2300

R. Kraft
Texas Instruments
13536 N Central Expressway 75243
P. O. Box 655012
MS 944
Dallas, Texas 75265

A. Krishnan
CFD Research Corporation
3325 Triana Blvd.
Huntsville, AL 35805

A. Labun
Digital Equipment Corporation
77 Reed Road, HLO2-3/J09
Hudson, MA 01749-2895

D. Lindholm
SEMATECH
2706 Montopolis Drive
Austin, TX 78741-6499

H. Liu
Hewlett Packard
370 W. Trimble Road, MS 91MJ
San Jose, CA. 95131-1008

P. McHugh
Semitool
Thermal Products Division
655 West Reserve Drive
Kalispell, Montana 59901

M. Meyyappan
Scientific Research Associates, Inc.
P. O. Box 1058
Glastonbury, CT 06033

S. Mokhtari
Watkins-Johnson Company
440 Kings Village Road
Scotts Valley, CA 95066-4081

J. Y. Murthy
Centerra Resource Park
10 Cavendish Court
Lebanon, NH 03766

Prof. S. V. Patankar
Dept. of Mech. Eng.
University of Minnesota
111 Church St.
Minneapolis, MN 55455

I. Raaijmakers
ASM America, Inc.
4302 E. Broadway
Phoenix, AZ 85040

Prof. A. A. Rodriguez
Dept. of Electrical Eng.
Center for Systems Science-GWC 612
Arizona State University
Box 877606
Tempe, AZ 85287-7606

Prof. D. Rosner
Chemical Engineering Dept.
Yale
P.O. Box 2159 YS
New Haven, CT 06520

S. Shankar
Intel
Mail Stop RN2-40
2200 Mission College Blvd.
Santa Clara, CA 95052

C. Shelton
3M Industrial & Electronic Sector
Bldg. 60-IN-01
St. Paul, MN 55144-1000

V. Singh
Lam Research Corp.
4650 Cushing Parkway
Fremont, CA 94538-6470

V. Starov
Gasonics Intl.
2730 Junction Ave.
San Jose, CA 95134-1909

P. Vitello
P. O. Box 808
L-296
Lawrence Livermore National Laboratory
Livermore, CA 94551

G. W. Weigand
U. S. DOE
1000 Independence Ave., SW
Room 4A-043 (DP1.1)
Washington, DC 20585

C. Werner
Siemens AG, Dept. ZFET ME3
D-81730 Munich
Germany

G. Wilson
Semitool
Thermal Products Division
655 West Reserve Drive
Kalispell, Montana 59901

P. Zawadzki
Emcore, Corporation
35 Elizabeth Avenue
Somerset, NJ 08873

INTERNAL DISTRIBUTION:

MS 0601	1126	W. G. Breiland	
MS 0601	1126	M. E. Coltrin	
MS 0601	1126	P. Ho	
MS 0601	1126	H. K. Moffat	
MS 0601	1126	J. Y. Tsao	
MS 0603	1314	P. Esherick	
MS 9001	8000	T. Hunter	
		Attn:	J. B. Wright, 2200
			M. E. John, 8100
			R. C. Wayne, 8400
			P. N. Smith, 8500
			P. E. Brewer, 8800
			D. L. Crawford, 8900
MS 9214	8117	W. E. Mason	
MS 9214	8117	J. Meza	
MS 9420	8200	A. L. West	
		Attn:	R. E. Stoltz, 8202
			J. M. Hruby, 8230
			A. J. West, 8240

MS 9409	8250	R. H. Stulen
MS 9054	8300	W. J. Mclean
		Attn: C. W. Robinson, 8301
		W. Bauer, 8302
		G. A. Fisk, 8355
MS 9051	8351	L. A. Rahn
MS 9051	8351	J. H. Chen
MS 9051	8351	A. R. Kerstein
MS 9055	8353	F. P. Tully
MS 9052	8361	D. R. Hardesty
MS 9052	8361	M. D. Allendorf
MS 9053	8362	R. W. Carling
MS 9053	8362	S. R. Vosen
MS 9053	8366	R. J. Gallagher
MS 9042	8345	C. W. Hartwig
MS 9042	8345	G. H. Evans (15)
MS 9042	8345	J. F. Grcar
MS 9042	8345	S. Griffiths
MS 9042	8345	W. G. Houf
MS 9042	8345	R. S. Larson
MS 9042	8345	A. E. Lutz
MS 9042	8345	E. Meeks
MS 9042	8345	C. D. Moen (7)
MS 9042	8345	B. Nilson
MS 9042	8345	F. Rupley
MS 9042	8345	P. Spence
MS 9042	8345	A. Ting
MS 9042	8345	W. S. Winters (20)
MS 9405	8700	M. T. Dyer
		Attn: M. W. Perra
		M. I. Baskes
		J. C. F. Wang
		K. L. Wilson
		W. G. Wolfer
		M. R. Birnbaum
MS 9042	8742	V. K. Gabrielson
MS 9405	8743	P. E. Nielan
MS 9405	8743	L. A. Bertram
MS 9405	8743	D. R. Chenoweth
MS 9405	8743	M. P. Kanouff
MS 9011	8920	K. Perano
MS 0151	9000	Gerold Yonas
		Attn: J. S. Rottler, 9003
		W. Camp, 9200
MS 0841	9100	P. L. Hommert
MS 0833	9103	J. H. Biffle
MS 0826	9111	D. K. Gartling
MS 0826	9111	W. L. Hermina
MS 0826	9111	S. Kempka

MS 0826	9111	J. Schutt
MS 0826	9111	R. Shunk
MS 0834	9112	A. C. Ratzel
MS 0834	9112	M. R. Baer
MS 0834	9112	M. Martinez
MS 0834	9112	R. R. Torczynski
MS 0835	9113	B. Blackwell
MS 0835	9113	R. Cochran
MS 0827	9114	T. J. Bartel
MS 0834	9114	R. B. Campbell
MS 0834	9114	A. S. Geller
MS 0827	9114	R. T. McGrath
MS 0827	9114	C. C. Wong
MS 0825	9115	F. Blottner
MS 0825	9115	B. Hassan
MS 0825	9115	W. H. Rutledge
MS 0836	9116	C. W. Peterson
MS 0836	9116	G. F. Homicz
MS 0836	9116	S. R. Tieszen
MS 0437	9118	L. M. Taylor
MS 1111	9221	S. S. Dosanjh
MS 1111	9221	J. N. Shadid
MS 1111	9221	A. G. Salinger

MS 9021	Technical Communications Department, 8815/for OSTI (10)
MS 9021	Technical Communications Department, 8815/Technical Library, MS0899, 4414
MS 0899	Technical Library, 4414 (4)
MS 9018	Central Technical Files, 8940-2 (3)