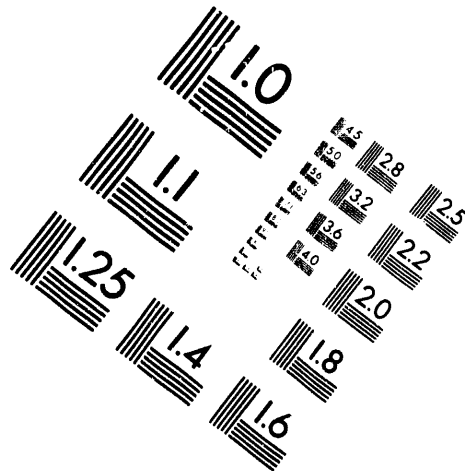
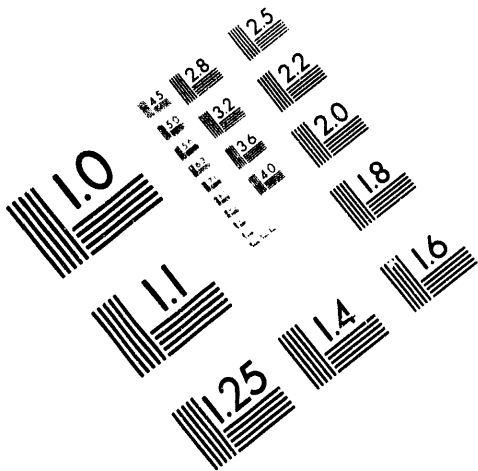




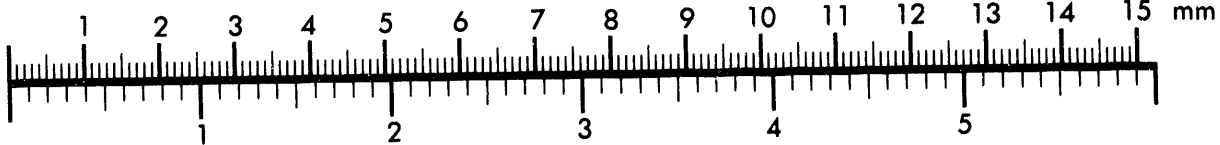
**AIM**

**Association for Information and Image Management**

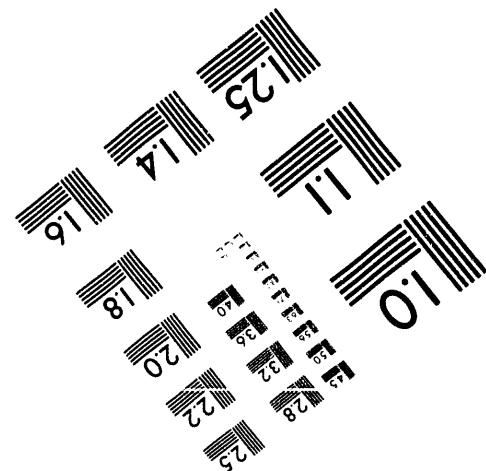
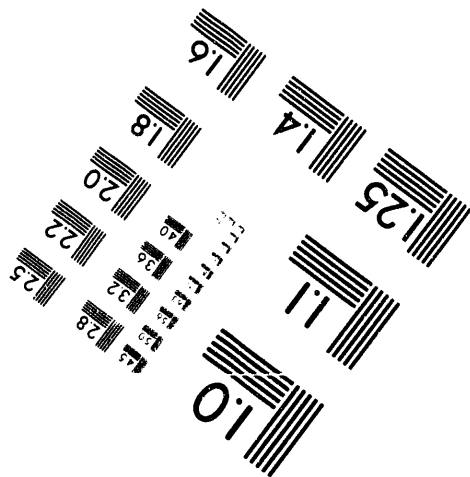
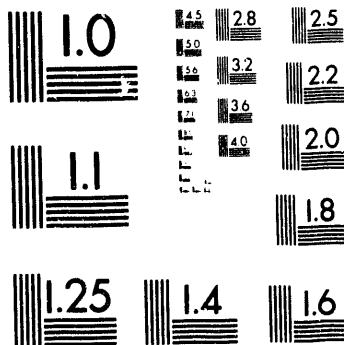
1100 Wayne Avenue, Suite 1100  
Silver Spring, Maryland 20910  
301/587-8202



Centimeter



Inches



MANUFACTURED TO AIM STANDARDS  
BY APPLIED IMAGE, INC.

**1 of 1**

2  
K & D

11-2  
3

DE-AC22-92PC91346-2

**DEVELOPMENT OF THE  
INTEGRATED ENVIRONMENTAL CONTROL MODEL**

**Quarterly Progress Report**

to

**Pittsburgh Energy Technology Center  
U.S. Department of Energy  
Pittsburgh, PA 15236**

from

**Center for Energy and Environmental Studies  
Carnegie Mellon University  
Pittsburgh, PA 15213**

**Prepared by  
E.S. Rubin, Principal Investigator  
M.B. Berkenpas  
J. Kalagnanam**

**April 1993**

**DISCLAIMER**

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

**MASTER**

**DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED**

EP

## Table of Contents

<b>1.0</b>	<b>Introduction</b>	<b>1</b>
<b>2.0</b>	<b>Personnel Training</b>	<b>1</b>
<b>3.0.</b>	<b>Model Software Enhancements</b>	<b>2</b>
<b>3.1</b>	<b>Demos Improvements</b>	<b>2</b>
	3.1.1 Terminology	2
	3.1.2 Navigation	3
	3.1.3 Modularity	5
	3.1.4 Development Tools	6
	3.1.5 Interface	7
<b>3.2</b>	<b>Model Use Improvements</b>	<b>8</b>
	3.2.1 Titles	8
	3.2.2 Integrated Modules	8
	3.2.3 Technology Models	8
	3.2.4 Configuring a Power Plant	8
<b>4.0</b>	<b>Technology Performance and Cost Models</b>	<b>9</b>
<b>4.1</b>	<b>FGD Performance Models</b>	<b>10</b>
<b>4.1.1</b>	<b>Limestone with Forced Oxidation</b>	<b>11</b>
<b>4.1.2</b>	<b>Limestone with Dibasic Acid</b>	<b>11</b>
<b>4.2</b>	<b>FGD Cost Models</b>	<b>12</b>

## **1.0 Introduction**

The purpose of this contract is to develop and refine the Integrated Environmental Control Model (IECM) created and enhanced by Carnegie Mellon University (CMU) for the U.S. Department of Energy's Pittsburgh Energy Technology Center (DOE/PETC) under contract Numbers DE-FG22-83PC60271 and DE-AC22-87PC79864.

