Mathematical Models of Water Quality Parameters for Rivers and Estuaries

R.W. Hann, Jr.
P.J. Young

Texas Water Resources Institute
Texas A&M University
RESEARCH PROJECT COMPLETION REPORT

Project Number A-014-TEX

October 1, 1969 -- September 30, 1971

Agreement Numbers
14-31-0001-3044
14-31-0001-3244
14-31-0001-3544

MATHEMATICAL MODELS OF WATER QUALITY PARAMETERS
FOR RIVERS AND ESTUARIES

Principal Investigators
Roy W. Hann, Jr.
Paul Jonathan Young

The work upon which this publication is based was supported in part by funds provided by the United States Department of the Interior, Office of Water Resources Research, as authorized under the Water Resources Research Act of 1964.

Technical Report No. 45
Water Resources Institute
Texas A&M University

October 1972
FOREWARD

The development of computer models for mass transport in estuaries has been an important engineering activity for the past decade. Initial surveys conducted under this project showed however that only a limited amount of work had been done in modeling two dimensional characteristics in partially stratified estuaries.

As a result the major thrust of this project was directed in this important area. The project effort was coordinated with an existing project dealing with the Houston Ship Channel specifically EPA Research and Development grant 16090 DQW and considerably field data was developed by this EPA project. Additional support was received in the form of a Miles Cox fellowship which partially supported Dr. Young.

This final report presents the results of the literature survey model development and model calibration.

Project personnel over project time period have included Dr. Roy W. Hann, Jr., the current project director, Dr. Donald Schaezler, Dr. Robert Irvine, Dr. P. Jonathan Young, Mr. Richard Withers, Dr. Richard Allison and other staff members of the Estuarine Systems Projects Research group of The Environmental Engineering Division of Texas A&M Universities Civil Engineering Department.
### TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.</td>
<td>INTRODUCTION AND GOALS</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>GOALS OF THIS STUDY</td>
<td>2</td>
</tr>
<tr>
<td>II.</td>
<td>CHARACTERISTICS OF THE HOUSTON SHIP CHANNEL</td>
<td>4</td>
</tr>
<tr>
<td>III.</td>
<td>DEVELOPMENT OF THE ONE- AND TWO-DIMENSIONAL MASS TRANSPORT EQUATIONS</td>
<td>9</td>
</tr>
<tr>
<td>IV.</td>
<td>RECENT MATHEMATICAL MODELS OF ESTUARIES</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>MODELING METHODS OF DONALD J. O'CONNOR</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>MODELING METHODS OF ROBERT V. THOMANN</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>MODELING METHOD OF WATER RESOURCES ENGINEERS, INC.</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>MODELING METHOD OF HANN AND EVETT</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>ANALYTICAL METHODS OF HARLEMAN AND IPPEN</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>TWO-DIMENSIONAL MODELS OF GALVESTON BAY</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>VERTICAL MODEL BY PRITCHARD AND WILSON</td>
<td>42</td>
</tr>
<tr>
<td>V.</td>
<td>DEVELOPMENT OF FINITE DIFFERENCE SCHEMES</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>ONE-DIMENSIONAL EQUATIONS WITH CONSTANT COEFFICIENTS</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>ONE-DIMENSIONAL EQUATIONS WITH VARYING COEFFICIENTS</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>TWO-DIMENSIONAL EQUATIONS WITH VARYING COEFFICIENTS</td>
<td>60</td>
</tr>
<tr>
<td>VI.</td>
<td>STABILITY CONSIDERATIONS</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>EXPLICIT METHOD</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>IMPLICIT METHOD</td>
<td>76</td>
</tr>
<tr>
<td>VII.</td>
<td>BOUNDARY CONDITIONS</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>BOUNDARIES ALLOWING NO TRANSFER</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>CONSTANT SLOPE EXTRAPOLATION</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>EXPONENTIAL EXTRAPOLATIONS</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>CONTINUED FRACTIONS AND INVERTED DIFFERENCES</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>EXTRAPOLATION BY PROPORTIONS</td>
<td>87</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS
(Continued)

<table>
<thead>
<tr>
<th>Chapter</th>
<th>MATHEMATICAL MODELS AND SUPPORTING COMPUTER PROGRAMS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIII.</td>
<td>IDEAL-I</td>
<td>88</td>
</tr>
<tr>
<td></td>
<td>IDEAL-II</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>MASTRANS-I</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>MASTRANS-II</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>OXTRANS-I</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td>STABLE-I and STABLE-II</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>EXACT-I and EXACT-II</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>PROFILE-I and PROFILE-II</td>
<td>96</td>
</tr>
<tr>
<td>IX.</td>
<td>ANALYSIS OF ACCURACY</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>STEADY-STATE PROFILES</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>ONE-DIMENSIONAL INSTANTANEOUS RELEASE</td>
<td>104</td>
</tr>
<tr>
<td></td>
<td>OF A SLUG LOAD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TWO-DIMENSIONAL ACCURACY ANALYSIS</td>
<td>124</td>
</tr>
<tr>
<td>X.</td>
<td>COMPUTATIONAL EXPERIENCE</td>
<td>130</td>
</tr>
<tr>
<td></td>
<td>TIME REQUIREMENTS</td>
<td>130</td>
</tr>
<tr>
<td></td>
<td>BOUNDARY CONDITIONS</td>
<td>132</td>
</tr>
<tr>
<td></td>
<td>CHOICE OF INPUT PARAMETERS</td>
<td>133</td>
</tr>
<tr>
<td>XI.</td>
<td>APPLICATION TO THE HOUSTONSHIP CHANNEL</td>
<td>135</td>
</tr>
<tr>
<td></td>
<td>MODELING BOD AND DISSOLVED OXYGEN IN THE HOUSTONSHIP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CHANNEL</td>
<td>147</td>
</tr>
<tr>
<td></td>
<td>APPLICABILITY OF THE MODELS</td>
<td>149</td>
</tr>
<tr>
<td>XII.</td>
<td>DISCUSSION</td>
<td>153</td>
</tr>
<tr>
<td>XIII.</td>
<td>SUMMARY AND CONCLUSIONS</td>
<td>157</td>
</tr>
<tr>
<td></td>
<td>APPENDIX I - REFERENCES</td>
<td>160</td>
</tr>
<tr>
<td></td>
<td>APPENDIX II - NOTATIONS</td>
<td>171</td>
</tr>
<tr>
<td></td>
<td>APPENDIX III - COMPUTER PROGRAMS AND DATA</td>
<td>179</td>
</tr>
<tr>
<td>Number</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>9-1</td>
<td>COMPARISON BETWEEN EXACT AND FINITE DIFFERENCE SOLUTIONS</td>
<td>101</td>
</tr>
<tr>
<td>9-2</td>
<td>BOD AND DO PROFILES</td>
<td>103</td>
</tr>
<tr>
<td>9-3</td>
<td>ACCURACY FOR BOD PROFILES FOR SEVERAL DISTANCE INCREMENTS</td>
<td>105</td>
</tr>
<tr>
<td>9-4</td>
<td>ACCURACY FOR DISSOLVED OXYGEN PROFILES FOR SEVERAL DISTANCE INCREMENTS</td>
<td>105</td>
</tr>
<tr>
<td>9-5</td>
<td>DISTRIBUTION FROM A SLUG LOAD AT TIME = 0.2 DAYS</td>
<td>109</td>
</tr>
<tr>
<td>9-6</td>
<td>DISTRIBUTION FROM A SLUG LOAD AT TIME = 1.0 DAYS</td>
<td>109</td>
</tr>
<tr>
<td>9-7</td>
<td>ONE DIMENSIONAL CONVECTIVE-DISPERSION OF A SLUG LOAD AFTER ONE DAY</td>
<td>110</td>
</tr>
<tr>
<td>9-8</td>
<td>INITIAL CONCENTRATION VALUES</td>
<td>113</td>
</tr>
<tr>
<td>9-9</td>
<td>PROFILES CALCULATED BY FINITE DIFFERENCE METHODS FOR A SLUG LOAD; ELAPSED TIME = 0.6 DAYS</td>
<td>114</td>
</tr>
<tr>
<td>9-10</td>
<td>ACCURACY ANALYSIS: EXPLICIT METHOD. PROFILE AFTER 0.6 DAYS FOR AN INITIAL SLUG LOAD OF 18939.39</td>
<td>116</td>
</tr>
<tr>
<td>9-11</td>
<td>ACCURACY ANALYSIS: EXPLICIT METHOD. PROFILE AFTER 0.6 DAYS FOR AN INITIAL KNOWN DISTRIBUTION AT 28 POINTS</td>
<td>117</td>
</tr>
<tr>
<td>9-12</td>
<td>ACCURACY ANALYSIS: CRANK-NICOLSON METHOD. PROFILE AFTER 0.6 DAYS FOR AN INITIAL SLUG LOAD OF 18939.39</td>
<td>118</td>
</tr>
<tr>
<td>9-13</td>
<td>ACCURACY ANALYSIS: CRANK-NICOLSON METHOD. PROFILE AFTER 0.6 DAYS FOR AN INITIAL KNOWN DISTRIBUTION AT 28 POINTS</td>
<td>119</td>
</tr>
<tr>
<td>9-14</td>
<td>CONCENTRATIONS AT TIME = 0.11 DAYS</td>
<td>125</td>
</tr>
<tr>
<td>Number</td>
<td>Table Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>9-15</td>
<td>INITIAL VALUES FOR NON-SYMmetrical DISTRIBUTION</td>
<td>127</td>
</tr>
<tr>
<td>9-16</td>
<td>TWO DIMENSIONAL DISTRIBUTION AT TIME = 0.77 DAYS</td>
<td>128</td>
</tr>
<tr>
<td>10-1</td>
<td>COMPUTATION TIMES</td>
<td>131</td>
</tr>
<tr>
<td>11-1</td>
<td>INITIAL CONCENTRATIONS</td>
<td>139</td>
</tr>
<tr>
<td>11-2</td>
<td>HORIZONTAL VELOCITIES AND VERTICAL DISPERSION COEFFICIENTS</td>
<td>140</td>
</tr>
<tr>
<td>11-3</td>
<td>WIDTHS AT 20-FOOT DEPTH</td>
<td>140</td>
</tr>
<tr>
<td>Number</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>2-1</td>
<td>GALVESTON BAY AREA</td>
<td>5</td>
</tr>
<tr>
<td>2-2</td>
<td>HOUSTON SHIP CHANNEL</td>
<td>6</td>
</tr>
<tr>
<td>4-1</td>
<td>RELATIONSHIP BETWEEN STRATIFICATION NUMBER AND ESTUARY NUMBER</td>
<td>35</td>
</tr>
<tr>
<td>4-2</td>
<td>CELL STRUCTURE FOR SALINITY MODEL OF SHANKAR AND MASCH</td>
<td>40</td>
</tr>
<tr>
<td>4-3</td>
<td>SCHEMATIC FOR THE TWO-DIMENSIONAL MODEL OF WILSON AND PRITCHARD INDICATING SEGMENTATION AND PARAMETERS USED TO CHARACTERIZE FLOW. ( Q_u, Q_v, ) AND ( Q_w ) ARE VOLUME RATES OF NET ADVECTIVE FLOW; ( E ) IS THE COEFFICIENT OF VERTICAL DIFFUSIVE EXCHANGE.</td>
<td>43</td>
</tr>
<tr>
<td>5-1</td>
<td>EXPLICIT ( (\theta = 0) )</td>
<td>49</td>
</tr>
<tr>
<td>5-2</td>
<td>CRANK-NICOLSON ( (\theta = 1/2) )</td>
<td>49</td>
</tr>
<tr>
<td>5-3</td>
<td>FULLY IMPLICIT ( (\theta = 1) )</td>
<td>49</td>
</tr>
<tr>
<td>5-4</td>
<td>EXPLICIT FINITE DIFFERENCE EQUATIONS FOR ONE-DIMENSIONAL MASS TRANSPORT WITH VARYING COEFFICIENTS</td>
<td>53</td>
</tr>
<tr>
<td>5-5</td>
<td>IMPLICIT CRANK-NICOLSON FINITE DIFFERENCE EQUATION FOR ONE-DIMENSIONAL MASS TRANSPORT WITH VARYING COEFFICIENTS</td>
<td>57</td>
</tr>
<tr>
<td>5-6</td>
<td>COEFFICIENTS FOR TRIDIAGONAL MATRIX</td>
<td>59</td>
</tr>
<tr>
<td>5-7</td>
<td>EXPLICIT FINITE DIFFERENCE EQUATIONS FOR TWO-DIMENSIONAL MASS TRANSPORT WITH VARYING COEFFICIENTS</td>
<td>61</td>
</tr>
<tr>
<td>5-8</td>
<td>TWO-DIMENSIONAL EXPLICIT COMPUTATION MOLECULE</td>
<td>64</td>
</tr>
<tr>
<td>5-9</td>
<td>ARRAY VALUES FOR THE FIRST PHASE OF THE IMPLICIT PROCEDURE</td>
<td>68</td>
</tr>
<tr>
<td>Number</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>5-10</td>
<td>ARRAY VALUES FOR THE SECOND PHASE OF THE IMPLICIT PROCEDURE</td>
<td>70</td>
</tr>
<tr>
<td>5-11</td>
<td>TWO-DIMENSIONAL IMPLICIT COMPUTATIONAL MOLECULE</td>
<td>72</td>
</tr>
<tr>
<td>5-12</td>
<td>TWO-DIMENSIONAL VERTICAL GRID</td>
<td>73</td>
</tr>
<tr>
<td>6-1</td>
<td>DISTANCE INCREMENTS FOR STABILITY</td>
<td>77</td>
</tr>
<tr>
<td>6-2</td>
<td>TIME INCREMENTS FOR STABILITY</td>
<td>78</td>
</tr>
<tr>
<td>9-1</td>
<td>SLUG LOAD &quot;SEEN&quot; BY FINITE DIFFERENCE MODEL</td>
<td>107</td>
</tr>
<tr>
<td>9-2</td>
<td>PROFILE FROM A SLUG LOAD</td>
<td>111</td>
</tr>
<tr>
<td>9-3</td>
<td>PERCENTAGE OF ERROR (see TABLE 9-10)</td>
<td>120</td>
</tr>
<tr>
<td>9-4</td>
<td>ERROR FOR A PEAK 689.7 PPM (see TABLE 9-10)</td>
<td>121</td>
</tr>
<tr>
<td>11-1</td>
<td>TIDE READINGS</td>
<td>136</td>
</tr>
<tr>
<td>11-2</td>
<td>MEASURED CONCENTRATIONS AT H + 4</td>
<td>137</td>
</tr>
<tr>
<td>11-3</td>
<td>MEASURED CONCENTRATIONS AT H + 15</td>
<td>142</td>
</tr>
<tr>
<td>11-4</td>
<td>CONCENTRATIONS PREDICTED BY MASSTRANS-I FOR H + 15</td>
<td>143</td>
</tr>
<tr>
<td>11-5</td>
<td>COMPARISON OF MEASURED AND SIMULATED PROFILES</td>
<td>144</td>
</tr>
<tr>
<td>11-6</td>
<td>COMPARISON OF PEAK CONCENTRATIONS</td>
<td>146</td>
</tr>
<tr>
<td>11-7</td>
<td>INITIAL HYPOTHETICAL VALUES FOR BOD\textsubscript{20} AND DO AT H + 4 HOURS</td>
<td>148</td>
</tr>
<tr>
<td>11-8</td>
<td>BOD\textsubscript{20} (ppm) AT H + 15 HOURS</td>
<td>150</td>
</tr>
<tr>
<td>11-9</td>
<td>DISSOLVED OXYGEN (ppm) AT H + 15 HOURS</td>
<td>151</td>
</tr>
</tbody>
</table>
CHAPTER I

INTRODUCTION AND GOALS

Estuaries represent an economically and biologically important part of our coast. These bodies of water are used as permanent homes or nursery grounds for many of the important forms of marine life. Estuaries also provide protected transportation routes for cargo ships and barges. Because of their biological productivity and their economically advantageous locations, estuaries often attract industrial and commercial development along with associated dense populations.

Because of industrial and metropolitan development, estuaries are often expected to accept large amounts of waste materials. Many estuaries are large in size and give the appearance of being able to assimilate large volumes of this waste material. In actuality, estuaries may act as natural traps for pollutants. The fresh-water flow into some estuaries is relatively small. Once a pollutant enters such an estuary, the material is carried back and forth by the tides and makes only slow progress toward the ocean. These conditions are especially characteristic of Gulf
Coast estuaries. For example, recent studies (92) indicate that the average flushing time (i.e., the time necessary for the volume of fresh water inflow to equal the volume of the estuary) for the upper 14 miles of the Houston Ship Channel is 26 days and the average flushing time for Galveston Bay is 155 days.

In regard to salinity, estuaries vary between two extremes: the homogeneous estuary and the highly stratified. At each section of a homogeneous estuary, there is complete vertical mixing of the salt and fresh water. In a stratified estuary, there are two layers of water: the upper layer contains fresh water flowing toward the sea, and the lower layer contains salt water moving away from the sea. A sharp change in salinity occurs at the interface between the upper and lower layers. The Delaware Estuary is representative of a homogeneous estuary, whereas the Mississippi Delta represents a highly stratified estuary. Many estuaries fall somewhere between these two extremes and include some mixing and some stratification at each section; hence, they are more difficult to analyze than the idealized homogeneous or stratified cases. The Houston Ship Channel is an example of a partially stratified estuary. A comprehensive discussion of the characteristics of these types of estuaries is available in Ippen (51).

GOALS OF THIS STUDY

Although the past decade has produced an extensive amount of
mathematical modeling of estuaries, little effort has gone into modeling of the two dimensional characteristics of partially stratified estuaries. A major goal of this research was to develop computer models which could calculate vertical and horizontal mass transport in partially stratified estuaries. These models were to be applicable for predicting pollutant dispersion and dissolved oxygen distributions.

Two types of finite difference techniques provided the basis for these mathematical models: an explicit method and a Crank-Nicolson implicit method. The second major goal of this research was to compare the accuracy and usefulness of these two techniques. Discussions in literature of this type of comparison are scanty. Where these comparisons do exist for the two-dimensional mass transport equation (28), the equation has been simplified and applied to non-estuary problems with limited success.

A third major goal of this research was to summarize existing one- and two-dimensional mathematical models that have been applied to significant estuary problems.

The final major goal of this study was to demonstrate the applicability of these computer models to the mass transport characteristics of the Houston Ship Channel.
CHAPTER II

CHARACTERISTICS OF THE HOUSTON SHIP CHANNEL

The portion of the Houston Ship Channel being studied in this report is a tributary to upper Galveston Bay (see Figure 2-1). The Houston Ship Channel, shown in Figure 2-2, is a fairly narrow estuary which has been dredged to a depth of about 40 feet to allow the passage of ocean-going vessels. This estuary can be considered laterally homogeneous, especially in the upper 14 miles where there is no pronounced influence from side bays. The channel receives a large amount of industrial and municipal wastes from outfalls which discharge into the channel at many points.

Salinity measurements have been made on a regular basis on the channel since April 1968. Early in the sampling program, salinity was recorded at every ten feet of depth at five points laterally across the channel. These measurements were taken every four miles up the 24-mile long channel. Eventually, it was decided to sample only the centerline of the channel, since the salinity seemed not to vary significantly across the channel. At that time, the sampling grid was tightened to include measurements at every five feet vertically and two miles longitudinally (29).

As can be seen from the raw data and from several Estuarine Systems Projects reports (92, 109), the salinity structure of the
FIGURE 2-1. - GALVESTON BAY AREA
FIGURE 2-2. HOUSTON SHIP CHANNEL
Houston Ship Channel is extremely variable. During some periods of heavy rainfall, the channel becomes highly stratified. At times of low inflow, the channel may become vertically well-mixed because of tidal agitation. However, the majority of the data indicates that the salinity structure is generally at some intermediate state of stratification, i.e., the salinity varies considerably from top to bottom with no sharp interface between fresh and saline water. Thus, the ship channel can best be classified as being "partially stratified." Calculations using typical values for inflow and tidal prism for the channel indicate that it usually falls within the "partially mixed" classification of previous estuary researchers (51, 107). The terms "partially mixed" and "partially stratified" are used synonymously in this report.

The channel can be adequately modeled by assuming a constant centerline depth and varying width; its behavior is essentially two-dimensional. However, several factors can be listed which keep the channel from being a simple, well-behaved hydraulic system:

1. The downstream end of the channel is a protected bay which is partially stratified and which has changeable salinity.

2. Part of the dredged channel is bordered by large, shallow side bays whose influence on flow patterns and dissolved oxygen patterns is difficult to determine.
3. A large tributary, the San Jacinto River, enters the channel about ten miles from the entrance to upper Galveston Bay and the flow of the river is erratic due to the operation of an upstream dam.

4. Tides in the 40-foot deep channel are erratic and of small amplitude, usually averaging about one foot in height.

5. Many chemical substances are added to the channel waters; their effects on flow patterns, salinity structure, and decay rates are difficult to determine.

6. Ship traffic on the channel has an undetermined effect on flow patterns and reaeration and interrupts the taking of measurements.

7. Benthal deposits and algae have a significant effect on water quality characteristics.
CHAPTER III

DEVELOPMENT OF THE ONE AND TWO DIMENSIONAL
MASS TRANSPORT EQUATIONS

Mass transport is governed by the principle of conservation of matter. For the case of materials dissolved in a fluid, conservation of matter must be satisfied for each constituent. For such a system, the conservation principle can be summarized as follows:

\[
\text{TIME RATE OF ACCUMULATION OF CONSTITUENT INSIDE A FLUID ELEMENT} = \begin{cases} \text{INFLOW OF CONSTITUENT TO FLUID ELEMENT} \\ \text{OUTFLOW OF CONSTITUENT FROM FLUID ELEMENT} \\ \text{TIME RATE OF PRODUCTION OF CONSTITUENT BY CHEMICAL AND BIOLOGICAL REACTION INSIDE THE FLUID ELEMENT} \end{cases}
\]
This principle can be expressed by the equation

\[ \frac{3p_A}{3t} = - \left( \frac{3N_x}{3x} + \frac{3N_y}{3y} + \frac{3N_z}{3z} \right) + r_A \quad \ldots \ldots \ldots \ldots \ldots \quad (3-1) \]

or in the vector form

\[ \frac{3p_A}{3t} = - \nabla . N_A + r_A \quad \ldots \ldots \ldots \ldots \ldots \quad (3-2) \]

where \( \rho_A \) is the mass density of the mixture; \( N_A \) is a mass flux of constituent A across a boundary and is expressed in the units mass / (Length)² (Time); \( N_x, N_y, N_z \) are the components of \( N_A \); and
\( r_A \) is the time rate of production of constituent A. This equation and its various forms are referred to by the names "equation of continuity for a constituent," "equation for conservation of a dissolved constituent," "mass transport equation," "mass balance equation," "diffusion equation," and the "convective dispersion equation."

The mass flux term, \( N_A \), represents transport by two phenomena: local fluid velocity and molecular diffusion:

\[ N_A = - \rho_A D_A \nabla C_A + \rho_A C_A \nabla \quad \ldots \ldots \ldots \ldots \ldots \quad (3-3) \]

where \( D_A \) is the molecular diffusion coefficient for constituent A in the fluid, \( \rho_A \) is the mass density of the mixture; \( C_A \) is the concentration of constituent A; and \( \nabla \) is the velocity.
vector. The negative diffusion term results from the tendency of a constituent in a mixture to move by molecular action in the direction of decreasing concentration of that constituent.

The velocities have both a time-averaged component, \( V_i \), and a perturbation component, \( V'_i \), which represents the random variations in velocity from its average value. The concentrations also have a perturbation component, \( C'_A \). A combination of these terms produces the following form for the turbulent mass flux (14):

\[
\text{Turbulent Mass Flux Per Unit Area} = \rho_A \left( \bar{V}_i C'_A \right) \hspace{1cm} (3-4)
\]

where a time average is denoted by a bar over the appropriate terms.

By analogy with Fick's first law of diffusion, it is often assumed that the turbulent flux is proportional to the gradient of the time-averaged concentration; for example, in the \( x \)-direction:

\[
\rho_A \left( \bar{V}_i C'_A \right)_x = \rho_A E_x \frac{\partial \bar{C}_A}{\partial x} \hspace{1cm} (3-5)
\]

where \( E_x \) is the turbulent diffusion coefficient.

If the isotropic molecular diffusion is assumed, these considerations lead to the following vector form of the mass transport equation for a single component, after the \( A \) subscripts are
dropped:

\[ \frac{\partial C}{\partial t} + \nabla \cdot CV = D(\nabla \cdot V) + \nabla \cdot (E \cdot V) + \frac{r}{\rho} \quad \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (3-6) \]

When continuity for an incompressible fluid is applied, a more useful form results:

\[ \frac{\partial C}{\partial t} + V_x \frac{\partial C}{\partial x} + V_y \frac{\partial C}{\partial y} + V_z \frac{\partial C}{\partial z} \]

\[ = \frac{\partial}{\partial x} \left( E_x \frac{\partial C}{\partial x} \right) + \frac{\partial}{\partial y} \left( E_y \frac{\partial C}{\partial y} \right) + \frac{\partial}{\partial z} \left( E_z \frac{\partial C}{\partial z} \right) \]

\[ + D \left( \frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} + \frac{\partial^2 C}{\partial z^2} \right) + \frac{r}{\rho} \quad \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (3-7) \]

The $E_x$, $E_y$, and $E_z$ are the turbulent diffusion coefficients and can be only equal for the case of isotropic turbulence. In a natural system, the turbulent diffusion coefficients are many orders of magnitude larger than the molecular diffusion coefficients; hence it is acceptable to ignore completely the molecular diffusion terms. Of course, the terms on the left side of this equation can be more compactly expressed by the material derivative, $DC/DT$.

Equation 3-6 can be simplified by dropping the molecular diffusion term, which is generally insignificant, and by eliminating the term $r/\rho$ by assuming that the material is conservative. This leaves the following form of the mass transport equation:
\[ \frac{\partial C}{\partial t} = - \frac{\partial (V_x C)}{\partial x} - \frac{\partial (V_y C)}{\partial y} - \frac{\partial (V_z C)}{\partial z} \]

\[ + \frac{\partial}{\partial x} \left( E_x \frac{\partial C}{\partial x} \right) + \frac{\partial}{\partial y} \left( E_y \frac{\partial C}{\partial y} \right) + \frac{\partial}{\partial z} \left( E_z \frac{\partial C}{\partial z} \right) \] ... (3-8)

Equation 3-8 is associated with the general three-dimensional equation of continuity for an incompressible fluid:

\[ \frac{\partial V_x}{\partial x} + \frac{\partial V_y}{\partial y} + \frac{\partial V_z}{\partial z} = 0 \] ... (3-9)

By expansion of a method suggested by Pritchard (78), Equations 3-8 and 3-9 can be treated to produce the two-dimensional and one-dimensional mass transport equations for a laterally homogeneous estuary. During this derivation, we assume that concentrations do not change with width and that the width does not change with time.

Assume that \(a(x, z)\) is the left-hand boundary of the estuary and \(b(x, z)\) is the right-hand boundary. By integrating Equation 3-8 between these boundaries we have

\[ \int_a^b \int_a^b \frac{\partial C}{\partial t} \, dy = \int_a^b \int_a^b \frac{\partial (V_x C)}{\partial x} \, dy - \int_a^b \int_a^b \frac{\partial (V_y C)}{\partial y} \, dy - \int_a^b \int_a^b \frac{\partial (V_z C)}{\partial z} \, dy + \text{etc.} 

\] ... (3-10)

By extension of the Leibnitz rule (57), we can write
\[ \int_{a}^{b} \frac{\partial (V_x C)}{\partial x} \, dy = \frac{\partial}{\partial x} \left( \int_{a}^{b} (V_x C) \, dy \right) + \left( (V_x C) \frac{\partial a}{\partial x} \right)_{a}^{b} - \left( (V_x C) \frac{\partial b}{\partial x} \right)_{a}^{b} \]

\[ \cdots \cdots \cdots \cdots \cdots (3-11) \]

and

\[ \int_{a}^{b} \frac{\partial}{\partial x} \left( E_x \frac{\partial C}{\partial x} \right) \, dy = \frac{\partial}{\partial x} \left( \int_{a}^{b} \left( E_x \frac{\partial C}{\partial x} \right) \, dy \right) + \left( E_x \frac{\partial C}{\partial x} \right)_{a}^{b} + \left( E_x \frac{\partial C}{\partial x} \right)_{a}^{b} \]

\[ \cdots \cdots \cdots \cdots \cdots (3-12) \]

and similarly for the other terms. We can integrate the \( y \) terms directly:

\[ \int_{a}^{b} \frac{\partial (V_y C)}{\partial y} \, dy = (V_y C)_{b}^{a} \]

\[ \cdots \cdots \cdots \cdots \cdots (3-13) \]

and
\[
\int_a^b \frac{\partial}{\partial y} \left( E_y \frac{\partial C}{\partial y} \right) dy = \left( E_y \frac{\partial C}{\partial y} \right)_b - \left( E_y \frac{\partial C}{\partial y} \right)_a \quad \ldots \ldots \ldots \ldots \quad (3-14)
\]

Since there can be no mass transfer across the boundaries and since the boundaries do not change with time, the following equations result from analysis of the derivatives:

\[
(V_y C) = (V_x C) \frac{\partial b}{\partial x} + (V_z C) \frac{\partial b}{\partial z} \quad \ldots \ldots \ldots \ldots \quad (3-15)
\]

\[
(V_y C) = (V_x C) \frac{\partial a}{\partial x} + (V_z C) \frac{\partial a}{\partial z} \quad \ldots \ldots \ldots \ldots \quad (3-16)
\]

\[
\left( E_y \frac{\partial C}{\partial y} \right)_b = \left( E_x \frac{\partial C}{\partial x} \right)_b + \left( E_z \frac{\partial C}{\partial z} \right)_b \quad \ldots \ldots \ldots \ldots \quad (3-17)
\]

\[
\left( E_y \frac{\partial C}{\partial y} \right)_a = \left( E_x \frac{\partial C}{\partial x} \right)_a + \left( E_z \frac{\partial C}{\partial z} \right)_a \quad \ldots \ldots \ldots \ldots \quad (3-18)
\]

If equations 3-11 through 3-18 are substituted into Equation 3-10, we are left with
\[
\frac{\partial}{\partial t} \int_a^b C dy = - \frac{\partial}{\partial x} \int_a^b \left( V_x C \right) dy - \frac{\partial}{\partial z} \int_a^b \left( V_z C \right) dy \\
+ \frac{\partial}{\partial x} \int_a^b \left( E_x \frac{\partial C}{\partial x} \right) dy + \frac{\partial}{\partial z} \int_a^b \left( E_z \frac{\partial C}{\partial z} \right) dy
\]

\hspace{1cm} \cdots \hspace{1cm} \cdots \hspace{1cm} (3-19)

Since we have assumed lateral homogeneity, this equation reduces to

\[
\frac{\partial (WC)}{\partial t} = - \frac{\partial (WV_x C)}{\partial x} - \frac{\partial (WV_z C)}{\partial z} \\
+ \frac{\partial}{\partial x} \left( E_x W \frac{\partial C}{\partial x} \right) + \frac{\partial}{\partial z} \left( E_z W \frac{\partial C}{\partial z} \right)
\]

\hspace{1cm} \cdots \hspace{1cm} \cdots \hspace{1cm} (3-20)

where \( W = b - a \), which is the estuary width.

If we apply the Leibnitz rule and the same boundary conditions to Equation 3-9, we find that the two-dimensional continuity equation with regard to width becomes

\[
\frac{\partial (WV_x)}{\partial x} + \frac{\partial (WV_z)}{\partial z} = 0 \hspace{1cm} \cdots \hspace{1cm} \cdots \hspace{1cm} (3-21)
\]

Combining Equation 3-20 and Equation 3-21 gives the two-dimensional mass transport equation which is used most frequently in this study:
\[
W \frac{\partial C}{\partial t} = -W_x \frac{\partial C}{\partial x} - W_z \frac{\partial C}{\partial z} \\
+ \frac{\partial}{\partial x} \left( E_x W \frac{\partial C}{\partial x} \right) + \frac{\partial}{\partial z} \left( E_z W \frac{\partial C}{\partial z} \right) \cdots \cdots \cdots \cdots \cdots (3-22)
\]

If we wish to derive the one-dimensional mass transport equation, we can apply the same type of analysis to Equation 3-8 using the cross-sectional area, \( A_x \); the following equation results:

\[
A_x \frac{\partial C}{\partial t} = -A_x V_x \frac{\partial C}{\partial x} + \frac{\partial}{\partial x} \left( E_x A_x \frac{\partial C}{\partial x} \right) \cdots \cdots \cdots \cdots \cdots (3-23)
\]

Appropriate source and sink terms can be added to Equations 3-22 and 3-23 for the various substances being transported.

For an estuary of constant depth, the cross-sectional area term can be replaced by a width term, \( W_x \):

\[
W_x \frac{\partial C}{\partial t} = -W_x V_x \frac{\partial C}{\partial x} + \frac{\partial}{\partial x} \left( E_x W_x \frac{\partial C}{\partial x} \right) \cdots \cdots \cdots \cdots \cdots (3-24)
\]
CHAPTER IV

RECENT MATHEMATICAL MODELS OF ESTUARIES

Because of increased concern over the degradation of estuaries, advanced techniques are needed to analyze the behavior of pollutants in these bodies of water. When developing a plan for controlling the quality of a body of water, one of the crucial problems is the formulation of a model which accurately simulates the behavior of the system. The most significant work relating to the mathematical modeling of estuaries has been done in the last decade. In the early 1960's, researchers concentrated on applying closed-form, analytical equations to estuaries. As experience has been gained, more complex methods have been developed with a heavier reliance on digital computers. The most recent models generally require a high degree of mathematical sophistication including numerical solution on a computer of complex systems of partial differential equations. Once a simulation model has been developed and verified, techniques of operations research can often be applied to determine the minimum cost solution to the water quality problems.

Most analyses dealing with water quality problems are based on some form of the mass transport equation. Some manipulation of terms in this equation is necessary for constituents which interact, such as biochemical oxygen demand (BOD) and dissolved
oxygen. For parameters which undergo progressive changes, such as in the nitrogen cycle, the mass transport equation may have to be used several times to describe the system adequately. When a steady-state condition in a river is assumed, the diffusion and dispersion terms can be dropped and formulations such as the classical Streeter-Phelps equation (94) can be derived directly.

Determining the success of any model in adequately representing a system is highly dependent upon the amount of data available from that system. Assuming a model has been formulated correctly, the accuracy of the model then depends on choosing the correct values for reaeration, decay, dispersion, salinity gradients, velocity of flow, volume of pollution, and various factors related to tidal action and currents. Attempts to use a model to predict water quality should be made only after that model has been shown to fit adequately a large amount of existing data.

Estuary modeling of particular interest has been done by many American researchers. Important work also has been done in foreign countries; for instance, the Water Pollution Research Laboratory of Great Britain has made a landmark study of the Thames (104) and the Delft Hydraulics Laboratory makes regular studies on estuaries in The Netherlands (35). The state of current research for estuary mathematical models can best be indicated by a brief review of the work of some of these research efforts.
MODELING METHODS OF DONALD J. O'CONNOR

O'Connor is the best known researcher in the field of mathematical modeling of estuaries. His 1960 paper (65) entitled, "Oxygen Balance of an Estuary" was the first comprehensive report on modeling of the behavior of non-conservative substances in these bodies of water. During the past decade, O'Connor has written many additional papers (66, 67, 68, 69) dealing with the modeling of various estuaries.

O'Connor's approach to the modeling of estuaries is generally characterized by steady-state analytical solutions to a one-dimensional form of the estuary mass transport equation. Two of the most useful of these equations will be discussed later in this chapter as Equations 4-5 and 4-6. O'Connor's use of analytical equations is most versatile when several of these equations are used simultaneously. This approach was applied to the East River, New York, where the system was divided into segments, each with its appropriate general solution. The concentration profile for the entire profile was then determined by solving a set of equations which matched at the segment boundaries (68). This approach requires that the variation in cross-sectional area of segments of the estuary be expressed as well-behaved functions of distance. O'Connor often uses expressions for cross-sectional area which are linear, monomial,
or exponential functions of distance. The dispersion term $E_x$ is often used to include the influence of tides as well as eddy diffusion.

O'Connor's methods have been used to obtain steady-state concentration profiles for many estuaries including the New York Harbor, East River, Delaware River, James River, and Raritan River. Time-varying equations also have been developed and applied.

MODELING METHODS OF ROBERT V. THOMANN

A widely used mathematical model for the Delaware Estuary is primarily the work of Thomann (97, 98, 99, 100, 101, 102, 75). Thomann's initial formulation of the model was presented in his doctoral dissertation, "The Use of Systems Analysis to Describe the Time Variation of Dissolved Oxygen in a Tidal Stream" (97). Thomann later became the technical director of the Delaware Estuary Comprehensive Study (DECS) under the Public Health Service and Federal Water Pollution Control Administration (FWPCA); for several years, a large amount of data was collected by DECS and the model was improved. In July, 1966, the FWPCA published a report (24) which recorded the status of the model and the recommendations of that government agency for the
uses of the river. Since that time, the Thomann model has been applied to the Potomac (42) and other estuaries.

A numerical solution to the basic equations in the model was obtained by the DECS from the Re-Entry Systems Department of the General Electric Company. The work at General Electric was supervised by Jeglic (54, 55). A recent digital computer program, called DECS III, is a refined, time-dependent version of the original model. It has been programmed in several versions of FORTRAN IV and can be used by many of the larger digital computers. A steady-state version of the Thomann model was been documented by Bunce and Hetling (9).

The basic form of the Thomann equation for BOD is as follows:

\[
\frac{dL_k}{dt} = \frac{Q_{k-1,k}}{VOL_k} \left[ \xi_{k-1,k} L_{k-1} + (1 - \xi_{k-1,k}) L_k \right] \\
- \frac{Q_{k,k+1}}{VOL_k} \left[ \xi_{k,k+1} L_k + (1 - \xi_{k,k+1}) L_{k+1} \right] \\
+ \frac{E'_{k-1,k}}{VOL_k} (L_{k-1} - L_k) + \frac{E'_{k,k+1}}{VOL_k} (L_{k+1} - L_k) \\
- d_k L_k \pm J_k \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (4-1)
\]

In this equation, \( k \) ranges from 1 to \( n \), where \( n \) = the number of sections, \( k-1 \) = the upstream section, \( k+1 \) = the downstream section, \( L \) = the ultimate carbonaceous biochemical oxygen demand concentration, \( Q \) = the net flow from section to section, \( VOL \) = the volume
of the section, \( \xi = \) an advective coefficient dependent upon the ratio of the dispersion to the advective forces, \( E' = \) an eddy exchange coefficient analogous to the classical eddy diffusion coefficient in non-tidal streams, \( d = \) the decay rate of biochemical oxygen demand, and \( J = \) the direct sources of biochemical oxygen demand.

It is obvious from this equation that Thomann attempts to separate some of the effects of tides from the eddy exchange coefficient, \( E' \). It should be noted that the dimensions of \( E' \) must be \( L^3/T \) to make the equation dimensionally correct. Thus, Thomann's \( E' \) can not be considered equal to the dispersion or diffusion coefficients (\( D \) or \( E \)) of other investigations.

The advective coefficient, \( \xi \), varies between a value of 0.5 for an environment dominated almost entirely by tides to a value of 1.0 for a non-tidal stream. For situations between these extremes, \( \xi \) is assumed to be some function of \( Q \) and \( E' \). In some applications, \( \xi \) has been assumed equal to either 0.5 (99) or to the ratio \( l_k / (l_k + l_{k-1}) \) where \( l \) is the length of the appropriate segment (9). If \( \xi \) is equal to 0.5, the Thomann equation becomes a one-dimensional finite-difference form of the mass transport equation when \( E' \) is set equal to \( EA/\Delta x \), where \( A \) is the cross-sectional area. A discussion of the problems associated with evaluating the various parameters in the Thomann equation is found in the proceedings of the Stanford ASCE Symposium on Estuarine Pollution (75, 76).
Recently, Hays (38, 39) has applied the steady-state version \( \frac{dL_k}{dt} = 0 \) of the Thomann model to the Houston Ship Channel. Hays applied non-linear programming techniques to this one-dimensional model and obtained least-cost solutions for producing certain dissolved oxygen profiles in the channel.

MODELING METHOD OF WATER RESOURCES ENGINEERS, INC.

The mathematical model most often applied to the San Francisco Bay and Delta region was developed for the FWPCA (now the Environmental Protection Agency, EPA) by Water Resources Engineers, Inc., under the supervision of Orlob. The first stage of this model was formulated in 1965 to represent the water quality in the Sacramento-San Joaquin Delta (90, 105). Verification of this phase of the model was achieved by comparison with past salinity and hydrological conditions and with the results of a prototype dye tracer study. During the following year, 1966, the model was extended to Suisun and San Pablo Bay (72, 106). The model in this form was the major tool used in predicting the effects of the proposed San Joaquin master drain (25) on the future water quality of the estuary.

The model uses finite-difference approximations to simplified, one-dimensional forms of the equation of motion, the equation of continuity, and the mass transport equation for conservative materials (106):
\[
\frac{\partial V_x}{\partial t} + V_x \frac{\partial V_x}{\partial x} + g \frac{\partial H}{\partial x} + K |V_x| V_x = 0 \quad \ldots \ldots \ldots \quad (4-2)
\]

\[
\frac{\partial H}{\partial t} + \frac{\partial Q}{\partial x} = 0 \quad \ldots \ldots \ldots \ldots \ldots \ldots \quad (4-3)
\]

\[
\frac{\partial C}{\partial t} = E_x \frac{\partial^2 C}{\partial x^2} - V_x \frac{\partial C}{\partial x} \quad \ldots \ldots \ldots \ldots \quad (4-4)
\]

where \(K\) represents a frictional resistance term; \(H\) is the tidal height; \(Q\) is the flow rate; \(V_x\) is the velocity; and the remaining terms are defined as in previous sections of this report. These one-dimensional equations were applied to a two-dimensional grid representing the Bay-Delta watercourse. The model used a network of channels which are linked at nodes to describe the area covered by the Delta, Suisun Bay, and San Pablo Bay. This grid system was a major distinction between the San Francisco Bay model and the Delaware Estuary model, which simply divided the Delaware into a series of reaches.

Water Resources Engineers chose to employ an explicit scheme in solving the above equations because of the simplicity of this type of formulation and because the step size that can be used is relatively independent of the size of the system (106). The water quality model uses intra-tidal velocities, and it was found that
the effects of advective transport were much more important than eddy dispersion. Thus, the dispersion term could be ignored and only the velocity term was used in the refined version of the conservative mass transport equation. Some problems developed because of the "numerical mixing" error which was a consequence of the numerical methods used to integrate the mass transport equation and the application of one-dimensional equations to a two-dimensional system. Techniques have been developed to circumvent these problems by applying a weighted-average factor to the concentrations, but generally either accuracy or stability of the solution must be sacrificed to a certain degree (72).

The Water Resources Engineers' model is operated as follows: The hydrodynamic model is operated for several equal tidal cycles until stable intra-tidal velocities are obtained; normally these velocities are obtained for 30-minute increments (72). These velocities are fed into the water quality model which is run for a large number of equal tidal cycles until a stable equilibrium pattern is developed. The rate of convergence of the quality model is somewhat dependent upon the initial concentrations used; thirty tidal cycles is a representative number of repetitions needed before equilibrium is obtained.

In the use of this model by the FWPCA, primary emphasis was placed on total dissolved solids and total nitrogen, where nitrogen was considered as a conservative element. The principal characteristics of the present nitrogen and salinity distributions in the
Bay-Delta have been reproduced by the model with a satisfactory degree of accuracy. Non-conservative substances can be directly represented by the model and several preliminary studies have been conducted which illustrate the capability of the model to handle BOD and DO distributions. However, the model is set up on the assumption of complete mixing at a cross-section and, therefore, cannot handle a vertically stratified body of water.

This model has been applied to other bodies of water, including the San Diego Bay (26) and the Columbia River estuary (10). Recent documentation of the model by Feigner and Harris (26) show the following characteristics: The hydrodynamic model is typically run for 1 to 4 tidal cycles with a resulting execution time of 7 to 23 minutes on a CDC 6600 computer. The water quality model is typically run for 20 to 56 tidal cycles with a resulting execution time of 3 to 14 minutes on the CDC 6600. Execution time for an IBM 360/65 computer was found to be 2 to 3 times greater than those reported for the CDC 6600. In the above applications of the model, the number of junctions ranged between 112 and 830, and the number of channels ranged between 170 and 1050.

MODELING METHOD OF HANN AND EVETT

A mathematical model developed at Texas A&M University under the supervision of Hann has been applied to several Gulf Coast
estuaries including the San Bernard, the Neches, and the Houston Ship Channel. The primary reference for this model is a doctoral dissertation written by Evett (23). A revised edition of this model has been written in a problem-oriented language (POL) format (30).

In comparison with all of the other models discussed thus far, Hann's approach is unique in that it can be applied to partially stratified estuaries. This model uses the following two solutions to a one-dimensional form of the mass transport equation:

\[ C = C_0 e^{-K_d x/U} \]  \hspace{1cm} (4-5) 

and

\[ C = C_0 e^{jx} \]  \hspace{1cm} (4-6) 

where

\[ j = \frac{U}{2E} \left( 1 \pm \sqrt{1 + \frac{4EK_d}{U^2}} \right) \]

For these equations, \( E \) = the dispersion coefficient, \( C \) = the waste concentration, \( C_0 \) = the concentration at the outfall, \( x \) = the horizontal distance, \( U \) = the advective velocity, and \( K_d \) = the removal coefficient.

Equation 4-6 represents concentration of a non-conservative
pollutant in a vertically mixed (homogeneous) estuary where the optional minus sign is used for the downstream direction. Equation 4-5 represents the concentration below an outfall in a river with no horizontal dispersion. Equation 4-5 is used in the upper, fresh-water layer of a stratified estuary where it is assumed that diffusion is insignificant in comparison to advective transport. Allowances also are made to account for salt water flow upward from the saline wedge. The development of both the homogeneous and stratified cases assume a steady-state condition.

The methods of modeling a homogeneous estuary or a highly stratified estuary proceed in essentially the same manner. Initially, the estuary is divided into segments. The lengths of these segments can vary; usually one-mile segments are most convenient. A calculation is then made using Equations 4-5 and 4-6 to determine the concentration that would appear in each segment under steady-state conditions as a result of injecting a unit waste load of one pound per day into a particular segment. After this calculation is repeated for each segment, a matrix of "unit loading coefficients" results. The appropriate unit loading coefficients can then be multiplied by the actual waste load being put into each segment and the waste distribution in the estuary is obtained.

The unit loading coefficients are adjusted for use in a channel with varying area by the following technique. Assume that a unit load (one pound of BOD per day) is introduced into segment 5 and a concentration of C5 results. The quality in the downstream
segment 6 is then calculated assuming that it has a volume equal to segment 5. The calculation is made using Equation 4-6 for a homogeneous estuary and Equation 4-5 for the top layer of a highly stratified estuary. The next step is to modify the quality parameter in segment 6 to reflect the changed volume by means of the following formula:

\[
C_6, \text{modified} = \frac{VOL_5}{VOL_6}C_6
\]

(4-7)

where VOL_5 add VOL_6 equal the volumes of segments 5 and 6, respectively. This procedure is then repeated using the modified C_6 value to calculate C_7 and so on down the estuary. Similar calculations are made in the upstream direction in the case of a homogeneous estuary.

The procedure used by Hann and Evett to analyze a partially mixed estuary is based on the assumption that the waste concentration in each segment will fall at some value between that for a highly stratified estuary and that for a homogeneous estuary. The actual value will depend upon the amount of stratification existing in the estuary. For each segment, a calculation is made to determine a "degree of stratification", DS. The calculation of this term requires the following segment input parameters: average top salinity, A; average bottom salinity, B; and salinity of the ocean or bay outside the estuary, C. The "degree of stratification" is then calculated by the following formula:
\[ DS = \frac{B-A}{C-A} \]

Thus, to determine the waste pattern in a partially stratified estuary, a calculation is made for a homogeneous estuary and for a completely stratified estuary, and then each calculation is weighted in accordance with the above "degree of stratification." Evett (23) applied linear programming techniques to this model to determine the maximum waste discharges which could be applied to the channel while satisfying certain dissolved oxygen criteria.

ANALYTICAL METHODS OF HARLEMAN AND IPPEN

Harleman and Ippen have produced some of the most significant American research in relating experimental findings to the behavior of actual estuaries. Their work is summarized briefly in this section to demonstrate an alternative approach to the numerical analysis techniques employed by other researchers. Likewise, the experiments of Harleman and Ippen give significant insight toward determining some of the important input parameters for mass transport models.

Most of the estuary studies performed by these investigators since 1960 were initiated by the U. S. Army Corps of Engineers Committee on Tidal Hydraulics. The experimental data were obtained in the rectangular salinity flume of the Waterways Experiment Station at Vicksburg and in a smaller flume at the Hydrodynamics
Laboratory of the Massachusetts Institute of Technology. The scope of these investigations covered the following four phases and were reported in the indicated references:

1. The extent of salinity intrusion and the mean salinity distribution in an idealized estuary (50, 51);
2. The vertical mixing of fresh and salt water and the resulting vertical salinity distribution (36);
3. The vertical distribution of current velocities as affected by salinity distribution (36); and
4. The movement and deposition of sediments as affected by the density current phenomena (37).

The estuary research of Harleman and Ippen has led to a number of unique parameters which they feel are important in analyzing estuaries. Much of their work has been presented in a non-dimensional format so that results can be applied to estuaries of widely varying dimensions and behavior.

Most of the work done by Harleman and Ippen has been correlated with a parameter known as the "stratification number." The use of this parameter was concisely explained in these authors' analysis of the Rotterdam waterway (35): "The stratification number, G/J, is defined as the following ratio.

$$\frac{G}{J} = \frac{\text{rate of energy dissipation per unit mass of fluid}}{\text{rate of gain of potential energy per unit mass of fluid}}$$

(4-9)
The numerator was evaluated from an analysis of the damping of the tidal wave in the channel. The denominator reflects the gain in potential energy due to increasing specific weight as the water becomes saline in moving down the estuary to the ocean. A large value of stratification number (of the order of 100 or more) indicates a well mixed estuary condition. Lower values indicate an increasing tendency toward stratification with a two-layer saline wedge as a limiting case. Although the stratification number was a useful parameter in correlating the Vicksburg Waterway Experiment Station (W.E.S.) channel tests, it is inconvenient to evaluate in actual estuaries. The difficulty is primarily due to the necessity of evaluating the rate of energy dissipation from tidal data."

During a reanalysis of the W.E.S. data it was found that a dimensionless parameter containing the tidal prism, Froude number (based on the maximum tidal velocity), fresh water discharge, and the tidal period was uniquely related to the stratification number. For convenience in later discussion this combination will be called "estuary number." In comparison with the stratification number, the estuary number is relatively easy to evaluate in actual estuaries. The estuary number is defined as follows:

\[ \text{estuary number} = \frac{P_t}{Q_f} \] ............................ (4-10)

where \( P_t \) = tidal prism, the volume of sea water entering the
estuary on the flood tide, \( F_o = \text{Froude number} = \frac{u_o}{\sqrt{gh}}, \) \( u_o \) is the maximum flood tide velocity at \( x = 0 \), and \( h \) is the mean depth at \( x = 0 \). \( Q_f \) = fresh water discharge, and \( T \) = tidal period. The relation between the stratification number and the estuary number is shown in Figure 4-1 for the W.E.S. tests.

In their one-dimensional treatment of estuary analysis, Harleman and Ippen use what they term an "apparent diffusion coefficient \( D'_x \) (50, 35, 51). Through this term, they attempt to take into consideration the effects of large scale internal circulation estuaries and the net effects of vertical transport. The ratio of \( D'_x \) to the actual longitudinal coefficient of turbulent diffusion (\( D_t \)) approaches unity for the well-mixed estuaries and approaches an order of 1000 for the highly stratified (51). The \( D'_x/D_t \) ratio ranged between 20 and 180 for the W.E.S. studies (50).

To solve the one-dimensional form of the mass transport equation, it is necessary to express the diffusion coefficient in terms of \( x \). Harleman and Ippen assumed that the apparent diffusion coefficient decreased linearly with \( x \) in the upland direction. This approach seemed to work well for a salinity flume with a constant cross-section. Since salinity at low water slack cannot equal "ocean" salinity, the variation of \( D'_x \) was extended seaward a distance \( b \) where ocean salinity was assumed to exist. The equation for the apparent diffusion coefficient becomes:

\[
D'_x = \frac{D'_0 b}{x + b} \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad (4-11)
\]
FIGURE 4 - 1. RELATIONSHIP BETWEEN STRATIFICATION NUMBER AND ESTUARY NUMBER
where \( D'_x = D'_0 \) at \( x = 0 \), the mouth of the channel. Thus, the governing equation for salinity in the one-dimensional analysis becomes:

\[
\frac{s_{1ws}}{s_0} = \exp \left( -\frac{V_f (x+B)^2}{2 D'_0 B} \right) \ldots \ldots \ldots \ldots (4-12)
\]

where \( s_{1ws} \) = local salinity at low water slack, \( s = \) ocean salinity, and \( V_f \) is the velocity due to fresh water discharge. If salinity is known at low water slack for at least two points, then the parameters \( D'_0 \) and \( B \) can be determined.

Other parameters which Harleman and Ippen feel are important in one-dimensional analysis are \( D'_0 / V_f B, 2 B/u_0 T, \) and \( Q_T / P_t \); all the parameters in these ratios have been defined previously. Analysis using these parameters has been applied to the Rotterdam Waterway by Harleman and Abraham (35). Several characteristics of the waterway were defined; unfortunately, most of the Vicksburg flume data falls outside the range of the waterway data.

Most recent studies of the two-dimensional behavior of estuaries have been performed by Harleman and Ippen to analyze vertical transport (36) and shoaling (37). A large amount of information was collected on tidal time-averaged values of vertical velocity, horizontal velocity, salinity profiles, and vertical dispersion in an idealized estuary of constant cross section. Vertical velocities were found to be in the downward
direction in the seaward half of the intrusion region and upward in the landward half (36). This finding is in contradiction to some other researchers who assume that a net vertical motion from bottom to surface exists throughout the intrusion length. In addition, certain parameters were found which proved to be useful in the qualitative analysis of estuary shoaling.

**TWO-DIMENSIONAL MODELS OF GALVESTON BAY**

Several studies of Galveston Bay, Texas, have been made recently to evaluate the bay's hydrodynamic and mass transport characteristics. Reid and Bodine modeled the behavior of the bay for storm surge conditions (84). Masch and Shankar extended this work to include mass transport considerations (61, 62, 88, 89). The Galveston Bay Study, through the consulting firm Tracor, Inc., is applying similar models to Galveston Bay (21, 22). In addition, the Texas Water Development Board is extending the use of the Masch/Shankar models to other estuaries along the Texas coast (96).

The work of Reid and Bodine is based on a vertically integrated form of the equations of motion and continuity. These equations allow for rainfall, wind stress, and quadratic bottom friction but ignore terms dealing with momentum advection and the Coriolis force:

\[
\frac{3U}{3t} + g \frac{3H}{3x} = kW^2 \cos \psi - fOU^2 \quad \ldots \quad (4-13)
\]
\[ \frac{\partial V}{\partial t} + gZ \frac{\partial H}{\partial y} = KW^2 \sin \psi - fQVZ^{-2} \] ........................ (4-14)

\[ \frac{\partial H}{\partial t} + \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} = R \] ........................ (4-15)

where \( U \) and \( V \) are the vertically integrated \( x \) and \( y \) horizontal components, respectively, of transport per unit width; \( H \) is the water level elevation relative to the local mean sea level datum; \( Z \) is the depth of water at position \( x, y \), at time \( t \); \( Q \) is equal to the vector average of transport per unit width and is obtained from the positive root of the radical \( Q = \sqrt{U^2 + V^2} \); \( R \) equals the rainfall rate; \( f \) is a dimensionless bed resistance coefficient; \( W \) is the wind speed 10 meters above the water; \( \psi \) is the angle between the wind velocity vector and the \( x \)-axis; \( K \) is the dimensionless Van Dorn coefficient for wind stress and is considered to be a function of \( W \). Reid and Bodine solve Equations 4-13, 4-14, and 4-15 for \( U, V, \) and \( H \), using a finite-difference recursion equation (84). The two-dimensional grid for Galveston Bay uses a two-nautical-mile spacing. The model has been verified for Hurricanes Carla of 1961 and Cindy of 1963.

Masch and his associates (61) have made certain revisions to the Reid/Bodine hydrodynamics model in order to make the model more applicable to conditions other than storm surges. Part of this refinement included changing the spacing to a one-mile grid. Shankar (88, 89, 62) added supplemental versatility to
the model by coupling it with a two-dimensional salinity model.

Shankar applies the following form of the mass transport equation, where the terms are defined as in previous examples:

\[
\frac{\partial C}{\partial t} = \frac{\partial}{\partial x} \left( E_x \frac{\partial C}{\partial x} \right) + \frac{\partial}{\partial y} \left( E_y \frac{\partial C}{\partial y} \right) - U \frac{\partial C}{\partial x} - V \frac{\partial C}{\partial y} \ldots \quad (4-16)
\]

The Bay was represented according to a repeating grid of cells shown in Figure 4-2. In this grid, the dispersion terms and concentration are cell-centered, whereas the velocities are defined at the cell boundaries. Thus, the following explicit finite-difference equation was developed \((88, 89)\) to approximate Equation 4-16:

\[
\frac{c_{i,j}^{(k+1)} - c_{i,j}^{(k)}}{\Delta t} = \frac{1}{2(\Delta x)^2} \left[ E_{x_{i,j}} c_{i+1,j}^{(k)} - 2E_{x_{i,j}} c_{i,j}^{(k)} 
+ E_{x_{i,j}} c_{i-1,j}^{(k)} + E_{x_{i+1,j}} \left( c_{i+1,j}^{(k)} - c_{i,j}^{(k)} \right) 
- E_{x_{i-1,j}} \left( c_{i,j}^{(k)} - c_{i-1,j}^{(k)} \right) \right]

+ \frac{1}{2(\Delta y)^2} \left[ E_{y_{i,j}} c_{i,j+1}^{(k)} - 2E_{y_{i,j}} c_{i,j}^{(k)} 
+ E_{y_{i,j}} c_{i,j-1}^{(k)} + E_{y_{i+1,j}} \left( c_{i,j+1}^{(k)} - c_{i,j}^{(k)} \right) \right]
\]
FIGURE 4-2. - CELL STRUCTURE FOR SALINITY MODEL OF SHANKAR AND MASCH
\[
- E_{x, i, j-1} \left( \frac{C_{x, i, j}^{(k)} - C_{x, i, j+1}^{(k)}}{2 \Delta x} \right)
\]

\[
- \left( \frac{U_{i, j} + U_{i-1, j}}{2} \right) \left[ \frac{C_{y+1, j}^{(k)} - C_{y-1, j}^{(k)}}{2 \Delta y} \right]
\]

\[
- \left( \frac{V_{i, j} + V_{i, j-1}}{2} \right) \left[ \frac{C_{z+1, j}^{(k)} - C_{z, j-1}^{(k)}}{2 \Delta z} \right]
\]

This equation can be rearranged to compute the unknown salinities, \(C_{i, j}^{(k+1)}\), at time \(t + \Delta t\) from the known salinities, \(C_{i, j}^{(k)}\), at time \(t\).

Tracor and the Galveston Bay Study (GBS) use a hydrodynamic model for the Bay which is basically the same as that used by Masch (21). Likewise, the one-mile grid used by the GBS for mass transport is defined in a manner similar to that of Shankar (Figure 4-2), except that the GBS expresses \(U, V,\) and \(C\) at the cell center and dispersion at the cell walls (22). In addition, the GBS model includes terms for various sources and sinks.

Tracor has applied the GBS model extensively to predict salinity, BOD, temperature, and nutrient patterns in Galveston Bay. Tracor has recently edited a noteworthy report which assesses the state-of-the-art of estuary modeling techniques (20) additional information on the Galveston Bay model can be found in that report.

Recently, Leendertse (59, 60) has also applied finite
difference schemes to hydrodynamic and water quality phenomena.

VERTICAL MODEL BY PRITCHARD AND WILSON

A two-layered, segmented model was suggested by Pritchard (81) in 1969. His approach has been applied recently by Wilson (108) at the Chesapeake Bay Institute under Pritchard's supervision. This analysis was directed towards the flushing of pollutants from a partially mixed estuary in which the flow is strongly two-layered. The segmentation method used by Wilson is shown in Figure 4-3. Horizontal dispersion was ignored on the assumption that mixing could be represented properly by the combined use of vertical mixing and horizontal velocity.

A set of equations was developed to represent the mass balances between the segments and these equations were integrated with time by a Hamming predictor-corrector scheme. This approach was applied to the Northwest Branch of the Baltimore Harbor.

Wilson (108) concluded that the two-layered, segmented model could describe adequately the gross features of advection and diffusion within the Northwest Branch. Use of the model led to estimates of steady-state concentrations; these estimates were considered to be in good accord with field data.
FIGURE 4-3. - SCHEMATIC FOR THE TWO-DIMENSIONAL MODEL OF WILSON AND PRITCHARD INDICATING SEGMENTATION AND PARAMETERS USED TO CHARACTERIZE FLOW. (Q_u, Q_v, and Q_y ARE VOLUME RATES OF NET ADVECTIVE FLOW; E IS THE COEFFICIENT OF VERTICAL DIFFUSIVE EXCHANGE.) (108)
CHAPTER V

DEVELOPMENT OF FINITE DIFFERENCE SCHEMES

As the advantages of high speed digital computers are becoming more widely recognized in the field of mathematical modeling, finite difference techniques are becoming more popular. Finite difference methods are based on the assumption that derivatives can be approximated by using the values of functions at points which are separated by finite increments of space or time. The finite difference approximations that are used to represent partial derivatives can be derived by truncating a Taylor series expansion of a function at a point or, more simply, by visualizing the relationships between slopes and values of a function at separated points. For instance, the following relationships are often used to approximate first and second-order derivatives:

\[
\frac{\partial C}{\partial x} = \frac{C_{x+1,t} - C_{x,t}}{\Delta x} + O(\Delta x) \quad \ldots \quad (5-1)
\]

\[
\frac{\partial C}{\partial x} = \frac{C_{x,t} - C_{x-1,t}}{\Delta x} + O(\Delta x) \quad \ldots \quad (5-2)
\]

\[
\frac{\partial C}{\partial x} = \frac{C_{x+1,t} - C_{x-1,t}}{2\Delta x} + O[(\Delta x)^2] \quad \ldots \quad (5-3)
\]

\[
\frac{\partial^2 C}{\partial x^2} = \frac{C_{x-1,t} - 2C_{x,t} + C_{x+1,t}}{(\Delta x)^2} + O[(\Delta x)^2] \quad \ldots \quad (5-4)
\]
The x and t subscripts denote location of the value C in a space grid and time grid, respectively, and the O(Δx) symbol represents the order of magnitude of the error. The Δx represents a dimensionless distance fraction. Equation 5-1 is a forward form of the finite difference representation and Equation 5-2 is a backward form; Equations 5-3 and 5-4 are central difference forms. By using more neighboring points, an unlimited number of other finite difference approximations can be obtained. However, the above forms are the most compact and were judged to be the most useful ones for this study.

The mass transport equation which we are dealing with in this research is a parabolic partial differential equation. When developing a numerical procedure to approximate this equation, a choice must often be made between and "explicit" formulation and an "implicit" formulation. An approach which expresses one unknown pivotal value directly in terms of known pivotal values is called an "explicit" formulation. An approach in which several unknowns are related to one or several unknowns by an equation is called an "implicit" formulation. The implicit approach requires the solution of a set of simultaneous equations, whereas the explicit approach requires the solution of only one equation for each point.

ONE DIMENSIONAL EQUATIONS WITH CONSTANT COEFFICIENTS

These concepts can be demonstrated by showing the derivation
of the finite difference equations for the one-dimensional, constant
coefficient case. The partial differential equation can be ex-
pressed as

$$\frac{\partial C}{\partial t} = E\frac{\partial^2 C}{\partial x^2} - U\frac{\partial C}{\partial x} - K_d C \ldots \ldots \ldots \ldots \ldots \ldots \ldots (5-5)$$

where \( C \) is concentration (ppm), \( E \) is a constant dispersion
(ft\(^2\)/sec), \( U \) is a constant velocity (ft/sec), and \( K_d \) is a constant
decay rate (/sec).

Explicit Formulation. In terms of the appropriate finite differ-
cequations, an explicit form of this equation is as follows:

$$\frac{C_{x,t+1} - C_{x,t}}{\Delta t} = E \left( \frac{C_{x+1,t} - 2C_{x,t} + C_{x-1,t}}{(\Delta x)^2} \right)$$

$$- U \left( \frac{C_{x+1,t} - C_{x-1,t}}{2\Delta x} \right) - K_d (C_{x,t}) \ldots \ldots (5-6)$$

After rearranging terms, the following equation results:

$$C_{x,t+1} = \frac{E\Delta t}{(\Delta x)^2} \left( C_{x+1,t} - 2C_{x,t} + C_{x-1,t} \right)$$

$$- \frac{U\Delta t}{2\Delta x} \left( C_{x+1,t} - C_{x-1,t} \right)$$

$$- \Delta t K_d (C_{x,t}) + C_{x,t} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (5-7)$$

This equation is used repeatedly at each time step until all the
new values for the t+1 set of points are calculated. Since all
the concentrations with a t subscript are known before the equation
is applied, the calculation of the t+1 concentrations is straight-
forward except at the boundaries. At the boundaries, certain pro-
cedures must be followed to obtain a solution. These boundary
conditions are discussed in a later chapter of this report.

*Implicit Formulation.* Using a more general approach, the mass
transport equation can be represented by the following weighted
average approximation (after Smith, 91):

\[
\frac{C_{x,t+1} - C_{x,t}}{\Delta t} = \frac{E}{(\Delta x)^2} \left\{ \begin{array}{l}
\omega \left( C_{x+1,t+1} - 2C_{x,t+1} + C_{x-1,t+1} \right) \\
+ (1 - \omega) \left( C_{x+1,t} - 2C_{x,t} + C_{x-1,t} \right) \\
- \frac{U}{2\Delta x} \left\{ \begin{array}{l}
\omega \left( C_{x+1,t+1} - C_{x-1,t+1} \right) \\
+ (1 - \omega) \left( C_{x+1,t} - C_{x-1,t} \right) \\
- K_d \left\{ \begin{array}{l}
\omega \left( C_{x,t+1} \right) + (1 - \omega) \left( C_{x,t} \right) \\
\end{array} \right. \end{array} \right. \right\} . \tag{5-8}
\]

This equation uses the same approximations for the derivatives as
were used previously for the explicit formulation. If we sub-
stitute \( \omega = 0 \), we obtain Equation 5-7 of the explicit scheme. If we
substitute \( \omega = 1/2 \), we obtain what is known as the Crank-Nicolson
implicit formulation. If we substitute $\theta = 1$, we obtain a fully implicit, backward difference formula. These formulations are compared schematically in Figures 5-1, 5-2, and 5-3. For any value of $\theta$ greater than zero, a set of simultaneous equations must be solved.

Using $\theta = 1/2$ in Equation 5-8, and collecting terms, the following Crank-Nicolson representation is obtained:

$$
-A_i \left( C_{x-1,t+1} \right) + B_i \left( C_{x,t+1} \right) - D_i \left( C_{x+1,t+1} \right) \\
= A_i \left( C_{x-1,t} \right) - B'_i \left( C_{x,t} \right) + D_i \left( C_{x+1,t} \right) \quad \cdots \cdots \cdots \cdots \ (5-9)
$$

where

$$
A_i = \left( \frac{E\Delta t}{2(\Delta x)^2} + \frac{U\Delta t}{4\Delta x} \right) \quad \cdots \cdots \cdots \cdots \ (5-10)
$$

$$
B_i = \left( \frac{E\Delta t}{(\Delta x)^2} + \frac{K\Delta t}{2} + 1 \right) \quad \cdots \cdots \cdots \cdots \ (5-11)
$$

$$
B'_i = B_i - 2
$$

$$
D_i = \left( \frac{E\Delta t}{2(\Delta x)^2} - \frac{U\Delta t}{4\Delta x} \right) \quad \cdots \cdots \cdots \cdots \ (5-12)
$$

for $i = 2, 3, 4, \ldots, N - 3$, where $N$ is the number of points.

For $N$ points, there are $N - 2$ equations. By arranging the knowns and unknowns for a series of distance steps, the following
**Figure 5-1.** - **Explicit** ($\theta = 0$)

- $\triangle$ = Known values
- $\circ$ = Unknown values

**Figure 5-2.** - **Crank-Nicolson** ($\theta = 1/2$)

**Figure 5-3.** - **Fully Implicit** ($\theta = 1$)
The \( W_i \) terms represent the collection of known concentrations and their coefficients, as shown by the right side of Equation 5-9. The values of the terms \( B_1, D_1, W_1, A_{N-2}, B_{N-2}, \) and \( W_{N-2} \) are dependent upon the boundary conditions which are applied to the problem; these considerations will be discussed in a later chapter of this report.

Fortunately, the coefficients of these simultaneous equations form a tridiagonal matrix which can be solved by a relatively simple Gauss elimination method, also known as the "Thomas Algorithm." If more than three unknowns are used in each equation, a more time-consuming matrix solution technique would be needed; this discourages the use of formulas which require more than three
points to approximate derivatives.

The Gauss method begins by using the first equation to eliminate \( C_{2,t+1} \) from the second equation. Then the new second equation is used to eliminate \( C_{3,t+1} \) from the third equation, and so on. Finally, the new last-but-one equation can be used to eliminate \( C_{N-2,t+1} \) from the last equation. This leaves one equation with only one unknown, \( C_{N-1,t+1} \). Thus, the unknown \( C_{i,t+1} \) values can be found by back-substitution. This procedure is summarized in the following steps (11, 91):

1. Define all \( A_{i}, B_{i}, D_{i}, \) and \( W_{i} \) values;
2. Define and calculate alpha's \( (\alpha_{i}) \);

\[
\alpha_{1} = B_{1} \quad \ldots \quad (5-14)
\]

\[
\alpha_{i} = B_{i} - \frac{A_{i} D_{i-1}}{\alpha_{i-1}} \quad \ldots \quad (5-15)
\]

for \( i = 2, 3, \ldots, N-2 \)

3. Define and calculate S's

\[
S_{1} = W_{1} \quad \ldots \quad (5-16)
\]

\[
S_{i} = W_{i} + \frac{A_{i} S_{i-1}}{\alpha_{i-1}} \quad \ldots \quad (5-17)
\]

for \( i = 2, 3, \ldots, N-2 \)

4. Calculate C's starting with \( C_{N-1,t+1} \)
\[ C_{N-1,t+1} = \frac{S_{N-2}}{\alpha_{N-2}} \] \hspace{0.5cm} (5-18)

\[ C_{i,t+1} = \frac{1}{\alpha_i} S_i + D_{i} C_{i+1,t+1} \] \hspace{0.5cm} (5-19)

for \( i = N-2, N-3, \ldots , 2. \)

**ONE DIMENSIONAL EQUATIONS WITH VARYING COEFFICIENTS**

The finite difference equations become more complicated when the coefficients are allowed to vary with time and distance. The appropriate partial differential equation which permits varying coefficients and varying width is the following:

\[ W \frac{\partial C}{\partial t} = - W_x \frac{\partial C}{\partial x} + \frac{\partial}{\partial x} \left( E_x W \frac{\partial C}{\partial x} \right) - W_x K D_x C + (\text{Sources} - \text{Sinks}) \] \hspace{0.5cm} (5-20)

The terms in this equation are as defined in Equation 3-24.

*Explicit Formulation.* - The explicit formulation for this equation uses the approximations listed in Equations 5-1, 5-3, and 5-4. In the following finite difference equations, a \( z \) subscript is used in order to be consistent with the two-dimensional formulations; this \( z \) is always equal to 1 in the one-dimensional formulas. The one-dimensional, explicit, finite difference approximation to Equation 5-20 is shown in Figure 5-4, as well as
Partial Differential Equation For One-Dimensional Mass Transport:

\[ \frac{\partial C}{\partial t} - W_x \frac{\partial C}{\partial x} + \frac{E_x W_x}{2\Delta x} \left( \frac{\partial C}{\partial x} \right) - W_x K D_x C + (\text{Sources} - \text{Sinks}) = 0 \]  

(5-20)

Explicit Finite Difference Formulation:

\[ \frac{W_x}{\Delta t} \left( C_{x,z,t+1} - C_{x,z,t} \right) = - \frac{W_x V_x z}{2\Delta x} \left( \frac{C_{x+1,z,t} - C_{x-1,z,t}}{\Delta x} \right) \]

\[ + \left( \frac{W_{x+1} E_{x+1,z} + W_x E_x}{2\Delta x} \right) \left( \frac{C_{x+1,z,t} - C_{x,z,t}}{\Delta x} \right) \]

\[ - \left( \frac{W_x E_x}{2\Delta x} \right) \left( \frac{C_{x,z,t} - C_{x-1,z,t}}{\Delta x} \right) \]

\[ - W_x K D_x C_{x,z,t} + (\text{Sources} - \text{Sinks}) \]  

(5-21)

FIGURE 5-4. - EXPLICIT FINITE DIFFERENCE EQUATIONS FOR ONE-DIMENSIONAL MASS TRANSPORT WITH VARYING COEFFICIENTS

(Continued on next page)
Rearranged Explicit Equation:

\[ C_{x,z,t+1} = C_{x-1,z,t} \left( \frac{\Delta t}{W_x} \right) \left( \frac{W_{VX,x,z}}{2\Delta x} + \frac{W_{EX,x,z} + W_{x-1EX,x-1,z}}{2(\Delta x)^2} \right) \]

\[ - C_{x,z,t} \left( \frac{\Delta t}{W_x} \right) \left( \frac{W_{x+1EX,x+1,z}}{2(\Delta x)^2} + \frac{W_{xEX,x,z} + W_{x-1EX,x-1,z}}{2(\Delta x)^2} \right) \]

\[ + C_{x+1,z,t} \left( \frac{\Delta t}{W_x} \right) \left( - \frac{W_{VX,x,z}}{2\Delta x} + \frac{W_{x+1EX,x+1,z} + W_{xEX,x,z}}{2(\Delta x)^2} \right) \]

\[ + C_{x,z,t} - KD_x \Delta t C_{x,z,t} + \text{(Sources - Sinks)} \]

(End of Figure 5-4)
a more useful rearranged version. The Equation 5-22 is solved repeatedly for each time step as in the explicit equation with constant coefficients discussed previously.

*Implicit Formulation.* - The Crank-Nicolson implicit finite difference formulation for the one-dimensional mass transport equation with varying coefficients is shown in Figure 5-5; the z subscript is equal to 1 in this one-dimensional framework and terms of the form "- - NEXT" refer to values at time t+1.

This set of equations can be arranged into a tridiagonal matrix. For easier programming, the indexing system is different than the previous one-dimensional, implicit array:

\[
\begin{align*}
B(2)C_{2,z,t+1} + G(2)C_{3,z,t+1} &= D(2) \\
A(3)C_{2,z,t+1} + B(3)C_{3,z,t+1} + G(3)C_{4,z,t+1} &= D(3) \\
A(4)C_{3,z,t+1} + B(4)C_{4,z,t+1} + G(4)C_{5,z,t+1} &= D(4) \\
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
A(N-1)C_{N-2,z,t+1} + B(N-1)C_{N-1,z,t+1} &= D(N-1)
\end{align*}
\]

This set of N-2 equations is used for a grid with N horizontal points. The values for B(2), G(2), D(2), A(N-1), B(N-1), and
D(N-1) depend upon the boundary conditions of the problems. The values for other D(I) are found by collecting all the C terms in Equation 5-23 with a subscript of t and their associated coefficients. This yields the values for the arrays A, B, G, and D shown in Figure 5-6. The tridiagonal matrix for this formulation is solved by the Gauss elimination scheme outlined previously in Equations 5-14 through 5-19.
Partial Differential Equation for One Dimensional Mass Transport:

\[ W \frac{\partial C}{\partial t} = - W_x \frac{\partial C}{\partial x} + \frac{\partial}{\partial x} \left( E_x W_x \frac{\partial C}{\partial x} \right) - W_x K_D x C + (\text{Sources} - \text{Sinks}) \]  \[ \text{(5-20)} \]

Implicit Crank-Nicolson Finite Difference Formulation:

\[ \frac{W_x}{\delta t} \left( C_{x,z,t+1} - C_{x,z,t} \right) = - \frac{W_{x,Vx,\text{NEXT}}_{x,z}}{2} \begin{Bmatrix} \frac{C_{x+1,z,t+1} - C_{x-1,z,t+1}}{2\Delta x} \\ \frac{C_{x+1,z,t} - C_{x-1,z,t}}{2\Delta x} \end{Bmatrix} \]

\[ + \frac{1}{2} \begin{Bmatrix} \frac{W_{x+1,EX,\text{NEXT}}_{x+1,z} + W_{x,EX,\text{NEXT}}_{x,z}}{2\Delta x} \frac{C_{x+1,z,t+1} - C_{x,z,t}}{\Delta x} \\ \frac{W_{x+1,EX,\text{NEXT}}_{x+1,z} + W_{x,EX,\text{NEXT}}_{x,z}}{2\Delta x} \frac{C_{x+1,z,t} - C_{x,z,t}}{\Delta x} \end{Bmatrix} \]

\[ + \frac{1}{2} \begin{Bmatrix} \frac{W_{x+1,EX,\text{NEXT}}_{x+1,z} + W_{x,EX,\text{NEXT}}_{x,z}}{2\Delta x} \frac{C_{x+1,z,t+1} - C_{x-1,z,t}}{\Delta x} \\ \frac{W_{x+1,EX,\text{NEXT}}_{x+1,z} + W_{x,EX,\text{NEXT}}_{x,z}}{2\Delta x} \frac{C_{x+1,z,t} - C_{x-1,z,t}}{\Delta x} \end{Bmatrix} \]

FIGURE 5-5. - IMPLICIT CRANK-NICOLSON FINITE DIFFERENCE EQUATION FOR ONE-DIMENSIONAL MASS TRANSPORT WITH VARYING COEFFICIENTS

(Continued on next page)
\[
- \frac{1}{2} \left( \frac{W_x E X_{x,z} + W_{x-1} E X_{x-1,z}}{2\Delta x} \right) \left\{ \frac{C_{x,z,t} - C_{x-1,z,t}}{\Delta x} \right\} \\
- \frac{1}{2} W_x K D_x \left\{ C_{x,z,t+1} \right\} - \frac{1}{2} W_x K D_x \left\{ C_{x,z,t} \right\} + \text{(Sources - Sinks)} \\
\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ ld
\[ A(x) = \left( \frac{\Delta t}{\Delta x} \right) \left( - \frac{w_{x \cdot vxnext, x \cdot z}}{4\Delta x} - \frac{w_{x \cdot exnext, x \cdot z} + w_{x \cdot l \cdot exnext, x \cdot l \cdot z}}{4(\Delta x)^2} \right) \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot (5-25) \]

\[ B(x) = 1 + \left( \frac{\Delta t}{\Delta x} \right) \left( \frac{w_{x+1 \cdot exnext, x+1 \cdot z} + 2w_{x \cdot exnext, x \cdot z} + w_{x-1 \cdot exnext, x-1 \cdot z}}{4(\Delta x)^2} \right) + \frac{w_K D_x}{2} \cdot \cdot \cdot (5-26) \]

\[ G(x) = \left( \frac{\Delta t}{\Delta x} \right) \left( \frac{w_{x \cdot vxnext, x \cdot z} - w_{x+1 \cdot exnext, x+1 \cdot z} + w_{x \cdot exnext, x \cdot z}}{4(\Delta x)^2} \right) \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot (5-27) \]

\[ D(x) = C_{x \cdot z \cdot t} + \left( \frac{\Delta t}{\Delta x} \right) \left( - \frac{w_{x \cdot vx, x \cdot z}}{4\Delta x} \right) \left\{ C_{x+1 \cdot z \cdot t} - C_{x-1 \cdot z \cdot t} \right\} + \left( \frac{\Delta t}{\Delta x} \right) \left( \frac{w_{x+1 \cdot ex, x+1 \cdot z} + w_{x \cdot ex, x \cdot z}}{4(\Delta x)^2} \right) \left\{ C_{x+1 \cdot z \cdot t} - C_{x \cdot z \cdot t} \right\} + \left( \frac{\Delta t}{\Delta x} \right) \left( \frac{w_{x \cdot ex, x \cdot z} + w_{x \cdot l \cdot ex, x \cdot l \cdot z}}{4(\Delta x)^2} \right) \left\{ C_{x \cdot z \cdot t} - C_{x-1 \cdot z \cdot t} - \frac{K D_x}{2} \left\{ C_{x \cdot z \cdot t} \right\} \right\} \cdot \cdot \cdot \cdot (5-28) \]

**Figure 5-6. - Coefficients for Tridiagonal Matrix**
TWO DIMENSIONAL EQUATIONS WITH VARYING COEFFICIENTS

The most complicated form of the mass transport equation which has been programmed in this research allows for two dimensional convective-dispersion with variable coefficients:

\[ W_x \frac{\partial C}{\partial t} = \frac{\partial}{\partial x} \left( E_x W_x \frac{\partial C}{\partial x} \right) + \frac{\partial}{\partial z} \left( E_z W_x \frac{\partial C}{\partial z} \right) \]

\[ - W_x V_x \frac{\partial C}{\partial x} - W_x V_z \frac{\partial C}{\partial z} - W_x K D_x C + \text{(Sources - Sinks)} \quad (5-29) \]

This equation was derived and discussed in Chapter III.

Explicit Formulation. The two dimensional explicit formulation takes the same approach as Equations 5-21 and 5-22 of the one-dimensional explicit method, but in addition includes terms in the vertical \( z \) direction. The two-dimensional explicit finite difference equation is shown in Figure 5-7. A more convenient rearranged version is shown also.