In its current configuration, the IECM provides a capability to model various conventional and advanced processes for controlling air pollutant emissions from coal-fired power plants before, during, or after combustion. The principal purpose of the model is to calculate the performance, emissions, and cost of power plant configurations employing alternative environmental control methods. The model consists of various control technology modules, which may be integrated into a complete utility plant in any desired combination. In contrast to conventional deterministic models, the IECM offers the unique capability to assign probabilistic values to all model input parameters, and to obtain probabilistic outputs in the form of cumulative distribution functions indicating the likelihood of different costs and performance results.

The most recent version of the IECM, implemented on a Macintosh II computer, was delivered to DOE/PETC at the end of the last contract in May 1991. The current contract will continue the model development effort to provide DOE/PETC with improved model capabilities, including new software developments to facilitate model use and new technical capabilities for analysis of environmental control technologies. Integrated environmental control systems involving pre-combustion, combustion, and post-combustion control methods will be considered.

The work in this contract is divided into two phases. Phase I deals with further developing the existing version of the IECM and training PETC personnel on the effective use of the model. Phase II deals with creating new technology modules, linking the IECM with PETC databases, and training PETC personnel on the effective use of the updated model.

The present report summarizes recent progress on the Phase I effort during the period January 1, 1993 through March 31, 1993. This report summarizes activities in three main areas: (1) personnel training (Task 2); (2) software development (Task 4); and (3) updates of analytical models (Task 5).

## **2.0 Personnel Training**

Task 2 of the current contract calls for the training of DOE/PETC personnel in the operation of the Integrated Environmental Control Model. These training efforts were initiated during the past quarter. Discussions with the PETC COTR indicated that the

best method of proceeding would be to initiate a training session focused on basic operation of the model, then follow this several months later with more intensive sessions that include model operation.

In February of 1993 a one day session was conducted by Michael Berkenpas of Carnegie Mellon University at the PETC site. This session was attended by the PETC COTR, four DOE trainees, and two DOE observers. The formal presentation covered an overview of the IECM models, the Demos modeling program, and recent changes made to the IECM models. The EPRI Chemical Assessment Model was demonstrated briefly as an example of an alternative interface to the IECM models. An informal discussion of interfaces ended the meeting. Current plans are to resume the personnel training in June or July 1993 for the four DOE trainees.

### **3.0 Model Software Enhancements**

This section of the report summarizes improvements to the Integrated Environmental Control Model (IECM) and underlying software. The IECM models are built and run with the Demos application (Decision Modeling System). Demos was developed during the 1980s on UNIX systems and was translated for use on Macintosh systems in the late 1980s. Demos is a versatile modeling environment that combines ease of use with advanced uncertainty analysis, graphics display, and multi-component integration.

Version 1.6 of the Demos program copyrighted to Carnegie Mellon University was delivered to PETC in May 1991 under contract DE-AC22-87PC79864. Since then, the Demos program has been copyrighted by Lumina Decisions, Inc. and developed into a commercial product that will be marketed in the middle of 1993. Licensing and purchase of the most current Demos version 2.5 thus must be negotiated by DOE/PETC with Lumina Decisions, Inc. Recent improvements to Demos have changed the way the IECM models are used and built. The changes to Demos and the impact on IECM models are discussed below.

### **3.1 Demos Improvements**

#### **3.1.1 Terminology**

Terminology changes in describing variables, submodels, models, modules, libraries, and overall projects in Demos now correspond directly with the IECM terminology. These names better describe the hierarchy of the IECM in a clear and concise manner. These terms are described below, using Figures 1 and 2 as illustrative examples.

*Variable* A variable is the simplest and most often used object. It describes the title, units, description, and definition of the object. The variable contains the relationships between other variables. For example, the front-most

window in Figure 1 shows the variable "Fluegaschar" which describes and defines a summary table of several flue gas performance characteristics. Variables are displayed in Demos by thin boundaries as shown in the second window in Figure 1.

*Model* A model embodies or groups a collection of variables and submodels together for the purpose of organization and structure. For example, the second window from the back of Figure 1 shows three models which separate the "Plant" submodels and variables into "Plant Performance Inputs", "Plant Performance" and "Plant Economics". Models are displayed in Demos by bold or thick boundaries.

*Submodel* A submodel is a model located inside another model. These submodels are more specific in nature than their parent models. A submodel is generally used to describe a subset of a larger model. For example, the middle window of Figure 1 shows numerous submodels of the "Plant Performance" model.

*Module* A module is a very special model that contains all the objects describing a technology system. This information is linked to a separate computer file. This allows the IECM to be used and modified in an integrated way. Figure 2 shows two modules, "Initial" and "Plant", which are noted graphically by a file symbol inside the model symbol.

*Library* A library is a module that contains functions only. In addition to being located in separate computer files, functions in the library appear under a separate menu in the Demos program. This facilitates their use by the user substantially.

*Project* A project is a file combining all the technology modules necessary to describe a complete power plant. The user executes projects when using the IECM. Figures 1 and 2 show an "Uncontrolled Plant" module. Projects are new to Demos. Previous versions of the IECM used "loadfiles" to mimic projects.

The addition of "module", "library", and "project" have made substantial structural advancements possible in the IECM models. Specific examples of the improvements to the IECM file structure are discussed below in the "Modularity" section.