The Equation 5-30 can be visualized as the computational molecule shown in Figure 5-8.
Partial Differential Equation for Two Dimensional Mass Transport:

\[
\frac{\partial C}{\partial t} + \frac{\partial}{\partial x} \left( W \frac{\partial C}{\partial x} \right) + \frac{\partial}{\partial z} \left( X \frac{\partial C}{\partial z} \right) = \omega (C) + W_D C + (\text{Sources} - \text{Sinks})
\]

\[
\frac{W_X}{\Delta t} \left\{ C_{x+1,z,t+1} - C_{x,z,t+1} \right\} - \frac{W_X}{\Delta z} \left\{ C_{x+1,z+1,t} - C_{x,z+1,t} \right\} - \frac{W_X E_X}{\Delta x} \left\{ C_{x+1,z,t} - C_{x,z,t} \right\} + \frac{W_X F_X}{\Delta x} \left\{ C_{x-1,z,t} - C_{x,z,t} \right\} + \frac{W_X E_z}{\Delta z} \left\{ C_{x,z+1,t} - C_{x,z,t} \right\} + \frac{W_X F_z}{\Delta x} \left\{ C_{x-1,z+1,t} - C_{x,z+1,t} \right\} = 0
\]

Explicit Finite Difference Formulation:

\[
\frac{W_X}{\Delta t} \left\{ C_{x,z,t+1} - C_{x,z,t} \right\} = - \frac{W_X}{\Delta x} \left( \frac{W_X + W_X E_x}{2 \Delta x} \right) \left\{ C_{x+1,z,t} - C_{x,z,t} \right\} + \frac{W_X}{\Delta z} \left( \frac{W_X + W_X E_z}{2 \Delta z} \right) \left\{ C_{x,z+1,t} - C_{x,z,t} \right\} + \frac{W_X}{\Delta x} \left( \frac{W_X + W_X E_z}{2 \Delta x} \right) \left\{ C_{x-1,z,t} - C_{x,z,t} \right\} + \frac{W_X F_X}{\Delta x} \left\{ C_{x-1,z,t} - C_{x,z,t} \right\} + \frac{W_X F_z}{\Delta z} \left\{ C_{x,z+1,t} - C_{x,z,t} \right\}
\]

(Continued on next page)
\[ -\left( \frac{W_x E x, z + W_x E x, z-1}{2\Delta z} \right) \left( \frac{C_{x, z, t} - C_{x, z-1, t}}{\Delta z} \right) \]

\[ - W_x K D x, x, z, t + (\text{Sources} - \text{Sinks}) \quad \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ ld - (Continued)
\[ + C_{x,z+1,t} \left( \frac{\Delta t}{W_x} \right) \left( \frac{W_x V Z_{x,z}}{2 \Delta z} + \frac{W_x E Z_{x,z+1}}{2(\Delta z)^2} \right) \]

\[ + C_{x,z,t} \cdot \Delta t \cdot K D_{x,z} \cdot C_{x,z,t} + (\text{Sources} - \text{Sinks}) \ldots \ldots \ldots \ldots \ldots \ (5-31) \]

(End of Figure 5-7)
UNKNOWNS

FIGURE 5-8. - TWO-DIMENSIONAL EXPLICIT COMPUTATION MOLECULE
**Implicit Formulation.** In order to avoid solving a large, two-dimensional matrix by a time-consuming iterative technique, a procedure known as the "implicit alternating-direction method" can be used. This approach, as discussed by Peaceman and Rachford (74), and later modified by Douglas (18), permits the application of the efficient tridiagonal matrix solution explained earlier in this report to a two-dimensional, Crank-Nicolson implicit approximation to the mass transport equation. Additional discussions of this approach are found in other sources (11, 28, 91). This method requires a two-phase procedure. During the initial phase, a Crank-Nicolson scheme is applied only to concentrations in the x-direction. The new concentrations, called CSTAR, which result from the solution of this system of equations are used in a second phase where the Crank-Nicolson scheme is applied to concentrations in the z-direction. The values of CSTAR can be considered as initial approximations which are used to find a more accurate solution during the second phase. This procedure can be expressed schematically as follows:

**First Phase.**

\[
\frac{w_x}{\Delta t} (\text{CSTAR}_{x,z} - C_{x,z,t}) = \frac{1}{2} \text{(Derivatives and coefficients of C in x-direction at time t+1)}
\]
\[ + \frac{1}{2} \text{ (Derivatives and coefficients of } C \text{ in } x\text{-direction at time } t) \]
\[ + \frac{1}{2} \text{ (Derivatives and coefficients of } C \text{ in } z\text{-direction at time } t) \]
\[ \text{................. (5-32)} \]

Second Phase.

\[ \frac{\Delta x}{\Delta t} \left( \frac{C_{x,z,t+1} - C_{x,z,t}}{2} \right) \]
\[ = \frac{1}{2} \text{ (Derivatives and coefficients of CSTAR in } x\text{ direction).} \]
\[ + \frac{1}{2} \text{ (Derivatives and coefficients of } C \text{ in } x\text{-direction at time } t) \]
\[ + \frac{1}{2} \text{ (Derivatives and coefficients of } C \text{ in } z\text{-direction at time } t+1) \]
\[ + \frac{1}{2} \text{ (Derivatives and coefficients of } C \text{ in } z\text{-direction at time } t) \]
\[ \text{................. (5-33)} \]

This two-phase approach requires the solution of two tri-diagonal matrices. The arrays are of the same form as Equation
5-24. The equations for these arrays are shown in Figures 5-9 and 5-10. As in the previous implicit cases, values for B(2), G(2), D(2), A(N-1), B(N-1), and D(N-1) are dependent upon the conditions at the boundaries. The computational molecule for the two-dimensional Crank-Nicolson scheme is shown in Figure 5-11. An example of a two-dimensional grid is shown in Figure 5-12. Values for dispersion, velocity, decay, and other variables must be defined for each grid point.
Where NX equals the number of points in the x-direction and for X = 3, 4, 5, ... NX-2:

\[ A(X) = \left( \frac{\Delta t}{W_X} \right) \left( \frac{W_{VX X+1, z} + W_{EX X, z} + W_{EX X, z}}{4(\Delta x)^2} \right) \]  
\[ B(X) = 1. + \left( \frac{\Delta t}{W_X} \right) \left( \frac{W_{X+1, z} + W_{EX X+1, z} + W_{X, z} + W_{X-1, z}}{4(\Delta x)^2} \right) \]  
\[ G(X) = \left( \frac{\Delta t}{W_X} \right) \left( \frac{W_{VX X, z} + W_{EX X, z} + W_{EX X, z}}{4(\Delta x)^2} \right) \]  

\[ D(X) = C_{X, z, t} + \left( \frac{\Delta t}{W_X} \right) \left[ - \frac{W_{VX X, z}}{4\Delta x} \left\{ C_{X+1, z, t} - C_{X-1, z, t} \right\} \right. \]  
\[ - \frac{2W_{VZ X, z}}{4\Delta z} \left\{ C_{X, z+1, t} - C_{X, z-1, t} \right\} + \left( \frac{W_{X+1, z} + W_{X, z}}{4(\Delta x)^2} \right) \left\{ C_{X+1, z, t} - C_{X, z, t} \right\} \]  
\[ - \left( \frac{W_{EX X, z} + W_{X-1, z}}{4(\Delta x)^2} \right) \left\{ C_{X, z, t} - C_{X-1, z, t} \right\} + 2 \left( \frac{W_{EZ X, z+1} + W_{EZ X, z}}{4(\Delta z)^2} \right) \]

**FIGURE 5-9.** - ARRAY VALUES FOR THE FIRST PHASE OF THE IMPLICIT PROCEDURE

(Continued on next page)
\[
\left\{ C_{x,z+1,t} - C_{x,z,t} \right\} \\
- 2 \left( \frac{W_x W_{x,z} + W_x W_{x,z-1}}{4(\Delta z)^2} \right) \left\{ C_{x,z,t} - C_{x,z-1,t} \right\} \\
\] 

(End of Figure 5-9)
Where NZ equals the number of points in the z-direction and for Z = 3, 4, 5, ..., NZ-2:

\[ A(Z) = \left( \frac{\Delta t}{W_x} \right) \left( \frac{W_x \text{VZNEXT}_{x,Z} - W_x \text{EZNEXT}_{x,Z} + W_x \text{EZNEXT}_{x,Z-1}}{4(\Delta z)^2} \right) \]  

\[ B(Z) = 1. + \left( \frac{\Delta t}{W_x} \right) \left( \frac{W_x \text{EZNEXT}_{x,Z+1} + W_x \text{EZNEXT}_{x,Z}}{4(\Delta z)^2} + \frac{W_x \text{EZNEXT}_{x,Z-1} + W_x \text{EZNEXT}_{x,Z}}{4(\Delta z)^2} \right) \]  

\[ G(Z) = \left( \frac{\Delta t}{W_x} \right) \left( \frac{W_x \text{VZNEXT}_{x,Z} - W_x \text{EZNEXT}_{x,Z+1} + W_x \text{EZNEXT}_{x,Z}}{4(\Delta z)^2} \right) \]  

\[ D(Z) = C_{x,z,t} + \left( \frac{\Delta t}{W_x} \right) - \frac{W_x \text{VXNEXT}_{x,z}}{4\Delta x} \left\{ C_{x+1,z,t} - C_{x-1,z,t} \right\} \]  

\[ - \frac{W_x \text{VX}_{x,z}}{4\Delta x} \left\{ C_{x+1,z,t} - C_{x-1,z,t} \right\} - \frac{W_x \text{VZ}_{x,z}}{4\Delta z} \left\{ C_{x,z+1,t} - C_{x,z-1,t} \right\} \]  

\[ + \left( \frac{W_x \text{EXNEXT}_{x,z+1} + W_x \text{EXNEXT}_{x,z}}{4(\Delta x)^2} \right) \left\{ C_{x+1,z,t} - C_{x,z,t} \right\} \]  

\[ - \left( \frac{W_x \text{EXNEXT}_{x,z} + W_x \text{EXNEXT}_{x-1,z}}{4(\Delta x)^2} \right) \left\{ C_{x,z,t} - C_{x-1,z,t} \right\} \]  

(Continued on next page)

**Figure 5-10.** - Array values for the second phase of the implicit procedure.
\[
+ \left( \frac{W_{x+1}E_x + W_x E_x}{4(\Delta x)^2} \right) \left\{ C_{x+1,z,t} - C_{x,z,t} \right\} - \left( \frac{W_x E_{x-1} + W_{x-1} E_{x-1}}{4(\Delta x)^2} \right)
\]
\[
\left\{ C_{x,z,t} - C_{x-1,z,t} \right\} + \left( \frac{W_x E_{x,z+1} + W_x E_{x,z}}{4(\Delta z)^2} \right) \left\{ C_{x,z,t+1} - C_{x,z,t} \right\} - \left( \frac{W_x E_{x,z} + W_x E_{x,z-1}}{4(\Delta z)^2} \right)
\]
\[
\left\{ C_{x,z,t} - C_{x,z-1,t} \right\} \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots (5-41)
\]

For mass transport problems with decay and other source/sink terms, this additional equation is used after the solution of the tridiagonal matrix:

\[
\text{FINAL} \quad C_{x,z,t+1} = C_{x,z,t+1} - \left( \frac{\Delta t K D_x}{2} \right) \left\{ C_{x,z,t+1} + C_{x,z,t} \right\} + \text{(Sources - Sinks)} \cdots \cdots \cdots \cdots (5-42)
\]

(End of Figure 5-10)
FIGURE 5-11. - TWO-DIMENSIONAL IMPLICIT COMPUTATIONAL MOLECULE
<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>X</td>
</tr>
<tr>
<td>X</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>o</td>
<td>X</td>
</tr>
</tbody>
</table>

- **O** Calculated Concentrations
- **X** Extrapolated Values and Image Points
- _____ Water Boundaries

**Figure 5-12. - Two-Dimensional Vertical Grid**
CHAPTER VI

STABILITY CONSIDERATIONS

Instability is an important problem which must be dealt with when applying finite difference techniques. Instability refers to the tendency of errors to grow without limit as the computation progresses. These errors can be generated by inexact initial conditions, inexact boundary conditions, oscillations generated in early time steps, or round-off errors. Stability, on the other hand, is the tendency to damp out errors as the computation progresses. If a solution is unstable, it becomes hopelessly inaccurate. On the other hand, a stable solution is not necessarily an accurate solution. The accuracy of a solution must be investigated after the stability criteria has been established.

EXPLICIT METHOD

Several sophisticated techniques are available for analyzing the stability of explicit finite difference formulations. However, in this study, the following simple rule was found to be the most convenient: stability is assured when the coefficient for each concentration value is greater than zero. Thus, examination of the Equation 5-31 leads to the following inequalities if variations with distance are considered to be small, and if the subscripts x, z, and t are dropped:
\[ \text{EX} \frac{\Delta t}{(\Delta x)^2} - \text{VX} \frac{\Delta t}{2\Delta x} > 0 \] (6-1)

\[ \text{EZ} \frac{\Delta t}{(\Delta z)^2} - \text{VZ} \frac{\Delta t}{2\Delta z} > 0 \] (6-2)

\[ 1 - 2\text{EX} \frac{\Delta t}{(\Delta x)^2} - 2\text{EZ} \frac{\Delta t}{(\Delta z)^2} - \Delta t \text{ KD} > 0 \] (6-3)

According to Equations 6-1 and 6-2, dispersion coefficients in a certain direction can not be set equal to zero when a velocity exists in that direction. Solution of these equations, in order, leads to the following stability criteria:

\[ \Delta x < \frac{2\text{EX}}{\text{VX}} \] (6-4)

\[ \Delta z < \frac{2\text{EZ}}{\text{VZ}} \] (6-5)

\[ \Delta t < \frac{(\Delta x)^2 (\Delta z)^2}{2\text{EX}(\Delta z)^2 + 2\text{EZ}(\Delta x)^2 + K\text{D}(\Delta x)^2 (\Delta z)^2} \] (6-6)

The corresponding stability criteria for one dimensional analysis is

\[ \Delta x < \frac{2\text{EX}}{\text{VX}} \] (6-7)

\[ \Delta t < \frac{(\Delta x)^2}{2\text{EX} + (\Delta x)^2 \text{KD}} \] (6-8)
One-dimensional stability restrictions can be examined rapidly by plotting coaxial graphs on log-log paper for Equations 6-7 and 6-8. Figures 6-1 and 6-2 display such graphs for a useful range of velocity and dispersion values.

When applying these criteria to the data in this study, the minimum numerators and maximum denominators were used. For example, when applying Equation 6-6, the maximum values for EX, EZ, and KD were used; likewise, when applying Equation 6-4, the minimum value for EX and the maximum value for VX were used. This approach leads to a conservative estimate of stability criteria, since maximum velocities and minimum dispersions rarely occur at the same point.

**IMPLICIT METHOD**

The Crank-Nicolson implicit method is not limited by the same stability criteria as the explicit formulation. However, literature and computation experience demonstrate that severe oscillations and inaccuracies can occur when the time increment exceeds twice the time criteria of the explicit approach. Thus, for the Crank-Nicolson formulation, the time increment should not exceed the following value when a new range of data is being analyzed:

\[
\Delta t < \frac{2(\Delta x)^2 (\Delta z)^2}{2EX(\Delta z)^2 + 2EZ(\Delta x)^2 + KD(\Delta x)(\Delta z)^2} \quad \cdots \cdots \cdots \quad (6-9)
\]

The corresponding one-dimensional limit is
FIGURE 6-1. DISTANCE INCREMENTS FOR STABILITY
FIGURE 6-2. TIME INCREMENTS FOR STABILITY
\[ \Delta t < \frac{2(\Delta x)^2}{2EX + (\Delta x)^2KD} \] (6-10)
CHAPTER VII

BOUNDARY CONDITIONS

Both the explicit and the implicit methods applied in this research use central difference formulas to represent first and second derivatives. As shown in Equations 5-3 and 5-4, these formulas require grid concentrations to the left and right of the point being analyzed. This requirement presents certain difficulties at the grid boundaries; concentration profiles must be extrapolated one point past the boundary to allow for the application of the central difference formula to a boundary concentration. Since inaccurate boundary concentrations can severely distort the entire concentration profile, care must be taken to choose the proper extrapolation technique. Profile distortion becomes a significant problem when concentration values are changing rapidly near the boundary or when initial concentration profiles are placed near the borders of the grid. One solution to these problems is to enlarge the grid to such a size that the concentrations at the boundary have an insignificant effect on the whole profile; this is the best solution when unlimited computer time is available. When economy requires that the grid must be as compact as possible, more care must be taken to choose a proper extrapolation technique.

Numerous extrapolation techniques were investigated during
the development of computer programs for this research; the most useful techniques are reviewed below.

BOUNDARIES ALLOWING NO TRANSFER

A common condition encountered in estuary modeling is a situation where no transfer of mass is allowed across the air-water surface or through the channel bottom. This condition is handled easily by including a set of "image" points at the border of the grid. At the air-water interface, these image concentrations are set equal to the corresponding concentrations at the surface. At the bottom of the channel, the image concentrations are set equal to the corresponding bottom-most concentrations.

For the two-dimensional explicit formulation, the upper image concentration is labelled $C_{x,1,t+1}$, and the surface concentration is labelled $C_{x,2,t+1}$. Thus, after Equation 5-31 has been applied to all the internal points, $C_{x,1,t+1}$ is set equal to the just-calculated value for $C_{x,2,t+1}$. The same procedure is followed for the bottom concentrations.

For the two-dimensional implicit formulation, a corresponding change must be made to the coefficient matrix represented by Equations 5-38 through 5-40. At the $z$-boundaries, the appropriate equations for a grid with NZ grid points would be

$$A(2) C_{x,1,t+1} + B(2) C_{x,2,t+1} + G(2) C_{x,3,t+1} = D(2) \ldots (7-1)$$
\[ A(NZ-1) C_{x,NZ-2,t+1} + B(NZ-1) C_{x,NZ-1,t+1} \]

\[ + G(NZ-1) C_{x,NZ,t+1} = D(NZ-1) \] \hspace{1cm} (7-2)

Since \( C_{x,1,t+1} \) and \( C_{x,NZ,t+1} \) are image concentrations, these equations become

\[ \left[ A(2) + B(2) \right] C_{x,2,t+1} + G(2) C_{x,3,t+1} = D(2) \] \hspace{1cm} (7-3)

\[ A(NZ-1) C_{x,NZ-2,t+1} + \left[ B(NZ-1) + G(NZ-1) \right] \]

\[ C_{x,NZ-1,t+1} = D(NZ-1) \] \hspace{1cm} (7-4)

Equations 7-3 and 7-4 become the first and last equations for the tridiagonal matrix.

These image concentrations block the dispersion of mass across a boundary. Loss of mass by advection is prevented by setting the vertical velocity term equal to zero at the surface and the bottom.

**CONSTANT SLOPE EXTRAPOLATION**

A convenient method of extrapolating profiles is to assume that a straight line can be extended from the two concentrations nearest the boundary. This approach is fairly accurate for profiles which have flattened out; it is extremely inaccurate for profiles whose slopes are changing rapidly near the boundaries. This
extrapolation technique is frequently used for the Crank-Nicolson implicit formulation because, unfortunately, only the two adjacent concentrations can be manipulated if the form of the tridiagonal matrix is to be preserved.

For the explicit method, concentrations are extended in the horizontal direction according to the following equations:

\[ C_{1,z,t+1} = 2 C_{2,z,t+1} - C_{3,z,t+1} \quad \cdots \cdots \cdots \cdots \quad (7-5) \]

\[ C_{NX,z,t+1} = 2 C_{NX-1,z,t+1} - C_{NX-2,z,t+1} \quad \cdots \cdots \cdots \cdots \quad (7-6) \]

Similar equations are applied if extrapolation in the vertical direction is needed.

The coefficient Equations 5-34 through 5-36 for the implicit method are represented as follows at the boundaries in the x-direction:

\[
[2A(2) + B(2)] C_{2,z,t+1} + [G(2) - A(2)] C_{3,z,t+1} = D(2) \quad \cdots \cdots \cdots \cdots \cdots \cdots \quad (7-7)
\]

\[
[A(NX-1) - G(NX-1)] C_{NX-2,z,t+1} + [B(NX-1) + 2G(NX-1)]
\]

\[ C_{NX-1,z,t+1} = D(NX-1) \quad \cdots \cdots \cdots \cdots \cdots \cdots \quad (7-8) \]
Extrapolations in the z-direction are represented by similar equations.

**EXPOENTIAL EXTRAPOLATIONS**

In an idealized estuary, where dispersion and velocity are constant throughout, the following two-dimensional equation can be applied to the instantaneous release of a slug of pollutant (16):

$$C_{x,z,t} = \left( \frac{m}{4\pi t^{1/2} \sqrt{EX \cdot EZ}} \right) \exp \left[ - \frac{(x-VX \cdot t)^2}{4EX \cdot t} - \frac{(z-VZ \cdot t)^2}{4EZ \cdot t} \right]. \quad (7-9)$$

where m equals amount per unit depth, and x and z are the distances from the source point. By comparing this equation for concentrations at $x - \Delta x$, $x + \Delta x$, and $x$, the following relationships are found:

$$C_{x-\Delta x,z,t} = C_{x,z,t} \cdot \exp \left[ \frac{2x\Delta x - (\Delta x)^2}{4EX \cdot t} - 2\Delta x \cdot VX \cdot t \right]. \quad (7-10)$$

$$C_{x+\Delta x,z,t} = C_{x,z,t} \cdot \exp \left[ - \frac{2x\Delta x - (\Delta x)^2}{4EX \cdot t} + 2\Delta x \cdot VX \cdot t \right]. \quad (7-11)$$

Similar expressions can be derived for concentrations at $z+\Delta z$ and $z-\Delta z$.

Equations 7-10 and 7-11 have obvious potential for extrapolating water quality profiles, especially if an idealized estuary is being modeled. For an estuary where average velocities and
dispersion coefficients are used throughout, this extrapolation provides the exact relationship between the final two concentrations on the grid at the appropriate boundaries. For an estuary with varying characteristics, these equations can give good approximations if the proper velocities and dispersion coefficients are provided.

For the explicit method, Equations 7-10 and 7-11 can be applied directly. For the implicit method, these equations can be easily represented in the coefficient matrix. For example, in the x-direction, the first line and last line of the tridiagonal matrix can be represented by Equations 7-12 and 7-13:

\[
\begin{align*}
\left[ -A(2) \cdot \exp(EXPON1) + B(2) \right] C_{2,z,t+1} \\
+ G(2) C_{3,z,t+1} &= D(2) \quad \ldots \quad \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (7-12)
\end{align*}
\]

\[
\begin{align*}
A(NX-1) C_{NX-2,z,t} + \\
+ \left[ B(NX-1) + G(NX-1) \cdot \exp(EXPON2) \right] C_{NX-1,z,t+1} \\
= D(NX-1) \quad \ldots \quad \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (7-13)
\end{align*}
\]

where EXPON1 stands for the bracketed expression in Equation 7-10 and EXPON2 stands for the bracketed expression in Equation 7-11.
Similar expressions can be inserted for extrapolation in the z-direction.

These equations can be applied only to instantaneous slug releases; formulas could be devised also for other types of profiles, such as a steady-state profile from a constant source.

CONTINUED FRACTIONS AND INVERTED DIFFERENCES

Several interpolation techniques are available in the IBM Scientific Subroutine package (48). With small modifications, these programs can be used for extrapolation. Three of these techniques were investigated during this study: the Aitkens scheme of Lagrange interpolation (ALI); the Aitkens scheme of Hermite interpolation (AHI); and the continued fractions and inverted differences scheme (ACFI). All of these techniques are explained in Hildebrand (43). These techniques were designed to interpolate equations containing low powers of the variable; thus, they are very accurate when applied to gradually changing profiles. However, serious inaccuracies can result when the slope of the profile is changing rapidly near the boundary: extrapolated concentrations can become negative or can become greater than the concentration at the adjacent point.

The continued fractions and inverted differences scheme (ACFI) was found to be the most reliable of these methods and was modified to be used as an extrapolation technique for the explicit
formulation. An advantage of this technique is that the programmer can choose the number of internal points to be included in the extrapolation formula.

**EXTRAPOLATION BY PROPORTIONS**

A final extrapolation method that was found to be useful was the simple proportion:

\[
\frac{C_{3,z,t+1}}{C_{2,z,t+1}} = \frac{C_{2,z,t+1}}{C_{1,z,t+1}} \quad \ldots \ldots \ldots \ldots \ldots \quad (7-14)
\]

For example, when this formula is rearranged to extrapolate in the \(x\)-direction, it becomes

\[
C_{1,z,t+1} = \frac{(C_{2,z,t+1})^2}{C_{3,z,t+1}} \quad \ldots \ldots \ldots \ldots \ldots \quad (7-15)
\]

This technique can be fairly accurate for extrapolating a curve whose slope is changing rapidly. The proportion technique and the continued fractions technique were both applied to the explicit method; neither was applied to the implicit method.
CHAPTER VIII

MATHEMATICAL MODELS AND SUPPORTING COMPUTER PROGRAMS

The estuary mass transport equation was programmed in finite difference form at several levels of complexity as part of this research. The simplest form deals with an idealized, one-dimensional estuary with constant values for dispersion, velocity, decay, and cross-section. The most complicated form is applicable to an estuary with characteristics that can be varied in two dimensions throughout the estuary. This progression toward increasing complexity brought about a more economical use of computer time. The numerical methods that are used in this study require considerable computational experience in order to establish accuracy and stability guidelines; this experience is more easily gained by working with less complex programs which take less computer time.

The basic computer programs that have been developed during this study are outlined below; they are written in FORTRAN IV and are discussed in detail in the sections which follow:

One-Dimensional Numerical Analysis With Constant Coefficients. - Two finite difference models were developed for evaluating mass transport in an idealized, one-dimensional estuary:

1. IDEAL - I: explicit method;
2. IDEAL - II: implicit method.

One- and Two-Dimensional Numerical Analysis With Varying
Coefficients. - Three finite difference models were developed for evaluating mass transport and oxygen transport in an estuary with varying characteristics:

1. MASSTRANS - I: mass transport by the explicit method;
2. MASSTRANS - II: mass transport by the implicit method;
3. OXTRANS - I: mass and oxygen transport by the explicit method.

Auxiliary Programs. - Several auxiliary computer programs were developed for accuracy and stability analysis:

1. STABLE - I: one-dimensional stability analysis;
2. STABLE - II: two-dimensional stability analysis;
3. EXACT - I: one-dimensional, time-changing, analytical solutions;
4. EXACT - II: two-dimensional, time-changing, analytical solutions for instantaneous releases;
5. PROFILE - I: steady-state, one-dimensional solutions for non-conservative substances; and
6. PROFILE - II: steady-state, one-dimensional solutions for BOD and dissolved oxygen profiles.

IDEAL - I

This program applies an explicit finite difference method to analyze mass transport in a one-dimensional estuary. The model is useful in evaluating transport in a flume or well-mixed estuary where velocity, dispersion, decay, and cross-sectional area can be
considered constant throughout.

The primary use of this program is to test the accuracy of the explicit finite difference method as compared with analytical, closed-form solutions for the estuary mass transport equation. The program can analyze instantaneous releases or steady-state profiles.

The basis of computation is Equation 5-7. IDEAL-I uses an exponential extrapolation or constant slope extrapolation at the boundaries. The program calculates and points out the appropriate stability criteria according to Equations 6-7 and 6-8.

**IDEAL - II**

This program has the same purpose and application as IDEAL - I except that the Crank-Nicolson implicit finite differences method is applied.

The basis of computation is Equations 5-9 through 5-19. IDEAL - II uses an exponential extrapolation or constant slope extrapolation at the boundaries. The program calculates and prints out the appropriate criteria to avoid severe oscillations; this criteria is based on Equation 6-10.

**MASSTRANS - I**

This program was developed to analyze estuaries whose characteristics do not vary significantly with width. The width of the estuary may vary throughout but the concentrations of
dissolved materials at each cross-section are considered to be unchanging in the lateral direction. All physical and hydrodynamic characteristics may vary with time and distance in the longitudinal and vertical directions. The program can be applied with equal ease to the two horizontal directions, allowing depth to vary rather than width.

Concentration profiles can be calculated by this program for continuous or instantaneous releases. The input data may include grid dimensions, distance increments, time increments, widths, loading parameters, velocities, dispersion coefficients, decay rates, sedimentation rates, and other source and sink terms. The time increment can be increased or decreased at any time during the calculation of the concentration profile. The distances between grid points can be increased at any time; a routine within the program will choose the appropriate values from the previously calculated profile and will place these values in the desired locations in the new grid system. The number of grid points may be increased or decreased at any time. Likewise, at any time during the calculation, a new set of data for physical and hydrodynamic conditions may be read into the program.

MASSTRANS-I applies an explicit finite difference method to analyze mass transport in an estuary. A user of this program must be familiar with the limitations on accuracy and stability inherent in this type of numerical procedure. A subroutine within the program prints out the proper increments to insure stability and
terminates the program if this criteria is violated by the input parameters. Another subroutine extrapolates concentrations at the boundaries; several methods can be used for these extrapolations depending on the type of profile being analyzed and the choice of the user. A subroutine is also included which prints out error messages and terminates the program if certain inconsistencies occur in the input data.

This program was developed primarily to analyze partially stratified estuaries which have been dredged out to a fairly constant depth at the centerline of the channel; these estuaries are common in the Gulf Coast region. Application of this program to partially stratified estuaries with variable depths would require moderate revisions to the program and would make the program estuary-dependent.

MASSTRANS-I also can be applied to estuaries which are well-mixed in the vertical direction. This option allows for varying width or varying depth and uses most of the routines available to the two-dimensional analysis.

The basis for one-dimensional applications is Equation 5-22. The basis for two-dimensional applications is Equation 5-31. Stability criteria is defined by Equations 6-4 through 6-8.

MASSTRANS-II

The essential difference between this program and MASSTRANS-I
is that a Crank-Nicolson implicit finite difference method is used
to solve the equations for estuary mass transport. Otherwise, the
programs operate and are applied in the same manner.

The basis for one-dimensional applications is Equations 5-20
through 5-28. The basis for two-dimensional applications is
Equations 5-32 through 5-42. Criteria to avoid severe oscillations
is generated through the use of Equations 6-9 and 6-10, but the
program is not terminated if this criteria is violated.

OXTRANS-I

This program analyzes the relationship between a primary
pollutant and a dissolved gas in a partially stratified estuary.
The principle application of OXTRANS-I is to the interaction of
biochemical oxygen demand (BOD) and dissolved oxygen (DO).

OXTRANS-I performs two basic calculations during each time
step. The first calculation is to determine the profile for the
primary pollutant, such as BOD, by the same methods available in
MASSTRANS-I. The second calculation uses information from the
first calculation to determine the values for the dissolved gas.

The input data for OXTRANS-I is the same as for MASSTRANS-I
except that additional data is required for reaeration co-
efficients, aerobic decay rates, anaerobic decay rates, and other
oxygen demands or sources. During the calculations for BOD and DO,
the program automatically switches to anaerobic decay rates
whenever the concentration of oxygen is zero. Reaeration terms can be applied to any point on the grid and, thus, the effects of mechanical aeration at any point or points can be investigated.

At the option of the user, the secondary calculations can be bypassed and the program can be operated as MASSTRANS-I.

STABLE-I and STABLE-II

These programs allow for a quick analysis of stability for the explicit finite difference method; they can process a large number of data combinations for dispersion, velocity, and decay with each run.

STABLE-I is applied to one-dimensional stability calculations through the use of Equations 6-7 and 6-8. STABLE-I generates the stability criteria for a large number of combinations of velocity and dispersion after one data set is read in.

STABLE-II is applied to two-dimensional stability calculations through the use of Equations 6-4 through 6-6. STABLE-II computes the stability criteria for each set of data that is read into the program.

EXACT-I and EXACT-II

These programs calculate concentration profiles which change with time according to known solutions to the partial differential equation for mass transfer in an estuary. These solutions can be
applied only to an idealized estuary where velocities, dispersion coefficients, decay rate, and cross-sectional area are constant throughout. EXACT-I and EXACT-II were used extensively in this study to evaluate the accuracy of the finite difference methods.

Solutions for three types of loading conditions were programmed for EXACT-I: a constant concentration defined at a source point; an instantaneous release; a continuous discharge at a point.

**Constant Concentration.** - The following solution is applicable to a condition where the concentration, \( C_0 \), is constant at and above a certain point in the estuary at time zero (51):

\[
\frac{C}{C_0} = \frac{1}{2} \exp \left( \frac{u \cdot x}{E} \right) \text{erfc} \left( \frac{x + u \cdot t}{2 \sqrt{E} \cdot t} \right) + \frac{1}{2} \text{erfc} \left( \frac{x - u \cdot t}{2 \sqrt{E} \cdot t} \right) \ldots \ldots (8-1)
\]

where \( \text{erfc} \) is the complementary error function, \( u \) is net velocity, \( E \) is the dispersion coefficient, \( x \) is distance, and \( t \) is time.

A second solution is applicable when the velocity equals zero:

\[
\frac{C}{C_0} = \text{erfc} \left( \frac{x}{2 \sqrt{E} \cdot t} \right) \ldots \ldots \ldots \ldots \ldots \ldots \ldots (8-2)
\]

**Instantaneous Release.** - The following equation represents the concentration resulting from an instantaneous release of amount \( m \) per unit cross-sectional area (16).

\[
C = \left( \frac{m}{\sqrt{4 \pi \cdot E \cdot t}} \right) \exp \left[ - \frac{(x - u \cdot t)^2}{4 \sqrt{E} \cdot t} \right] \cdot K_d \cdot t \ldots \ldots (8-3)
\]
This equation was used extensively to test the accuracy of IDEAL-I and IDEAL-II.

*Continuous Release.* - The following equation is applicable to a conservative substance released at a rate \( m \) per unit cross-sectional area per unit time into an estuary with no net velocity (13).

\[
C = \left( \frac{m \cdot t^2}{\sqrt{m \cdot E}} \right) \exp \left[ - \frac{x^2}{4E \cdot t} \right] - \left( \frac{mx}{2E} \right) \text{erfc} \left[ \frac{x}{2\sqrt{E} \cdot t} \right] \ldots \ldots (8-4)
\]

EXACT-II was used extensively to test the accuracy of MASTTRANS-I and MASTTRANS-II. The appropriate equation for two-dimensional convective-dispersion from an instantaneous release of amount \( m \) per unit depth is the following:

\[
C = \left( \frac{m}{4\pi tE_x \cdot E_y} \right) \exp \left[ - \frac{(x-V_x t)^2}{4E_x \cdot t} - \frac{(y-V_y t)^2}{4E_x \cdot E_y} - K_d \cdot t \right]. \ (8-5)
\]

where \( x \) and \( y \) represent perpendicular axes, \( V \) is velocity, and \( E \) is dispersion.

**PROFILE I and PROFILE II**

PROFILE-I calculates the steady-state profile for a non-conservative substance which is continuously released into an idealized estuary at the rate of \( W \) pounds per day. PROFILE-II calculates the steady-state profiles for BOD, oxygen deficit, and
dissolved oxygen (DO). These programs were used to test steady-state applications of MASSTRANS-I, MASSTRANS-II, and OXTRANS-I.

The following equations were programmed (67):

\[
BOD = C_0 \cdot \exp \left( \frac{u \cdot x}{2E} (1 \pm m_1) \right) \quad \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (8-6)
\]

\[
\text{DEFICIT} = \frac{K_d \cdot W}{(K_2 - K_d) Q} \left\{ \frac{1}{m_1} \exp \left[ \frac{u \cdot x}{2E} (1 \pm m_1) \right] \right. \\
\left. - \frac{1}{m_2} \exp \left[ \frac{u \cdot x}{2E} (1 \pm m_2) \right] \right\} \quad \ldots \ldots \ldots (8-7)
\]

\[
DO = C_{\text{SAT}} - \text{DEFICIT} \quad \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (8-8)
\]

in which

\[
m_1 = \sqrt{1 + \frac{4K_d \cdot E}{u^2}} \quad \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (8-9)
\]

\[
m_2 = \sqrt{1 + \frac{4K_2 \cdot E}{u^2}} \quad \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (8-10)
\]

\[
C_0 = \frac{W}{Qm_1} \quad \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (8-11)
\]

and \(Q\) equals flowrate in pounds per day, \(K_d\) equals the decay rate,
$K_2$ equals the reaeration rate, $u$ is the net velocity, and $C_{SAT}$ is the saturation value for dissolved oxygen. The positive sign refers to values of $x < 0$ and the negative sign refers to values of $x > 0$. Care must be taken to insure consistency of units. An equation similar to Equation 8-7 was applied for the special case where $K_d = K_2$. 
CHAPTER IX

ANALYSIS OF ACCURACY

Establishing the accuracy of a modeling technique is of utmost importance before that technique can be used as a basis for engineering decisions. The most straightforward and appropriate manner in which to test the accuracy of finite difference estuary models is to compare their results with exact solutions to the partial differential equations for mass transport in an estuary; these solutions are readily available for certain idealized types of estuary behavior. Such a comparison should be a requirement for any type of estuary model before the model is applied to the oftentimes inexact data of real estuaries.

The finite difference models were tested for the following cases for which exact, analytical solutions are available: one-dimensional, steady-state profiles for non-conservative substances and dissolved oxygen; one-dimensional, instantaneous release of a slug load; and two-dimensional, instantaneous release of a slug load. The accuracy characteristics of the two-dimensional models were found to be the same as the one-dimensional models and, therefore, most of the testing was done on the one-dimensional models in order to conserve computer time.

STEADY-STATE PROFILES

Steady-state profiles are obtained in the finite difference
models by setting a concentration at a point and then iterating the calculations until an unchanging profile develops. This is an inefficient way of obtaining such a profile but, as it turns out, a very accurate way.

The exact analytical solutions for steady-state profiles were obtained through the computer programs PROFILE-I and PROFILE-II which apply Equations 8-6 through 8-8.

Profiles for Non-Conservative Substances. - Excellent accuracy for steady-state profiles can be obtained by finite difference methods, even when large distance increments are used. Comparison between an exact solution and the profile calculated by the explicit method is shown in Table 9-1. The input values were as follows: dispersion coefficient = 10.0 mi$^2$/day; velocity = 1.0 mile / day; decay rate (base e) = 0.25/day. For the explicit program, IDEAL-I, the distance increment was 2.0 miles and the time increment was 0.01 days. The implicit solution for the same increments differed from the explicit solution by only one unit in the fourth decimal place. A constant slope extrapolation was applied at the downstream boundary and this appears to significantly affect the accuracy of only the last four concentrations.

In the Table 9-1, two columns are included which refer to the time at which steady-state was achieved in the finite difference model at different points. The first of these columns refers to the time it took for the solution to stabilize to four
### Table 9-1: Comparison between Exact and Finite Difference Solutions

<table>
<thead>
<tr>
<th>Distance (miles)</th>
<th>Exact (ppm)</th>
<th>Finite Difference (ppm)</th>
<th>Days till steady-state</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 places</td>
</tr>
<tr>
<td>0</td>
<td>10.0000</td>
<td>10.0000</td>
<td>23.40</td>
</tr>
<tr>
<td>2</td>
<td>7.9321</td>
<td>7.9398</td>
<td>24.30</td>
</tr>
<tr>
<td>4</td>
<td>6.2919</td>
<td>6.3040</td>
<td>25.25</td>
</tr>
<tr>
<td>6</td>
<td>4.9908</td>
<td>5.0052</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3.9588</td>
<td>3.9740</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3.1402</td>
<td>3.1552</td>
<td>27.60</td>
</tr>
<tr>
<td>12</td>
<td>2.4908</td>
<td>2.5051</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1.9758</td>
<td>1.9888</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1.5672</td>
<td>1.5789</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>1.2431</td>
<td>1.2534</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0.9861</td>
<td>0.9948</td>
<td>30.95</td>
</tr>
<tr>
<td>22</td>
<td>0.7822</td>
<td>0.7893</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>0.6204</td>
<td>0.6258</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>0.4921</td>
<td>0.4956</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>0.3904</td>
<td>0.3915</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>0.3096</td>
<td>0.3078</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>0.2456</td>
<td>0.2396</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>0.1948</td>
<td>0.1830</td>
<td>32.15</td>
</tr>
<tr>
<td>36</td>
<td>0.1548</td>
<td>0.1342</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>0.1226</td>
<td>0.0894</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>0.0972</td>
<td>0.0446</td>
<td>32.65</td>
</tr>
</tbody>
</table>
decimal points. The second of these columns refers to the time needed to reach a point in the calculations where the values remain essentially unchanged to one decimal point. Since the time increment was 0.01 days, one day in these columns would represent 100 time iterations.

The implicit method tended to take 1.75 times as much computer time as the explicit method for the same time and distance increments. When a time step of 0.1 days was used, the steady-state solutions were only slightly less accurate for both finite difference methods than for a time increment of 0.01 days.

*BOD - DO Profiles.* OXTRANS-I, the explicit finite difference model for dissolved oxygen, was investigated to determine its ability to calculate steady-state profiles for BOD and dissolved oxygen. Table 9-2 shows a comparison between the exact solution and the finite difference calculation for BOD and DO profiles based on the following input data: dispersion coefficient = 400 ft$^2$/sec. (1.24 mi$^2$/day); velocity = 0.2 ft/sec (3.27 miles/day); decay rate (base e) = 0.23/day; reaeration rate = 0.1/day; $C_{SAT} = 8.0$ ppm; and $C_O = 8.817$. OXTRANS-I used a distance increment of 0.5 miles and a time increment of 0.05 days.

OXTRANS-I provided excellent accuracy downstream from the source point at Mile 3.5. The extrapolation at the downstream boundary was based on a continued fractions and inverted differences scheme and this method provided outstanding accuracy at the final
TABLE 9-2.—BOD AND DO PROFILES

<table>
<thead>
<tr>
<th>Mile</th>
<th>BOD (ppm) Exact</th>
<th>BOD (ppm) OXTRANS-I</th>
<th>DO (ppm) Exact</th>
<th>DO (ppm) OXTRANS-I</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.001</td>
<td>0.000</td>
<td>8.00</td>
<td>8.00</td>
</tr>
<tr>
<td>0.5</td>
<td>0.003</td>
<td>0.001</td>
<td>8.00</td>
<td>8.00</td>
</tr>
<tr>
<td>1.0</td>
<td>0.010</td>
<td>0.003</td>
<td>8.00</td>
<td>8.00</td>
</tr>
<tr>
<td>1.5</td>
<td>0.039</td>
<td>0.014</td>
<td>7.99</td>
<td>8.00</td>
</tr>
<tr>
<td>2.0</td>
<td>0.152</td>
<td>0.068</td>
<td>7.98</td>
<td>7.99</td>
</tr>
<tr>
<td>2.5</td>
<td>0.588</td>
<td>0.345</td>
<td>7.93</td>
<td>7.96</td>
</tr>
<tr>
<td>3.0</td>
<td>2.276</td>
<td>1.745</td>
<td>7.81</td>
<td>7.85</td>
</tr>
<tr>
<td>3.5</td>
<td>8.817</td>
<td>8.817</td>
<td>7.56</td>
<td>7.55</td>
</tr>
<tr>
<td>4.0</td>
<td>8.520</td>
<td>8.520</td>
<td>7.27</td>
<td>7.27</td>
</tr>
<tr>
<td>4.5</td>
<td>8.233</td>
<td>8.233</td>
<td>7.00</td>
<td>7.00</td>
</tr>
<tr>
<td>5.0</td>
<td>7.956</td>
<td>7.956</td>
<td>6.75</td>
<td>6.74</td>
</tr>
<tr>
<td>5.5</td>
<td>7.688</td>
<td>7.688</td>
<td>6.50</td>
<td>6.50</td>
</tr>
<tr>
<td>6.0</td>
<td>7.429</td>
<td>7.430</td>
<td>6.27</td>
<td>6.26</td>
</tr>
<tr>
<td>6.5</td>
<td>7.179</td>
<td>7.179</td>
<td>6.05</td>
<td>6.04</td>
</tr>
<tr>
<td>7.0</td>
<td>6.937</td>
<td>6.938</td>
<td>5.84</td>
<td>5.84</td>
</tr>
<tr>
<td>7.5</td>
<td>6.704</td>
<td>6.704</td>
<td>5.64</td>
<td>5.64</td>
</tr>
<tr>
<td>8.0</td>
<td>6.478</td>
<td>6.478</td>
<td>5.46</td>
<td>5.45</td>
</tr>
<tr>
<td>8.5</td>
<td>6.260</td>
<td>6.260</td>
<td>5.28</td>
<td>5.28</td>
</tr>
<tr>
<td>9.0</td>
<td>6.049</td>
<td>6.050</td>
<td>5.12</td>
<td>5.11</td>
</tr>
<tr>
<td>9.5</td>
<td>5.845</td>
<td>5.846</td>
<td>4.96</td>
<td>4.96</td>
</tr>
<tr>
<td>10.0</td>
<td>5.649</td>
<td>5.649</td>
<td>4.81</td>
<td>4.81</td>
</tr>
<tr>
<td>10.5</td>
<td>5.458</td>
<td>5.459</td>
<td>4.67</td>
<td>4.67</td>
</tr>
<tr>
<td>11.0</td>
<td>5.275</td>
<td>5.275</td>
<td>4.54</td>
<td>4.54</td>
</tr>
<tr>
<td>11.5</td>
<td>5.097</td>
<td>5.098</td>
<td>4.42</td>
<td>4.42</td>
</tr>
<tr>
<td>12.0</td>
<td>4.925</td>
<td>4.926</td>
<td>4.31</td>
<td>4.30</td>
</tr>
<tr>
<td>12.5</td>
<td>4.759</td>
<td>4.760</td>
<td>4.20</td>
<td>4.19</td>
</tr>
<tr>
<td>13.0</td>
<td>4.599</td>
<td>4.600</td>
<td>4.10</td>
<td>4.09</td>
</tr>
<tr>
<td>13.5</td>
<td>4.444</td>
<td>4.445</td>
<td>4.00</td>
<td>4.00</td>
</tr>
<tr>
<td>14.0</td>
<td>4.295</td>
<td>4.295</td>
<td>3.92</td>
<td>3.91</td>
</tr>
<tr>
<td>14.5</td>
<td>4.150</td>
<td>4.151</td>
<td>3.84</td>
<td>3.83</td>
</tr>
<tr>
<td>15.0</td>
<td>4.010</td>
<td>4.011</td>
<td>3.76</td>
<td>3.76</td>
</tr>
<tr>
<td>15.5</td>
<td>3.875</td>
<td>3.876</td>
<td>3.69</td>
<td>3.69</td>
</tr>
<tr>
<td>16.0</td>
<td>3.745</td>
<td>3.746</td>
<td>3.63</td>
<td>3.63</td>
</tr>
<tr>
<td>16.5</td>
<td>3.619</td>
<td>3.619</td>
<td>3.57</td>
<td>3.57</td>
</tr>
<tr>
<td>17.0</td>
<td>3.497</td>
<td>3.498</td>
<td>3.52</td>
<td>3.51</td>
</tr>
<tr>
<td>17.5</td>
<td>3.379</td>
<td>3.380</td>
<td>3.47</td>
<td>3.47</td>
</tr>
<tr>
<td>18.0</td>
<td>3.265</td>
<td>3.266</td>
<td>3.43</td>
<td>3.42</td>
</tr>
<tr>
<td>18.5</td>
<td>3.156</td>
<td>3.156</td>
<td>3.39</td>
<td>3.38</td>
</tr>
</tbody>
</table>
point, Mile 18.5. The steady-state solution at Mile 4.0 stabilized after 60 time iterations and the solution at Mile 18.0 stabilized after 180 time iterations.

The profile upstream of the source point is fairly inaccurate, but this problem did not adversely affect the downstream profile. The inaccuracy in the upstream profile was caused by attempting to use a large distance increment in an area where the slope of the concentration profile was changing rapidly with distance. This rapid change in the slope of the steady-state profile is produced when the velocity is in the opposite direction from dispersion. This inaccuracy can be corrected by using a smaller distance increment in areas where concentration profiles have steep slopes. Tables 9-3 and 9-4 demonstrate the change in accuracy when the distance increment is changed from 0.5 miles to 0.1 miles. The time increment was changed to 0.0025 days to insure stability. Even better accuracy would be attained if the distance increment were decreased even further.

These results demonstrate that the choice of a distance increment has a limiting effect on the accuracy which can be obtained. For steady-state profiles, several distance increments should be tried in order to insure that an accurate profile has been calculated.

**ONE-DIMENSIONAL INSTANTANEOUS RELEASE OF A SLUG LOAD**

An important application of the finite difference models that
TABLE 9-3.-ACCURACY FOR BOD PROFILES FOR SEVERAL DISTANCE INCREMENTS

<table>
<thead>
<tr>
<th>Mile</th>
<th>Exact</th>
<th>OXTRANS-I</th>
<th>OXTRANS-I</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\Delta x = 0.5$</td>
<td>$\Delta t = 0.05$</td>
</tr>
<tr>
<td>0.0</td>
<td>0.001</td>
<td>0.000</td>
<td>-</td>
</tr>
<tr>
<td>0.5</td>
<td>0.003</td>
<td>0.001</td>
<td>-</td>
</tr>
<tr>
<td>1.0</td>
<td>0.010</td>
<td>0.003</td>
<td>-</td>
</tr>
<tr>
<td>1.5</td>
<td>0.039</td>
<td>0.014</td>
<td>0.029</td>
</tr>
<tr>
<td>2.0</td>
<td>0.152</td>
<td>0.068</td>
<td>0.140</td>
</tr>
<tr>
<td>2.5</td>
<td>0.588</td>
<td>0.345</td>
<td>0.571</td>
</tr>
<tr>
<td>3.0</td>
<td>2.276</td>
<td>1.745</td>
<td>2.252</td>
</tr>
<tr>
<td>3.5</td>
<td>8.817</td>
<td>8.817</td>
<td>8.817</td>
</tr>
</tbody>
</table>

TABLE 9-4.-ACCURACY FOR DISSOLVED OXYGEN PROFILES FOR SEVERAL DISTANCE INCREMENTS

<table>
<thead>
<tr>
<th>Mile</th>
<th>Exact (ppm)</th>
<th>OXTRANS-I</th>
<th>OXTRANS-I</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\Delta x = 0.5$</td>
<td>$\Delta t = 0.05$</td>
</tr>
<tr>
<td>0.0</td>
<td>8.00</td>
<td>8.00</td>
<td>-</td>
</tr>
<tr>
<td>0.5</td>
<td>8.00</td>
<td>8.00</td>
<td>-</td>
</tr>
<tr>
<td>1.0</td>
<td>8.00</td>
<td>8.00</td>
<td>-</td>
</tr>
<tr>
<td>1.5</td>
<td>7.99</td>
<td>8.00</td>
<td>8.00</td>
</tr>
<tr>
<td>2.0</td>
<td>7.98</td>
<td>7.99</td>
<td>7.98</td>
</tr>
<tr>
<td>2.5</td>
<td>7.93</td>
<td>7.96</td>
<td>7.94</td>
</tr>
<tr>
<td>3.0</td>
<td>7.81</td>
<td>7.85</td>
<td>7.83</td>
</tr>
<tr>
<td>3.5</td>
<td>7.56</td>
<td>7.55</td>
<td>7.55</td>
</tr>
</tbody>
</table>
have been developed in this study will be to calculate the concentration profiles resulting from slug loads released into an estuary. Such loading might result from spills of polluting materials or dye releases. Therefore, a considerable amount of time has been spent during this study to evaluate the ability of the models to predict accurately the concentration profiles produced by slug loads.

Finite difference methods are generally regarded as being accurate for diffusion predictions, as in heat transfer problems. However, when velocity is also included as a parameter, a phenomenon known as "numerical dispersion" results. This phenomenon caused significant difficulties in the finite difference methods used in the San Francisco Bay-Delta model (72). Numerical dispersion is related to the fact that the distance increment used in a finite difference model is rarely a multiple of the velocity at which the material is moving.

In this study, numerical dispersion was found to produce distortions and severe inaccuracies if improper choices were made for distance increments and time increments. However, excellent accuracy was achieved and the effects of numerical dispersion were minimized when a small enough distance increment was chosen; for the dispersion coefficients and velocities used in this study, good accuracy was obtained for horizontal distance increments between 0.1 miles and 0.25 miles.
When comparing finite difference solutions with exact analytical solutions, initial concentration values for the model must be chosen to represent accurately the mass of the slug load. These initial values can be chosen in two ways: either by obtaining an initial distribution of values at several points from the analytical solution or by calculating an equivalent "slug" value for a single starting value in the model. The analytical solution, as represented by Equation 8-3, calculates the concentration profile for a mass \( m \) per unit cross-sectional area. For a single initial concentration, the finite difference model "sees" this mass as a triangle, as shown in Figure 9-1, below:

\[
\text{SLUG}
\]

\[
i - 1 \quad i \quad i + 1
\]

\[\Delta x\]

**FIGURE 9-1.-SLUG LOAD "SEEN" BY FINITE DIFFERENCE MODEL**

The value for SLUG is equal to \( m/\Delta x \), where the \( m \) used for accuracy investigations was \( 10^7 \). Thus, for a distance increment of 0.25 miles (1,320 feet), \( \text{SLUG} = 10^7/1320 = 7575.8 \). For a distance increment of 0.10 miles (528 feet), \( \text{SLUG} = 10^7/528 = 18939.39 \). The \( \Delta x \) must be expressed in feet.

*Analysis of Dispersion Without Velocity.* - When using a single
value of a finite difference model, distortions develop in the early calculations of the profile. When a sufficiently small distance increment is chosen along with a proper time increment, these distortions are rapidly damped out. Tables 9-5 and 9-6 demonstrate the ability of the models to converge on the correct solution. For this example, the dispersion coefficient $= 1 \text{ mi}^2/\text{day} (322.7 \text{ ft}^2/\text{sec})$, the time increment $= 0.01$ days, the distance increment $= 0.25$ miles, and SLUG $= 7575.8$ at time $= 0.0$ days. The velocity and decay rate were set equal to zero. The profiles were essentially symmetrical; therefore only half of the profiles are shown.

Effect of Velocity on the Concentration Profile from a Slug Load. - Table 9-7 shows the effects of a velocity of 5 miles/day ($0.306 \text{ ft/sec}$) on the distribution described in the previous example. Velocity distorts the concentration profiles; however, this distortion is minor when proper time and distance increments are chosen, except at the leading and trailing edges. A comparison between the exact profile after one day elapsed time and the profile predicted by the implicit method is shown in Figure 9-2, for a distance increment of 0.25 miles. Although the explicit method is more accurate for dispersion alone, the implicit method appears to be more accurate when velocity is included.

Error Analysis for a Slug Load. - The finite difference models
TABLE 9-5.—DISTRIBUTION FROM A SLUG LOAD AT TIME = 0.2 DAYS

<table>
<thead>
<tr>
<th>Mile</th>
<th>Exact</th>
<th>Explicit</th>
<th>Implicit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>1194.7</td>
<td>1196.1</td>
<td>1219.9</td>
</tr>
<tr>
<td>0.25</td>
<td>1104.9</td>
<td>1105.7</td>
<td>1120.4</td>
</tr>
<tr>
<td>0.50</td>
<td>874.0</td>
<td>873.8</td>
<td>870.6</td>
</tr>
<tr>
<td>0.75</td>
<td>591.4</td>
<td>590.5</td>
<td>577.0</td>
</tr>
<tr>
<td>1.00</td>
<td>342.3</td>
<td>341.6</td>
<td>329.8</td>
</tr>
<tr>
<td>1.25</td>
<td>169.4</td>
<td>169.3</td>
<td>164.5</td>
</tr>
<tr>
<td>1.50</td>
<td>71.75</td>
<td>71.88</td>
<td>72.46</td>
</tr>
<tr>
<td>1.75</td>
<td>25.98</td>
<td>26.15</td>
<td>28.49</td>
</tr>
<tr>
<td>2.00</td>
<td>8.05</td>
<td>8.15</td>
<td>10.10</td>
</tr>
<tr>
<td>2.25</td>
<td>2.13</td>
<td>2.17</td>
<td>3.26</td>
</tr>
<tr>
<td>2.50</td>
<td>0.48</td>
<td>0.49</td>
<td>0.96</td>
</tr>
<tr>
<td>2.75</td>
<td>0.09</td>
<td>0.09</td>
<td>0.26</td>
</tr>
<tr>
<td>3.00</td>
<td>0.02</td>
<td>0.02</td>
<td>0.04</td>
</tr>
</tbody>
</table>

TABLE 9-6.—DISTRIBUTION FROM A SLUG LOAD AT TIME = 1.0 DAYS

<table>
<thead>
<tr>
<th>Mile</th>
<th>Exact</th>
<th>Explicit</th>
<th>Implicit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>534.3</td>
<td>534.3</td>
<td>536.2</td>
</tr>
<tr>
<td>0.25</td>
<td>526.0</td>
<td>526.1</td>
<td>527.8</td>
</tr>
<tr>
<td>0.50</td>
<td>501.9</td>
<td>502.0</td>
<td>503.3</td>
</tr>
<tr>
<td>0.75</td>
<td>464.2</td>
<td>464.2</td>
<td>464.9</td>
</tr>
<tr>
<td>1.00</td>
<td>416.1</td>
<td>416.1</td>
<td>416.1</td>
</tr>
<tr>
<td>1.25</td>
<td>361.5</td>
<td>361.5</td>
<td>360.9</td>
</tr>
<tr>
<td>1.50</td>
<td>304.4</td>
<td>304.4</td>
<td>303.3</td>
</tr>
<tr>
<td>1.75</td>
<td>248.5</td>
<td>248.4</td>
<td>247.0</td>
</tr>
<tr>
<td>2.00</td>
<td>196.6</td>
<td>196.5</td>
<td>195.1</td>
</tr>
<tr>
<td>2.25</td>
<td>150.7</td>
<td>150.6</td>
<td>149.3</td>
</tr>
<tr>
<td>2.50</td>
<td>112.0</td>
<td>112.0</td>
<td>110.8</td>
</tr>
<tr>
<td>2.75</td>
<td>80.66</td>
<td>80.63</td>
<td>79.77</td>
</tr>
<tr>
<td>3.00</td>
<td>56.31</td>
<td>56.29</td>
<td>55.69</td>
</tr>
<tr>
<td>Mile</td>
<td>Exact</td>
<td>IDEAL-II (Implicit)</td>
<td>IDEAL-I (Explicit)</td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
<td>---------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td></td>
<td>ΔT = .01 Days</td>
<td>ΔT = .01</td>
<td>ΔT = .0025</td>
</tr>
<tr>
<td>-5.00 (Input point)</td>
<td>1.03</td>
<td>0.54</td>
<td>0.40</td>
</tr>
<tr>
<td>-4.75</td>
<td>1.90</td>
<td>1.11</td>
<td>0.82</td>
</tr>
<tr>
<td>-4.50</td>
<td>3.38</td>
<td>2.16</td>
<td>1.62</td>
</tr>
<tr>
<td>-4.25</td>
<td>5.84</td>
<td>4.07</td>
<td>3.07</td>
</tr>
<tr>
<td>-4.00</td>
<td>9.79</td>
<td>7.34</td>
<td>5.61</td>
</tr>
<tr>
<td>-3.75</td>
<td>15.88</td>
<td>12.73</td>
<td>9.87</td>
</tr>
<tr>
<td>-3.50</td>
<td>25.00</td>
<td>21.21</td>
<td>16.74</td>
</tr>
<tr>
<td>-3.25</td>
<td>38.10</td>
<td>33.97</td>
<td>27.35</td>
</tr>
<tr>
<td>-3.00</td>
<td>56.31</td>
<td>52.34</td>
<td>43.05</td>
</tr>
<tr>
<td>-2.75</td>
<td>80.66</td>
<td>77.55</td>
<td>65.30</td>
</tr>
<tr>
<td>-2.50</td>
<td>112.0</td>
<td>110.6</td>
<td>95.46</td>
</tr>
<tr>
<td>-2.25</td>
<td>150.7</td>
<td>151.9</td>
<td>134.5</td>
</tr>
<tr>
<td>-2.00</td>
<td>196.6</td>
<td>201.0</td>
<td>182.7</td>
</tr>
<tr>
<td>-1.75</td>
<td>248.5</td>
<td>256.1</td>
<td>239.2</td>
</tr>
<tr>
<td>-1.50</td>
<td>304.4</td>
<td>315.7</td>
<td>301.9</td>
</tr>
<tr>
<td>-1.25</td>
<td>361.5</td>
<td>375.1</td>
<td>364.5</td>
</tr>
<tr>
<td>-1.00</td>
<td>416.1</td>
<td>430.5</td>
<td>431.4</td>
</tr>
<tr>
<td>-0.75</td>
<td>464.2</td>
<td>477.6</td>
<td>488.4</td>
</tr>
<tr>
<td>-0.50</td>
<td>501.9</td>
<td>512.4</td>
<td>533.4</td>
</tr>
<tr>
<td>-0.25</td>
<td>526.0</td>
<td>532.1</td>
<td>561.9</td>
</tr>
<tr>
<td>-0.00 (Peak)</td>
<td>534.3</td>
<td>535.1</td>
<td>571.1</td>
</tr>
<tr>
<td>0.25</td>
<td>526.0</td>
<td>521.6</td>
<td>560.1</td>
</tr>
<tr>
<td>0.50</td>
<td>501.9</td>
<td>492.9</td>
<td>530.1</td>
</tr>
<tr>
<td>0.75</td>
<td>464.2</td>
<td>452.1</td>
<td>484.3</td>
</tr>
<tr>
<td>1.00</td>
<td>416.1</td>
<td>402.5</td>
<td>427.0</td>
</tr>
<tr>
<td>1.25</td>
<td>361.5</td>
<td>348.2</td>
<td>363.5</td>
</tr>
<tr>
<td>1.50</td>
<td>304.4</td>
<td>292.9</td>
<td>298.7</td>
</tr>
<tr>
<td>1.75</td>
<td>248.5</td>
<td>239.6</td>
<td>237.1</td>
</tr>
<tr>
<td>2.00</td>
<td>196.6</td>
<td>190.8</td>
<td>181.7</td>
</tr>
<tr>
<td>2.25</td>
<td>150.7</td>
<td>147.9</td>
<td>134.5</td>
</tr>
<tr>
<td>2.50</td>
<td>112.0</td>
<td>111.7</td>
<td>96.15</td>
</tr>
<tr>
<td>2.75</td>
<td>80.66</td>
<td>81.95</td>
<td>66.71</td>
</tr>
<tr>
<td>3.00</td>
<td>56.31</td>
<td>57.21</td>
<td>46.57</td>
</tr>
</tbody>
</table>
FIGURE 9-2. - PROFILE FROM A SLUG LOAD
developed in this study should be useful in predicting the concentration profiles resulting from slug loads; therefore, a more extensive analysis of stability and accuracy characteristics of the finite difference methods was believed to be worthwhile.

The explicit and implicit models were applied to a slug loading using several different time increments. The profiles were calculated until 0.6 days had elapsed and then the results were compared. The following parameters were the same for all runs: distance increment = 0.1 miles; dispersion coefficient = 1.0 mi²/ day (322.7 ft²/sec); velocity = 5.0 miles/day (.306 ft/sec); and decay rate = 0.0. Two initial starting conditions were applied. One starting condition placed a single concentration value of 18939.39 ppm at the source point at time = 0.0 days. The second starting condition applied a known concentration profile calculated by the analytical solution for an elapsed time of 0.1 days; this known initial distribution included 28 points and is shown in Table 9-8.

The accuracy of the computed profiles were compared for a series of different time increments. The maximum time increment for stability of the explicit method was calculated to be 0.005 days.

Table 9-9 shows the profiles for the Crank-Nicolson implicit method with a time increment of 0.002 days and the profile for the explicit method with a time increment of
<table>
<thead>
<tr>
<th>Mile</th>
<th>Concentration (ppm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3.0</td>
<td>904.3</td>
</tr>
<tr>
<td>-2.9</td>
<td>1132.5</td>
</tr>
<tr>
<td>-2.8</td>
<td>1349.1</td>
</tr>
<tr>
<td>-2.7</td>
<td>1528.7</td>
</tr>
<tr>
<td>-2.6</td>
<td>1647.8</td>
</tr>
<tr>
<td>-2.5</td>
<td>1689.5</td>
</tr>
<tr>
<td>-2.4</td>
<td>1647.8</td>
</tr>
<tr>
<td>-2.3</td>
<td>1528.7</td>
</tr>
<tr>
<td>-2.2</td>
<td>1349.1</td>
</tr>
<tr>
<td>-2.1</td>
<td>1132.5</td>
</tr>
<tr>
<td>-2.0</td>
<td>904.3</td>
</tr>
<tr>
<td>-1.9</td>
<td>686.9</td>
</tr>
<tr>
<td>-1.8</td>
<td>496.3</td>
</tr>
<tr>
<td>-1.7</td>
<td>341.1</td>
</tr>
<tr>
<td>-1.6</td>
<td>223.0</td>
</tr>
<tr>
<td>-1.5</td>
<td>138.7</td>
</tr>
<tr>
<td>-1.4</td>
<td>82.04</td>
</tr>
<tr>
<td>-1.3</td>
<td>46.17</td>
</tr>
<tr>
<td>-1.2</td>
<td>24.71</td>
</tr>
<tr>
<td>-1.1</td>
<td>12.58</td>
</tr>
<tr>
<td>-1.0</td>
<td>6.09</td>
</tr>
<tr>
<td>-0.9</td>
<td>2.81</td>
</tr>
<tr>
<td>-0.8</td>
<td>1.23</td>
</tr>
<tr>
<td>-0.7</td>
<td>0.51</td>
</tr>
<tr>
<td>-0.6</td>
<td>0.20</td>
</tr>
<tr>
<td>-0.5</td>
<td>0.08</td>
</tr>
<tr>
<td>-0.4</td>
<td>0.03</td>
</tr>
<tr>
<td>-0.3</td>
<td>0.01</td>
</tr>
</tbody>
</table>
### Table 9-9: Profiles Calculated by Finite Difference Methods for a Slug Load; Elapsed Time = 0.6 Days

<table>
<thead>
<tr>
<th>Distance from peak (miles)</th>
<th>Exact</th>
<th>Implicit $\Delta T = .002$ Days</th>
<th>$%$ Error</th>
<th>Explicit $\Delta T = .0005$ Days</th>
<th>$%$ Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2.5</td>
<td>51.0</td>
<td>50.1</td>
<td>-1.7</td>
<td>49.7</td>
<td>-2.5</td>
</tr>
<tr>
<td>-2.4</td>
<td>62.6</td>
<td>61.7</td>
<td></td>
<td>61.2</td>
<td></td>
</tr>
<tr>
<td>-2.3</td>
<td>76.1</td>
<td>75.2</td>
<td></td>
<td>74.7</td>
<td></td>
</tr>
<tr>
<td>-2.2</td>
<td>91.8</td>
<td>91.0</td>
<td></td>
<td>90.4</td>
<td></td>
</tr>
<tr>
<td>-2.1</td>
<td>109.8</td>
<td>109.2</td>
<td></td>
<td>108.5</td>
<td></td>
</tr>
<tr>
<td>-2.0</td>
<td>130.3</td>
<td>129.8</td>
<td>-0.37</td>
<td>129.0</td>
<td>-0.97</td>
</tr>
<tr>
<td>-1.9</td>
<td>153.3</td>
<td>153.0</td>
<td></td>
<td>152.2</td>
<td></td>
</tr>
<tr>
<td>-1.8</td>
<td>178.8</td>
<td>178.9</td>
<td>~0.0</td>
<td>177.9</td>
<td></td>
</tr>
<tr>
<td>-1.7</td>
<td>206.9</td>
<td>207.3</td>
<td></td>
<td>206.3</td>
<td>~0.0</td>
</tr>
<tr>
<td>-1.6</td>
<td>237.4</td>
<td>238.2</td>
<td></td>
<td>237.1</td>
<td></td>
</tr>
<tr>
<td>-1.5</td>
<td>270.1</td>
<td>271.3</td>
<td></td>
<td>270.2</td>
<td></td>
</tr>
<tr>
<td>-1.4</td>
<td>304.8</td>
<td>306.4</td>
<td></td>
<td>305.4</td>
<td></td>
</tr>
<tr>
<td>-1.3</td>
<td>341.1</td>
<td>343.2</td>
<td></td>
<td>342.1</td>
<td></td>
</tr>
<tr>
<td>-1.2</td>
<td>378.5</td>
<td>381.0</td>
<td></td>
<td>380.1</td>
<td></td>
</tr>
<tr>
<td>-1.1</td>
<td>416.6</td>
<td>419.5</td>
<td></td>
<td>418.7</td>
<td></td>
</tr>
<tr>
<td>-1.0</td>
<td>454.7</td>
<td>457.9</td>
<td>+0.70</td>
<td>457.2</td>
<td>+0.55</td>
</tr>
<tr>
<td>-0.9</td>
<td>492.2</td>
<td>495.6</td>
<td></td>
<td>495.1</td>
<td></td>
</tr>
<tr>
<td>-0.8</td>
<td>528.3</td>
<td>531.9</td>
<td></td>
<td>531.6</td>
<td></td>
</tr>
<tr>
<td>-0.7</td>
<td>562.4</td>
<td>566.0</td>
<td></td>
<td>566.0</td>
<td></td>
</tr>
<tr>
<td>-0.6</td>
<td>593.7</td>
<td>597.2</td>
<td></td>
<td>597.5</td>
<td></td>
</tr>
<tr>
<td>-0.5</td>
<td>621.5</td>
<td>624.8</td>
<td></td>
<td>625.4</td>
<td></td>
</tr>
<tr>
<td>-0.4</td>
<td>645.3</td>
<td>648.2</td>
<td></td>
<td>649.1</td>
<td></td>
</tr>
<tr>
<td>-0.3</td>
<td>664.4</td>
<td>666.8</td>
<td></td>
<td>668.0</td>
<td></td>
</tr>
<tr>
<td>-0.2</td>
<td>678.3</td>
<td>680.2</td>
<td>~0.0</td>
<td>681.7</td>
<td></td>
</tr>
<tr>
<td>-0.1</td>
<td>686.9</td>
<td>688.1</td>
<td></td>
<td>689.9</td>
<td></td>
</tr>
<tr>
<td>+0.0</td>
<td>689.7</td>
<td>690.2</td>
<td>+0.07</td>
<td>692.3</td>
<td>+0.37</td>
</tr>
<tr>
<td>+0.1</td>
<td>686.9</td>
<td>686.6</td>
<td>~0.0</td>
<td>688.8</td>
<td></td>
</tr>
<tr>
<td>+0.2</td>
<td>678.3</td>
<td>677.4</td>
<td></td>
<td>679.7</td>
<td></td>
</tr>
<tr>
<td>+0.3</td>
<td>664.4</td>
<td>662.7</td>
<td></td>
<td>665.1</td>
<td></td>
</tr>
<tr>
<td>+0.4</td>
<td>645.3</td>
<td>642.9</td>
<td></td>
<td>645.4</td>
<td>~0.0</td>
</tr>
<tr>
<td>+0.5</td>
<td>621.5</td>
<td>618.6</td>
<td></td>
<td>621.0</td>
<td></td>
</tr>
<tr>
<td>+0.6</td>
<td>593.7</td>
<td>590.4</td>
<td></td>
<td>592.7</td>
<td></td>
</tr>
<tr>
<td>+0.7</td>
<td>562.4</td>
<td>558.7</td>
<td></td>
<td>560.9</td>
<td></td>
</tr>
<tr>
<td>+0.8</td>
<td>528.3</td>
<td>524.5</td>
<td></td>
<td>526.5</td>
<td></td>
</tr>
<tr>
<td>+0.9</td>
<td>492.2</td>
<td>488.4</td>
<td></td>
<td>490.3</td>
<td></td>
</tr>
<tr>
<td>+1.0</td>
<td>454.7</td>
<td>451.2</td>
<td>~0.77</td>
<td>453.0</td>
<td>~0.39</td>
</tr>
</tbody>
</table>
0.0005 days. The percentage of error is shown at certain points. Similar analyses of profiles for other time increments are summarized in Tables 9-10, 9-11, 9-12, and 9-13. In these four tables, execution time is included to demonstrate the accuracy which can be obtained for a certain expenditure of computer time.

The percentage of error for various time increments in Table 9-10 are shown graphically in Figure 9-3. Error in terms of concentration is shown for the same data in Figure 9-4; the concentrations should be compared with the peak concentration of 689.7 ppm for 0.6 days lapsed time.

Discussion of Accuracy for Slug Loads. - The analyses in this section demonstrate that the Crank-Nicolson implicit method generally provides better accuracy and more freedom from oscillations than the explicit method. In addition, for one-dimensional applications the implicit method requires less computer time than the explicit method for an accurate solution. However, excellent accuracy also can be obtained by the explicit method when the time increment is chosen properly. The choice of a distance increment is also important in limiting the degree of accuracy which can be obtained by either method.

The accuracy of the explicit method improves noticeably for smaller and smaller time increments. This accuracy will have a lower limit because round-off errors become more significant as
<table>
<thead>
<tr>
<th>Time increment in days $\Delta T$</th>
<th>Execution time for 0.6 days (seconds)</th>
<th>% Error at peak</th>
<th>No error distance from peak (miles)</th>
<th>% Error at certain distances from peak</th>
<th>Oscillations and stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0005</td>
<td>33.72</td>
<td>+0.37</td>
<td>+0.4</td>
<td>-1.6</td>
<td>-0.39 +0.55 -0.97 -2.5 Stable</td>
</tr>
<tr>
<td>0.001</td>
<td>17.90</td>
<td>+0.67</td>
<td>+1.0</td>
<td>-1.3</td>
<td>0.00 +0.40 -1.6 -3.2 Stable</td>
</tr>
<tr>
<td>0.00167</td>
<td>10.62</td>
<td>+1.06</td>
<td>-</td>
<td>-1.1</td>
<td>+0.52 +0.19 -2.4 -4.2 Stable</td>
</tr>
<tr>
<td>0.002</td>
<td>9.69</td>
<td>+1.25</td>
<td>-</td>
<td>-1.1</td>
<td>+0.78 +0.07 -2.8 -4.7 Stable</td>
</tr>
<tr>
<td>0.003</td>
<td>6.01</td>
<td>+1.85</td>
<td>-</td>
<td>-0.9</td>
<td>+1.60 -0.26 -4.0 -6.2 Stable</td>
</tr>
<tr>
<td>0.004</td>
<td>4.55</td>
<td>+2.45</td>
<td>-</td>
<td>-0.8</td>
<td>+2.40 -0.62 -5.2 -7.6 Early oscillations quickly damped out</td>
</tr>
<tr>
<td>0.0045</td>
<td>4.20</td>
<td>+2.80</td>
<td>-</td>
<td>-0.7</td>
<td>+1.0 -1.9 -7.3 -9.5 Early oscillations damped out</td>
</tr>
<tr>
<td>0.005</td>
<td>3.84</td>
<td>+205.70</td>
<td>-</td>
<td>-</td>
<td>Unstable oscillations</td>
</tr>
<tr>
<td>Time increment in days</td>
<td>Execution time for 0.5 days (seconds)</td>
<td>% Error at peak</td>
<td>No error distance from peak (miles)</td>
<td>% Error at certain distances from peak (miles)</td>
<td>Oscillations and stability after .2 days</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------</td>
<td>----------------</td>
<td>-----------------------------------</td>
<td>-----------------------------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>0.0005</td>
<td>29.37</td>
<td>+0.42</td>
<td>+0.5 -2.3</td>
<td>-0.41 +0.82 +0.28 -1.8</td>
<td>Stable</td>
</tr>
<tr>
<td>0.002</td>
<td>7.63</td>
<td>+1.16</td>
<td>+1.2 -1.3</td>
<td>+0.32 +0.45 -1.20 -2.0</td>
<td>Stable</td>
</tr>
<tr>
<td>0.004</td>
<td>3.59</td>
<td>+2.16</td>
<td>+1.3 -1.0</td>
<td>+1.30 -0.08 -3.10 -4.3</td>
<td>Stable</td>
</tr>
<tr>
<td>0.005</td>
<td>3.37</td>
<td>+2.67</td>
<td>+1.3 -0.9</td>
<td>+1.80 -0.36 -4.10 -5.5</td>
<td>Stable</td>
</tr>
<tr>
<td>0.006</td>
<td>3.15</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Unstable oscillations</td>
</tr>
</tbody>
</table>
### TABLE 9-12 - ACCURACY ANALYSIS: CRANK-NICOLSON IMPLICIT METHOD
PROFILE AFTER 0.6 DAYS FOR AN INITIAL SLUG LOAD OF 18939.39

<table>
<thead>
<tr>
<th>Time increment in days $\Delta T$</th>
<th>Execution time for 0.6 days (seconds)</th>
<th>% Error at peak</th>
<th>No error distance from peak (miles)</th>
<th>% Error at certain distance from peak +1.0 miles -1.0 miles -2.0 miles -2.5 miles</th>
<th>Oscillations and stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>.002</td>
<td>16.83</td>
<td>+ .07</td>
<td>+0.1 -1.8</td>
<td>-.77 +0.70 -0.37 -1.7</td>
<td>Stable</td>
</tr>
<tr>
<td>.005</td>
<td>6.81</td>
<td>+ .13</td>
<td>+0.1 -1.8</td>
<td>-.74 +0.75 -0.35 -1.7</td>
<td>Stable</td>
</tr>
<tr>
<td>.01</td>
<td>3.80</td>
<td>+ .17</td>
<td>+0.1 -1.8</td>
<td>-.79 +0.77 -0.46 -1.9</td>
<td>Stable</td>
</tr>
<tr>
<td>.015</td>
<td>2.67</td>
<td>+ 2.00</td>
<td>- -2.4</td>
<td>+.77 +2.70 +1.30 -1.7</td>
<td>Early oscillations</td>
</tr>
<tr>
<td>.02</td>
<td>1.92</td>
<td>+24.00</td>
<td>- -</td>
<td>- - - -</td>
<td>severely distort profile</td>
</tr>
<tr>
<td>Time increment in days $\Delta T$</td>
<td>Execution time for 0.5 days (seconds)</td>
<td>% Error at peak</td>
<td>No error distance from peak (miles)</td>
<td>% Error at certain distance from peak miles</td>
<td>Oscillations and stability</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>--------------------------------------</td>
<td>----------------</td>
<td>-------------------------------------</td>
<td>---------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>.001</td>
<td>30.93</td>
<td>+ 0.22</td>
<td>+.2</td>
<td>&gt;-3.0</td>
<td>- 0.61 +0.98 +0.81 +0.45</td>
</tr>
<tr>
<td>.002</td>
<td>14.54</td>
<td>+ 0.17</td>
<td>+.2</td>
<td>&gt;-3.0</td>
<td>- 0.66 +0.94 +0.76 +0.39</td>
</tr>
<tr>
<td>.01</td>
<td>5.66</td>
<td>+ 0.26</td>
<td>+.2</td>
<td>-3.0</td>
<td>- 0.70 +0.99 +0.70 +0.29</td>
</tr>
<tr>
<td>.02</td>
<td>3.40</td>
<td>+ 0.45</td>
<td>+.3</td>
<td>-2.4</td>
<td>- 0.94 +1.10 +0.43 -0.12</td>
</tr>
<tr>
<td>.05</td>
<td>1.94</td>
<td>+ 1.80</td>
<td>+.4</td>
<td>-1.6</td>
<td>- 2.70 +1.80 -1.40 -2.90</td>
</tr>
<tr>
<td>.10</td>
<td>1.18</td>
<td>+ 6.90</td>
<td>+.4</td>
<td>-1.3</td>
<td>-10.40 +4.00 -9.20 -11.80</td>
</tr>
<tr>
<td>.20</td>
<td>0.61</td>
<td>+19.50</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
FIGURE 9-3. - PERCENTAGE OF ERROR (see TABLE 9-10)
FIGURE 9-4. ERROR FOR A PEAK 689.7 PPM (see TABLE 9-10)
the time increment becomes extremely small. Likewise, the distance increment limits the final accuracy which may be obtained, even when small time increments are used.

The implicit method has a more consistent accuracy throughout the useful range of time increments; this allows the use of larger time increments with the assurance that good accuracy can be obtained.

Starting the calculations with an initial distribution rather than a slug concentration at a single point has certain advantages. Although the accuracy near the peak is not significantly affected by the initial conditions, the accuracy of the leading and trailing edges is noticeably improved by using a distribution of values rather than a single concentration; this difference would become less significant as the dispersion coefficient is increased. Also, it should be noted that the leading and trailing edges of a profile for a slug load are generally not of much importance because they represent such a small percentage of the total mass of material.

Another advantage of using an initial distribution is that the implicit method remains accurate for much larger time increments. Table 9-12 shows that the implicit method is very accurate for a single starting concentration up to the time increment of 0.01 days, which is twice the value for stability for the explicit method. For an initial known distribution,
Table 9-13 shows that the implicit method is accurate up to a time increment of 0.05 days, which is ten times the stability criteria of the explicit method.

The distance increment of 0.1 miles provided good accuracy for any reasonable time increment applied in the error analysis. For example, with the explicit method a maximum error of 18.2 ppm is shown in Figure 9-4 near the peak concentration of 689.7 ppm and an error of about 4 ppm is shown for the trailing edge of the concentration. Thus, if a similar distribution were encountered from real data with a maximum BOD concentration of 6.89 mg/l, the maximum expected error from the explicit model would be 0.18 mg/l near the peak and about 0.04 mg/l near the edges of the profile; this is an acceptable error for real data. It should be noted that this magnitude of error resulted from a time increment of 0.004 days, which is larger than the time increment that would probably be used for these circumstances.

The following conclusions are clear from the calculations presented in this section: when a careful choice is made for the distance increment, the explicit method can provide good accuracy until the stability criteria is exceeded; the implicit method can provide good accuracy up to three times the stability criteria of the explicit method. When an initial distribution is known, the implicit method may remain accurate for up to ten times the stability criteria for time of the explicit method. Some authors recommend that the time increment for the explicit method be
restricted to one-half the stability criteria and that the Crank-Nicolson method be limited to twice this time increment in order to avoid inaccuracies caused by oscillations; these recommendations appear to be overly conservative for the examples presented in this section.

TWO-DIMENSIONAL ACCURACY ANALYSIS

The explicit and implicit finite difference models for two dimensions exhibited the same type of accuracy and stability as the one-dimensional models.

Steady-State Solutions. - Excellent accuracy can be obtained for steady-state solutions in two dimensions. However, iterating until the steady-state is reached can consume a large amount of computer time unless an initial distribution is used which is fairly close to the final values.

Instantaneous Releases of Slug Loads. - Excellent accuracy can be obtained in two dimensions when only dispersion is modeled. The introduction of velocity causes the same type of skewing and 'numerical dispersion' problems as in the one-dimensional cases; these inaccuracies can be minimized greatly by the proper choice of time and distance increments.