### **3.1.2 Navigation**

The IECM models are complex in nature due to the large number of objects that define the models. The average number of objects in an IECM project exceeds 750. Previous versions of Demos limited the navigation paths possible, forcing the user to memorize the hierarchy structure and variable names. These navigation changes have substantially improved the "use-ability" of the IECM models and require less knowledge of the structure and contents of the IECM. To quickly and efficiently navigate to or between the

object(s) of interest, several important improvements have been made in Demos. These navigation changes can be classified in five groups: a hierarchy diagram, object alias', common navigation buttons, a "Find" menu command, and input/output links.

### *Hierarchy Diagram*

Figure 2 shows a typical hierarchy diagram. This is referred to as a "list view" and resembles an outline. It mimics the list view used by the Macintosh operating system in displaying disk and folder contents. The triangles shown in the left column open and close models when clicked. For example, the "Plant" and "Plant\_perf" models were clicked in Figure 2 to display their contents. When a model or variable of interest is found, double-clicking its symbol or name will open its diagram.

### *Alias*

An alias is a symbolic link to another object, essentially representing the original object in another location and used as if it were the original object. This allows the user to have quick access to an object from any location. This is also a feature of the Macintosh operating system. Alias objects are represented with italicized titles.

### *Navigation Buttons*

Figure 3 shows the two navigation buttons present on every window in Demos for consistency. The top button opens the parent diagram window. For example, clicking on the top navigation button of the top-most window in Figure 2 would bring the second window in Figure 2 to the top.

The bottom navigation button opens the object window of the selected object. For example, the top-most window (an object window) in Figure 2 was opened by clicking on "Flue Gas Summary Table" followed by clicking on the bottom navigation button in the "Plant Performance Summary Tables" window of Figure 2.

### *Find Command*

The "Find" command in Demos has been improved. It is accessed from a menu in the Demos program. This command requires the user to type the name of the object entirely. The result is that the diagram window containing the object opens, with the object highlighted.

### *Input/Output Links*

Every object window contains a list of input and output variables which relate to that object. For example, the top window in Figure 2 shows the inputs to the "Fluegaschar" variable. To navigate to any one of these input variables, the user simply double clicks the symbol or title of the input variable. This particular example shows that the "Fluegaschar" variable has no output variables (output variables are variables which reference the particular object being viewed).

The input/output links feature is perhaps the most useful navigation tool available. It allows the user to go to variables directly related to the object of interest. As will be seen later in the “Development Tools” section, these input and output lists for navigation purposes can also be accessed from the diagram window.

### **3.1.3 Modularity**

#### *Modules*

Previous versions of the IECM were integrated in principle but still limited in flexibility. Several modifications were necessary to link files together into modules and also to link them into a final structure. Although these technologies were integrated, they could only be linked together in a few prescribed configurations. The most difficult constraint in previous versions was the inability to modify models directly in Demos. Permanent changes required the user to open technology files with a word processor such as “Microsoft Word:”, search for the proper object, make the changes, save the file, and finally open the proper loadfile with Demos to test the changes. In short, the IECM was previously a “read-only” environment when using Demos. This required a substantial investment in time by each user to understand the structure of the files in addition to the model structure when loaded in Demos.

With the addition of module support in Demos, files can now be linked together and still retain their individual nature. The following example illustrates this point.

The ESP technology was split between two IECM files, “ESP.perf” and “ESP.econ”. These two files together were referred to as the “ESP module” in the documentation. However, after these two files were loaded into the Demos project via a “loadESP.mod” loadfile, they could never be saved back to their individual files. Their contents were permanently fused with other module files into one monolithic project file. In the current version of Demos, these two files can be modified to be fully self-contained and retain their distinct relationship to their original files even when saving the total project. This new working environment allows the user to make permanent changes to the ESP module and see these changes reflected in the proper IECM files for future use with other configurations.

#### *Function Libraries*

As described in the “Terminology” section above, function libraries are very similar to modules. The difference is that they contain functions only and display these functions in a menu of Demos. Many functions are defined in the IECM and can be found in the “Initial” module. Previously, IECM functions were more difficult to use, primarily due to the user’s inability to access them easily. Now, these functions are easily accessible with the necessary parameters self-documented.

### 3.1.4 Development Tools

The IECM models require many advanced tools to help the user make changes and test hypotheses quickly. Demos has added several very important tools to facilitate this need. With these additions, an inexperienced IECM user can now use and even develop models with relatively little effort. Although these improvements have little impact on recent changes to the IECM structure, they do impact heavily on the expectations of the model developer for the future. The model developer has increased flexibility and functionality using these tools. Several of the most important tools are itemized below.

#### *Uncertainty Editor*

Several uncertainty functions have been improved or added directly to Demos. Together with the other Demos and IECM functions, these are now accessed either from a menu directly or through an uncertainty editor. Figure 4 shows this editor. The editor shows an example diagram of the function, a description of the function, the necessary parameters, and text-entry area to enter the parameter values.

#### *Table Editor*

Tables are one of the most complex and difficult objects in Demos to use and understand. Due to their complex nature, a specific editor has been added to Demos. Figure 5 shows an example of the table editor with some data entered. The user must choose the indices or title groups to be used and then enter the definitions for each index item. These definitions can contain other variable names, functions, operators, and constants.

#### *Diagram/Node Styles*

Diagram windows display the models and variables of a particular model. Four diagram windows are shown in Figure 1. Each object is referred to as a “node”. In previous versions of Demos, the form or appearance of these could not be changed. The current version of Demos allows the arrows, font, borders, and label to be modified with the diagram and node editors shown in Figures 6 and 7.

#### *Definition View Editor*

The object window displays the variable name, units, title, description, definition, value, inputs, and outputs of a particular object. Figure 8 shows an example of an object window. Because definitions exhibit many forms, they can now be viewed in ways that facilitate these forms. As shown in Figure 8, a pop-up editor selection menu allows the definition to be displayed as an expression (the only option in previous versions of Demos), a number, a list, a table editor, an uncertainty editor, or a function editor.