Table 9-14 shows the degree of accuracy which can be expected when both velocity and dispersion are included in the calculations. To obtain this distribution, an initial symmetrical
TABLE 9-14. CONCENTRATIONS AT TIME = 0.11 DAYS

<table>
<thead>
<tr>
<th>Miles</th>
<th>-.5</th>
<th>-.3</th>
<th>-.1</th>
<th>+.1</th>
<th>+.3</th>
<th>+.5</th>
<th>+.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>-.5</td>
<td>77</td>
<td>174</td>
<td>329</td>
<td>518</td>
<td>681</td>
<td>745</td>
<td>681</td>
</tr>
<tr>
<td>-.3</td>
<td>174</td>
<td>394</td>
<td>745</td>
<td>1174</td>
<td>1543</td>
<td>1689</td>
<td>1543</td>
</tr>
<tr>
<td>-.1</td>
<td>329</td>
<td>745</td>
<td>1408</td>
<td>2219</td>
<td>2915</td>
<td>3192</td>
<td>2915</td>
</tr>
<tr>
<td>+.1</td>
<td>518</td>
<td>1174</td>
<td>2219</td>
<td>3496</td>
<td>4592</td>
<td>5029</td>
<td>4592</td>
</tr>
<tr>
<td>+.3</td>
<td>681</td>
<td>1543</td>
<td>2915</td>
<td>4592</td>
<td>6032</td>
<td>6606</td>
<td>6032</td>
</tr>
<tr>
<td>+.5</td>
<td>745</td>
<td>1689</td>
<td>3192</td>
<td>5029</td>
<td>6606</td>
<td>7234</td>
<td>6606</td>
</tr>
<tr>
<td>+.7</td>
<td>681</td>
<td>1543</td>
<td>2915</td>
<td>4592</td>
<td>6032</td>
<td>6606</td>
<td>6032</td>
</tr>
</tbody>
</table>

(a) Exact Solution

<table>
<thead>
<tr>
<th>Miles</th>
<th>-.5</th>
<th>-.3</th>
<th>-.1</th>
<th>+.1</th>
<th>+.3</th>
<th>+.5</th>
<th>+.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>-.5</td>
<td>75</td>
<td>171</td>
<td>327</td>
<td>519</td>
<td>680</td>
<td>739</td>
<td>669</td>
</tr>
<tr>
<td>-.3</td>
<td>171</td>
<td>392</td>
<td>750</td>
<td>1188</td>
<td>1558</td>
<td>1693</td>
<td>1532</td>
</tr>
<tr>
<td>-.1</td>
<td>327</td>
<td>750</td>
<td>1434</td>
<td>2273</td>
<td>2980</td>
<td>3238</td>
<td>2931</td>
</tr>
<tr>
<td>+.1</td>
<td>519</td>
<td>1188</td>
<td>2273</td>
<td>3603</td>
<td>4723</td>
<td>5131</td>
<td>4645</td>
</tr>
<tr>
<td>+.3</td>
<td>680</td>
<td>1558</td>
<td>2980</td>
<td>4723</td>
<td>6193</td>
<td>6727</td>
<td>6090</td>
</tr>
<tr>
<td>+.5</td>
<td>739</td>
<td>1693</td>
<td>3238</td>
<td>5131</td>
<td>6727</td>
<td>7308</td>
<td>6616</td>
</tr>
<tr>
<td>+.7</td>
<td>669</td>
<td>1532</td>
<td>2931</td>
<td>4645</td>
<td>6090</td>
<td>6616</td>
<td>5989</td>
</tr>
</tbody>
</table>

(b) Implicit Solution for $\Delta T = 0.0025$ Days

<table>
<thead>
<tr>
<th>Miles</th>
<th>-.5</th>
<th>-.3</th>
<th>-.1</th>
<th>+.1</th>
<th>+.3</th>
<th>+.5</th>
<th>+.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>-.5</td>
<td>73</td>
<td>167</td>
<td>320</td>
<td>507</td>
<td>669</td>
<td>734</td>
<td>670</td>
</tr>
<tr>
<td>-.3</td>
<td>167</td>
<td>259</td>
<td>541</td>
<td>941</td>
<td>1529</td>
<td>1678</td>
<td>1532</td>
</tr>
<tr>
<td>-.1</td>
<td>320</td>
<td>730</td>
<td>1393</td>
<td>2212</td>
<td>2920</td>
<td>3203</td>
<td>2925</td>
</tr>
<tr>
<td>+.1</td>
<td>507</td>
<td>1159</td>
<td>2212</td>
<td>3513</td>
<td>4635</td>
<td>5084</td>
<td>4641</td>
</tr>
<tr>
<td>+.3</td>
<td>669</td>
<td>1529</td>
<td>2920</td>
<td>4635</td>
<td>6113</td>
<td>6700</td>
<td>6112</td>
</tr>
<tr>
<td>+.5</td>
<td>734</td>
<td>1678</td>
<td>3203</td>
<td>5084</td>
<td>6700</td>
<td>7334</td>
<td>6684</td>
</tr>
<tr>
<td>+.7</td>
<td>670</td>
<td>1532</td>
<td>2925</td>
<td>4641</td>
<td>6112</td>
<td>6684</td>
<td>6085</td>
</tr>
</tbody>
</table>

(c) Explicit Solution for $\Delta T = 0.0010$ Days
distribution was read into the models for a starting time of 0.01 days. The models then calculated the distribution until the elapsed time was 0.11 days. The explicit method used a time increment of 0.0025 days. The stability limits for the explicit method were 0.0025 days and 0.4 miles. The models used a distance increment of 0.1 miles; however, Table 9-14 presents the results for only every 0.2 miles. The exact solution was obtained by applying Equation 8-5 with \( m = 10,000 \); dispersion coefficient = 1 mi\(^2\)/day (322.7 ft\(^2\)/sec); and velocity = 5 miles/day (.306 ft/sec). Almost equivalent accuracy was obtained by the models when a time increment of 0.002 days was used for the explicit method and a time increment of 0.005 days was used for the implicit method.

Good accuracy was also obtained when the dispersion coefficients and velocities varied in the two dimensions. The explicit model was applied to the following parameters: \( EX = 1 \) mi\(^2\)/day (322.7 ft\(^2\)/sec); \( EZ = 0.025 \) ft\(^2\)/sec \( (0.775 \times 10^{-4} \) mi\(^2\)/day\); \( VX = 5 \) miles/day (.306 ft/sec); \( VZ = 0.0 \) ft/sec. The initial profile at 0.01 days is shown in Table 9-15 for selected values, and the distribution calculated for 0.11 days is shown in Table 9-16. The vertical distance increment is 5 feet and the horizontal distance increment is 0.1 miles. The model used a time increment 0.002 days and the stability criteria for time was 0.0027 days. The exact solution is based on a value \( m = 10 \) in Equation 8-5.
### TABLE 9-15.-INITIAL VALUES FOR NON-SYMMETRICAL DISTRIBUTION

<table>
<thead>
<tr>
<th>Depth (Feet)</th>
<th>-0.5</th>
<th>-0.4</th>
<th>-0.3</th>
<th>-0.2</th>
<th>-0.1</th>
<th>L.P.*</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 ft.</td>
<td>17.</td>
<td>166</td>
<td>953.</td>
<td>3325.</td>
<td>7040.</td>
<td>9039.</td>
<td>7040.</td>
<td>3325.</td>
<td>953.</td>
<td>166.</td>
<td>17.</td>
</tr>
<tr>
<td>5 ft.</td>
<td>13.</td>
<td>6769.</td>
<td>2842.</td>
<td>669.</td>
<td>2.</td>
<td>9.</td>
<td>32.</td>
<td>69.</td>
<td>88.</td>
<td>69.</td>
<td>32.</td>
</tr>
<tr>
<td>10 ft.</td>
<td>5.</td>
<td>669.</td>
<td>2842.</td>
<td>669.</td>
<td>2.</td>
<td>9.</td>
<td>32.</td>
<td>69.</td>
<td>88.</td>
<td>69.</td>
<td>32.</td>
</tr>
<tr>
<td>15 ft.</td>
<td>1.</td>
<td>669.</td>
<td>2842.</td>
<td>669.</td>
<td>2.</td>
<td>9.</td>
<td>32.</td>
<td>69.</td>
<td>88.</td>
<td>69.</td>
<td>32.</td>
</tr>
<tr>
<td>20 ft.</td>
<td>0.</td>
<td>2.</td>
<td>9.</td>
<td>32.</td>
<td>69.</td>
<td>88.</td>
<td>69.</td>
<td>32.</td>
<td>9.</td>
<td>2.</td>
<td>0.</td>
</tr>
<tr>
<td>25 ft.</td>
<td>0.</td>
<td>7.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>30 ft.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>35 ft.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
</tbody>
</table>

*Loading Point
<table>
<thead>
<tr>
<th>Depth (Feet)</th>
<th>L.P.*</th>
<th>0.1 Miles</th>
<th>0.2 Miles</th>
<th>0.3 Miles</th>
<th>0.4 Miles</th>
<th>0.5 Miles</th>
<th>0.6 Miles</th>
<th>0.7 Miles</th>
<th>0.8 Miles</th>
<th>0.9 Miles</th>
<th>1.0 Miles</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>466.</td>
<td>571.</td>
<td>670.</td>
<td>750.</td>
<td>803.</td>
<td>822.</td>
<td>803.</td>
<td>750.</td>
<td>670.</td>
<td>571.</td>
<td>466.</td>
</tr>
<tr>
<td>5 ft.</td>
<td>446.</td>
<td>571.</td>
<td>670.</td>
<td>750.</td>
<td>803.</td>
<td>822.</td>
<td>803.</td>
<td>750.</td>
<td>670.</td>
<td>571.</td>
<td>466.</td>
</tr>
<tr>
<td></td>
<td>453.</td>
<td>805.</td>
<td>800.</td>
<td>461.</td>
<td>453.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 ft.</td>
<td>413.</td>
<td>571.</td>
<td>670.</td>
<td>750.</td>
<td>803.</td>
<td>822.</td>
<td>803.</td>
<td>750.</td>
<td>670.</td>
<td>571.</td>
<td>466.</td>
</tr>
<tr>
<td></td>
<td>419.</td>
<td>745.</td>
<td>740.</td>
<td>426.</td>
<td>419.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>306.</td>
<td>440.</td>
<td>495.</td>
<td>531.</td>
<td>545.</td>
<td>553.</td>
<td>498.</td>
<td>444.</td>
<td>379.</td>
<td>309.</td>
<td>306.</td>
</tr>
<tr>
<td></td>
<td>241.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>241.</td>
</tr>
<tr>
<td></td>
<td>181.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>181.</td>
</tr>
<tr>
<td></td>
<td>128.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Loading Point

Figure Key: 458. = Exact
            466. = Model
For the examples shown in this chapter, the accuracy was poorest during early time steps and continued to improve as time progressed.
CHAPTER X

COMPUTATIONAL EXPERIENCE

Three aspects of the use of the computer models will be discussed briefly in this chapter: time requirements; use of boundary conditions; and choice of input parameters.

TIME REQUIREMENTS

The computer programs developed in this study are very economical in their use of computer time when compared to some other models for large estuaries (26). However, the programs are time-consuming with regard to the computer budget of a student; this is especially true for the two-dimensional programs when a large number of alternatives must be investigated to insure accuracy.

Table 10-1 presents data on the time requirements for some typical runs on an IBM 360/65 of the most recent versions of the models. The information includes the size of the grid, the number of iterations, the time to compile and initialize the data, the execution time for the computation phase, the number of seconds of execution time per iteration, and the number of seconds of execution time per iteration per grid point. This final parameter allows for the computation of an approximate run time of the models on an IBM 360/65. For instance, the approximate time requirement for MASSTRANS-1 would be:
<table>
<thead>
<tr>
<th>Grid</th>
<th>Iterations</th>
<th>Initialization (sec)</th>
<th>Execution Time (Seconds)</th>
<th>Seconds Iteration</th>
<th>Seconds Inter.-Pt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 x 15</td>
<td>102</td>
<td>22.</td>
<td>111.5</td>
<td>1.09</td>
<td>$484 \times 10^{-5}$</td>
</tr>
<tr>
<td>15 x 15</td>
<td>52</td>
<td>22.</td>
<td>55.2</td>
<td>1.06</td>
<td>$472 \times 10^{-5}$</td>
</tr>
<tr>
<td>36 x 11</td>
<td>42</td>
<td>22.</td>
<td>81.6</td>
<td>1.94</td>
<td>$490 \times 10^{-5}$</td>
</tr>
<tr>
<td>42 x 1</td>
<td>122</td>
<td>20.</td>
<td>21.0</td>
<td>0.172</td>
<td>$409 \times 10^{-5}$</td>
</tr>
<tr>
<td>36 x 1</td>
<td>182</td>
<td>20.</td>
<td>21.6</td>
<td>0.119</td>
<td>$331 \times 10^{-5}$</td>
</tr>
</tbody>
</table>

(b) MASTRANS-II (Implicit)

<table>
<thead>
<tr>
<th>Grid</th>
<th>Iterations</th>
<th>Initialization (sec)</th>
<th>Execution Time (Seconds)</th>
<th>Seconds Iteration</th>
<th>Seconds Inter.-Pt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>15 x 15</td>
<td>42</td>
<td>25.</td>
<td>115.2</td>
<td>2.75</td>
<td>$1225 \times 10^{-5}$</td>
</tr>
<tr>
<td>15 x 15</td>
<td>22</td>
<td>25.</td>
<td>58.2</td>
<td>2.65</td>
<td>$1180 \times 10^{-5}$</td>
</tr>
</tbody>
</table>

(c) IDEAL-I (Explicit)

<table>
<thead>
<tr>
<th>Grid</th>
<th>Iterations</th>
<th>Initialization (sec)</th>
<th>Execution Time (Seconds)</th>
<th>Seconds Iteration</th>
<th>Seconds Inter.-Pt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>51 x 1</td>
<td>1200</td>
<td>1.</td>
<td>33.72</td>
<td>0.0281</td>
<td>$55 \times 10^{-5}$</td>
</tr>
<tr>
<td>51 x 1</td>
<td>600</td>
<td>1.</td>
<td>17.90</td>
<td>0.0298</td>
<td>$58 \times 10^{-5}$</td>
</tr>
<tr>
<td>51 x 1</td>
<td>300</td>
<td>1.</td>
<td>9.69</td>
<td>0.0323</td>
<td>$63 \times 10^{-5}$</td>
</tr>
</tbody>
</table>

(d) IDEAL-II (Implicit)

<table>
<thead>
<tr>
<th>Grid</th>
<th>Iterations</th>
<th>Initialization (sec)</th>
<th>Execution Time (Seconds)</th>
<th>Seconds Iteration</th>
<th>Seconds Inter.-Pt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>51 x 1</td>
<td>600</td>
<td>1.</td>
<td>30.93</td>
<td>0.0516</td>
<td>$101 \times 10^{-5}$</td>
</tr>
<tr>
<td>51 x 1</td>
<td>300</td>
<td>1.</td>
<td>16.83</td>
<td>0.0562</td>
<td>$110 \times 10^{-5}$</td>
</tr>
<tr>
<td>51 x 1</td>
<td>120</td>
<td>1.</td>
<td>6.81</td>
<td>0.0567</td>
<td>$111 \times 10^{-5}$</td>
</tr>
</tbody>
</table>

(e) OXTRANS-I (Explicit)

<table>
<thead>
<tr>
<th>Grid</th>
<th>Iterations</th>
<th>Initialization (sec)</th>
<th>Execution Time (Seconds)</th>
<th>Seconds Iteration</th>
<th>Seconds Inter.-Pt.</th>
</tr>
</thead>
<tbody>
<tr>
<td>42 x 1</td>
<td>204</td>
<td>21.</td>
<td>82.80</td>
<td>.406</td>
<td>$965 \times 10^{-5}$</td>
</tr>
<tr>
<td>42 x 1</td>
<td>102</td>
<td>21.</td>
<td>34.20</td>
<td>.335</td>
<td>$798 \times 10^{-5}$</td>
</tr>
<tr>
<td>31 x 1</td>
<td>402</td>
<td>21.</td>
<td>106.00</td>
<td>.265</td>
<td>$853 \times 10^{-5}$</td>
</tr>
<tr>
<td>33 x 7</td>
<td>68</td>
<td>23.</td>
<td>147.00</td>
<td>2.160</td>
<td>$935 \times 10^{-5}$</td>
</tr>
</tbody>
</table>
Total Computer Time in Seconds

\[ = 25 + (500 \times 10^{-5} \times \text{iterations} \times \text{total number of grid points}) \]

Since different options and boundary conditions were used for the various programs, the time requirements are not completely comparable. However, it is clear that the one-dimensional version of the Crank-Nicolson implicit method tends to take 1.75 times as long as the explicit method for the same time and distance increments. Likewise, the two-dimensional version of the implicit method takes 2.5 times as long as the explicit method. However, since the implicit method remains accurate for much larger time increments than the explicit method, the implicit method often proves to be the more economical choice.

OXTRANS-I, the largest program, has the following storage requirements:

Object Code = 66,000 bytes; Array Area = 59,000 bytes.

BOUNDARY CONDITIONS

For steady-state profiles and for leading and trailing edges of a concentration profile, inaccurate boundary conditions significantly affect only the closest three internal concentrations. However, as the concentrations near the boundary become a high percentage of the peak value, the entire profile can be distorted by inaccurate boundary extrapolations. The most straightforward
way of avoiding this "feedback" is to keep the peak far away from the boundaries where values must be extrapolated; this can be accomplished by using a large grid or by moving the grid when the peak gets too close to the boundaries.

CHOICE OF INPUT PARAMETERS

Two-dimensional, time-varying data is difficult and expensive to obtain. However, this type of data is required to completely calibrate and verify two-dimensional models.

In spite of the logistical problems involved, velocity is one of the more easily involved parameters because it can be measured directly with sensitive current meters. If horizontal velocities are carefully measured, vertical velocities can be calculated through the two-dimensional continuity equation. For estuaries whose width and velocities vary throughout the estuary, care must be taken to insure that input values for velocity satisfy continuity. If measured data is not available, approximate intra-tidal velocity values also can be obtained by coupling the finite difference models to computer models for tidal velocities; a one-dimensional tidal model is available for the Houston Ship Channel (110).

Dispersion coefficients are particularly troublesome to obtain in two dimensions. As shown by Ippen (51), the value of the horizontal dispersion coefficient for one-dimensional models is
much larger than the horizontal dispersion coefficients appropriate for two-dimensional models. Horizontal dispersion coefficients for one-dimensional models must represent the effects of the more localized eddy diffusion; in two-dimensional analysis, the two extra variables of vertical velocity and vertical dispersion are available to aid in representing these effects. More research needs to be done to determine representative values for vertical dispersion for individual estuaries. Work by Prichard (80) and Bowden (5) suggests that vertical dispersion coefficients can be expected to be in the range of 0.001 to 0.050 \text{ ft}^2/\text{sec}, where the higher values indicate a higher degree of vertical mixing.

Adequate source and sink terms are sometimes difficult to obtain. Decay rates in highly polluted estuaries must be chosen with care because of possible inhibition effects (87, 85). Re-aeration rates are particularly elusive for slowly moving, highly contaminated estuaries.

As Callaway has said (20), this is truly "the age of the great coefficient hunt."
CHAPTER XI

APPLICATION TO THE HOUSTON SHIP CHANNEL

An important aspect in proving the accuracy and usefulness of a model comes through testing the ability of the model to reproduce actual data. On May 24-26, 1971, a dye study was carried out on the Houston Ship Channel by the Estuarine Systems Projects of Texas A&M University during a moderate rainfall on the watershed surrounding the channel. This study provided sufficient data (31) to determine the ability of the model to reproduce the two-dimensional behavior of the channel during dynamic conditions.

Two hundred pounds of Rhodamine WT dye were released at Mile 16.5 on May 24, 1971, at 12 noon (hereafter called H-hour). Two-dimensional measurements for dye were taken periodically on the channel during the 44 hours following the release. The gauged runoff into the channel peaked at H + 7 hours (7:00 p.m.) on May 24 at a discharge of approximately 4750 cfs (31). The channel was partially stratified during the study period; for example, on the morning of May 25, the salinity at Mile 12 ranged from 12.6 ppt at the surface to 15.6 ppt at the bottom. The tide readings for the study period are shown in Figure 11-1.

The first two-dimensional data collection for dye fluorescence was carried out at H + 4 hours. The fluorescence contours for that sampling run are shown in Figure 11-2; the channel
FIGURE 11-1. - TIDE READINGS
had a background fluorescence of about 10 units.

The concentrations from this first sampling run were used as the input concentrations for the model. Concentrations were chosen at intervals of 0.25 miles in the longitudinal direction and 5.0 feet in the vertical. These initial concentration values are shown in Table 11-1.

Data was available also from a two-dimensional sampling run at H + 15 hours. The aim of the simulation was to compare the concentrations predicted by the model with the concentrations measured at H + 15 hours.

Average values were used in the model for velocity, dispersion, and decay during the 11-hour simulation period. Horizontal velocities were obtained by channel measurements and observation of fluorescence contours. Vertical velocity was considered to be insignificant and was set equal to zero. Horizontal dispersion was set at 450 ft²/sec (1.39 miles²/day) at all points. Vertical dispersion was considered to be proportional to horizontal velocity and had a maximum value of 0.004 ft²/sec (1.24 x 10⁻⁵ miles²/day). Values for horizontal velocity and vertical dispersion coefficients are shown in Table 11-2.

Widths were approximated from Corps of Engineer cross-section data and from work by Hutton (45, 46). These values are shown in Table 11-3 and represent widths at a 20-foot depth. The sampling data indicated that a considerable
### TABLE 11-1. INITIAL CONCENTRATIONS

<table>
<thead>
<tr>
<th>Mile</th>
<th>0'</th>
<th>5'</th>
<th>10'</th>
<th>15'</th>
<th>20'</th>
<th>25'</th>
<th>30'</th>
<th>35'</th>
<th>40'</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.00</td>
<td>13.2</td>
<td>12.0</td>
<td>11.7</td>
<td>12.0</td>
<td>13.0</td>
<td>13.0</td>
<td>11.8</td>
<td>10.2</td>
<td>10.0</td>
</tr>
<tr>
<td>16.75</td>
<td>14.0</td>
<td>13.5</td>
<td>12.0</td>
<td>13.0</td>
<td>14.0</td>
<td>14.0</td>
<td>12.1</td>
<td>10.3</td>
<td>10.0</td>
</tr>
<tr>
<td>16.50</td>
<td>16.2</td>
<td>15.0</td>
<td>14.0</td>
<td>14.0</td>
<td>15.0</td>
<td>15.0</td>
<td>12.3</td>
<td>10.4</td>
<td>10.0</td>
</tr>
<tr>
<td>16.25</td>
<td>19.5</td>
<td>17.0</td>
<td>16.0</td>
<td>16.0</td>
<td>17.0</td>
<td>16.0</td>
<td>12.5</td>
<td>10.5</td>
<td>10.0</td>
</tr>
<tr>
<td>16.00</td>
<td>100.0</td>
<td>19.0</td>
<td>19.0</td>
<td>19.0</td>
<td>20.0</td>
<td>17.0</td>
<td>12.7</td>
<td>10.6</td>
<td>10.0</td>
</tr>
<tr>
<td>15.75</td>
<td>125.0</td>
<td>80.0</td>
<td>50.0</td>
<td>40.0</td>
<td>35.0</td>
<td>19.0</td>
<td>13.0</td>
<td>10.8</td>
<td>10.0</td>
</tr>
<tr>
<td>15.50</td>
<td>175.0</td>
<td>220.0</td>
<td>300.0</td>
<td>130.0</td>
<td>65.0</td>
<td>22.0</td>
<td>13.5</td>
<td>11.0</td>
<td>10.0</td>
</tr>
<tr>
<td>15.25</td>
<td>300.0</td>
<td>500.0</td>
<td>450.0</td>
<td>175.0</td>
<td>80.0</td>
<td>30.0</td>
<td>14.0</td>
<td>10.8</td>
<td>10.0</td>
</tr>
<tr>
<td>15.00</td>
<td>750.0</td>
<td>900.0</td>
<td>330.0</td>
<td>150.0</td>
<td>65.0</td>
<td>24.0</td>
<td>13.3</td>
<td>10.6</td>
<td>10.0</td>
</tr>
<tr>
<td>14.75</td>
<td>2000.0</td>
<td>550.0</td>
<td>225.0</td>
<td>100.0</td>
<td>40.0</td>
<td>20.0</td>
<td>12.6</td>
<td>10.4</td>
<td>10.0</td>
</tr>
<tr>
<td>14.50</td>
<td>1000.0</td>
<td>300.0</td>
<td>130.0</td>
<td>60.0</td>
<td>25.0</td>
<td>17.0</td>
<td>12.0</td>
<td>10.3</td>
<td>10.0</td>
</tr>
<tr>
<td>14.25</td>
<td>300.0</td>
<td>200.0</td>
<td>85.0</td>
<td>30.0</td>
<td>18.0</td>
<td>14.0</td>
<td>11.6</td>
<td>10.2</td>
<td>10.0</td>
</tr>
<tr>
<td>14.00</td>
<td>175.0</td>
<td>85.0</td>
<td>35.0</td>
<td>19.0</td>
<td>15.0</td>
<td>12.0</td>
<td>11.2</td>
<td>10.1</td>
<td>10.0</td>
</tr>
<tr>
<td>13.75</td>
<td>100.0</td>
<td>20.0</td>
<td>19.0</td>
<td>15.0</td>
<td>12.0</td>
<td>11.5</td>
<td>10.8</td>
<td>10.0</td>
<td>10.0</td>
</tr>
<tr>
<td>13.50</td>
<td>30.0</td>
<td>18.0</td>
<td>17.0</td>
<td>12.0</td>
<td>11.7</td>
<td>11.0</td>
<td>10.4</td>
<td>10.0</td>
<td>10.0</td>
</tr>
<tr>
<td>13.25</td>
<td>19.0</td>
<td>17.0</td>
<td>15.0</td>
<td>11.8</td>
<td>11.5</td>
<td>10.7</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
</tr>
<tr>
<td>13.00</td>
<td>17.0</td>
<td>16.0</td>
<td>14.0</td>
<td>11.6</td>
<td>11.2</td>
<td>10.4</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
</tr>
<tr>
<td>12.75</td>
<td>15.0</td>
<td>15.0</td>
<td>13.0</td>
<td>11.4</td>
<td>11.0</td>
<td>10.2</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
</tr>
<tr>
<td>12.50</td>
<td>14.0</td>
<td>14.0</td>
<td>12.0</td>
<td>11.3</td>
<td>10.7</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
</tr>
<tr>
<td>12.25</td>
<td>13.0</td>
<td>13.0</td>
<td>11.8</td>
<td>11.2</td>
<td>10.5</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
</tr>
<tr>
<td>*12.00</td>
<td>12.5</td>
<td>12.0</td>
<td>11.6</td>
<td>11.1</td>
<td>10.2</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
</tr>
</tbody>
</table>

*All concentrations not shown were equal to 10.0. The background fluorescence of 10.0 was subtracted from the concentrations shown in the table and the resulting values were used as the input data.*
### TABLE 11-2.-HORIZONTAL VELOCITIES AND VERTICAL DISPERSION COEFFICIENTS

<table>
<thead>
<tr>
<th>Depth (ft.)</th>
<th>Horizontal Velocity (ft/sec)</th>
<th>Vertical Dispersion (ft^2/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.3667</td>
<td>0.3333</td>
</tr>
<tr>
<td>5'</td>
<td>0.4200</td>
<td>0.3820</td>
</tr>
<tr>
<td>10'</td>
<td>0.4400</td>
<td>0.4000</td>
</tr>
<tr>
<td>15'</td>
<td>0.4400</td>
<td>0.4000</td>
</tr>
<tr>
<td>20'</td>
<td>0.4333</td>
<td>0.3940</td>
</tr>
<tr>
<td>25'</td>
<td>0.3893</td>
<td>0.3540</td>
</tr>
<tr>
<td>30'</td>
<td>0.2733</td>
<td>0.2485</td>
</tr>
<tr>
<td>35'</td>
<td>0.2333</td>
<td>0.2120</td>
</tr>
<tr>
<td>40'</td>
<td>0.2333</td>
<td>0.2120</td>
</tr>
</tbody>
</table>

### TABLE 11-3.-WIDTHS AT 20-FOOT DEPTH

<table>
<thead>
<tr>
<th>Mile</th>
<th>Width (feet)</th>
<th>Mile</th>
<th>Width (feet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.00</td>
<td>497</td>
<td>12.75</td>
<td>872</td>
</tr>
<tr>
<td>16.75</td>
<td>510</td>
<td>12.50</td>
<td>931</td>
</tr>
<tr>
<td>16.50</td>
<td>524</td>
<td>12.25</td>
<td>863</td>
</tr>
<tr>
<td>16.25</td>
<td>537</td>
<td>12.00</td>
<td>796</td>
</tr>
<tr>
<td>16.00</td>
<td>550</td>
<td>11.75</td>
<td>728</td>
</tr>
<tr>
<td>15.75</td>
<td>562</td>
<td>11.50</td>
<td>661</td>
</tr>
<tr>
<td>15.50</td>
<td>575</td>
<td>11.25</td>
<td>657</td>
</tr>
<tr>
<td>15.25</td>
<td>562</td>
<td>11.00</td>
<td>654</td>
</tr>
<tr>
<td>15.00</td>
<td>550</td>
<td>10.75</td>
<td>650</td>
</tr>
<tr>
<td>14.75</td>
<td>537</td>
<td>10.50</td>
<td>647</td>
</tr>
<tr>
<td>14.50</td>
<td>525</td>
<td>10.25</td>
<td>633</td>
</tr>
<tr>
<td>14.25</td>
<td>568</td>
<td>10.00</td>
<td>619</td>
</tr>
<tr>
<td>14.00</td>
<td>611</td>
<td>9.75</td>
<td>605</td>
</tr>
<tr>
<td>13.75</td>
<td>654</td>
<td>9.50</td>
<td>591</td>
</tr>
<tr>
<td>13.50</td>
<td>697</td>
<td>9.25</td>
<td>586</td>
</tr>
<tr>
<td>13.25</td>
<td>756</td>
<td>9.00</td>
<td>582</td>
</tr>
</tbody>
</table>
amount of dye was lost by sediment adsorption and other influences. An exponential decay rate of 2.5 / day was found to adequately represent this loss of dye for the period being simulated.

The simulation run used a time increment of 600 seconds; the maximum allowable time increment for stability would have been 1155 seconds. The simulation required 66 iterations; the compile time plus the execution time was 2.5 minutes on an IBM 360/65 for MASSTRANS-I.

The measured dye concentration profiles for hour H + 15 are shown in Figure 11-3. The concentration profiles predicted by MASSTRANS-I are shown in Figure 11-4. Agreement between the profiles is very good considering that only average values were used as input data in the model.

Comparison between the dye concentrations and model predictions are shown at 5-foot increments in Figure 11-5. Peak values are accurately matched and the mass of dye appears to be conserved by the model. The skewing of the measured profiles was caused by local velocity fluctuations which were not included in the model input data.

Figure 11-6 shows a comparison between peak concentrations in the vertical direction for the model and the sampled data. This comparison demonstrates that the model can be especially accurate in representing the peak concentrations at the appropriate depths.
FIGURE 11-3. - MEASURED CONCENTRATIONS AT HOUR H + 15
Figure 11-4a - Concentrations predicted by MASTRANS-1 for hour H + 15

Horizontal Distance (m/ft)

Depth (feet)
FIGURE 11-5. COMPARISON OF MEASURED AND SIMULATED PROFILES

(Continued on next page)
FIGURE 11-6. - COMPARISON OF PEAK CONCENTRATIONS
These comparisons demonstrate that the two-dimensional models can accurately represent dynamic conditions on the Houston Ship Channel, even when averaged input data is used. When more precise data for velocity, dispersion, and other parameters are fed into the model, more accurate simulations can be expected.

MODELING BOD AND DISSOLVED OXYGEN IN THE HOUSTON SHIP CHANNEL

A major use of OXTRANS-I will be to model the BOD and dissolved oxygen distributions in the Channel for rapidly changing conditions. The ability of the model to accomplish this task was demonstrated by applying the velocity and dispersion values which existed during the dye study to a hypothetical dissolved oxygen distribution. A simple initial distribution of dissolved oxygen and BOD was chosen in order that changes in the distribution would be more apparent. The hypothetical initial values for BOD$_{20}$ and dissolved oxygen are shown in Figure 11-7; this figure shows a saturated value of 8 ppm of oxygen and no BOD below Mile 15. Above Mile 15, a BOD$_{20}$ concentration of 13 ppm is shown along with a zero value for dissolved oxygen.

The velocity and dispersion values from the dye study simulation were applied to the initial DO-BOD values, and the simulation was run again until hour H + 15. The following parameters were used: aerobic decay rate (base e) = 0.25/day; anaerobic decay rate (base e) = 0.08/day; and reaeration rate (base e) =
**Figure 11-7.** Initial hypothetical values for BOD$_{20}$ and DO at H + 4 hours.
0.10/day. The results of this simulation are shown in Figure 11-8, for BOD\textsubscript{20} and Figure 11-9, for dissolved oxygen. During the 11 hours of simulation, the saturated value of 8 ppm is pushed from Mile 15 to Mile 9.

This example demonstrates that the most pronounced change during the early stages of increased flow are due mainly to the downstream movement of the low oxygen conditions by velocity and dispersion. Several different initial BOD concentrations were tried in the model and the dissolved oxygen pattern at the end of the simulation was essentially the same as the values already shown in Figure 11-9. However, over a longer period of time, the persistence of the low oxygen conditions would obviously depend upon the amount of organic material that remained in the part of the channel being studied.

The conditions modeled in this simulation represent only moderate runoff and moderate pollution conditions. Extensive studies by Reynolds and Eckenfelder (85) during 1968 and 1969 show that the BOD\textsubscript{20} in the vicinity of Mile 16 can rise to at least 38 ppm and the decay rate (base e) can range between .07/day and .44/day. Likewise, much higher runoff conditions are common to the channel.

APPLICABILITY OF THE MODELS

The examples of this chapter and the previous chapter demonstrate the reliability of the models when applied to the Houston
FIGURE 11-8. - BOD$_{20}$ (ppm) AT H + 15 HOURS
FIGURE 11-9 - DISSOLVED OXYGEN (ppm) AT H + 15 HOURS
Ship Channel and similar estuaries. These models can use approximate input parameters to provide estimates of concentration distributions in one or two dimensions; these approximations can be used for rough engineering estimates of conditions in the estuary for various hydrodynamic and loading conditions. The models can also accept input parameters which are much more exact; this type of approach is useful for simulation or predictive purposes.
CHAPTER XII

DISCUSSION

When modeling a partially stratified estuary, the following steps should be taken in utilizing the computer programs that were developed in this study. Representative parameter values for the estuary being studied should be used in the one-dimensional explicit and implicit models and the computed concentrations should be checked against exact solutions. Several time and distance steps should be tried until good accuracy is obtained for the range of conditions being studied. These initial computations should indicate the approximate increments to use in the two-dimensional models. In turn, the two-dimensional models should be run several times with different increments. The best approach is to use both the explicit and implicit models to cross-check each other. Special care should be taken when studying instantaneous releases; inaccurate initial conditions can invalidate the entire computation.

One of the real dangers in mathematical modeling is that almost any type of estuary model can generate profiles that appear to be reasonable. Some estuary models have gained acceptance through repeated use and promotion, not through repeated proof of their accuracy. For this reason, this study has taken great care to establish the accuracy of its finite difference models and has
provided means of checking their accuracy when applied to parameters other than those used in this report. The models developed in this study include no weighting factors and depend solely on the estuary data provided as input; all of this data can be obtained independently of the models.

In this study, good accuracy was obtained for horizontal distance increments up to 0.25 miles. For finite difference methods, increasing the distance increment generally decreases the accuracy of the model and increases interference by numerical dispersion; this is especially true when studying time-changing behavior. Thus, a model whose grid dimensions can be changed easily has a great advantage: the model can be run several times with different distance increments until results of sufficient accuracy are obtained.

The computer programs that were developed in this study have numerous applications. The two-dimensional models can be applied directly to partially stratified estuaries of constant center-line depth; dredged channels in the Gulf Coast region are the most obvious examples of this type of estuary. The one-dimensional models can be applied directly to well-mixed estuaries.

The accuracy requirements of the input data for the models is determined by the needs of the user. Rough estimates of concentration profiles can be obtained by inputting a representative range of values for the estuary being studied. If more accurate
results are required, data for the model should be obtained from dye releases, velocity measurements, reaeration rate measurements, decay rate studies, and benthal deposit studies. Examples of these types of investigations are available in several sources (2, 8, 40, 44, 46, 56, 71, 85, 103).

The usefulness of the present form of the computer programs is limited in several respects. The two-dimensional models require computers with large core storage; in addition, the computer programs require a compiler which accepts FORTRAN IV and unformatted READ and PRINT statements. These shortcomings can be overcome by straightforward programming techniques: discs and tapes can be used as alternate storage locations and format statement can be rewritten according to requirements of the computer being used.

The finite difference methods discussed in this study can be extended to estuaries with varying depths. Likewise, the finite difference equations can be expanded to three dimensions. However, in both these cases, the computational grids must be matched to the geometry of the particular estuary; this would require extensive additional programming and would result in models which were estuary-dependent. Additional work is needed on models of this type.

Additional verification of the present models is also needed. The ultimate usefulness of these methods can be determined only
after extensive computations are made for a wide variety of inter-tidal and intra-tidal conditions.
CHAPTER XIII

SUMMARY AND CONCLUSIONS

The following goals were accomplished in this study: (a) computer models were developed which can calculate time-varying vertical and horizontal mass transport in partially stratified estuaries; (b) the accuracy and usefulness of explicit finite difference models were compared with that of Crank-Nicolson implicit finite difference models; (c) the applicability of finite difference models to the mass transport characteristics of the Houston Ship Channel was demonstrated; and (d) a summary was made of existing information on one- and two-dimensional mathematical models that have been applied to significant estuary problems.

Explicit and Crank-Nicolson finite difference models were developed for the one- and two-dimensional estuary equations with varying coefficients. The models were constructed to allow for the varying of parameters at any time at any grid point and were programmed in FORTRAN-IV computer language. Good accuracy was obtained by both types of models when proper time and distance increments were used. The Crank-Nicolson approach was found to be more accurate for a wider range of these increments. The concentration profiles for instantaneous releases and for steady-state conditions were analyzed. Accuracy was determined by comparison with analytical closed-form solutions.
Models also were developed to analyze the profiles for biochemical oxygen demand and dissolved oxygen under time-changing conditions. These models can analyze both aerobic and anaerobic conditions and included provisions for analyzing the effects of mechanical reaeration.

Applicability of these models to partially stratified estuaries was established by comparisons with dye study data from the Houston Ship Channel.

The Crank-Nicolson implicit method programmed in this study tends to take 1.75 times as much computer time as the explicit method in one-dimensional applications and 2.5 times as much time for two-dimensional applications when equal time and distance increments are used; however, the implicit method is sometimes the more economical choice since it remains stable and accurate for larger time increments than the explicit method.

The accuracy of the finite difference models is particularly sensitive to the distance increments. Horizontal grid increments of 0.25 miles or less can provide excellent accuracy when finite difference models are applied to instantaneous releases. Grid increments of 2.0 miles or less can provide excellent accuracy when finite difference models are applied to the downstream portions of steady-state profiles. Finite difference models should have grid sizes which can be varied easily to enable the choice of a distance increment which provides good accuracy for
the conditions being studied.

The finite difference methods outlined in this study can be extended to partially stratified estuaries with varying depth and to three-dimensional estuary calculations; these applications would require extensive modifications to the existing programs. Further verification of the present two-dimensional models also is needed to determine their general applicability.

This study and similar studies demonstrate that techniques are presently available to model a wide range of hydrodynamic and mass transport conditions in estuaries. A major future task is to determine for two and three dimensions the appropriate values for parameters such as velocity, dispersion, decay, re-aeration, photosynthesis, and benthal demands.
APPENDIX I. REFERENCES


8. Bunce, Ronald E., "A Discussion and Tabulation of Diffusion Coefficients for Tidal Waters Computed as a Function of Velocity," CB-SRBP Technical Paper No. 9, Middle Atlantic Region, Federal Water Pollution Control Administration, Charlottesville, Virginia, February 1967.


28. Gebhard, Thomas, Jr., "Numerical Solutions of the Two-Dimensional Convective Dispersion Equation," thesis presented to The University of Texas, at Austin, Texas, in May 1968, in partial fulfillment of the requirements for the degree of Doctor of Philosophy, Department of Civil Engineering.


44. Hung, Y. T., "Biological Reaction Rate Constants in Natural Water," thesis presented to The University of Texas, at Austin, Texas, in May 1970, in partial fulfillment of the requirements for the degree of Doctor of Philosophy, Civil Engineering Department.

45. Hutton, Welford S., "A Study of Benthic Deposits in the Houston Ship Channel," thesis presented to Texas A&M University, at College Station, Texas, in 1970, in partial fulfillment of the requirements for the degree of Master of Science, Department of Civil Engineering.


87. Schneider, Peter W., "Reaction Rates and Apparent Toxicity of Houston Ship Channel Water," thesis presented to Texas A&M University, at College Station, Texas, in May 1969, in partial fulfillment of the requirements for the degree of Master of Science, Department of Civil Engineering.


89. Shankar, Nelakantan J., "Influence of Tidal Inlets on Salinity and Related Phenomena in Estuaries," thesis presented to The University of Texas, at Austin, Texas, in January 1971, in partial fulfillment of the requirements for the degree of Doctor of Philosophy, Department of Civil Engineering.