#### *Attribute Editor*

The object window displays all the attributes of an object. An alternative method to opening an object window is available on each diagram window. A small toggle lever is present on the bottom-left of every diagram window which opens and closes an attribute editor when activated. Figure 9 shows this modified diagram window. A pop-up menu is

used to select the particular attribute of the highlighted object in the window. A text entry area appears below.

#### *Inter-Application Communications*

Demos can now communicate directly with other Macintosh applications such as Microsoft Excel or Hypercard and is referred to as inter-application communications (IAC). Commands and results can be sent to and from either Demos or the other application.

#### *Graph Editor*

The most effective method to display results is through graphs of important variables. Due to its importance, several improvements have been made to the Demos graph editor. Several new graph types have been added to the previous list. These are shown in the graph selection editor on the graph window shown in Figure 10. Statistics are shown in bar graph form in Figure 10 and the probability density diagram in Figure 11.

Graph styles are more versatile in the new version of Demos. Double-clicking the graph in Figure 11 opens one of two style selection views. These style changes allow the user to fine-tune a graph into a final form ready for presentation. The final graph can be saved or copied.

Table and Graph windows can be used to alter the indices used for multi-dimensioned variables. The graph shown in Figure 11 has only one index, therefore does not show an index editor as Figure 13 does. The flue gas summary table shown in Figure 13 has several indices and allows the user to specify which ones to use in displaying the table.

### **3.1.5 Interface**

Several smaller changes have been added to the Demos interface. These improvements are listed below. A short discussion of their importance for use of the IECM with Demos is discussed in each item.

#### *Icons*

Icons are small pictorial representations of a variable, table, or model. They convey very quickly to a user the concept or use of the object to which it is tied. These icons are presently not used in the IECM Demos interface. Although icon development would present a more intuitive visual aide, the large number of objects in the IECM would demand considerable time and effort to implement them.

#### *Intelligent Title Use*

Demos now uses titles in a very intelligent manner, using them to label tables and graphs in addition to the objects in every object window. Recently, titles have been created for some of the IECM module files. These titles are now used by the Demos interface as displayed in Figure 1. A further benefit is that the number of objects in the IECM

modules may be reduced. Tables required numerous index variables to be assigned to contain the titles as demonstrated in Figure 13. Now, the information in the index variables can be included in the table definition itself, requiring fewer index variables in the IECM modules.

#### *Print Reports*

Every window can be printed or saved to a disk file in Demos. A report combining the information from multiple windows is also available.

#### *User Input/Output Objects*

One of the most important interface additions to Demos is the input and output objects. These allow numbers to be entered or displayed in a diagram window without the use of a table or object window.

### **3.2 Model Use Improvements**

#### **3.2.1 Titles**

As discussed briefly in the “Intelligent Title Use” section above, titles are being added to all the objects in the IECM modules. These titles are used throughout the Demos interface and simplify the use of the IECM greatly.

#### **3.2.2 Integrated Modules**

Because a real power plant is modular in nature, the IECM files also should be modular. Recent changes described above in the “Terminology” section have enhanced the modular structure of the model.

#### **3.2.3 Technology Models**

The IECM was designed primarily for a person using the typescript. Variables were positioned in the hierarchy to minimize key strokes. For example, all the input parameters from all the technology files were put into separate submodels of one common model: “Allinputs”. The “allinputs” model was initialized with “initial.perf”, and reopened with each successive technology file. The contents of the “Allinputs” model were then “Plant”, “ESP”, “Fabric Filter”, etc. It was easier to change the inputs of several technologies this way, rather than navigating to an entirely new section of the hierarchy each time a different technology was needed. With the flexibility of the object windows and hierarchy diagram, it is no longer necessary to place all the input parameters together; in fact it is less efficient. All input submodels are now nested inside their primary technology model. This is fully compatible with the current version of Demos.

#### **3.2.4 Configuring a Power Plant**

The various plant configurations were assembled via “loadfiles” in previous versions of the IECM. The particular loadfile entered files in the proper order. The first file loaded (“initial.perf”) defined 3 nodes to connect the technology models to the skeleton flue gas

structure. The difficulty was determining how to place a component at a different node. For example, the fabric filter files could be attached to nodes 1-2 or 2-3. So, "link" or "patch" files were created that connected the 3 or 4 variables to the proper node number. Presently, even more alternatives to node placement of the same technologies are possible. These still cannot be implemented in the current version of Demos. A way to accomplish this is to create a project for each possible configuration. This project would contain the links to the necessary module files and the "link" parameters. Alternatives and new Demos features are currently being worked out. So too is the addition of more nodes to expand the flexibility for configuring new power plant designs.

#### **4.0 Technology Performance and Cost Models**

Task 5 of the current contract calls for updating and enhancing several of the existing performance and cost models in the IECM, especially several of the models for conventional coal-fired power plants are now out of date. In particular, the model of FGD system performance and cost, which is a major component of the conventional power plant environmental control costs, is to be updated under the current contract. In addition, models of the selective catalytic reduction (SCR) system and other elements of the conventional power plant will be studied later in this project. Over the past quarter our efforts have focused on conventional wet limestone flue gas desulfurization (FGD) systems. The main objective of the IECM modeling effort is to reflect the improvements in performance and changes in the cost experienced in the U.S. over the past five or six years.

The performance and cost models currently embedded in the IECM originally were derived from a detailed computer model developed by the Tennessee Valley Authority (TVA) in the late 1970s and early 1980s. The TVA model was last updated in about 1984, and used as the basis for development of the current IECM algorithms for performance and cost. In the late 1980s the IECM performance algorithms were revised to reflect the generally lower stoichiometry used in FGD systems relative to the higher values predicted by the TVA model. No revisions to the cost algorithms, however, were undertaken at that time. Recent work by the Electric Power Research Institute (EPRI) and others, however, now suggests that FGD costs are generally lower than costs estimated in the mid 1980s because of a number of key design changes incorporated into modern FGD systems. At the same time, the performance of FGD systems has improved to the point where removal efficiencies significantly in excess of 90 percent now are routine (in contrast to a maximum removal efficiency of 90 percent assumed by most studies during the 1980s). For this reason, it is necessary to revise both the performance and cost models of the IECM to properly reflect the capabilities and cost of modern FGD systems.

## 4.1 FGD Performance Models

Two FGD systems are analyzed: Limestone with Forced Oxidation (LSFO) and Dibasic Acid enhanced LSFO (LS/DBA). Limestone with forced oxidation is becoming the most common type of FGD system. A limestone slurry is used in an open spray tower with in-situ oxidation to remove  $\text{SO}_2$  and form a gypsum sludge. The main advantages as compared to conventional systems is easier dewatering, more economical disposal of scrubber products, and decreased scaling on tower walls. Limestone with DBA (LS/DBA) is a modification to LSFO where dibasic acid (DBA) is added to act as a buffer/catalyst in the open spray tower. The main advantages are increased  $\text{SO}_2$  removal and decreased liquid to gas ratio.

Advanced wet limestone FGD systems are now designed to achieve  $\text{SO}_2$  removal efficiencies in excess of 95%. The single loop countercurrent spray tower is the most commonly used device for the removal of  $\text{SO}_2$ . The design of spray towers for high efficiency without additives is achieved by using high liquid-to-gas (L/G) ratios and improving gas/liquid contact by spray nozzle design. On the other hand, organic acids such as dibasic acid or adipic acid are added as buffers to improve performance. In this section, brief descriptions and approximate relations for high performance LSFO and LS/DBA systems are provided. The approximate relations are gleaned from the literature to reflect US experience with high performance wet limestone FGD systems. The paragraph below provides some background for the scrubbing process which is useful in understanding the bases for the relations provided later.

The scrubbing process in the absorber is a mass transfer operation between streams in different phases.  $\text{SO}_2$  from the flue gas is absorbed into the scrubbing liquor and the acid formed is neutralized continuously by the limestone. The rate of this neutralization has a major effect on how quickly  $\text{SO}_2$  can be absorbed into the liquor. Processes for which the rate of neutralization of absorbed  $\text{SO}_2$  does not limit the rate at which  $\text{SO}_2$  is absorbed from the flue gas are said to be "gas-phase limited". This is because the rate of  $\text{SO}_2$  absorption depends largely on liquid/gas contact, which is a function of the total droplet area. On the other hand, the primary source of alkalinity in limestone FGD processes is in the solid phase as  $\text{CaCO}_3$ . The rate at which the solid phase dissolves determines the rate at which absorbed  $\text{SO}_2$  is neutralized and becomes the rate limiting step. Such systems are "liquid-phase limited. Efforts to improve performance are directed at (a) making the bulk of the liquor available to provide the low liquid-phase alkalinity to the gas by increasing the L/G ratio; (b) increasing the residence time of liquor in the absorber; and (c) buffering the effect of low pH using organic additives.

#### 4.1.1 Limestone with Forced Oxidation

LSFO is a wet limestone FGD process. Since it is a liquid-phase limited system, performance is improved by increasing the L/G ratio to improve the availability of the low alkalinity bulk liquor. Typically sulfur removal efficiency is related to L/G in an exponential relation. Industry practice is based on a log-linear relation between  $\eta_{SO_2}$  and L/G ratio. Typical values for  $\eta_{SO_2}$  and L/G are provided in the table below.

$\eta_{SO_2}$	L/G(gal/1000 acfm)
90%	90
95%	130

Based on these values an approximate relation between sulfur removal efficiency and L/G assuming a log-linear relationship is:

$$\eta_{SO_2} = 1 - \exp\{-(0.725 + 0.0175 \times L / G)\} \quad (1)$$

where L/G is measured in gal/1000 acfm. Notice that the constant in the exponential includes the effect of stoichiometry. The stoichiometry for wet limestone systems is generally fixed at either 1.05 or 1.10 and is not often used as a parameter to improve performance. Most advanced FGD systems are designed to utilize limestone efficiently.

#### 4.1.2 Limestone with Dibasic Acid

Several utilities have used organic buffers to improve the performance of wet limestone FGD systems. Dibasic acid is the most commonly used buffer in the US, and is composed of glutaric acid, succinic acid and adipic acid. Organic buffers suppress the normal decrease in pH in the liquid and, hence, maintain the driving force for absorbing SO<sub>2</sub>. Therefore organic additives lower the L/G required for a given level of performance. Once again we provide a very approximate relation to relate  $\eta_{SO_2}$  to the concentration of DBA in the bulk liquor. Based on current industry practice in the US, typically addition of 1000 ppm of DBA in the liquor reduces the L/G by about 30 for a given removal efficiency. This seems to be a fairly robust estimate for wide variations (1000-3000 ppm) inlet SO<sub>2</sub> concentrations. Based on this number, Equation (1) for  $\eta_{SO_2}$  can be modified to reflect the effect of DBA as follows:

$$\eta_{SO_2} = 1 - \exp\{-(0.725 + 0.0175 \times L / G + 0.00042 \times DBA)\} \quad (2)$$

where DBA represents the concentration of dibasic acid in ppm. Note that for a fixed L/G ratio the addition of DBA will increase the SO<sub>2</sub> removal efficiency. This is the most common way of achieving high SO<sub>2</sub> removal with limestone systems.