APPENDIX II. - NOTATION

Dimensions are shown in brackets and are represented as follows: \( M = \text{mass}; \ L = \text{length}; \) and \( T = \text{time}. \)

\[
\begin{align*}
a &= \text{abbreviated notation for } a(x,z) \\
a(x,z) &= \text{left-hand boundary of estuary; lower limit of integration across the width, } [L] \\
A &= \text{average top salinity (equation 4-8 only)} \\
A_i &= \text{constant coefficient for Crank-Nicolson method} \\
A_x &= \text{cross-sectional area at location } x, \ [L^2] \\
A(X) &= \text{coefficient array for Crank-Nicolson method} \\
b &= \text{abbreviated notation for } b(x,z) \\
b(x,z) &= \text{right-hand boundary of estuary; upper limit of integration across the width, } [L] \\
B &= \text{seaward distance, } [L] \\
B &= \text{average bottom salinity (equation 4-8 only)} \\
B_i &= \text{constant coefficient for Crank-Nicolson method} \\
B'_i &= B_i - 2 \\
\text{BOD}_{20} &= \text{ultimate or 20-day BOD} \\
B(X) &= \text{coefficient array for Crank-Nicolson method} \\
\text{BOD} &= \text{concentration of biochemical oxygen demand} \\
C &= \text{salinity of the ocean or bay outside the estuary being studied (equation 4-8 only)}
\end{align*}
\]
C = concentration
C_6 = concentration in segment 6
C_A = concentration of constituent A
\( \bar{C}_A \) = time averaged concentration of constituent A
C'_A = perturbation component of the concentration of constituent A
C_{(k)}_{i,j} = concentration at location i,j, at time t (equation 4-17 only)
C_o = concentration at an outfall
C_{x,t} = concentration at location x, at time t
C_{x,z,t} = concentration at location x,z, at time t
CSTAR_{x,z} = intermediate concentration value at location x,z, at completion of the first phase of the implicit alternating-direction method
C_{SAT} = saturation value for dissolved oxygen
D = molecular diffusion coefficient, \([L^2T^{-1}]\)
D_A = molecular diffusion coefficient for constituent A, \([L^2T^{-1}]\)
D_i = constant coefficient for Crank-Nicolson method
D'_o = apparent diffusion coefficient at x = 0, the mouth of the estuary, \([L^2T^{-1}]\)
D_t = longitudinal coefficient of turbulent diffusion, \([L^2T^{-1}]\)
D'_x = apparent diffusion coefficient, \([L^2T^{-1}]\)
DO = dissolved oxygen concentration
DS = degree of stratification
D(X) = array for Crank-Nicolson method
e, exp = base of natural logarithms, equals 2.718..., usually raised to a power
erfc = error function

\[ E = \text{dispersion coefficient, generally for the x-direction, } [LT^{-1}] \]

\[ E_i = \text{turbulent diffusion component } [L^2T^{-1}] \]

\[ E'_{k,k+1} = \text{eddy exchange coefficient at interface between segment } k \text{ and segment } k+1, [L^3T^{-1}] \]

\[ E_x, E_y, E_z = \text{directional components of turbulent diffusion coefficient, } [L^2T^{-1}] \]

\[ E_x_{i,j} = \text{dispersion component in the x-direction at location } i,j, [L^2T^{-1}] \]

\[ E_y_{i,j} = \text{dispersion component in the y-direction at location } i,j, [L^2T^{-1}] \]

\[ EX = \text{dispersion coefficient for the x-direction, } [L^2T^{-1}] \]

\[ EX_{x,z} = \text{dispersion coefficient for the x-direction at location } x,z, \text{ at time } t, [L^2T^{-1}] \]

\[ EX_{x,z} = \text{dispersion coefficient for the x-direction at location } x,z, \text{ at time } t, [L^2T^{-1}] \]

\[ \text{EXPON1 = exponent expression occurring in an extrapolation technique} \]

\[ \text{EXPON2 = exponent expression occurring in an extrapolation technique} \]

\[ EZ = \text{dispersion coefficient for the z-direction, } [L^2T^{-1}] \]

\[ EZ_{x,z} = \text{dispersion coefficient for the z-direction at location } x,z, \text{ at time } t, [L^2T^{-1}] \]

\[ EZ_{x,z} = \text{dispersion coefficient for the z-direction at location } x,z, \text{ at time } t, [L^2T^{-1}] \]

\[ f = \text{bed resistance coefficient, [dimensionless]} \]

\[ Fo = \text{Froude number, [dimensionless]} \]

\[ g = \text{acceleration of gravity, } [LT^{-2}] \]

\[ G = \text{rate of energy dissipation per unit mass of fluid, } [L^2T^{-3}] \]
\(G(X)\) = coefficient array for Crank-Nicolson method

\(h\) = mean depth, [L]

\(H\) = tidal height, [L]

\(t_0\) = time zero, point in time at which tracer is released

\(H\) = water level elevation relative to the local mean sea level datum (equations 4-13 and 4-15 only), [L]

\(J\) = rate of gain of potential energy per unit mass of fluid, [L^2 T^{-3}]

\(J_k\) = direct sources of biochemical oxygen demand in segment \(k\); added concentration per unit time

\(K\) = frictional resistance term, [L^{-1}]

\(K\) = Van Dorn coefficient for wind stress (equations 4-13 and 4-14 only), [dimensionless]

\(k_d\) = decay rate; removal coefficient to the base e, [T^{-1}]

\(kD\) = decay rate to the base e, [T^{-1}]

\(kD_x\) = decay rate, to the base e, at location \(x\), [T^{-1}]

\(L_k\) = concentration of ultimate carbonaceous biochemical oxygen demand in estuary segment \(k\)

\(m\) = amount per unit cross-sectional area for an instantaneous release into a one-dimensional estuary

\(m\) = amount per unit cross-sectional area per unit time for a continuous release into a one-dimensional estuary

\(m\) = amount per unit depth for an instantaneous release into a two-dimensional estuary

\(N\) = number of points

\(\dot{N}\) = mass flux, [ML^{-2} T^{-1}]

\(\dot{N}_A\) = mass flux of constituent A across a boundary, [ML^{-2} T^{-1}]

\(N_x, N_y, N_z\) = directional components of the mass flux, [ML^{-2} T^{-1}]

\(NX\) = number of points in the x-direction
NZ = number of points in the z-direction

ppm = parts per million parts

\( P_t \) = tidal prism, \([L^3]\)

\( Q \) = flowrate, \([L^3T^{-1}]\)

\( Q \) = flowrate of water in pounds per day (equations 8-6 and 8-11 only)

\( Q \) = vector average of transport per unit width (equations 4-13 and 4-14 only), \([L^2T^{-1}]\)

\( Q_f \) = fresh water discharge, \([L^3T^{-1}]\)

\( Q_{k,k+1} \) = net flow across the interface between segment \( k \) and segment \( k+1 \), \([L^3T^{-1}]\)

\( r \) = time rate of production, \([ML^{-3}T^{-1}]\)

\( r_A \) = time rate of production of constituent \( A \), \([ML^{-3}T^{-1}]\)

\( R \) = rainfall rate, \([LT^{-1}]\)

\( s_{lws} \) = local salinity at low water slack

\( s_o \) = ocean salinity

\( S_i \) = term defined for Gaussian elimination method

SLUG = point concentration used for an instantaneous release

\( t \) = time, \([T]\)

\( T \) = tidal period, \([T]\)

\( u \) = net velocity, \([LT^{-1}]\)

\( u_0 \) = maximum flood tide velocity at \( x = 0 \), \([LT^{-1}]\)

\( U \) = advective velocity, generally in the x-direction, \([LT^{-1}]\)

\( U \) = vertically integrated x-component of transport per unit width (equations 4-13 to 4-15 only), \([L^2T^{-1}]\)

\( U_{i,j} \) = velocity component in the x-direction at location \( i,j \), \([LT^{-1}]\)
\( V = \text{advective velocity, generally in the y-direction, [LT}^{-1}] \)

\( V = \text{vertically integrated y-component of transport per unit width (equations 4-13 to 4-15 only), [L^2T}^{-1}] \)

\( V = \text{velocity vector, [LT}^{-1}] \)

\( V_i = \text{a time-averaged component of velocity, [LT}^{-1}] \)

\( V'_{i} = \text{perturbation component for turbulent velocity, [LT}^{-1}] \)

\( V_{i,j} = \text{velocity component in the y-direction at location i,j, [LT}^{-1}] \)

\( V'_{x} = \text{perturbation component in the x-direction for turbulent velocity, [LT}^{-1}] \)

\( V_x, V_y, V_z = \text{directional velocity components, [LT}^{-1}] \)

\( V_X = \text{velocity component in the x-direction, [LT}^{-1}] \)

\( V_{X_x,z} = \text{velocity component in the x-direction at location x,z, at time } t, [LT}^{-1}] \)

\( V_{X,\text{NEXT} x,z} = \text{velocity component in the x-direction at location x,z, at time } t + \Delta t, [LT}^{-1}] \)

\( V_Z = \text{velocity component in the z-direction, [LT}^{-1}] \)

\( V_{Z_x,z} = \text{velocity component in the z-direction at location x,z, at time } t, [LT}^{-1}] \)

\( V_{Z,\text{NEXT} x,z} = \text{velocity component in the z-direction at location x,z, at time } t + \Delta t, [LT}^{-1}] \)

\( \text{VOL}_{k} = \text{volume of segment } k \)

\( W = b(x,z) - a(x,z); \text{width of estuary at location } x,y, [L] \)

\( W = \text{wind speed 10 meters above the water (equations 4-13 and 4-14 only), [LT}^{-1}] \)

\( W = \text{loading rate in pounds per day (equations 8-6 to 8-11 only) \}

\( W_i = \text{collection of terms for Crank-Nicolson method} \)

\( W_x = \text{width at location } x, [L] \)
x = longitudinal direction or distance, [L]
y = lateral direction or distance, [L]
z = vertical direction or distance, [L]
Z = depth of water, [L]

Symbols

\( a_i \) = term defined for Gaussian elimination method
\( \Delta t \) = time increment, [T]
\( \Delta x \) = x-increment, [L]
\( \Delta y \) = y-increment, [L]
\( \Delta z \) = z-increment, [L]
\( \Theta \) = weighting factor for finite difference equation

\( \xi_{k,k+1} \) = advective coefficient dependent upon the ratio of dispersion to advective forces at interface between segment \( k \) and segment \( k+1 \), [dimensionless]

\( \pi = 3.14159 \ldots \)
\( \rho \) = mass density, [ML\(^{-3}\)]
\( \rho_A \) = mass density of a mixture with constituent A, [ML\(^{-3}\)]
\( \psi \) = angle between the wind velocity vector and the x-axis

Subscripts

1, 2, etc. = location
A = constituent A
d = decay
f = fresh water
i = a general index
i = location i
j = location j
k = segment designation in a segmented estuary

NX = value at location NX, the final grid point in the x-direction
NZ = value at location NZ, the final grid point in the z-direction

o = origin; x = o
o = outfall
o = ocean (equation 4-12 only)

t = tide (equation 4-10 only)
t = time
t = turbulence

x = the x-component; in the longitudinal direction

y = the y-component; in the lateral direction

z = the z-component; in the vertical direction

Superscripts

(k) = at time t
(k+1) = at time t + Δt

' = a perturbation component resulting from turbulent fluctuations

' = apparent value

' = dispersion term for Thomann method
APPENDIX III. - COMPUTER PROGRAMS AND DATA

IDEAL-I
IDEAL-II
MASSTRANS-I
MASSTRANS-II
OXTRANS-I
STABLE-I
STABLE-II
EXACT-I
EXACT-II
PROFILE-I
PROFILE-II
COMPUTER PROGRAM FOR

IDEAL-I

Object Code = 7912 bytes
Array Area = 1616 bytes
Total = 9528 bytes
**************  

*  
*I**DEAL-I  
*  
*ONE DIMENSIONAL MASS TRANSPORT MODEL*  
*  
*EXPLICIT FINITE DIFFERENCE METHOD*  
*  
*CONSTANT COEFFICIENTS*  
*  
*--------------------------------------*  
*  
*LOGIC AND PROGRAMMING BY*  
*  
*JONATHAN YOUNG*  
*ENVIRONMENTAL ENG. DIVISION*  
*CIVIL ENGINEERING DEPT.*  
*Texas A&M University*  
*COLLEGE STATION, TEXAS*  
*  
*--------------------------------------*  

THIS PROGRAM USES AN EXPLICIT FINITE DIFFERENCE METHOD TO ANALYZE MASS TRANSPORT IN A ONE DIMENSIONAL ESTUARY. THE MODEL IS USEFUL IN EVALUATING TRANSPORT IN A FLUME OR WELL-MIXED ESTUARY WHERE VELOCITY, DISPERSION, DECAY, AND CROSS-SECTIONAL AREA CAN BE CONSIDERED CONSTANT THROUGHOUT.
THIS PROGRAM WAS WRITTEN PRIMARILY TO TEST THE ACCURACY OF A FINITE DIFFERENCE METHOD AS COMPARED WITH ANALYTICAL, CLOSED FORM SOLUTIONS FOR THE ESTUARY MASS TRANSPORT EQUATION. REPRESENTATIVE VALUES FOR DISPERSION, VELOCITY, AND DECAY RATES SHOULD BE CHOSEN FOR THE ESTUARY BEING STUDIED. THESE VALUES CAN BE USED SEVERAL TIMES IN THIS MODEL TO DETERMINE WHICH TIME AND DISTANCE INCREMENTS GIVE SATISFACTORY ACCURACY. STABILITY CRITERIA IS PRINTED BY THE PROGRAM AND SEVERE INACCURACIES WILL RESULT IF THESE LIMITS ARE EXCEEDED.

THE PROGRAM CAN ANALYZE INSTANTANEOUS RELEASES, CONTINUOUS RELEASES, OR STEADY-STATE PROFILES. EXPONENTIAL EXTRAPOLATION AT THE BOUNDARIES IS USED FOR INSTANTANEOUS RELEASES. FOR OTHER TYPES OF LOADING, A CONSTANT SLOPE EXTRAPOLATION IS USED.

INFORMATION REGARDING THIS COMPUTER PROGRAM CAN BE OBTAINED FROM JONATHAN YOUNG AT HYDROSCIENCE, INC., WESTWOOD, NEW JERSEY.
***INPUT VARIABLES***

N = NUMBER OF POINTS
DELT = TIME INCREMENT (DAYS)
DELX = DISTANCE INCREMENT (MILES)
E = DISPERSION COEFFICIENT (MILES**2/DAY)
XD = DECAY RATE (1/DAY)
U = VELOCITY (MILES/DAY)
NC = NUMBER OF INITIAL CONCENTRATIONS
NS = NUMBER OF SOURCES (MAXIMUM OF 2)
LTYPE = TYPE OF LOADING
1 MEANS CONSTANT CONCENTRATION AT OUTFALL
( CAN BE USED FOR STEADY-STATE CALCULATIONS )
2 MEANS AN INSTANTANEOUS RELEASE
3 MEANS CONTINUOUS LOADING
NDTPR = PRINTING INCREMENT FOR TIME STEPS (INTEGER)
NSKP = PRINTING INCREMENT FOR DISTANCE STEPS (INTEGER)
TBEGIN = TIME OF FIRST CALCULATION (DAYS)
TSTOP = TIME OF FINAL CALCULATION (DAYS)
JJ = POINT OF AN INITIAL CONCENTRATION
LOAD(JJ) = INITIAL CONCENTRATION
MASS = MASS RATE BEING ADDED AT OUTFALL (OPTIONAL)
***OTHER VARIABLES***

MSU = POINT OF PEAK OF UPPERMOST SOURCE
MSL = POINT OF PEAK OF LOWERMOST SOURCE
UPPER = MILEAGE AT MSU
LOWER = MILEAGE AT MSL
UMAX = CONCENTRATION AT MSU
LMAX = CONCENTRATION AT MSL
NSEG = NUMBER OF SEGMENTS (=N-1)
COUNT = A COUNTER TO COMPARE WITH PRINTING INCREMENT
X(IJ) = DISTANCES TO BE PRINTED
NN1, IJ, K, L = DO LOOP INDICES
QUIT = NUMBER OF TIME INCREMENTS
NOMORE = DO LOOP PARAMETER (=QUIT+1)
T = TIME (DAYS)
COEF1 = DISPERSION TERM COEFFICIENT
COEF2 = VELOCITY TERM COEFFICIENT
COEF3 = DECAY TERM COEFFICIENT
CT(K) = CONCENTRATION AT PREVIOUS TIME INCREMENT
CT1(K) = CONCENTRATION BEING CALCULATED
EXPON = EXPONENT FOR BOUNDARY CONDITION CALCULATIONS
XUP = ORIGINAL LOCATION OF UPPERMOST PEAK OR OUTFALL
XLOW = ORIGINAL LOCATION OF LOWERMOST PEAK OR OUTFALL
STABLX = MAXIMUM X INCREMENT TO INSURE STABILITY
STABLT = MAXIMUM TIME INCREMENT TO INSURE STABILITY
CNTINU = SUBROUTINE TO CALCULATE CONTINUOUS LOADING VALUES
NCALC = APPROXIMATE NUMBER OF PROGRAM CALCULATIONS
INDEX = N*NOMORE
REAL LOAD, LOWER, LMAX, MASS
INTEGER COUNT
DIMENSION CT(101), CT1(101), LOAD(101), X(101)

C
C----------------------------------------
C ***INITIALIZE TERMS AND READ INPUT DATA***
C----------------------------------------

DATA CT, CT1, LOAD, X/404*0.0/
READ, N, DELT, DELX, E, XKO, U, NC, NS, LTYPE, NDTPR, NSkip, TBEGIN, TSTOP
PRINT, N, DELT, DELX, E, XKO, U, NC, NS, LTYPE, NDTPR, NSkip, TBEGIN, TSTOP
DO 3 J=1, NC
READ, JJ, LOAD(JJ)
CT(JJ)=LOAD(JJ)
3 CONTINUE
IF(LTYPE.NE.3) GO TO 4
READ, MASS
4 CONTINUE
NSEG=N-1
MSU=1
MSL=N
UPPER=0.0
LOWER=(N-1)*DELX
UMAX=LOAD(1)
LMAX=LOAD(N)

C
C----------------------------------------
C ***ESTABLISH THE LOCATIONS OF THE UPPERMOST
C AND LOWERMOST WASTE SOURCES***
C----------------------------------------
DO 6 I=1,NSEG
   IF(LOAD(I+1).LT.LOAD(I)) GO TO 7
   MSU=I+1
   UPPER=(I)*DELX
   UMAX=LOAD(I+1)
6 CONTINUE
7 DO 8 I=1,NSEG
   IF(LOAD(N-I).LT.LOAD(N+1-I)) GO TO 9
   MSL=N-I
   LOWER=(MSL-1)*DELX
   LMAX=LOAD(N-I)
8 CONTINUE
9 CONTINUE

C------------------------------------------------------------------
C ***DETERMINE THE INITIAL VALUE OF COUNT***
C------------------------------------------------------------------
COUNT=(TBEGIN/DELT + 0.01)
COUNT=COUNT-(COUNT/NDTPR)*NDTPR
C------------------------------------------------------------------
C ***DETERMINE STABILITY CRITERIA***
C------------------------------------------------------------------
STBLX=DELX
STBLT=DELT
IF(E.GT.0.0.AND.U.GT.0.0) STBLX=2.*E/U
TERM=2.*E+DELX*DELX*XXD
IF(TERM.NE.0.0) STBLT=(DELX*DELX)/TERM
C
C------------------------------------------
C ***PRINT DATA VALUES***
C------------------------------------------
854 PRINT 10
855  10 FORMAT(1H1)
856 PRINT 11
857  11 FORMAT(1H1,///, 15X, '*****DISPERSION CALCULATIONS BY THE EXPLICI
858 $T METHOD*****', ///)
859 PRINT 12, N
860  12 FORMAT(22X, ' NUMBER OF POINTS = ',13,///)
861 SECS=DELT*86400.
862 PRINT 13, DELT,SECS
863  13 FORMAT(25X, 'TIME INCREMENT = ',F7.4, ' DAYS ( ', F8.2, ' SECONDS )
864 $',///)
865 FEET=DELT*5280.
866 PRINT 14, DELT, FEET
867  14 FORMAT(21X, 'DISTANCE INCREMENT = ', F7.3, ' MILES ( ', F8.2, ' FE
868 $ET )',///)
869 EFEE=E*5280.*5280./86400.
870 PRINT 15, E,EFEE
871  15 FORMAT(17X, 'DISPERSION COEFFICIENT = ',F7.2, ' MILES SQUARED/DAY
872 $$(', F7.2, ' FEET SQUARED/SEC )',///)
873 PRINT 16, XKD
874  16 FORMAT(22X, 'DECAY COEFFICIENT = ', F7.2, ' PER DAY',///)
875 UF=U*5280./86400.
876 PRINT 17, U,UF
878 $ ')',///)
879 PRINT 18, NC
880  18 FORMAT(1X, 'NUMBER OF KNOWN INITIAL CONCENTRATIONS = ', I3,///)
876 PRINT 185, NS
877 185 FORMAT(22X, 'NUMBER OF SOURCES = ', I3, /)
878 PRINT 19, LTYPE
879 19 FORMAT(16X, 'TYPE OF LOADING (LTYPE) = ', I2, /)
880 DTPR=NDTPR*DELT
881 WRITE(6,20)DTPR
882 20 FORMAT(16X, 'PRINTOUT TIME INCREMENT = ', F7.4, ' DAYS', /)
883 DXPR=NSKIP*DELX
884 WRITE(6,22) DXPR
885 22 FORMAT(12X, 'PRINTOUT DISTANCE INCREMENT = ', F7.4, ' MILES', /)
886 PRINT 221, STABLT
887 221 FORMAT(13X, 'MAXIMUM STABLE X INCREMENT = ', F7.3, ' MILES', /)
888 PRINT 222, STABLT
889 222 FORMAT(13X, 'MAXIMUM STABLE T INCREMENT = ', F7.4, ' DAYS', /)
890 PRINT 23, TBEGIN, TSTOP
891 23 FORMAT(26X, 'RANGE OF TIME = ', F5.3, ' TO ', F8.3, ' DAYS', /, 1H1)
892 T=TBEGIN
893 WRITE(6,24)
894 24 FORMAT(1X, 119('*'), /)

C
C********************************************************
C ***CALCULATE DISTANCES TO BE PRINTED***
C********************************************************

895 DO 25 IJ=1,N,NSKIP
896 X(IJ)=(IJ-1)*DELX
897 25 CONTINUE
898 NN1=1+NSKIP

C
C********************************************************
C ***PRINT DISTANCES***
C********************************************************
WRITE(6,27) (X(IJ), IJ=NN1,N,NSKI)

WRITE(6,24)

***PRINT INITIAL CONCENTRATIONS***

WRITE(6,32)
WRITE(6,35)T,(CT(J),J=1,N,NSKI)

***CALCULATE CONSTANT COEFFICIENTS FOR THE FINITE DIFFERENCE EQUATION***

COEF1=E*DELT/(DELEX**2.)
COEF2=U*DELT/(2.*DELEX)
COEF3=XKD*DELT
QUIT=(TSTOP-TBEGIN)/DELT
NOMORE=QUIT+1

***BEGIN LOOP WHICH IS REPEATED FOR EACH TIME INCREMENT***

DO 200 JJ=1,NOMORE
T=T+DELT
COUNT=COUNT+1
C

**APPLY EXPLICIT FINITE DIFFERENCE EQUATION
TO EACH INTERNAL POINT**

DO 50 K=2,NSEG
CT1(K)=COEF1*(CT(K+1)-2*CT(K)+CT(K-1))-COEF2*(CT(K+1)-CT(K-1))-COE
2F3*CT(K)*CT(K)
50 CONTINUE

C

**DETERMINE IF CURVE IS TO BE EXTENDED UPSTREAM AND
APPLY THE APPROPRIATE BOUNDARY CONDITIONS**

IF(MSU.EQ.1.AND.LTYPE.EQ.1) GO TO 55
IF(LTYPE.NE.2) GO TO 53
XUP=UPPER-U*TBEGIN
EXPON= (+2.*(-XUP+DELX)*DELX-(DELX**2.)) - 2.*DELX*U*T) /
$ (4.*E*T)
CT1(1)=CT1(2)*EXP(EXPON)
53 CONTINUE
CT1(1)=2.*CT1(2)-CT1(3)
54 CONTINUE
IF(CT1(3).LE.0.001) GO TO 55
IF(CT1(1).LT.0.0) CT1(1)=CT1(2)*CT1(2)/CT1(3)
55 CONTINUE
IF(CT1(1).LT.0.0) CT1(1)=0.0

C

**DETERMINE IF CURVE IS TO BE EXTENDED DOWNSRETAM AND
APPLY THE APPROPRIATE BOUNDARY CONDITIONS**
IF(MSL.EQ.N.AND.LTYPE.EQ.1) GO TO 59
IF(LTYPE.NE.2) GO TO 57
XLOW=LOWER-U*TBEGIN
EXPON= (-2.*(N-2)*DELF-XLCW ) * DELF- (DELF**2.) + 2.*DELF*U*T)/
$ \quad (4.*E*T)
CT1(N)=CT1(N-1)*EXP(EXPON)
GO TO 58
57 CONTINUE
CT1(N)=2.*CT1(N-1)-CT1(N-2)
58 CONTINUE
IF(CT1(N-2).LE.0.001) GO TO 59
IF(CT1(N).LT.0.0) CT1(N)=CT1(N-1)*CT1(N-1)/CT1(N-2)
59 CONTINUE
IF(CT1(N).LT.0.0) CT1(N)=0.0

C
C-----------------------------------------------
C***APPLY APPROPRIATE LOADING CONDITIONS***
C-----------------------------------------------

IF(LTYPE.EQ.1) GO TO 60
IF(LTYPE.EQ.3) CALL CONTINU(E,T,MAS,CT1(MSU))
GO TO 70
60 CONTINUE
CT1(MSU)=UMAX
CT1(MSL)=LMAX
70 CONTINUE
IF(COUNT.GE.NDTPR) GO TO 75
GO TO 100

C
C-----------------------------------------------
C***PRINT CONCENTRATIONS FOR EACH DESIRED TIME INCREMENT***
C-----------------------------------------------

75 WRITE(6,80) T,(CT1(M),M=1,N,NSKIP)
80 FORMAT(/, F8.3, F10.4, 10F10.4, /, 1(18X, 10F10.4))
COUNT=0
100 DO 150 L=1,N
150 IF(CT1(L).LT.0.001) CT1(L)=0.0
150 CT(L)=CT1(L)
200 CONTINUE
1000 CONTINUE

C

***DETERMINE THE NUMBER OF CALCULATIONS MADE BY THE PROGRAM

NCALC=3*NS + 10*NSEG + 2*N/NSKP + NOMORE*(5*NSEG + 32)
INDEX=N*NOMORE
PRINT 205, N,NOMORE,NCALC, INDEX
205 FORMAT(///, 1X, 'N=', I3, 5X, 'NOMORE=', I5, 5X, 'NCALC=', I7,
$ 5X, 'INDEX=', I6,///)
STOP
END

SUBROUTINE CNTINU(EX TIME XMASS CONCIN)

C

THIS SUBROUTINE SHOULD BE WRITTEN ACCORDING TO THE
PARTICULAR LOADING RATE OF THE GUTFALL BEING STUDIED.
THE SUBROUTINE WRITTEN BELOW IS VALID ONLY IN THE CASE OF
ONE GUTFALL AND WHERE NO VELOCITY IS PRESENT.

THIS SUBROUTINE IS USED ONLY IF LTYPE EQUALS 3.

C
C

RAD=3.14159*EX
CONCIN=(XMASS*TIME**.5)/SQRT(RAD)
RETURN
END
Input Data for IDEAL-I

Each line represents a new card unless single spaced.

51, 0.002, 0.10, 1., 0., 5., 1, 1, 2, 25, 1, 0., 0.6
11
18939.4
COMPUTER PROGRAM FOR
IDEAL-II

Object Code = 10,968 bytes
Array Area = 2,828 bytes
Total = 13,796 bytes
**IDEAL-II**

**ONE DIMENSIONAL MASS TRANSPORT MODEL**

**IMPLICIT FINITE DIFFERENCE METHOD**

**CONSTANT COEFFICIENTS**

**--------------------------------**

**LOGIC AND PROGRAMMING BY**

**JONATHAN YCUNG**

**ENVIRONMENTAL ENG. DIVISION**

**CIVIL ENGINEERING DEPT.**

**TEXAS A&M UNIVERSITY**

**COLLEGE STATION, TEXAS**

**--------------------------------**

THIS PROGRAM USES AN CRANK-NICOLSON IMPLICIT FINITE
DIFFERENCE METHOD TO ANALYZE MASS TRANSPORT IN A ONE DIMEN-
SIONAL ESTUARY. THE MODEL IS USEFUL IN EVALUATING TRANSPORT IN
A FLUME OR WELL-MIXED ESTUARY WHERE VELOCITY, DISPERSION,
DECAY, AND CROSS-SECTIONAL AREA CAN BE CONSIDERED CONSTANT
THROUGHOUT.
THIS PROGRAM WAS WRITTEN PRIMARILY TO TEST THE ACCURACY
OF A FINITE DIFFERENCE METHOD AS COMPARED WITH ANALYTICAL
SOLUTIONS FOR THE ESTUARY MASS TRANSPORT EQUATION.

REPRESENTATIVE VALUES FOR THE ESTUARY, MASS TRANSPORT, AND DECAY RATES
SHOULD BE CHOSEN FOR THE ESTUARY BEING STUDIED. THESE VALUES
CAN BE USED SEVERAL TIMES IN ORDER TO DETERMINE WHICH
STABILITY CRITERIA IS PRINTED BY THE PROGRAM AND SEVERE
INACCURACIES MAY RESULT IF THESE LIMITS ARE EXCEEDED.

THE PROGRAM CAN ANALYZE INSTANTANEOUS RELEASES OR CONTINUOUS
RELEASES, OR STEADY-STATE PROFILES. EXPONENTIAL EXTRAPOLATION
AT THE BOUNDARIES IS USED FOR INSTANTANEOUS RELEASES. FOR
OTHER TYPES OF LOADING, A CONSTANT SLOPE EXTRAPOLATION IS USED.

INFORMATION REGARDING THIS COMPUTER PROGRAM CAN BE
OBTAINED FROM JONATHAN YOUNG AT HYDROSCIENCE, INC., WESTWOOD,
NEW JERSEY.
***INPUT VARIABLES***

N = NUMBER OF POINTS
DELT = TIME INCREMENT (DAYS)
DELX = DISTANCE INCREMENT (MILES)
E = DISPERSION COEFFICIENT (MILES/DAY)
XKD = DECAY RATE (/DAY)
U = VELOCITY (MILES/DAY)
NC = NUMBER OF INITIAL CONCENTRATIONS
NS = NUMBER OF SOURCES (MAXIMUM OF 2)
LTYPE = TYPE OF LOADING
1 MEANS CONSTANT CONCENTRATION AT OUTFALL
   (CAN BE USED FOR STEADY-STATE CALCULATIONS)
2 MEANS AN INSTANTANEOUS RELEASE
3 MEANS CONTINUOUS LOADING
NDTPR = PRINTING INCREMENT FOR TIME STEPS (INTEGER)
NSKIP = PRINTING INCREMENT FOR DISTANCE STEPS (INTEGER)
TBEGIN = TIME OF FIRST CALCULATION (DAYS)
TSTOP = TIME OF FINAL CALCULATION (DAYS)
JJ = POINT OF AN INITIAL CONCENTRATION
LOAD(JJ) = INITIAL CONCENTRATION
MASS = MASS RATE BEING ADDED AT OUTFALL (OPTIONAL)
***OTHER VARIABLES***

MSU = POINT OF PEAK OF UPPERMOST SOURCE
MSL = POINT OF PEAK OF LOWERMOST SOURCE
UPPER = MILEAGE AT MSU
LOWER = MILEAGE AT MSL
UMAX = CONCENTRATION AT MSU
LMAX = CONCENTRATION AT MSL
NSEG = NUMBER OF SEGMENTS (=N-1)
COUNT = A COUNTER TO COMPARE WITH PRINTING INCREMENT
X[IJ] = DISTANCES TO BE PRINTED
QUIT = NUMBER OF TIME INCREMENTS
NOMORE = DO LOOP PARAMETER (=QUIT+1)
T = TIME (DAYS)
A,B*,D = COEFFICIENTS FOR SOLUTION MATRIX
W(M) = RIGHT SIDE ARRAY FOR SOLUTION MATRIX
ALPHA(M) = INTERMEDIATE ARRAY FOR MATRIX SOLUTION
S(M) = INTERMEDIATE ARRAY FOR MATRIX SOLUTION
CT(K) = CONCENTRATION AT PREVIOUS TIME INCREMENT
CTL(K) = CONCENTRATION BEING CALCULATED
EXPON = EXPONENT FOR BOUNDARY CONDITION CALCULATIONS
XUP = ORIGINAL LOCATION OF UPPERMOST PEAK OR OUTFALL
XLOW = ORIGINAL LOCATION OF LOWERMOST PEAK OR OUTFALL
STABLX = MAXIMUM X INCREMENT TO INSURE STABILITY
STABLT = MAXIMUM TIME INCREMENT TO INSURE STABILITY
CNTINU = SUBROUTINE TO CALCULATE CONTINUOUS LOADING VALUES
NCALC = APPROXIMATE NUMBER OF PROGRAM CALCULATIONS
INDEX = N*NOMORE
REAL LOAD, LOWER, LMAX, MASS
INTEGER COUNT
DIMENSION CT(101), CL(101), LOAD(101), X(101), W(101), ALPHA(101), $ S(101)

C

------------------------------------------------------------------------
C
**INITIALIZE TERMS AND READ INPUT DATA**
C------------------------------------------------------------------------
DATA CT, CL, LOAD, X, W, ALPHA, S/707*0.0/
READ, N, DELT, DELX, E, XKD, U, NC, NS, LTYPE, NDTPR, NSKIP, TBEGIN, TSTOP
PRINT, N, DELT, DELX, E, XKD, U, NC, NS, LTYPE, NDTPR, NSKIP, TBEGIN, TSTOP
DO 3 J=1, NC
READ, JJ, LOAD(JJ)
CT(JJ)=LOAD(JJ)
3 CONTINUE
IF(LTYPE.NE.3) GO TO 4
READ, MASS
4 CONTINUE
NSEG=N-1
MSU=1
MSL=N
UPPER=0.0
LOWER=(N-1)*DELX
UMAX=LOAD(1)
LMAX=LOAD(N)

C

------------------------------------------------------------------------
C
**ESTABLISH THE LOCATIONS OF THE UPPERMOST
C AND LOWERMOST WASTE SOURCES**
C------------------------------------------------------------------------
3017      DO 6 I=1,NSEG
3018      IF(LOAD(I+1).LT.LOAD(I)) GO TO 7
3019      MSU=I+1
3020      UPPER(I)=DELX
3021      UMAX=LOAD(I+1)
3022      6 CONTINUE
3023      DO 8 I=1,NSEG
3024      IF(LOAD(N-I).LT.LOAD(N*1-I)) GO TO 9
3025      MSL=N-I
3026      LOWER=(MSL-1)*DELX
3027      LMAX=LOAD(N-I)
3028      8 CONTINUE
3029      9 CONTINUE

C-----------------------------------------------
C ***DETERMINE THE INITIAL VALUE OF COUNT***
C-----------------------------------------------
3030      COUNT=(TBEGIN/DELT + 0.01)
3031      COUNT=COUNT-(COUNT/NDTPr)*NDTPr

C-----------------------------------------------
C ***DETERMINE STABILITY CRITERIA***
C-----------------------------------------------
3032      STABLX=DELX
3033      STABLT=DELT
3034      IF(E.GT.0.0.AND.U.GT.0.0) STABLX=2.*E/U
3035      TERM=2.*E+DELX*DELX*KD
3036      IF(TERM.NE.0.0) STABLT=2.*(DELX*DELX)/TERM
C
C ***PRINT DATA VALUES***
C

3037     PRINT 10
3038     10 FORMAT(1H1)
3039       PRINT 11
3040     11 FORMAT(1H1, ///, 15X, '******DISPERSION CALCULATIONS BY THE CRANK-N
3041       SCOLSON IMPLICIT METHOD******', ///)
3042     PRINT 12, N
3043     12 FORMAT(22X, 'NUMBER OF POINTS = ', I3, //)
3044       SECS=DELT*86400.
3045     PRINT 13, DELT, SECS
3046     13 FORMAT(25X, 'TIME INCREMENT = ', F7.4, ' DAYS ( ', F8.2, ' SECONDS )
3047       $', //)
3048       FEET=DELT*5280.
3049     PRINT 14, DELT, FEET
3050     14 FORMAT(21X, 'DISTANCE INCREMENT = ', F7.3, ' MILES ( ', F8.2, ' FE
3051       $ET )', //)
3052       EFEEF=E*5280.*5280.*86400.
3053     PRINT 15, E, EFEEF
3054     15 FORMAT(17X, 'DISPERSION COEFFICIENT = ', F7.2, ' MILES SQUARED/CAY
3055       $(', F7.2, ' FEET SQUARED/SEC )', //)
3056     PRINT 16, XKD
3057     16 FORMAT(22X, 'DECAY COEFFICIENT = ', F7.2, ' PER DAY', //)
3058     UF=U*5280.*86400.
3059     PRINT 17, U, UF
3061       $ )', //)
3062     PRINT 18, NC
3063     18 FORMAT(1X, 'NUMBER OF KNOWN INITIAL CONCENTRATIONS = ', I3, //)
3059 PRINT 185, NS
3060 185 FORMAT(22X, 'NUMBER OF SOURCES = ', I3,/) 
3061 PRINT 19, LTYPE
3062 19 FORMAT(16X, 'TYPE OF LOADING (LTYPE) = ', I2,/) 
3063 DTPR=NOTPR*DELT
3064 WRITE(6,20)DTPR
3065 20 FORMAT(16X, 'PRINTOUT TIME INCREMENT = ', F7.4, ' DAYS',/) 
3066 DXPR=NSKP*DELX
3067 WRITE(6,22) DXPR
3068 22 FORMAT(12X, 'PRINTOUT DISTANCE INCREMENT = ', F7.4, ' MILES',/) 
3069 PRINT 221, STABLX
3070 221 FORMAT(13X, 'MAXIMUM STABLE X INCREMENT = ', F7.3, ' MILES',/) 
3071 PRINT 222, STABLT
3072 222 FORMAT(13X, 'MAXIMUM STABLE T INCREMENT = ', F7.4, ' DAYS',/) 
3073 PRINT 23, TBEGIN, TSTOP
3074 23 FORMAT(26X, 'RANGE OF TIME = ', F5.3, ' TO ', F8.3, ' DAYS',/,1H1)
3075 T=TBEGIN
3076 WRITE(6,24)
3077 24 FORMAT(1X, 119(***)/,)

C........................................................................
C ***CALCULATE DISTANCES TO BE PRINTED***
C........................................................................

3078 DO 25 IJ=1,N,NSKP
3079 X(IJ)=(IJ-1)*DELX
3080 25 CONTINUE
3081 NN1=1+NSKP

C........................................................................
C ***PRINT DISTANCES***
C........................................................................
WRITE(6,27) (X(IJ), IJ=NN1,N,NSKIP)
27 FORMAT('  X VALUES IN MILES', 10F10.2,1(18X,10F10.2))
WRITE(6,24)

***PRINT INITIAL CONCENTRATIONS***

WRITE(6,32)
32 FORMAT(4X, 'TIME', 45X, 'CONCENTRATIONS (PPM)', 15X, 4F9.1, 2X, 4F9.1, 2X)
WRITE(6,35)T,(CT(J),J=1,N,NSKIP)
35 FORMAT(18X,10F10.4,1(18X,10F10.4))

***CALCULATE CONSTANT COEFFICIENTS FOR THE FINITE DIFFERENCE EQUATION***

A=E*DELT/(2.*DELT**2.1)+U*DELT/(4.*DELT)
B1=E*DELT/(DELT**2.)+XKD*DELT/2.+1.
D=E*DELT/(2.*DELT**2.)-U*DELT/(4.*DELT)
QUIT=(TSTOP-TBEGIN)/DELT
NOMORE=QUIT+1

***BEGIN LOOP WHICH IS REPEATED FOR EACH TIME INCREMENT***

DO 200 JK=1,NOMORE
T=T+DELT
COUNT=COUNT+1
***SET THE UPPER BOUNDARY CONDITIONS***

Statement Number 36 for exponential extrapolation
Statement Number 37 for constant slope extrapolation
Statement Number 39 for constant concentration at the boundary

3098 IF(MSU.EQ.1.AND.LTYPE.EQ.1) GO TO 39
3099 IF(LTYPE.NE.2) GO TO 37
3100 36 CONTINUE
3101 XUP=UPPER-U*TBEGIN
3102 EXPON= (+2.*(-XUP +DELX) * DELX-(DELX**2.) - 2.*DELX*U*T) /
       $ (4.*E*T)
3103 IF(EXPON.LT.-170.) EXPCN=-170.
3104 ALPHA(1)=-A*EXP(EXPON)+B1
3105 W(1)=A*CT(1) -B2*CT(2)+D*CT(3)
3106 ALPHA(2)=B1-A*D/ALPHA(1)
3107 GO TO 41
3108 37 CONTINUE
3109 ALPHA(1)=-2.*A+B1
3110 W(1)=A*CT(1)-B2*CT(2)+D*CT(3)
3111 ALPHA(2)=B1-A*(D-A)/ALPHA(1)
3112 GO TO 41
3113 39 CONTINUE
3114 ALPHA(1)=B1
3115 W(1)=A*2.*CT(1)-B2*CT(2)+D*CT(3)
3116 ALPHA(2)=B1-A*D/ALPHA(1)
3117 41 CONTINUE
**Build Arrays for Matrix Solution**

```c
3118  MOST=NSEG-2
3119  DO 40 M=2,MOST
3120  W(M)=A*CT(M)-B2*CT(M+1)+D*CT(M+2)
3121  40 CONTINUE
```

**Solve Tridiagonal Matrix**

```c
3122  S(1)=W(1)
3123  S(2)=W(2)+A*S(1)/ALPHA(1)
3124  DO 42 MM=3,MOST
3125  ALPHA(MM)=B1-A*D/ALPHA(MM-1)
3126  S(MM)=W(MM)+A*S(MM-1)/ALPHA(MM-1)
3127  42 CONTINUE
```

**Set Lower Boundary Conditions**

```c
3128  IF(LTYPE.EQ.1.AND.MSL.EQ.N) GO TO 43
3129  IF(LTYPE.NE.2) GO TO 44
3130  421 CONTINUE
3131  XLOW=LOWER-U*TBEGIN
```
EXPO N = (-2. * (N-2) * DELX-XLOW) * DELX - (DELX**2.) + 2.*DELX*U*T) / 
(4.*E*T)

IF (EXPO N.LT.-170.) EXPO N = -170.
B1F = B1-D*EXP(EXPO N)
ALPHA(NSEG-1) = B1F-A*D/ALPHA(NSEG-2)
W(NSEG-1) = A*CT(NSEG-1)-B2*CT(NSEG) + D*CT(NSEG+1)
S(NSEG-1) = W(NSEG-1) + A*S(NSEG-2)/ALPHA(NSEG-2)
GO TO 45

43 CONTINUE
ALPHA(NSEG-1) = B1-A*D/ALPHA(NSEG-2)
W(NSEG-1) = A*CT(NSEG-1)-B2*CT(NSEG) + D*CT(NSEG+1)
S(NSEG-1) = W(NSEG-1) + A*S(NSEG-2)/ALPHA(NSEG-2)
GO TO 45

44 CONTINUE
ALPHA(NSEG-1) = (B1-2.*D) + (D-A)*D/ALPHA(NSEG-2)
W(NSEG-1) = A*CT(NSEG-1)-B2*CT(NSEG) + D*CT(NSEG+1)
S(NSEG-1) = W(NSEG-1) - (D-A)*S(NSEG-2)/ALPHA(NSEG-2)

45 CONTINUE
CT1(NSEG) = S(NSEG-1)/ALPHA(NSEG-1)
NFINAL = NSEG-2
DO 46 II = 1,NFINAL

CT1(NSEG-II) = (S(NSEG-II-1)+D*CT1(NSEG-II+1))/ALPHA(NSEG-II-1)

46 CONTINUE

C-----------------------------------------------
C  ***DETERMINE IF CURVE IS TO BE EXTENDED UPSTREAM AND
C  APPLY THE APPROPRIATE BOUNDARY CONDITIONS***
C-----------------------------------------------

IF (MSU.EQ.1.AND.LTYPE.EQ.1) GO TO 55
IF (LTYPE.NE.2) GO TO 53
XUP = UPPER-U*TBEGIN
3157 EXPON = (+2.*(-XUP +DELX) * DELX-(DELX**2.*) - 2.*DELX*U*T) /
$ (4.*E*T)
3158 CT1(1)=CT1(2)*EXP(EXPON)
3159 GO TO 55
3160 53 CONTINUE
3161 CT1(1)=2.*CT1(2)-CT1(3)
3162 55 CONTINUE

C --------------------------------------------------------------
C ***DETERMINE IF CURVE IS TO BE EXTENDED DOWNSTREAM AND APPL
C --------------------------------------------------------------
3163 IF(MSL**EQ.N AND LTYPE**EQ.1) GO TO 59
3164 IF(LTYPE**NE.2) GO TO 57
3165 XLOW=LOWER-U*TBEGIN
3166 EXPON = (-2.*((N-2)*DELX-XLOW) * DELX-(DELX**2.*) * 2.*DELX*U*T) /
$ (4.*E*T)
3167 CT1(N)=CT1(N-1)*EXP(EXPCN)
3168 GO TO 59
3169 57 CONTINUE
3170 CT1(N)=2.*CT1(N-1)-CT1(N-2)
3171 59 CONTINUE

C --------------------------------------------------------------
C ***APPLY APPROPRIATE LOADING CONDITIONS***
C --------------------------------------------------------------
3172 IF(LTYPE**EQ.1) GO TO 60
3173 IF(LTYPE**EQ.3) CALL CNINTU(E,T,MASS,CT1(MSU))
3174 GO TO 70
3175 60 CONTINUE
CTI(MSU)=UMAX
CTI(MSL)=LMAX
70 CONTINUE
DO 73 L=1,N
IF(CTI(L).LT.0.001) CTI(L)=0.0
CTI(L)=CTI(L)
73 CONTINUE
IF(COUNT.GE.NDTPR) GO TO 75
GO TO 200
C
C---------------------------------------------------------------
C
C                ***PRINT CONCENTRATIONS FOR EACH DESIRED TIME INCREMENT***
C---------------------------------------------------------------
75 WRITE(6,80) T,(CTI(M),M=1,N,NSkip)
80 FORMAT(//,F8.3,F10.4,10F10.4,/,1(18X,10F10.4))
COUNTr=0
200 CONTINUE
1000 CONTINUE
C
C---------------------------------------------------------------
C
C                ***DETERMINE THE NUMBER OF CALCULATIONS MADE BY THE PROGRAM***
C---------------------------------------------------------------
NCALC=3*NS + 10*NSEG + 2*N/NSkip + NOMORE*(10*NSEG+46)
INDEX=N*NOMORE
PRINT 205, N,NOMORE,NCALC, INDEX
205 FORMAT(///,1X,'N=',13,5X,'NOMORE=',15,5X,'NCALC=',17,
$      5X,'INDEX=',16,///)
STOP
END
SUBROUTINE CNTINU(EX,TIME,XMASS,CONCIN)

This subroutine should be written according to the particular loading rate of the outfall being studied. The subroutine written below is valid only in the case of one outfall and where no velocity is present.

This subroutine is used only if LTYPE equals 3.

RAD=3.14159*FX
CONCIN=(XMASS*TIME**.5)/SQRT(RAD)
RETURN
END
Input Data for IDEAL-II

Each line represents a new card unless single spaced.

15, 0.002, 0.1, 1., .23, 5.0, 2, 2, 1, 1, 1, 0.0, 0.022
4
1000.
15
1000.
COMPUTER PROGRAM FOR
MASSTRANS-I

Object Code = 54,728 bytes
Array Area = 40,196 bytes
Total = 94,924 bytes
*****

MASSTRANS-I

ESTUARY MASS TRANSPORT MODEL

TWO DIMENSIONAL NUMERICAL ANALYSIS - EXPLICIT METHOD

*****

LOGIC AND PROGRAMMING BY

P. JONATHAN YOUNG

C/O ENVIRONMENTAL ENGINEERING DIVISION

CIVIL ENGINEERING DEPARTMENT

TEXAS A&M UNIVERSITY

COLLEGE STATION, TEXAS 77840

*****
THIS COMPUTER PROGRAM WAS DESIGNED PRIMARILY TO ANALYZE THE
MASS TRANSPORT OF DISSOLVED MATERIALS IN A PARTIALLY STRATIFIED
ESTUARY. THESE MATERIALS MAY BE CONSERVATIVE OR NON-CONSERVATIVE.
AN EXPLICIT FINITE DIFFERENCE APPROXIMATION TO THE ESTUARY MASS
TRANSPORT EQUATION FORMS THE BASIS OF THE COMPUTATIONS.

THIS PROGRAM WAS DEVELOPED TO ANALYZE ESTUARIES WHOSE
CHARACTERISTICS DO NOT VARY SIGNIFICANTLY WITH WIDTH. THE WIDTH
OF THE ESTUARY MAY BE VARIED THROUGHOUT BUT THE CONCENTRA-
TIONS OF DISSOLVED MATERIALS AT EACH CROSS SECTION ARE CONSIDERED
TO BE UNCHANGING IN THE LATERAL DIRECTION. ALL PHYSICAL AND
HYDRODYNAMIC CHARACTERISTICS MAY VARY WITH TIME IN THE LONGITU-
DINAL AND VERTICAL DIRECTIONS. THE PROGRAM CAN BE APPLIED WITH
EQUAL EASE TO THE TWO HORIZONTAL DIRECTIONS, ALLOWING DEPTH TO
VARY RATHER THAN WIDTH.

CONCENTRATION PROFILES CAN BE CALCULATED BY THIS PROGRAM FOR
CONTINUOUS OR INSTANTANEUS RELEASES. INPUT DATA MAY INCLUDE GRID
DIMENSIONS, DISTANCE INCREMENTS, TIME INCREMENTS, WIDTHS, LOADING
PARAMETERS, VELOCITIES, DISPERSION COEFFICIENTS, DECAY RATES,
BENTHAL DEMANDS, AND OTHER SOURCE OR SINK TERMS.

THE TIME INCREMENTS CAN BE INCREASED OR DECREASED AT ANY TIME
DURING THE CALCULATION OF THE CONCENTRATION PROFILE. THE DIS-
TANCES BETWEEN GRID POINTS CAN BE INCREASED AT ANY TIME, AND A
ROUTINE WITHIN THE PROGRAM WILL CHOOSE THE APPROPRIATE VALUES FROM
THE PREVIOUSLY CALCULATED PROFILE AND WILL PLACE THESE VALUES IN
THE DESIRED LOCATIONS IN THE NEW GRID SYSTEM. THE NUMBER OF GRID
POINTS MAY BE INCREASED OR DECREASED AT ANY TIME. LIKewise, AT
ANY TIME DURING THE CALCULATION, A NEW SET OF DATA FOR PHYSICAL
AND HYDRODYNAMIC CONDITIONS MAY BE READ INTO THE PROGRAM.
A user of this program must be familiar with the limitations on accuracy and stability inherent in the type of numerical procedure used in these calculations. A subroutine within the program prints out the proper increments to insure stability and terminates the program if this criteria is violated by the input parameters. Another subroutine extrapolates concentrations at the boundaries—several methods can be used for these extrapolations depending on the type of profile being analyzed and the choice of the user. A subroutine is also included which prints out error messages and terminates the program if certain inconsistencies occur in the input data.

This program was developed primarily to analyze partially stratified estuaries which have been dredged out to a fairly constant depth at the centerline of the channel—these estuaries are common in the Gulf Coast region. Application of this program to partially stratified estuaries with variable depths would require moderate revisions to the program and would make the program estuary-dependent.

This computer program can also be applied to estuaries which are well-mixed in the vertical direction. This option allows for varying width or varying depth and uses most of the routines available to the two-dimensional analysis.

Questions regarding this program may be referred to Jonathan Young at Hydroscience, Inc, 363 Old Hook Road, Westwood, New Jersey 07675 Phone 201 / 666-2600

----------------------------------
***INPUT VARIABLES***

SET = DATA SET NUMBER
     -- CAN BE EQUAL TO 1 IF NO NEW GRID DIMENSIONS
     ARE TO BE READ IN -- MUST BE GREATER THAN 1 IF
     GRID DIMENSIONS ARE TO BE CHANGED

************NOTE--OPTIONAL GRID ADJUSTMENT PARAMETERS ARE
     INSERTED HERE. THESE ARE EXPLAINED BELOW

NX = NUMBER OF GRID POINTS IN THE X DIRECTION
NZ = NUMBER OF GRID POINTS IN THE Z DIRECTION
DELMX = X INCREMENT (MILES)
DELTZ = Z INCREMENT (FEET)
DELT1 = FIRST TIME INCREMENT (DAYS)
ITER1 = NUMBER OF ITERATIONS USING DELT1
DELT2 = SECOND TIME INCREMENT (DAYS)
ITER2 = NUMBER OF ITERATIONS USING DELT2
NC = NUMBER OF GRID POINTS IN THE X DIRECTION WHERE
     CONCENTRATIONS ARE ADDED WITH THE DATA SET
NS = NUMBER OF POINT SOURCES
XSTART = MILEAGE AT UPPERMOST GRID POINT
TSTART = TIME TO BE ADDED TO THE TOTAL TIME AT THE FIRST
     ITERATION OF THE DATA SET BEING PROCESSED (DAYS)
XPRINT = PRINTING INTERVAL FOR X DIRECTION (INTEGER)
TPRINT = PRINTING INTERVAL FOR TIME (INTEGER)
NPPAGE = NUMBER OF GRIDS PRINTED PER PAGE OF OUTPUT
LTYPE = TYPE OF LOADING (INSTANTANEOUS OR CONTINUOUS)
1 MEANS CONSTANT CONCENTRATION AT OUTFALL
   (THIS CAN BE USED FOR STEADY-STATE PROFILES)
2 MEANS AN INSTANTANEOUS RELEASE
3 MEANS CONTINUOUS LOADING

OPTION = OPTION FORBOUNDARY CONDITIONS
1 MEANS NO TRANSFER ACROSS Z BOUNDARIES AND
   EXPONENTIAL EXTRAPOLATION IN X DIRECTION
2 MEANS NO TRANSFER ACROSS Z BOUNDARIES AND
   CONSTANT SLOPE EXTRAPOLATION IN X DIRECTION
3 MEANS A SYMMETRICAL DISTRIBUTION WITH
   EXPONENTIAL EXTRAPOLATION
4 MEANS A CONSTANT SLOPE EXTRAPOLATION IN
   BOTH DIRECTIONS
5 MEANS A SPECIAL CASE DEFINED BY THE USER
6 MEANS NO TRANSFER ACROSS Z BOUNDARIES AND
   EXTRAPOLATION IN X DIRECTION BY INVERTED
   DIFFERENCES (ACFI)
7 MEANS EXTRAPOLATION IN BOTH DIRECTIONS BY
   INVERTED DIFFERENCES (ACFI)

VX(X,Z) = HORIZONTAL VELOCITY (FT/SEC)
EX(X,Z) = HORIZONTAL DISPERSION (FT**2/SEC)
W(X) = WIDTH (FEET) FOR THE TWO DIMENSIONAL CASE OR
   CROSS SECTIONAL AREA (FT**2) FOR THE ONE
   DIMENSIONAL CASE
KDAYS(X) = DECAY RATE (PER DAY)
DEMAND(X,Z) = OTHER SOURCES AND SINKS (CONCENTRATION UNITS)
INSECT(I) = SECTIONS INTO WHICH ARE ADDED LOADS
   OR CONCENTRATIONS
CONCIN(I,J) = INITIAL CONCENTRATIONS ASSOCIATED WITH INSECT(I)

***************NOTE---INITIAL CONCENTRATIONS MAY ALSO BE INPUT
   THROUGH THE BLOCK DATA SUBPROGRAM
*********NOTE--VALUES FOR VZ AND EZ ARE TO BE INCLUDED
WITH THE DATA ONLY IF NZ IS GREATER THAN 1

VZ(X,Z) = VERTICAL VELOCITIES (FT/SEC)
EZ(X,Z) = VERTICAL DISPERSION (FT**2/SEC)

FMT1(I) = FORMAT FOR HEADING FOR COMPUTED PROFILES
FMT2(I) = FORMAT FOR DEPTH NOTATION FOR PROFILES
FMT3(I) = FORMAT FOR OUTPUT OF COMPUTED PROFILES

*** GRID ADJUSTMENT PARAMETERS ***

NXSKIP = NUMBER OF X INTERVALS SKIPPED IN OLD GRID
NZSKIP = NUMBER OF Z INTERVALS SKIPPED IN OLD GRID
NXTAKE = LOWEST GRID NUMBER IN X DIRECTION IN THE OLD GRID
FROM WHICH A CONCENTRATION IS TAKEN
NZTAKE = LOWEST GRID NUMBER IN Z DIRECTION IN THE OLD GRID
FROM WHICH A CONCENTRATION IS TAKEN
NXPUT = LOWEST GRID NUMBER IN THE X DIRECTION IN THE NEW
GRID INTO WHICH A CONCENTRATION IS PLACED
NZPUT = LOWEST GRID NUMBER IN THE Z DIRECTION IN THE NEW
GRID INTO WHICH A CONCENTRATION IS PLACED
********** MAIN PROGRAM *****

THE MAIN PROGRAM CALLS FOR THE DATA TO BE READ IN, THEN CALLS FOR THE STABILITY CONDITIONS TO BE EVALUATED, AND NEXT CALLS FOR THE PRINTING OF A SUMMARY OF INPUT DATA. THE NUMERICAL ANALYSIS SUBROUTINE IS THEN CALLED FOR EITHER THE ONE-DIMENSIONAL OR TWO-DIMENSIONAL CASE. IF THE TIME INCREMENT IS TO BE CHANGED DURING THIS PART OF THE PROGRAM, THE APPROPRIATE PARAMETERS ARE ADJUSTED. STABILITY IS AGAIN CHECKED AND THE NUMERICAL ANALYSIS IS CONTINUED. NEXT, A NEW DATA SET MAY BE READ TO CONTINUE THE ANALYSIS. ANY NUMBER OF DATA SETS MAY BE PROCESSED. WHEN NO MORE DATA SETS ARE AVAILABLE, THE PROGRAM IS TERMINATED.


THE FORTRAN USED IN THIS PROGRAM USES SEVERAL OF THE OPTIONS AVAILABLE IN WATFOR AS DESCRIBED IN THE FOLLOWING TEXT**FORTRAN IV WITH WATFOR AND WATFIV**BY CRESS, DIRKSEN, & GRAHAM, PRENTICE-HALL, INC., 1970. THE PROGRAM CAN BE RUN ON A COMPUTER WITH A WATFOR OR WATFIV COMPILER. THE SYNTAX HAS BEEN KEPT COMPATIBLE WITH FORTRAN G EXCEPT FOR THE UNFORMATTED READ AND PRINT STATEMENTS.
INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,COUNT,OPTION
REAL KD,KMAX,KMIN,KDAYS,LOWER
COMMON/ARRAYS/ C(51,21,2),VX(51,21),VZ(51,21),EX(51,21),EZ(51,21),
1 W(51),CONCIN(51,21),INSECT(51),KDAYS(51),KD(51),
2 DEMAND(51,21),FMT1(40),FMT2(40),FMT3(20)
COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,WMAX,WMIN,
1 KMAX,KMIN,ITER,ITER1,ITER2,DELT1,DELT2,TNEXT,
2 TPRINT,XPRINT,STABLX,STABLY,STABLZ,STABL,T1,T2,VZMAX,SET,
3 VXMIN,EZMAX,EZMIN,DEMIX,DEMIX,NPPAGE,DELTAX,DELTAX,
4 DELTAT,XSTART,XSF,TSTART,COUNT,IPAGE,UPPER,LOWER,
5 MSU,MSL,TIME,NX1,NX2,NZ1,NZ2,ZSF,LTYPE,OPTION

10 CALL DATA(&1000)
11 CALL STABLE
12 CALL PRINT1
13 CALL PRINT2
14 CALL ERROR(&1000)
15 25 IF(NZ.EQ.1) GO TO 310
16 CALL TWODEX
17 GO TO 315
18 310 CALL ONEDEX
19 315 IF(ITER2.EQ.0) GO TO 10
20 ITER=ITER2
21 ITER2=0
22 TPRINT=TPRINT*(DELT1/DELT2+.00001)
23 DELTAT=DELT2*86400.
24 CALL STABLE
25 IF(ITER.EQ.0) CALL ERRCR(&1000)
26 GO TO 25
27 1000 CONTINUE
28 STOP
29 END
**BLOCK DATA**

************************************

***** BLOCK DATA *****

************************************

THIS ROUTINE INITIALIZES SELECTED VARIABLES IN THE
COMMON BLOCKS. THIS SUBPROGRAM MAY BE USED TO ESTABLISH
INITIAL CONCENTRATION PROFILES IF THE USER DESIRES.

----------------------------------------------

INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,COUNT,OPTION

REAL KD,KDAX,KDMIN,KDAYS,LOWER

COMMON/ARRAYS/ C(51,21,21),VX(51,21),VZ(51,21),EX(51,21),EZ(51,21),
1 W(51),CONCIN(51,21),INSECT(51),KDAYS(51),KD(51),
2 DEMAND(51,21),FMT1(40),FMT2(40),FMT3(20)

COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,WMAX,WMIN,
1 KDMAX,KDMIN,ITER,ITER1,ITER2,DELT1,DELT2,TNEXT,
2 TPRINT,XPRINT,STBLX,STBLZ,STBLT,R1,R2,VZMAX,SET,
3 VZMIN,EZMAX,EZMIN,DEMAX,DEMIN,NPPAGE,DELTAX,DELTAZ,
4 DELTAT,XSTART,XSF,TSTART,COUNT,IPAGE,UPPER,LOWER,
5 MSU,MSL,TIME,NXM1,NXM2,NZM1,NZM2,ZSF,LTYPE,OPTION

----------------------------------------------

ESTABLISH INITIAL CONCENTRATION IF DESIRED

----------------------------------------------

DATA C /2142*0.0/
DATA VX,VZ,EX,EZ,CONCIN,DEMAND/6426*0.0/
DATA W,KDAYS,KD/153*0.0/
DATA EXAXM,EZMAX,VXMAX,VZMAX,WMAX,KDMIN,DEMAX,TIME/8*0.0/
DATA EXMIN,EZMIN,VXMIN,VZMIN,WMIN,KDMIN,DEMIN/7*1000000.0/
DATA INSECT/51*0/
DATA IPAGE/1/
DATA T/1/,TNEXT/2/
DATA END
C----------------- SUBROUTINE DATA(*)
C
***************
***** SUBROUTINE DATA *****
***************
C
THIS SUBROUTINE READS IN THE APPROPRIATE DATA SET AND ADJUSTS
THE GRID SIZE IF NECESSARY. THE LOCATIONS OF THE UPPERMOST AND
LOWEST PEAKS ARE DETERMINED, AND THE MINIMUM AND MAXIMUM VALUES
FOR THE INPUT PARAMETERS ARE CALCULATED. ALL OF THE INPUT DATA IS
PRINTED OUT UNFORMATTED. THE FORMAT IS READ IN FOR THE PRINTING
OUT OF THE CALCULATED CONCENTRATIONS.
C-----------------

INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,CCOUNT,OPTION
REAL KD,KDAX,KDMIN,KDAYS,LOWER
COMMON/ARRAYS/ C(51,21,2),VX(51,21),VZ(51,21),EX(51,21),EZ(51,21),
 1 W(51),CCNCIN(51,21),INSECT(51),KDAYS(51),KD(51),
 2 DEMAND(51,21),FMT1(40),FMT2(40),FMT3(20)
COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,WMAX,WMIN,
 1 KDAX,KDMIN,ITER,ITER1,ITER2,DELT1,DELT2,TNEXT,
 2 TPRINT,XPRINT,STABLX,STABLT,STABLT,R1,R2,VZMAX,SET,
 3 VZMIN,EZMAX,EZMIN,DMAX,DMIN,NPPAGE,DELTAX,DELTAZ,
 4 DELTAT,XSTART,XSF,TSTART,COUNT,IPAGE,UPPER,LOWER,
 5 MSU,MSL,TIME,NXM1,NXM2,NZM1,NZM2,ZSF,LTYPE,OPTION
READ(5,12,END=13) SET
12 FORMAT(F10.0)
GO TO 14
13 CONTINUE
RETURN1
14 CONTINUE
IF(SET.EQ.1.) GO TO 400
CHOSE CONCENTRATIONS FROM THE OLD GRID TO PUT INTO THE
APPROPRIATE PLACES IN THE NEW GRID. THE GRID CAN ONLY BE
CHANGED IF LTYPE EQUALS 2. OTHERWISE, THE DATA CARD IS READ
BUT IS NOT IMPLEMENTED.

51 READ,NXSKIP,NZSKIP,NXTAKE,NZTAKE,NXPUT,NZPUT
52 IF(LTYPE.NE.2) GO TO 400
53 NXSTOP=(NX-1)/NXSKIP+1
54 NZSTOP=(NZ-1)/NZSKIP+1
55 IF(NZ.EQ.1) NZSTOP=1
56 DO 350 I=1,NXSTOP
57 X=NXPUT-I+1
58 II=NXTAKE+NXSKIP*(I-1)
59 DO 350 J=1,NZSTOP
60 Z=NZPUT-J+1
61 JJ=NZTAKE+NZSKIP*(J-1)
62 C(X,Z,TNEXT)=C(II,JJ,T)
63 350 CONTINUE
64 400 CONTINUE

READ AND PRINT INPUT DATA FOR HORIZONTAL DIRECTION
READ,NX,NZ
READ,DELMX,DELTAZ
READ,DELT1,ITER1
READ,DELT2,ITER2
READ,NC,NS
READ,XSTART,TSTART,XPRINT,TPRINT,NPPAGE
READ,LTYPE,OPTION
PRINT 1001
1001 FORMAT( 1H1,10X,'INPUT DATA*','//,
1   1X,'NX,NZ, DELMX,DELTAZ, DELT1,ITER1*)
PRINT,NX,NZ,DELMX,DELTAZ,DELT1,ITER1
PRINT 1002
1002 FORMAT(',//,1X,'DELT2,ITER2, NC,NS, XSTART,TSTART*)
PRINT, DELT2,ITER2,NC,NS,XSTART,TSTART
PRINT 1003
1003 FORMAT(',//,1X,'XPRINT,TPRINT,NPPAGE, LTYPE,OPTION*)
PRINT, XPRINT,TPRINT,NPPAGE,LTYPE,OPTION
IF(SET.EQ.1.) GO TO 500
DO 375 Z=1,NZ
DO 375 X=1,NX
C X,Z,T=C(X,Z,TNEXT)
C X,Z,TNEXT=0.0
375 CONTINUE
500 CONTINUE
NXM1=NX-1
NXM2=NX-2
NZM1=NS-1
NZM2=NS-2
DELTAT=DELT1*86400.
DELTAX=DELTX*5280.
XSF=XSTART*5280.
ITER=ITER1
TIME=TIME+TSTART*86400.
COUNT=(TSTART/DELTAT+0.01)
COUNT=COUNT-(COUNT/TPRINT)*TPRINT
IF(SECC.EQ.2) RETURN
READ*((VX(X,Z),X=1,NX),Z=1,NZ)
PRINT 2
2 FORMAT(/,,1X,'((VX(X,Z),X=1,NX),Z=1,NZ)')
PRINT*((VX(X,Z),X=1,NX),Z=1,NZ)
READ*((EX(X,Z),X=1,NX),Z=1,NZ)
PRINT 3
3 FORMAT(/,,1X,'((EX(X,Z),X=1,NX),Z=1,NZ)')
PRINT*((EX(X,Z),X=1,NX),Z=1,NZ)
READ*(W(X),X=1,NX)
PRINT 4
4 FORMAT(/,,1X,'(W(X),X=1,NX)')
PRINT*,(W(X),X=1,NX)
READ*(KDA(YS(X),X=1,NX)
PRINT 5
5 FORMAT(/,,1X,'(KDA(YS(X),X=1,NX)')
PRINT*,(KDA(YS(X),X=1,NX)
READ*((DEMAND(X,Z),X=1,NX),Z=1,NZ)
PRINT 6
6 FORMAT(/,,1X,'((DEMAND(X,Z),X=1,NX),Z=1,NZ)')
PRINT* ((DEMAND(X,Z),X=1,NX),Z=1,NZ)
IF(NC.EQ.0) GO TO 24
DO 10 I=1,NC
READ*INSECT(I)
PRINT 7
7 FORMAT(//, 1X, 'INSECT(I)')
PRINT, INSECT(I)
READ, (CONCIN(I,J), J=1,NZ)
PRINT 8
8 FORMAT(//, 1X, '(CONCIN(I,J),J=1,NZ)')
PRINT, (CONCIN(I,J), J=1,NZ)
10 CONTINUE
DO 15 I=1,NC
   DO 15 Z=1,NZ
   C(INSECT(I),Z,T)=CONCIN(I,Z) + C(INSECT(I),Z,T)
15 CONTINUE
C---------------------------------------------------------------
C DETERMINE LOCATIONS OF UPPERMOST AND LOWERMOST C PEAK CONCENTRATIONS C---------------------------------------------------------------
AMAX1=0.0
PEAKU=0.0
AMAX2=0.0
PEAKL=0.0
UPPER=XSF
LOWER=XSF
MSU=1
MSL=NX
DO 17 I=1,NC
   DO 17 J=1,NZ
      IF(CONCIN(I,J).LT.PEAKU) GO TO 16
      PEAKU=CONCIN(I,J)
17 CONTINUE
16 CONTINUE
IF(PEAKU.LE.AMAX1) GO TO 20
AMAX1=PEAKU
MSU=INSECT(I)

UPPER=(MSU-1)*DELTAX + XSF

CONTINUE

DO 22 II=1,NC
I=NC+1-II
DO 21 J=1,NZ
IF(CONCIN(I,J),LT,PEAKL) GO TO 21
PEAKL=CONCIN(I,J)

CONTINUE

IF(PEAKL.LE.AMAX2) GO TO 24
AMAX2=PEAKL
MSL=INSECT(I)
LOWER=(MSL-1)*DELTAX + XSF

CONTINUE

CONTINUE

C-----------------------------------------------
C DETERMINE MAXIMUM AND MINIMUM VALUES FOR INPUT DATA
C-----------------------------------------------

DO 25 X=1,NX
KD(X)=KDAYS(X)/86400.
IF(W(X),GT,WMAX)WMAX=W(X)
IF(W(X),LT,WMIN)WMIN=W(X)
IF(KD(X),GT,KDMAX)KDMAX=KD(X)
IF(KD(X),LT,KDMIN)KDMIN=KD(X)
DO 25 Z=1,NZ
ABSVX=ABS(VX(X,Z))
IF(ABSVX,GT,VXMAX)VXMAX=ABSVX
175 IF(ABS VX.LT.VXMIN)VXMIN=ABS VX
176 IF(EX(X,Z).GT.EXMAX)EXMAX=EX(X,Z)
177 IF(EX(X,Z).LT.EXMIN)EXMIN=EX(X,Z)
178 IF(DEMAND(X,Z).GT.DEMAX)DEMAX=DEMAND(X,Z)
179 IF(DEMAND(X,Z).LT.DEMIN)DEMIN=DEMAND(X,Z)
180 25 CONTINUE
181 IF(NZ.EQ.1)GO TO 200

C-----------------------------------------------------------------------
C READ AND PRINT DATA FOR VERTICAL DIRECTION
C-----------------------------------------------------------------------
182 READ((VZ(X,Z),X=1,NX),Z=1,NZ)
183 PRINT 30
184 30 FORMAT(//,1X,*(VZ(X,Z),X=1,NX),Z=1,NZ*)
185 PRINT((VZ(X,Z),X=1,NX),Z=1,NZ)
186 READ((EZ(X,Z),X=1,NX),Z=1,NZ)
187 PRINT 32
188 32 FORMAT(//,1X,*(EZ(X,Z),X=1,NX),Z=1,NZ*)
189 PRINT((EZ(X,Z),X=1,NX),Z=1,NZ)

C-----------------------------------------------------------------------
C DETERMINE MAXIMUM AND MINIMUM VALUES FOR INPUT DATA
C-----------------------------------------------------------------------
190 DO 120 X=1,NX
191 DO 120 Z=1,NZ
192 ABSVZ=ABS(VZ(X,Z))
193 IF(ABSVZ.GT.VZMAX)VZMAX=ABSVZ
194 IF(ABSVZ.LT.VZMIN)VZMIN=ABSVZ
195 IF(EZ(X,Z).GT.EZMAX)EZMAX=EZ(X,Z)
196 IF(EZ(X,Z).LT.EZMIN)EZMIN=EZ(X,Z)
197 120 CONTINUE
198 200 CONTINUE
READ AND PRINT FORMATS FOR OUTPUT

199       READ 205, (FMT1(I), I=1,40)
200       PRINT 202
201       202 FORMAT(//$,1X,'FORMAT FOR OUTPUT$)
202       PRINT 205, (FMT1(I), I=1,40)
203       205 FORMAT (20A4,//$,20A4)
204       READ 205, (FMT2(I), I=1,40)
205       PRINT 205, (FMT2(I), I=1,40)
206       READ 207, (FMT3(I), I=1,20)
207       PRINT 207, (FMT3(I), I=1,20)
208       207 FORMAT (20A4)
209       PRINT 259
210       259 FORMAT (1H1)
211       RETURN
212       END
SUBROUTINE STABLE

THE RELATIONSHIP BETWEEN DISTANCE INCREMENTS, TIME INCREMENTS,
DISPERSION COEFFICIENTS, AND VELOCITIES IS NEEDED TO DETERMINE
THE STABILITY OF THE EXPPLICIT FINITE-DIFFERENCE PROCEDURE. THIS
SUBROUTINE CALCULATES THE MAXIMUM ALLOWABLE INCREMENTS FOR TIME
AND DISTANCE. THE PROGRAM IS TERMINATED IF THE INPUT PARAMETERS
VIOLATE THIS CRITERIA.

INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,COUNT,OPTION
REAL KD,KDMAX,KDMIN,KDAYS,LOWER
COMMON/ARRAYS/ C(51,21,2), VX(51,21), VZ(51,21), EX(51,21), EZ(51,21),
               W(51), CONCIN(51,21), INSECT(51), KDAYS(51), KD(51),
               DEMAND(51,21), FMT1(40), FMT2(40), FMT3(20)
COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,VZMAX,VZMIN,
               KDMAX,KDMIN,ITER,ITER1,ITER2,DELT1,DELT2,TNEXT,
               TPRINT,XPRINT,STABLX,STABLZ,STABLT,R1,R2,VMAX,SET,
               VZMIN,EZMAX,EZMIN,DEMAX,DEMIN,NPPAGE,DELTA,T,DELTAZ,
               DELTAT,XSTART,XSF,TSTART,COUNT,IPAGE,UPPER,LOWER,
               MSU,MSL,TIME,NXM1,NXM2,NZM1,NZM2,ZSF,LTYPE,OPTION

STABLX = DELTAX
STABLZ = DELTAZ
STABLT = DELTAT
C
CALCULATE STABILITY CRITERIA FOR ONE DIMENSIONAL CASE

221 IF(NZ.NE.1) GO TO 500
222 IF(EXMIN.GT.0.0.AND.VXMAX.GT.0.0) STABLX=2.*EXMIN/VXMAX
223 IF(STABLX.LT.DELTAX) ITER=0
224 TERM=2.*EXMAX+DELTAX**2.*KMAX
225 IF(TERM.NE.0.0) STABL=(DELTAX**2.)*TERM
226 IF(STABL.LT.DELTAT) ITER=0
227 R1=DELTAT/(DELTAX**2.)
228 R2=0.
229 GO TO 800
230 500 CONTINUE

C
CALCULATE STABILITY CRITERIA FOR TWO DIMENSIONAL CASE

231 IF(EXMIN.GT.0.0.AND.VXMAX.GT.0.0) STABLX=2.*EXMIN/VXMAX
232 IF(STABLX.LT.DELTAX) ITER=0
233 IF(EXMIN.GT.0.0.AND.VZMAX.GT.0.0) STABLZ=2.*EZMIN/VZMAX
234 IF(STABLZ.LT.DELTAZ) ITER=0
235 TERM=2.*((EXMAX*DELTAZ**2.+EZMAX*DELTAX**2.)+KMAX*(DELTAX**2.)*
$ (DELTAZ**2.2))
236 IF(TERM.NE.0.0) STABL=(DELTAZ**2.)*(DELTAX**2.)*TERM
237 IF(STABL.LT.DELTAT) ITER=0
238 R1=DELTAT/(DELTAX**2.0)
239 R2=DELTAT/(DELTAZ**2.0)
240 800 CONTINUE
241 RETURN
242 END
SUBROUTINE ONEDEX

***************
**** SUBROUTINE ONEDEX ****
***************

THIS SUBROUTINE CALCULATES THE ONE DIMENSIONAL CONCENTRATION
PROFILE FOR THE ASSIGNED GRID AND THE ASSIGNED NUMBER OF ITERA-
TIONS. A FINITE-DIFFERENCE, EXPLICIT SCHEME IS USED TO SOLVE THE
PARTIAL DIFFERENTIAL EQUATION FOR MASS TRANSPORT.

INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,COUNT,OPTION
REAL KD,KDMIN,KDAYS,KDAYS,LLOW
COMMON/ARRAYS/ C(51,21,21), VX(51,21), VZ(51,21), EX(51,21), EZ(51,21),
  1 W(51), CONCIN(51,21), INSECT(51), KDAISY(51), KD(51),
  2 DEMAND(51,21), FMT1(40), FMT2(40), FMT3(20)
COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,WMAX,WMIN,
  1 KDMIN,KDMIN, ITER, ITER1, ITER2, DELT1, DELT2, TNEXT,
  2 TPRINT, XPRINT, STABLX, STABLY, STABLT, R1, R2, VZMAX, SET,
  3 VZMIN, EMAX, EMIN, DEMAX, DEMIN, NPAGE, DELTAX, DELTAY,
  4 DELTAT, XSTART, XSF, TSTART, COUNT, IPAGE, UPPER, LOWER,
  5 MSU, MSL, TIME, XMIN, XMAX, NMX1, NMX2, NZMIN, NMX2, ZSF, LTYPE, OPTION
248  Z=1
249  NXM1=NX-1

C===============================================================================
C CALCULATE TERMS WHICH ARE USED REPEATEDLY
C===============================================================================
250  TWOXSQ=2.*DELTAX**2.
251  TWOX=2.*DELTAX
252  TWOZSQ=2.*DELTAZ**2.
253  TWOZ=2.*DELTAZ

C===============================================================================
C CALCULATE NEW CONCENTRATION AT EACH POINT
C===============================================================================
254  DO 300 IT=1,ITER
255    TIME=TIME+DELTAT
256    DO 200 X=2,NXM1
257    TERM1A=W(X)*VX(X,Z)/TWOX
258    TERM1B=(W(X)*EX(X,Z)+W(X-1)*EX(X-1,Z))/TWOXSQ
259    TERM2B=(W(X+1)*EX(X+1,Z)+W(X)*EX(X,Z))/TWOXSQ
260    TERM3A=(-W(X)*VX(X,Z))/TWOX
261    TERM3B=(W(X+1)*EX(X+1,Z)+W(X)*EX(X,Z))/TWOXSQ
262    TERM1=C(X-1,Z,T)*TERM1A*TERM1B
263    TERM2=-C(X,Z,T)*TERM2B*KD(X)*W(X)
264    TERM3=C(X+1,Z,T)*TERM3A*TERM3B
265    C(X,Z,TNEXT)=C(X,Z,T)+(DELTAT/W(X))*TERM1+TERM2+TERM3
$  DEMAND(X,Z)
266    IF(C(X,Z,TNEXT).LT.0.0) C(X,Z,TNEXT) = 0.0
267  200 CONTINUE
C

CALCULATE BOUNDARY VALUES

268       CALL BOUND
269       Z=1
270     DO 250 X=1,NX
271     C(X,Z,T)=C(X,Z,TNEXT)
272     C(X,Z,TNEXT)=0.0
273   250 CONTINUE
274       COUNT=COUNT+1
275     IF(COUNT.GE.TPRINT) GO TO 292
276       GO TO 300

C

PRINT CONCENTRATIONS AT APPROPRIATE INTERVALS

277   292 CONTINUE
278       COUNT=0
279     IPAGE=IPAGE+1
280     IF(IPAGE.GE.NPPAGE) GO TO 295
281       GO TO 297
282   295 PRINT 296
283     296 FORMAT(1H1)
284       IPAGE = 0
285   297 CALL PRINT2
286   300 CONTINUE
287       RETURN
288       END
SUBROUTINE TWODEX

******************************************************************************
****** SUBROUTINE TWODEX ******
******************************************************************************

THIS SUBROUTINE CALCULATES THE TWO-DIMENSIONAL CONCENTRATION
PROFILE FOR THE ASSIGNED GRID AND THE ASSIGNED NUMBER OF ITERA-
TIONS. A FINITE-DIFFERENCE, EXPLICIT SCHEME IS USED TO SOLVE THE
PARTIAL DIFFERENTIAL EQUATION FOR MASS TRANSPORT.