## 4.2 FGD Cost Models

The cost models that are being developed are centered largely around the recent work for EPRI which provides the most up to date cost models for 15 FGD processes. The cost development and breakdown follow the EPRI economic premises. Cost is broken down in terms of capital and operating cost. The bases and items included for each component of the capital estimate are provided in Table 1 from the EPRI manual. The process capital estimate is broken down on an area basis for each process. The cost areas are provided in Table 2 based on the EPRI manual. Operating costs for FGD systems are separated into fixed and variable operating costs. Fixed costs include operating and maintenance labor, maintenance materials and administrative/support labor. Various factors based on EPRI premises are used to estimate these costs based on capital cost estimate. Variable operating costs include consumables such as fuel, water, power, chemicals, and solids disposal. Table 3 provides the operating cost criteria used by EPRI.

In the work to follow, current information on capital and O&M costs will be used to revise the current IECM models using EPRI economic premises and nomenclature. Results of this work will be reported in the next quarterly report.

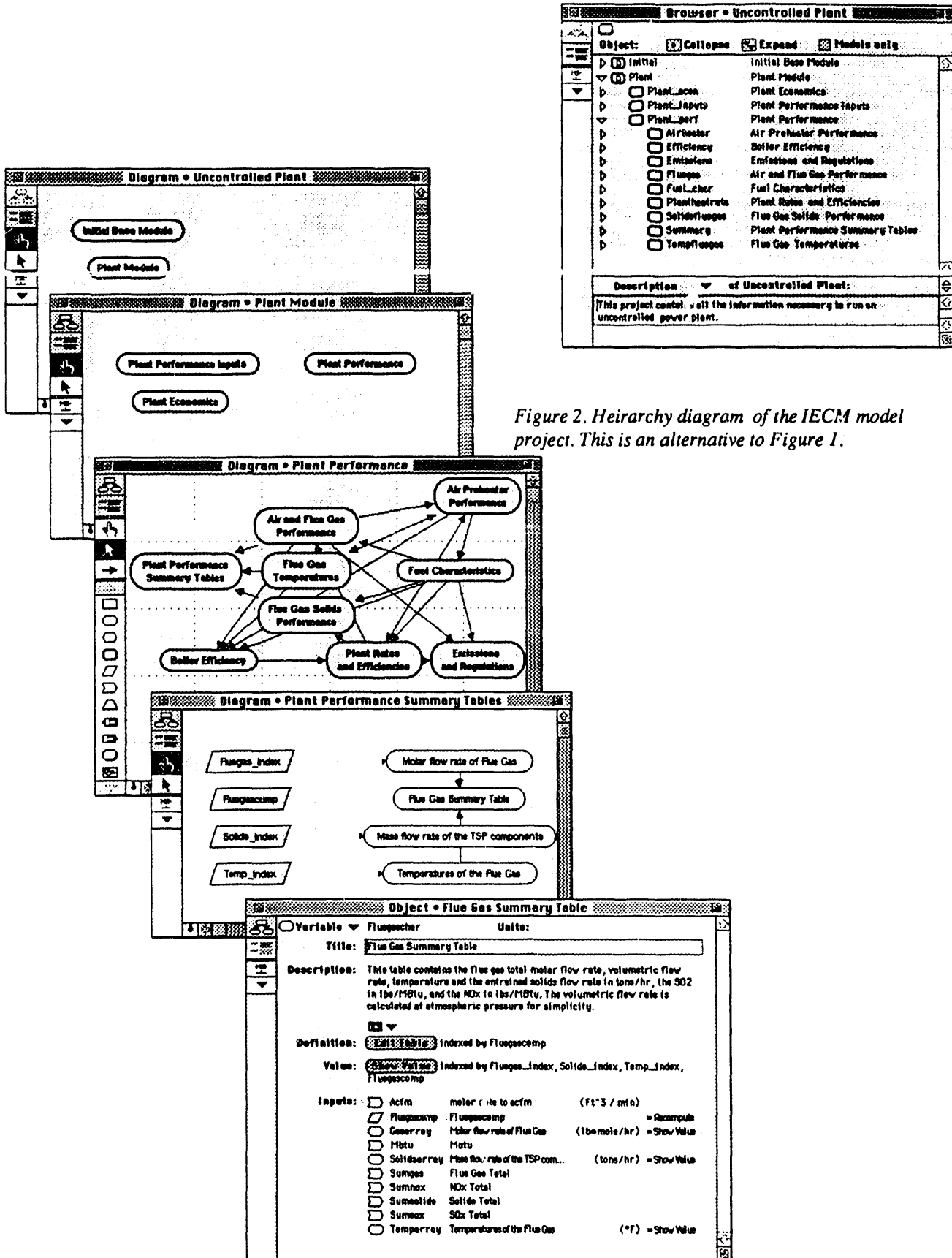


Figure 2. Hierarchy diagram of the IECM model project. This is an alternative to Figure 1.

Figure 1. Diagram windows displaying the hierarchy of the Uncontrolled Plant. Windows are opened in succession until the desired object is found.

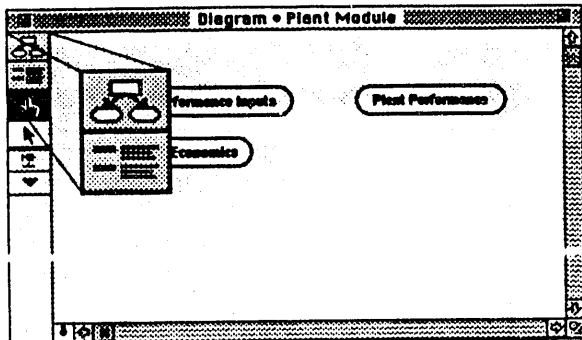


Figure 3. Two navigation buttons are present on every window.

Figure 4. An uncertainty editor is used to develop uncertainty functions.

Variable Name	Definition
FG, lbmole/hr	Sumgas (Gasarray)
FG, acfm	Acta (Sumgas (Gasarray), (Temparray+460), 0)
FG, mf	Temparray
Sulfur, lb/hr	Sumsoilds (Solidsarray)
SO <sub>2</sub> , lb/HBTU	Hb tu ((Sumsox (Gasarray) * 64.06))
NO <sub>x</sub> , lb/HBTU	Hb tu ((Sumnox (Gasarray) * 46.01))

Figure 5. A table editor is used to build tables of variables.

Figure 6. The diagram style editor applies to the front-most diagram window.

Figure 7. The diagram node editor applies to nodes in the front-most diagram window.

Input	Value
Loss (%)	0
Gas Loss (%)	5.7251
on Loss (%)	5.3114
unttd Losses (%)	0.1641
unttd Losses (%)	0.5

Figure 8. Variable definitions can be viewed or edited in many ways.

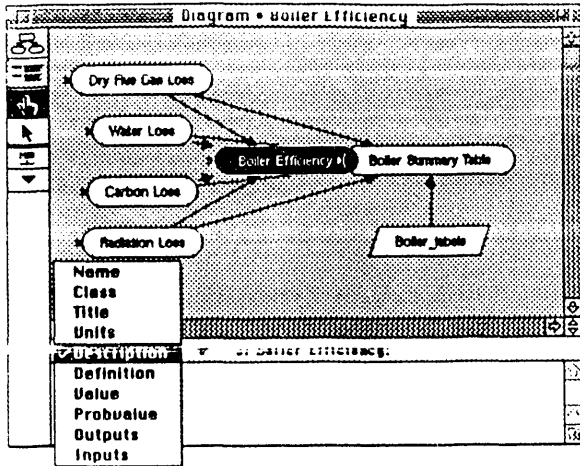


Figure 9. An attribute editor can be opened on each diagram window.

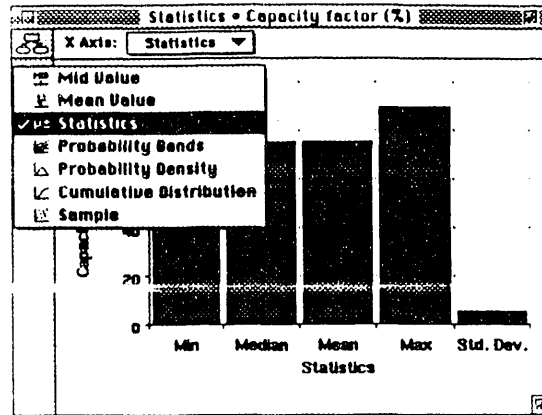


Figure 10. Uncertainty can be viewed easily in many forms with the graph editor.

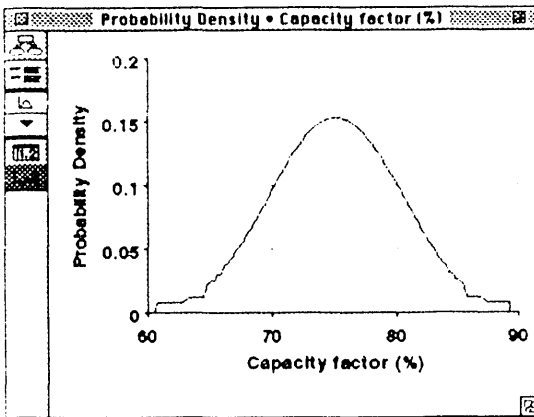


Figure 11. Graphs are dynamic, automatically reflecting data present in the variable title and units.

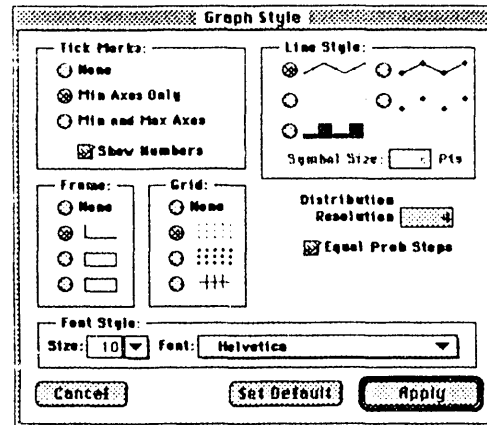


Figure 12. The style of the graph is easily modified.

	FG, lbmoles/hr	FG, scfm	FG, °f	Solids, tons/hr	SO <sub>2</sub> , lb/MBtu	NO <sub>x</sub> , lb/MBtu
Air into FD Fans	179.3K	1.178M	80	0	0	0
Air exiting APH	154.8K	1.017M	80	0	0	0
Air Leakage across APH	24.51K	161K	80	0	0	0
Gas to Furnace	163.6K	1.074M	80	0	8.3324	0.7357
Gas exiting Economizer	163.6K	1.074M	80	0	8.3324	0.7357
Gas into Air Preheater	163.6K	1.074M	80	0	8.3324	0.7357
Gas exiting Air Preheater	188.1K	1.235M	80	0	8.3324	0.7357
Fg_point1	188.1K	1.235M	80	0	8.3324	0.7357
Fg_point2	188.1K	1.235M	80	0	8.3324	0.7357
Fg_point3	188.1K	1.235M	80	0	8.3324	0.7357
Flue Gas into Stack	188.1K	1.235M	80	0	8.3324	0.7357

Figure 13. Tables display any combination of dimensions.

Table 1

ITEMS INCLUDED IN PROCESS CAPITAL COST ESTIMATE

Earthwork  
 Concrete  
 buildings and Structures  
 Process Equipment  
 Piping  
 Electrical  
 Painting  
 Instruments and Controls  
 Insulation  
 Direct Field Labor  
 Indirect Field Costs  
     Payroll Taxes  
     Insurance, Bonds  
     Construction Supplies  
     Temporary Facilities  
     Construction Equipment  
     Vendor Fees

FGD SYSTEM CAPITAL COST ESTIMATE COMPONENTS

<u>Capital Investment</u>	<u>\$/kw</u>
Process Capital (includes sales tax)	A
General Facilities	B
Engineering and Home Office Fees	C
Project Contingency	D
Process Contingency	E
Total Plant Cost, TPC	= A+B+C+D+E
Total Cash Expended, TCE	TPC x Adjust factor*
AFDC (Allowance for Funds During Construction)	F
Total Plant Investment, TPI	= TCE + F
Royalty Allowance	G
Preproduction Costs	H
Inventory Capital	I
Initial Catalyst and Chemicals	J
Land	K
Total Capital Requirement, TCR (Capital Investment Jan., 1990, Startup Jan., 1990)	<u>TPI+G+H+I+J+K</u>

\*Adjustment Factor for TCE per table 3-4 of EPRI TAG (P6587-L,  
 Reprinted in Section 4 of this report)

Table 2

COST AREAS FOR PROCESS CAPITAL BREAKDOWN

<u>Area</u>	<u>Description</u>
10	Reagent Feed System
20	SO <sub>2</sub> Removal System
30	Flue Gas System
40	Regeneration System
50	By-product System
60	Solids Handling System
70	General Support Area
80	Miscellaneous Equipment

- 10: Reagent Feed System - all equipment required for storage, handling and preparation of raw materials, reagents, and additives used in each process.
- 20: SO<sub>2</sub> Removal System - equipment required for SO<sub>2</sub> scrubbing, such as the absorption tower, recirculation pumps and other associated equipment.
- 30: Flue Gas System - duct work and fans required for flue gas distribution to the SO<sub>2</sub> scrubbing system, plus gas reheat as required.
- 40: Regeneration System - specific to regenerable reagent systems, equipment used to regenerate spent absorbent for return to the process, plus any preconditioning system for SO<sub>2</sub> or H<sub>2</sub>S off gas.
- 50: By-product System - production equipment for salable process by-products and storage facilities for the final products.
- 60: Solids Handling System - equipment required for fixation, treatment, and transportation of all sludge/dry solids materials produced by each scrubbing process.
- 70: General Support Area - additional equipment required to support FGD system operation such as makeup water and instrument air.
- 80: Miscellaneous Equipment - This area will include plant modifications necessitated by the addition of the FGD system. Also included are costs for electrical equipment tie-ins and other associated systems.

Table 3

## OPERATING COST CRITERIA

Fixed Operating Costs		Unit	Rate		
Operating Labor		Man-hrs	\$20.00 (Jan. 1990)		
Maintenance Labor - Slurry Handling		\$/yr	3.2% of Process Capital		
- Liquid Handling		\$/yr	1.2% of Process Capital		
Maintenance Material - Slurry Handling		\$/yr	4.8% of Process Capital		
- Liquid Handling		\$/yr	1.8% of Process Capital		
Administrative & Support Labor		\$/yr	30% of Operating and Maintenance Labor		

Variable Operating Costs	Unit	Jan. 1990 Cost/Unit	Additional Freight \$/t	30-Yr Level Factor	30-Yr Levelized Cost/Unit
Fuel Oil (No. 6)	gal	\$ 0.41	Incl	1.613	\$ 0.66
Raw water	1000 gal	\$ 0.60	Incl	1.613	\$ 0.97
Cooling water	1000 gal	\$ 0.16	Incl	1.613	\$ 0.26
Power	kwh	50 mills	Incl	1.668	83 mills
Methane	1000 ft <sup>3</sup>	\$ 3.00	Incl	1.668	\$ 5.00
Lime	ton	\$ 55.00	Incl	1.613	\$ 88.72
Limestone	ton	\$ 15.00	Incl	1.613	\$ 24.20
Soda Ash	ton	\$ 93.00	\$ 43.00	1.613	\$ 219.37
Magnesia	ton	\$ 232.00	\$135.00	1.613	\$ 591.97
Ammonia	ton	\$ 145.00	\$ 5.50	1.613	\$ 242.76
Sulfur Emulsion	ton	\$ 220.00	Incl	1.613	\$ 354.86
Dibasic Acid	ton	\$ 360.00	Incl	1.613	\$ 580.68
Formic Acid	ton	\$ 800.00	\$20 (est.)	1.613	\$1322.66
Allied Catalyst	ton	\$2500.00	\$45.00	1.613	\$4105.09
Claus Catalyst	ton	\$1000.00	Incl	1.613	\$1613.00
Disposal Charges,					
Dry Solids (lined)	ton (dry)	\$ 9.29	Incl	1.613	\$ 14.98
Fly ash (unlined)	ton (dry)	\$ 8.00	Incl	1.613	\$ 12.90
Fly ash with Nahcolite	ton (dry)	\$ 11.14	Incl	1.613	\$ 17.97
Sludge (trucked to lined landfill)	ton (dry)	\$ 9.25	Incl	1.613	\$ 14.92
Sludge (trucked to unlined landfill)	ton (dry)	\$ 8.15	Incl	1.613	\$ 13.15
Sludge (ponded)	ton (dry)	\$ 6.00	Incl	1.613	\$ 9.68
Gypsum (pumped & stacked)	ton (dry)	\$ 4.75	Incl	1.613	\$ 7.66
Condensate	1000 lb.	\$ 0.77	Incl	1.613	\$ 1.24
Steam					
0-70 psia	1000 lb.	\$ 2.85	Incl	1.668	\$ 4.75
70-250 psia	1000 lb.	\$ 3.50	Incl	1.668	\$ 5.84
250-400 psia	1000 lb.	\$ 5.30	Incl	1.668	\$ 8.84
By-product Credit					
Sulfur	Long ton	\$ 90.00	Incl	1.613	\$ 145.17
Sulfuric Acid	ton	\$ 50.00	Incl	1.613	\$ 80.65
Liquid Sulfur Dioxide	ton	\$ 230.00	Incl	1.613	\$ 371.00
Gypsum	ton	\$ 2.00	Incl	1.613	\$ 3.23

Note: 1000 gal = 3.785 m<sup>3</sup>, t (short) = 0.9072 t (SI),  
 1000 ft<sup>3</sup> = 28.32 m<sup>3</sup>, 1000 lb = 453.6 Kg

Assumed 100 mile shipping distance for all chemical reagents except where noted.

**DATE  
FILMED**

8 / 20 / 93

**END**