INTEGER X, L, T, TNEXT, TPRINT, XPRT, COUNT, OPTION
REAL KD, KMAX, KMIN, KDAYS, LOWER
COMMON/ARRAYS/ C(51,21,21), VX(51,21), VZ(51,21), EX(51,21), EZ(51,21),
1 W(51), CONCIN(51,21), INSECT(51), KDAYS(51), KD(51),
2 DEMAND(51,21), FMT1(40), FMT2(40), FMT3(20)
COMMON/NAMES/X, Z, T, NX, NZ, NC, NS, EXMAX, EXMIN, VXMAX, VXMIN, WMAX, WMIN,
1 KMAX, KMIN, ITER, ITER1, ITER2, DELT1, DELT2, TNEXT,
2 TPRINT, XPRT, STABLX, STABLZ, STABL, R1, R2, VZMAX, SET,
3 VXMIN, EZMAX, EZMIN, DEMAX, DEMIN, NPPAGE, DELTAX, DELTAZ,
4 DELTAT, XSTART, XSF, TSTART, COUNT, IPAGE, UPPER, LOWER,
5 MSU, MSL, TIME, NXM1, NXM2, NZM1, NZM2, ZSF, LTYPE, OPTION

NXM1=NX-1
NZM1=NZ-1
NZ1=2
C

C -----------------------------------------------
C CALCULATE TERMS WHICH ARE USED REPEATEDLY
C -----------------------------------------------

297  TWOXSQ=2.*DELTA**2.
298  TWOX=2.*DELTA
299  TWOZSQ=2.*DELTA**2.
300  TWOZ=2.*DELTA

C

C -----------------------------------------------
C CALCULATE NEW CONCENTRATION AT EACH POINT
C -----------------------------------------------

301  DO 300 IT=1,ITER
302  TIME=TIME+DELTAT
303  DO 200 X=2,NXM1
304  DO 100 Z=NZ1,NZM1
305  TERM1A=W(X)*VX(X,Z)/TWOX
306  TERM18=(W(X)*EX(X,Z)+W(X-1)*EX(X-1,Z))/TWOXSQ
307  TERM2B = (W(X+1)*EX(X+1,Z) + W(X)*EX(X,Z))/TWOXSQ
308     *(W(X)*EX(X,Z) + W(X-1)*EX(X-1,Z))/TWOXSQ
309  TERM3A=(-W(X)*VX(X,Z))/TWOX
310  TERM3B=(W(X+1)*EX(X+1,Z)+W(X)*EX(X,Z))/TWOXSQ
311  TERM4A=W(X)*VZ(X,Z)/TWCZ
312  TERM4B=(W(X)*EZ(X,Z)+W(X)*EZ(X,Z-1))/TWOZSQ
313  TERM5A=(-W(X)*VZ(X,Z))/TWOZ
314  TERM5B=(W(X)*EZ(X,Z+1)+W(X)*EZ(X,Z))/TWOZSQ
315  TERM6B = (W(X) * EZ(X,Z+1) + W(X)*EZ(X,Z))/TWOZSQ
316     *(W(X)*EZ(X,Z) + W(X)*EZ(X,Z-1))/TWOZSQ

235
TERM2 = -C(X,Z,T)*(TERM2B+TERM6B+KDX)*W(X))
TERM1 = C(X-1,Z,T)*(TERM1A+TERM1B)
TERM3 = C(X+1,Z,T)*(TERM3A+TERM3B)
TERM4 = C(X,Z-1,T)*(TERM4A+TERM4B)
TERM5 = C(X,Z+1,T)*(TERM5A+TERM5B)
C(X,Z,TNEXT) = C(X,Z,T) + (DELTAT/W(X))*(TERM1+TERM2+TERM3+TERM4+TERM5
$ I-DEMAND(X,Z)
IF(C(X,Z,TNEXT).LT.0.0) C(X,Z,TNEXT) = 0.0
100 CONTINUE
200 CONTINUE
C
C---------------------------------------------
C CALCULATE BOUNDARY VALUES
C---------------------------------------------
CALL ROUND
DO 250 Z = 1,NZ
DO 250 X = 1,NX
C(X,Z,T) = C(X,Z,TNEXT)
C(X,Z,TNEXT) = 0.0
250 CONTINUE
COUNT = COUNT + 1
IF (COUNT .GE. TPRINT) GO TO 292
GO TO 300

C
------------------------------------------------------------------------
C
PRINT CONCENTRATIONS AT APPROPRIATE INTERVALS
C
------------------------------------------------------------------------

292 CONTINUE
COUNT = 0
IPAGE = IPAGE + 1
IF (IPAGE .GE. NPAGE) GO TO 295
GO TO 297
295 PRINT 296
296 FORMAT (I11)
IPAGE = 0
297 CALL PRINT2
300 CONTINUE
RETURN
END
SUBROUTINE BOUND

*******************************************************************************
***** SUBROUTINE BOUND *****
*******************************************************************************

SUBROUTINE BOUND PROVIDES FOR EXTRAPOLATION OF CONCENTRATIONS
AT THE BOUNDARIES. THE TYPE OF EXTRAPOLATION DEPENDS ON THE VALUE
OF THE VARIABLE OPTION.

346 INTEGER X, Z, T, TNEXT, TPRINT, XPRINT, CCOUNT, OPTION
347 REAL KD, KD MAX, KD MIN, KDAYS, LOWER
348 COMMON/ARRAYS/ C(51,21,2), VX(51,21), VZ(51,21), W(51), CONCIN(51,21), INSECT(51), KDAYS(51), KD(51),
349 DEMAND(51,21), FMT1(40), FMT2(40), FMT3(20)
350 COMMON/NAMES/X, Z, T, NX, NZ, NC, NS, EXMAX, EXMIN, VXMAX, VXMIN, WMAX, WMIN,
351 KD MAX, KD MIN, ITER, ITER1, ITER2, DELT1, DELT2, TNEXT,
352 TPRINT, XPRINT, STABLX, STABLZ, STABLT, R1, R2, VZMAX, SET,
353 VZMIN, EZMAX, EZMIN, DEMAX, DEMIN, NPPAGE, DELTAX, DELTAY, DELTAZ,
354 DELTAT, XSTART, XSF, TSTART, COUNT, IPAGE, UPPER, LOWER,
355 MSU, MSL, TIME, NXMI, NXM2, NZM1, NZM2, ZSF, LTYPE, OPTION
356 IF(NZ.EQ.1) NZM1=1
357 ZUP=UPPER
358 ZLOW=LOWER
359 ZSF=XSF
354  GO TO (1001, 1002, 1003, 1004, 1005, 1006, 1007), OPTION
C
C--------------------------------------------------------------------------------------
C EXPONENTIAL EXTRAPOLATION
C--------------------------------------------------------------------------------------
C 1003 CONTINUE
356  IF(NZ.EQ.1) GO TO 1001
357  DO 140 X=2,NXM1
358  IF(C(X,2,T+1).LE.0.0) GO TO 140
359  VZ1=VZ(X,1)
360  ZZ=-ZUP+DELTAZ*ZSF+VZ1*TSTART*86400.
361  EZ1=EZ(X,1)
362  C2=4.*EZ1*TIME
363  EXPON=( 2.*ZZ*DELTAZ-(DELTAZ*DELTAZ)-2.*DELTAZ*VZ1*TIME)/C2
364  C(X,1,T+1)=C(X,2,T+1)*EXP(EXPON)
365  140 CONTINUE
366  DO 190 X=2,NXM1
367  IF(C(X,NZM1,T+1).LE.0.0) GO TO 190
368  VZN2=VZ(X,NZ)
369  ZZ=(NZ-2)*DELTAZ-ZLOW+ZSF+VZN2*TSTART*86400.
370  EZN2=EZ(X,NZ)
371  C2=4.*EZN2*TIME
372  EXPON=(-2.*ZZ*DELTAZ-(DELTAZ*DELTAZ)+2.*DELTAZ*VZN2*TIME)/C2
373  C(X,NZ,T+1)=C(X,NZM1,T+1)*EXP(EXPON)
374  190 CONTINUE
375  1001 CONTINUE
376  IF(LTYPE.EQ.1.AND.MSU.EQ.1) GO TO 350
377  DO 340 Z=1,NZ
378  IF(C(2,Z,T+1).LE.0.0) GO TO 340
379  VX1=VX(1,Z)
380  XX=UPPER+DELTAX*XSF+VX1*TSTART*86400.
381  EX1=EX(1,Z)
C2=4.*EX1*TIME
EXPON=1 2.*XX*DELTAX-(DELTAX*DELTAX)-2.*DELTAX*VX1*TIME)/C2
C(1,Z,T+1)=C(2,Z,T+1)*EXP(EXPON)

340 CONTINUE
350 CONTINUE
387 IF(LTYPE.EQ.1.AND.MSL.EQ.NZ) GO TO 400
388 DO 390 Z=1,NZ
389 IF(C(NX-1,Z,T+1).LE.0.0) GO TO 390
390 VXNX=VX(NX,Z)
391 XX=(NX-2)*DELTAX-LOWER+XSF+VXNX*TSTART*86400.
392 EPNX=EX(NX,Z)
393 C2=4.*EPNX*TIME
394 EXPON=(-2.*XX*DELTAX-(DELTAX*DELTAX)+2.*DELTAX*VXNX*TIME)/C2
395 C(NX,Z,T+1)=C(NX-1,Z,T+1)*EXP(EXPON)
396 390 CONTINUE
397 400 CONTINUE
398 GO TO 2000

C

C-----------------------------------------------------------------------------------
C                   CONSTANT SLOPE EXTRAPOLATION
C-----------------------------------------------------------------------------------

399 1004 CONTINUE
400 IF(NZ.EQ.1) GO TO 1002
401 DO 540 X=2,NXM1
402 C(X,1,T+1)=2.*C(X,2,T+1)-C(X,3,T+1)
403 IF(C(X,1,T+1).LT.0.0) C(X,1,T+1)=0.0
404 C(X,NZ,T+1)=2.*C(X,NZM1,T+1)-C(X,NZM2,T+1)
405 IF(C(X,NZ,T+1).LT.0.0) C(X,NZ,T+1)=0.0
406 540 CONTINUE
1002 CONTINUE
   DO 590 Z=1,NZ
      C(1,Z,T+1)=2.*C(2,Z,T+1)-C(3,Z,T+1)
   IF(C(1,Z,T+1).LT.0.0) C(1,Z,T+1)=0.0
   C(NX,Z,T+1)=2.*C(NXM1,Z,T+1)-C(NXM2,Z,T+1)
   IF(C(NX,Z,T+1).LT.0.0) C(NX,Z,T+1)=0.0
590 CONTINUE
   GO TO 2000
1005 CONTINUE
C------------------------------------------------------------------------
C THIS OPTION ALLOWS THE PROGRAM USER TO SUBSTITUTE HIS OWN
C EXTRAPOLATION ROUTINE
C------------------------------------------------------------------------
   ZZ=-40.
   EZ1=0.025
   VZ1=0.
   C2=4.*EZ1*TIME
   EXPON=(*2.*ZZ*DELTAZ-(DELTAZ**2.)-2*DELTAZ*VZ1*TIME)/C2
   DO 240 X=1,NX
      C(X,1,T+1)=C(X,2,T+1)*EXP(EXPON)
      C(X,NZ,T+1)=C(X,1,T+1)
240 CONTINUE
   GO TO 1001
C------------------------------------------------------------------------
C INVERTED DIFFERENCES EXTRAPOLATION (ACFI)
C------------------------------------------------------------------------
1006 CALL EXTRAP
   GO TO 2000
2000 CONTINUE
IF(NZ.EQ.1) GO TO 2005
IF(OPTION.EQ.3.OR.OPTION.EQ.4.OR.OPTION.EQ.5) GO TO 2005
DO 700 X=1,NX
   C(X,1,T+1)=C(X,2,T+1)
   C(X,NZ,T+1)=C(X,NZM1,T+1)
700 CONTINUE
2005 CONTINUE

C
C IF LTYPE EQUALS 1, SET CONCENTRATIONS AT SOURCE POINTS.
C IF LTYPE EQUALS 3, THE USER CAN INCLUDE A SPECIAL SUBROUTINE
C OR SET OF CALCULATIONS IN THIS PART OF THE PROGRAM.
C

IF(LTYPE.EQ.1.AND.NS.LE.2) GO TO 275
IF(LTYPE.EQ.1.AND.NS.GT.2) GO TO 281
GO TO 290
275 CONTINUE
DO 280 Z=1,NZ
   C(MSU,Z,T+1)=C(MSU,Z,T)
   C(MSL,Z,T+1)=C(MSL,Z,T)
280 CONTINUE
GO TO 290
281 CONTINUE
DO 285 I=1,NC
   DO 285 Z=1,NZ
   C(INSECT(I),Z,T+1)=CONCIN(I,Z)
285 CONTINUE
GO TO 290
RETURN
END
SUBROUTINE EXTRAP

*******************************************
**** SUBROUTINE EXTRAP ****
*******************************************

THIS SUBROUTINE EXTRAPOLATES THE CONCENTRATION PROFILE BY
USING A CONTINUED FRACTIONS AND INVERTED DIFFERENCES SCHEME. IT IS
A MODIFIED VERSION OF THE IBM SCIENTIFIC SUBPROGRAM ACFI.

IF THE EXTRAPOLATED CONCENTRATION IS GREATER THAN THE
ADJACENT CONCENTRATION OR IF IT IS NEGATIVE, THEN THE BOUNDARY
CONCENTRATION IS EXTRAPOLATED ACCORDING TO THE PROPORTION
BETWEEN THE CONCENTRATIONS AT THE TWO ADJACENT INTERNAL POINTS
(I.E. C1/C2 = C2/C3).

INTEGER X, Z, T, TNEXT, TPRINT, XPRINT, COUNT, OPTION
REAL KD, KDMAX, KDMIN, KDAYS, LOWER
COMMON/ARRAYS/ C(51,21,2), VX(51,21), VZ(51,21), EX(51,21), EZ(51,21),
1 W(51), CONCIN(51,21), INSECT(51), KDAYS(51), KD(51),
2 DEMAND(51,21), FMT1(40), FMT2(40), FMT3(20)
COMMON/NAMES/X, Z, T, NX, NZ, NC, NS, EXMAX, EXMIN, VXMAX, VXMIN, WMAX, WMIN,
1 KDMAX, KDMIN, ITER, ITER1, ITER2, DELT1, DELT2, TNEXT,
2 TPRINT, XPRINT, STABLX, STABLZ, STABLRT, R1, R2, VZMAX, SET,
3 VZMIN, EZMAX, EZMIN, DEMAX, DEMIN, NPPAGE, DELTAX, DELTAZ,
4 DELTAT, XSTART, XSF, TSTART, COUNT, IPAGE, UPPER, LOWER,
5 MSU, MSL, TIME, NXM1, NXM2, NZM1, NZM2, ZSF, LTYPE, OPTION
DIMENSION ARG(10), VALY(10)
NZ1 = 2

IF(NZ.EQ.1) NZ1 = 1
IF(NZ.EQ.1)NZM1 = 1
IF(MSU.EQ.1.AND.LTYPE.EQ.1) GO TO 228

C
-----------------------------------------------
C  EXTRAPOLATE THE PROFILE IN THE X DIRECTION
C
-----------------------------------------------
DO 225 Z = NZ1,NZM1
IF(C(2, Z, T+1) . LE. C.00) GO TO 220

C
-----------------------------------------------
C  CHOOSE THE NUMBER OF POINTS (NDIM) TO BE USED IN THE
C  EXTRAPOLATION.
C
-----------------------------------------------
NDIM = 3
NSTOP = NDIM + 1

C
-----------------------------------------------
C  PLACE CONCENTRATIONS IN PROPER ORDER FOR SUBROUTINE ACFI
C
-----------------------------------------------
DO 210 X = 2, NSTOP
ARG(X - 1) = FLOAT(X)
VALY(X-1) = C(X, Z, T+1)
IF(VALY(X-1) . LE. 0.0) GO TO 205
GO TO 210
205 NDIM = X - 1
GO TO 211
210 CONTINUE
211 CONTINUE
EPS = VALY(1)/1000.
CALL ACFI ( 1.0, ARG, VALY, Y, NDIM, EPS, IER, J )
C(I, Z, T+1) = Y

C
C------------------------------------------------------------------------
C IF THE EXTRAPOLATED VALUE IS TOO HIGH OR IS NEGATIVE, THEN USE
C A PROPORTION EQUATION.
C------------------------------------------------------------------------

479 IF(Y .LE. 0.0 OR Y .GT. C(2,Z,T+1) ) C(1,Z,T+1) = C(2,Z,T+1)*C(2,Z,T+1)
$ /C(3,Z,T+1)
480 GO TO 225
481 220 C(I, Z, T+1) = 0.0
482 225 CONTINUE
483 228 CONTINUE
484 IF(LTYPE.EQ.1.AND.MSL.EQ.NX) GO TO 251
485 DO 250 Z=NZ1,NZM1
486 IF ( C(NX-1, Z, T+1) . LE. 0.00 ) GO TO 245
487 NDIM=3
488 NSTOP=NDIM
489 DO 235 I=1,NSTOP
490 IBACK = NX - I
491 ARG(I) = FLOAT(I+1)
492 VALY(I) = C( IBACK, Z, T+1 )
493 IF( VALY(I) . LE. 0.00 ) GO TO 230
494 GO TO 235
495 230 NDIM = I
496 GO TO 236
497 235 CONTINUE
498 236 CONTINUE
499 EPS = VALY(1)/1000.
CALL ACFI ( 1.0, ARG, VALY, Y, NDIM, EPS, IER, J )
C( NX, Z, T+1) = Y
IF(Y.LE.0.0.OR.Y.GT.C(NX-1,Z,T+1))C(NX,Z,T+1)=C(NX-1,Z,T+1)*
$ C(NX1,Z,T+1)/C(NXZ1,Z,T+1)
GO TO 250
245 C(NX, Z, T+1) = 0.0
250 CONTINUE
251 CONTINUE
IF(NZ.EQ.1.OR.OPTION.EQ.6) RETURN

C-----------------------------------------------
C EXTRAPOLATE THE PROFILE IN THE Z DIRECTION
C-----------------------------------------------
DO 275 X = 1, NX
IF(C(X, Z, T+1) . LE. 0.00) GO TO 270
NDIM=3
NSTOP=NDIM
DO 260 Z = 1, NSTOP
ARG(Z) = FLOAT(Z+1)
VALY(Z) = C(X, Z+1, T+1)
IF( VALY(Z) . LE. 0.00) GO TO 255
GO TO 260
255 NDIM = Z
260 CONTINUE
261 CONTINUE
EPS = VALY(1)/1000.
CALL ACFI ( 1.0, ARG, VALY, Y, NDIM, EPS, IER, J )
C ( X, 1, T+1) = Y
IF(Y.LE.0.0.OR.Y.GT.C(X,2,T+1))C(X,1,T+1)=C(X,2,T+1)*C(X,2,T+1)
$ /C(X,3,T+1)
IF(C(X,1,T+1).LT.0.0) C(X,1,T+1)=0.0
GO TO 275
270 C(X,1,T+1)=0.00
275 CONTINUE
DO 295 X = 1, NX
IF (C(X, NZ-1, T+1) . LE. 0.00) GO TO 292
NDIM=3
NSTOP=NDIM
DO 285 I = 1, NSTOP
ARG(I) = FLOAT(I+1)
IBACK = NZ - I
VALY(I) = C(X, IBACK, T+1)
IF(VALY(I) . LE. 0.00) GO TO 280
GO TO 285
280 NDIM = I
GO TO 286
285 CONTINUE
286 CONTINUE
EPS = VALY(1)/ 1000.
CALL ACFI ( 1.0, ARG, VALY, Y, NDIM, EPS, IER, J)
C(X, NZ, T+1) = Y
IF(Y.LE.0.0.OR.Y.GT.C(X,NZ-1,T+1)) C(X,NZ,T+1)=C(X,NZM1,T+1)*
$ C(X,NZM1,T+1)/C(X,NZM2,T+1)
GO TO 295
292 C(X, NZ, T+1)=0.0
295 CONTINUE
RETURN
END
SUBROUTINE ACFI ( X, ARG, VALY, Y, NDIM, EPS, IER, J )

************ SUBROUTINE ACFI ******
************

PURPOSE
TO INTERPOLATE FUNCTION VALUE Y FOR A GIVEN ARGUMENT VALUE X USING A GIVEN TABLE (ARG,VAL) OF ARGUMENT AND FUNCTION VALUES.

DESCRIPTION OF PARAMETERS
X - THE ARGUMENT VALUE SPECIFIED BY INPUT.
ARG - THE INPUT VECTOR (DIMENSION NDIM) OF ARGUMENT VALUES OF THE TABLE (POSSIBLY DESTROYED).
VAL - THE INPUT VECTOR (DIMENSION NDIM) OF FUNCTION VALUES OF THE TABLE (DESTROYED).
Y - THE RESULTING INTERPOLATED FUNCTION VALUE.
NDIM - AN INPUT VALUE WHICH SPECIFIES THE NUMBER OF POINTS IN TABLE (ARG,VAL).
EPS - AN INPUT CONSTANT WHICH IS USED AS UPPER BOUND FOR THE ABSOLUTE ERROR.
IER - A RESULTING ERROR PARAMETER.
REMARKS

(1) TABLE (ARG, VAL) SHOULD REPRESENT A SINGLE-VALUED
FUNCTION AND SHOULD BE STORED IN SUCH A WAY, THAT THE
DISTANCES ABS(ARG(I) - X) INCREASE WITH INCREASING
SUBSCRIPT I. TO GENERATE THIS ORDER IN TABLE (ARG, VAL),
SUBROUTINES ATSG, ATSM OR ATSE COULD BE USED IN A
PREVIOUS STAGE.

(2) NO ACTION BESIDES ERROR MESSAGE IN CASE NDIM LESS
THAN 1.

(3) INTERPOLATION IS TERMINATED EITHER IF THE DIFFERENCE
BETWEEN TWO SUCCESSIVE INTERPOLATED VALUES IS
ABSOLUTELY LESS THAN TOLERANCE EPS, OR IF THE ABSOLUTE
VALUE OF THIS DIFFERENCE STOPS DIMINISHING, OR AFTER
(NDIM-1) STEPS (THE NUMBER OF POSSIBLE STEPS IS
DIMINISHED IF AT ANY STAGE INFINITY ELEMENT APPEARS IN
THE DOWNWARD DIAGONAL OF INVERTED-DIFFERENCES-SCHEME
AND IF IT IS IMPOSSIBLE TO ELIMINATE THIS INFINITY
ELEMENT BY INTERCHANGING OF TABLE POINTS).
FURTHER IT IS TERMINATED IF THE PROCEDURE DISCOVERS TWO
ARGUMENT VALUES IN VECTOR ARG WHICH ARE IDENTICAL.
DEPENDENT ON THESE FOUR CASES, ERROR PARAMETER IER IS
CODED IN THE FOLLOWING FORM

IER=0 - IT WAS POSSIBLE TO REACH THE REQUIRED
ACCURACY (NO ERROR).
IER=1 - IT WAS IMPOSSIBLE TO REACH THE REQUIRED
ACCURACY BECAUSE OF ROUNDING ERRORS.
IER=2 - IT WAS IMPOSSIBLE TO CHECK ACCURACY BECAUSE
NDIM IS LESS THAN 2, OR THE REQUIRED ACCURACY
COULD NOT BE REACHED BY MEANS OF THE GIVEN
TABLE. NDIM SHOULD BE INCREASED.
IER=3 - THE PROCEDURE DISCOVERED TWO ARGUMENT VALUES
IN VECTOR ARG WHICH ARE IDENTICAL.
METHOD

INTERPOLATION IS DONE BY CONTINUED FRACTIONS AND INVERTED DIFFERENCES SCHEME. ON RETURN Y CONTAINS AN INTERPOLATED FUNCTION VALUE AT POINT X, WHICH IS IN THE SENSE OF REMARK (3) OPTIMAL WITH RESPECT TO GIVEN TABLE. FOR REFERENCE, SEE F. B. HILDEBRAND, INTRODUCTION TO NUMERICAL ANALYSIS, McGRAW-HILL, NEW YORK/TORONTO/LONDON, 1956, PP. 395-406.

DIMENSION ARG(10), VALY(10), VAL(10)
DO 100 M = 1, NDIM
  VAL(M) = VALY(M)
100 CONTINUE
IER=2
IF(NDIM)20, 20, 1
  Y=VAL(1)
DELT2=0.
IF(NDIM-1)20, 20, 2

PREPARATIONS FOR INTERPOLATION LOOP
P2=1.
P3=Y
Q2=0.
Q3=1.
C START INTERPOLATION LLOCIP
566 DO 16 I=2,NDIM
567 II=0
568 P1=P2
569 P2=P3
570 Q1=Q2
571 Q2=Q3
572 Z=Y
573 DELT1=DELT2
574 JEND=I-1

C

C COMPUTATION OF INVERTEC DIFFERENCES
575 3 AUX=VAL(I)
576 DO 10 J=1,JEND
577 H=VAL(I)-VAL(J)
578 IF(ABS(H)-1.E-6*ABS(VAL(I)))4,4,9
579 4 IF(ARG(I)-ARG(J))5,17,5
580 5 IF(J-JEND)8,6,6

C

C INTERCHANGE ROW I WITH ROW I+II
581 6 II=II+1
582 III=I+II
583 IF(III-NDIM)7,7,19
584 7 VAL(I)=VAL(III)
585 VAL(III)=AUX
586 AUX=ARG(I)
587 ARG(I)=ARG(III)
588 ARG(III)=AUX
589 GOTO 3

C
C COMPUTATION OF VAL(I) IN CASE VAL(I) = VAL(J) AND J LESS THAN I-1
590  8 VAL(I) = 1.0E75
591  9 GOTO 10
C
C COMPUTATION OF VAL(I) IN CASE VAL(I) NOT EQUAL TO VAL(J)
592  9 VAL(I) = (ARG(I) - ARG(J)) / H
593 10 CONTINUE
C INVERTED DIFFERENCES ARE COMPUTED
C
C COMPUTATION OF NEW Y
594 11 P3 = VAL(I) * P2 + (X - ARG(I-1)) * P1
595 12 Q3 = VAL(I) * Q2 + (X - ARG(I-1)) * Q1
596 13 IF(Q3) 11, 12, 11
597 14 Y = P3 / Q3
598 15 GOTO 13
599 16 Y = 1.0E75
600 17 DELT2 = ABS(Z - Y)
601 18 IF(DELT2 - EPS) 19, 19, 14
602 19 IF(1 - EPS) 16, 15, 15
603 20 IF(DELT2 - DELT1) 16, 18, 18
604 21 CONTINUE
C END OF INTERPOLATION LCOP
C
C RETURN
C
C THERE ARE TWO IDENTICAL ARGUMENT VALUES IN VECTOR ARG
606 22 IER = 3
607 23 RETURN
C TEST VALUE DELT2 STARTS OSCILLATING

18 Y = Z
19 IER = 0
20 RETURN

C THERE IS SATISFACTORY ACCURACY WITHIN NDIM-1 STEPS

608
609
610
611
612
613
SUBROUTINE PRINT1

************************************************************
***** SUBROUTINE PRINT1 *****
************************************************************

THIS SUBROUTINE CALCULATES THE CONVERSION VALUES FOR MANY OF
THE INPUT PARAMETERS AND PRINTS OUT A SUMMARY OF THE INPUT DATA
AND STABILITY CRITERIA.

-------------------------------------------------------------------
614

INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,CCOUNT,OPTION
REAL KD,KDMAX,KDMIN,KDAYS,LOWER
COMMON/ARRAYS/ C(51,21,2),VX(51,21),VZ(51,21),EX(51,21),EZ(51,21),
1            W(51),CONCIN(51,21),INSECT(51),KDAYS(51),KD(51),
2            DEMAND(51,21),FMT1(40),FMT2(40),FMT3(20)
COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,VMAX,VMIN,
1            KDMAX,KDMIN,ITER,ITER1,ITER2,DELT1,DELT2,TNEXT,
2            TPRINT,XPRINT,STABLX,STABLY,STABLT,R1,R2,VZMAX,SET,
3            VZMIN,EZMAX,EZMIN,DEMEN,DEMIN,NPPAGE,DELTAX,DELTAZ,
4            DELTAT,XSTART,XSF,TSTART,COUNT,IPAGE,UPPER,LOWER,
5            MSU,MSL,TIME,NXM1,NXM2,NZM1,NZM2,MSF,LTYPE,OPTION

619   UNITS1 = 86400. / 5280.
620   UNITS2 = 86400. / (5280. * 5280.)
621   DELDAY = DELTAT / 86400.
622   SECS1 = DELT1 * 86400.
623   SECS2 = DELT2 * 86400.
624   DELMX = DELTAX / 5280.
625   DELMZ = DELTAZ / 5280.
TOTLX=(NX-1)*DELTAX
TOTMX=TOTALX/5280.
TOTALZ=(NZ-1)*DELTAZ
TOTMZ=TOTALZ/5280.
VX1=VXMAX*UNITS1
VX2=VXMIN*UNITS1
VZ1=VZMAX*UNITS1
IF(VZMIN.EQ.1000000.) VZMIN=0.
VZ2=VZMIN*UNITS1
EX1=EXMAX*UNITS2
EX2=EXMIN*UNITS2
EZ1=EZMAX*UNITS2
IF(EZMIN.EQ.1000000.) EZMIN=0.
EZ2=EZMIN*UNITS2
XKD1=KDMAX*86400.
XKD2=KDMIN*86400.
SXM=STABLX/5280.
SZM=STABLZ/5280.
STD=STABL/86400.
PRINT 500
PRINT 505
PRINT 507
PRINT 510
PRINT 507
IF(NZ.EQ.1) PRINT 515
IF(NZ.NE.1) PRINT 520
PRINT 507
IF(LTYPE.EQ.1) GO TO 7
IF(LTYPE.EQ.2) GO TO 5
655    PRINT 530
656    GO TO 10
657    5 PRINT 525
658    GO TO 10
659    7 PRINT 523
660    10 CONTINUE
661    PRINT 507
662    PRINT 535
663    PRINT 507
664    PRINT 505
665    PRINT 540, DELTAX, DELMX
666    PRINT 545, DELTAZ, DELMZ
667    PRINT 550, SECS1, DELT1
668    PRINT 551, ITER1
669    PRINT 552, SECS2, DELT2
670    PRINT 553, ITER2
671    PRINT 555, NX
672    PRINT 560, N2
673    PRINT 565, VXMAX, VX1
674    PRINT 570, VXMIN, VX2
675    PRINT 572, VZMAX, VZ1
676    PRINT 573, VZMIN, VZ2
677    PRINT 575, EXMAX, EX1
678    PRINT 580, EXMIN, EX2
679    PRINT 585, EZMAX, EZ1
680    PRINT 590, EZMIN, EZ2
681    PRINT 600, WMAX
682    PRINT 605, WMIN
683    PRINT 610, XKD1, KOMAX
684    PRINT 615, XKD2, KOMIN
685    PRINT 620, DMAX
686    PRINT 630, NC
C
C XUPEAK AND XLPEAK ARE CALCULATED ASSUMING DISTANCES ARE
C DECREASING IN THE DOWNSTREAM DIRECTION
C
687 XUPEAK = XSTART - (MSU-1)*DELMX
688 XLPEAK = XSTART - (MSL-1)*DELMX
689 PRINT 635, XUPEAK, MSU
690 PRINT 637, XLPEAK, MSL
691 PRINT 500
692 PRINT 505
693 PRINT 502
694 PRINT 640
695 PRINT 645, STABLX, SXM
696 PRINT 650, STABLZ, SZM
697 PRINT 655, STABLT, STD
698 PRINT 660, R1
699 PRINT 665, R2
700 PRINT 505
701 IF(NC.EQ.0) GO TO 100
702 PRINT 500
703 PRINT 700
704 IF(NZ.EQ.1) GO TO 50
705 DO 25 I=1, NC
C
C WHERE IS CALCULATED ASSUMING DISTANCES ARE DECREASING IN
C THE DOWNSTREAM DIRECTION
C
C
WHERE=XSTART-(INSECT(I)-1)*DELMX
PRINT 710, INSECT(I), WHERE, DELTAZ
DO 25 Z=1,NZ
PRINT 715, INSECT(I), Z, CONCIN(I,Z)
25 CONTINUE
GO TO 100
50 DO 75 I=1,NC
WHERE=XSTART-(INSECT(I)-1)*DELMX
PRINT 705, INSECT(I), WHERE, CONCIN(I,1)
75 CONTINUE
100 CONTINUE
PRINT 500
500 FORMAT(IH1)
502 FORMAT(/)
505 FORMAT(T45, 38('**'))
507 FORMAT(T45, '**', T82, '**')
510 FORMAT(T45, '**', T56, 'ESTUARY SIMULATION', T82, '**')
515 FORMAT(T45, '**', T48, 'ONE DIMENSIONAL EXPLICIT METHOD', T82, '**')
520 FORMAT(T45, '**', T48, 'TWO DIMENSIONAL EXPLICIT METHOD', T82, '**')
523 FORMAT(T45, '**', T53, 'CONSTANT CONCENTRATION', T82, '**')
525 FORMAT(T45, '**', T54, 'INSTANTANEOUS RELEASE', T82, '**')
530 FORMAT(T45, '**', T56, 'CONTINUOUS LOADING', T82, '**')
535 FORMAT(T45, '**', T51, 'PROGRAMMER -- JONATHAN YOUNG', T82, '**')
540 FORMAT( '/', T53, 'X INCREMENT =', F7.0, ' FEET (', F8.5, ' MILES $)'), /)
545 FORMAT(T53, 'Z INCREMENT =', F7.0, ' FEET (', E10.3, ' MILES )')
550 FORMAT(T42, 'INITIAL TIME INCREMENT =', F7.0, ' SECONDS (', F6.4, '$ DAYS )')
551 FORMAT(T44, 'NUMBER OF ITERATIONS = ', I5, '/)
552 FORMAT(T42, 'REVISED TIME INCREMENT = ', F7.0, ' SECONDS(', F6.4, 
    $ ' DAYS )'/)
553 FORMAT(T44, 'NUMBER OF ITERATIONS = ', I5, '/)
555 FORMAT(T37, 'NUMBER OF HORIZONTAL POINTS = ', I4, '/)
560 FORMAT(T39, 'NUMBER OF VERTICAL POINTS = ', I4, '/)
565 FORMAT(T37, 'MAXIMUM HORIZONTAL VELOCITY = ', F5.2, ' FEET/SECOND 
    ($', F6.2, ' MILES/DAY )', '/)
570 FORMAT(T37, 'MINIMUM HORIZONTAL VELOCITY = ', F5.2, ' FEET/SECOND 
    ($', F6.2, ' MILES/DAY )', '/)
572 FORMAT(T39, 'MAXIMUM VERTICAL VELOCITY = ', E10.3, ' FEET/SECOND (' 
    $', E10.3, ' MILES/DAY )', '/)
573 FORMAT(T39, 'MINIMUM VERTICAL VELOCITY = ', E10.3, ' FEET/SECOND (' 
    $', E10.3, ' MILES/DAY )', '/)
575 FORMAT(T35, 'MAXIMUM HORIZONTAL DISPERSION = ', F6.0, ' FEET SQUARE 
    $/ SECOND ( ', F5.2, ' MILES SQUARED / DAY )', '/)
580 FORMAT(T35, 'MINIMUM HORIZONTAL DISPERSION = ', F6.0, ' FEET SQUARE 
    $/ SECOND ( ', F5.2, ' MILES SQUARED / DAY )', '/)
585 FORMAT(T37, 'MAXIMUM VERTICAL DISPERSION = ', F9.3, ' FEET SQUARED 
    $/ SECOND ( ', E10.3, ' MILES SQUARED / DAY )', '/)
590 FORMAT(T37, 'MINIMUM VERTICAL DISPERSION = ', F9.3, ' FEET SQUARED 
    $/ SECOND ( ', E10.3, ' MILES SQUARED / DAY )', '/)
600 FORMAT(T51, 'MAXIMUM WIDTH = ', F5.0, ' FEET', '/)
605 FORMAT(T51, 'MINIMUM WIDTH = ', F5.0, ' FEET', '/)
610 FORMAT(T46, 'MAXIMUM DECAY RATE = ', F5.3, ' PER DAY ( ', E10.3, 
    $ ' PER SECOND )', '/)
615 FORMAT(T46, 'MINIMUM DECAY RATE = ', F5.3, ' PER DAY ( ', E10.3, 
    $ ' PER SECOND )', '/)
620 FORMAT(T42, 'OTHER DEMANDS, MAXIMUM = ', F5.2, '/)
630 FORMAT(T26, 'NUMBER OF INITIAL CONCENTRATION VALUES = ', I2, '/)
635 FORMAT(T16, 'INITIAL LOCATION OF UPPERMOST PEAK CONCENTRATION = ',
  $ F6.2, ' MILES (SECTION NUMBER ', I2, ')', '/
637 FORMAT(T16, 'INITIAL LOCATION OF LOWERMOST PEAK CONCENTRATION = ',
  $ F6.2, ' MILES (SECTION NUMBER ', I2, ')'
640 FORMAT(T56, 'STABILITY CRITERIA', '/
645 FORMAT(T35, 'MAXIMUM ALLOWABLE X INCREMENT = ', F7.0, ' FEET (',
  $ F8.5, ' MILES ), /
650 FORMAT(T35, 'MAXIMUM ALLOWABLE Z INCREMENT = ', F7.0, ' FEET (',
  $ F8.5, ' MILES ), /
655 FORMAT(T32, 'MAXIMUM ALLOWABLE TIME INCREMENT = ', F7.0, ' SECONDS
  $ (', E10.3, ' DAYS ),/
660 FORMAT(T38, 'ACTUAL DELTAT/(DELTAX**2.) = ', E10.3, '/
665 FORMAT(T38, 'ACTUAL DELTAT/(DELTAZ**2.) = ', E10.3, '/
700 FORMAT(T25, 'LOCATIONS OF INITIAL CONCENTRATIONS', '/
705 FORMAT(' 'A WASTE SOURCE IS LOCATED AT STATION ', I2, ' (MILE ',
710 FORMAT(' 'AN INITIAL CONCENTRATION IS FOUND AT STATION ', I2,
  1 ' (MILE ', F6.3, ' )', ',', ' THE CONCENTRATIONS AT ', F7.1,
  2 ' FOOT INTERVALS WITH DEPTH ARE')
715 FORMAT(T15, 'C(', I2, ',', I2, ',', I2, ',', I2, ') = ', F9.2, ' PPM')
752 RETURN
753 END
SUBROUTINE PRINT2

************ SUBROUTINE PRINT2 ************

THIS SUBROUTINE PRINTS OUT THE TIME AND THE CONCENTRATION
PROFILE ACCORDING TO A FORMAT PREVIOUSLY READ INTO THE PROGRAM.
THIS SUBROUTINE CAN BE CHANGED ACCORDING TO THE NEEDS OF THE USER.
***WARNING—DO NOT CHANGE ANY VARIABLE OCCURRING IN A COMMON BLOCK.

INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,COUNT,OPTION
REAL KD,KMAX,KDMIN,KDAYS,LOWER
COMMON/ARRAYS/ C(51,21,2),VX(51,21),VZ(51,21),EX(51,21),EZ(51,21),
  W(51),CONCIN(51,21),INSECT(51),KDAYS(51),KD(51),
  DEMAND(51,21),FMT1(40),FMT2(40),FMT3(20)
COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,WMAX,WMIN,
  KMAX,KDMIN,ITER,ITER1,ITER2,DELT1,DELT2,TNEXT,
  TPRINT,XPRINT,STABLX,STABLZ,STABLT,RI,R2,VZMAX,SET,
  VZMIN,EZMAX,EZMIN,DEMAX,DEMIN,NPPAGE,DELTAX,DELTAZ,
  DELTAT,XSTART,XSF,TSTART,COUNT,IPAGE,UPPER,LOWER,
  MSU,MSL,TIME,NXM1,NXM2,NZM1,NZM2,ZSF,LTYPE,OPTION
DIMENSION NZREV(20)
DIMENSION R(51,21)
DAYS = TIME/86400.
HOURS=DAYS*24.
WRITE(6,FMT1) HOURS,DAYS
DO 100 IZ = 1, NZ
NZREV(IZ) = IFIX(DELTZ*(IZ-2))
100 CONTINUE
WRITE(6,FMT2) (NZREV(I), I=2, NZM1)

C-----------------------------------------------
C ALLOW FOR A BACKGROUND CONCENTRATION OF 10.0
C-----------------------------------------------

DO 150 X=1,NX
DO 150 IZ=1,NZ
R(X,IZ) = C(X,IZ,T) + 10.0
150 CONTINUE
XFIRST = XSTART - (NX-1)*DELTAX/5280.
DO 200 X=1,NX,XPRINT
M = NX+1-X.
XMILE = XFIRST + ((X-1)*DELTAX*XPRINT)/5280.
WRITE(6,FMT3) XMILE, {R(M,IZ), IZ=1,NZ}
200 CONTINUE
RETURN
END
SUBROUTINE ERROR(*)

******************************************************************************
**** SUBROUTINE ERROR *****
******************************************************************************

THIS SUBROUTINE CORRECTS SOME OF THE MOST COMMON ERRORS
IN THE INPUT DATA AND PRINTS OUT APPROPRIATE ERROR MESSAGES.

INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,COUNT,OPTION
REAL KD,KDMAX,KDMIN,KDAYS,LOWER
COMMON/ARRAYS/ C(51,21,2),VX(51,21),VZ(51,21),EX(51,21),EZ(51,21),
1 W(51),CONCIN(51,21),INSECT(51),KDAYS(51),KD(51),
2 DEMAND(51,21),FMT1(40),FMT2(40),FMT3(20)
COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,WMAX,WMIN,
1 KDMAX,KDMIN,ITER,ITER1,ITER2,DELT1,DELT2,TNEXT,
2 TPRINT,XPRINT,STABLX,STABLY,STABLT,R1,R2,VZMAX,SET,
3 VZMIN,EZMAX,EZMIN,DEMINS,DEMPAGE,DELTAX,DELTAY,
4 DELTAT,XSTART,XSF,TSTART,COUNT,IPAGE,UPPER,LOWER,
5 MSU,MSL,TIME,NXM1,NXM2,NZM1,NZM2,ZSF,LTYPE,OPTION

IF(OPTION.LT.8) GO TO 200
PRINT 181
181 FORMAT(///,** PROGRAM TERMINATED BECAUSE OPTION IS GREATER THAN 7****,//,1H1)
RETURN
900 200 IF(LTYPE.LT.3) GO TO 300
PRINT 201
201 FORMAT(/,, '****PROGRAM TERMINATED BECAUSE LTYPE IS GREATER THAN OR EQUAL TO 3.,/, 'IF LTYPE EQUALS 3, THE USER MUST SUPPLY A $ SUBROUTINE FOR THE LOADING CONDITIONS****',/,1H1)

RETURN

300 CONTINUE

IF(ITER.NE.0) GO TO 820

C

C "TERMINATE PROGRAM WITH PRINTED MESSAGE IF INPUT DATA
C "VIOLATES STABILITY CRITERIA
C"

PRINT 815

815 FORMAT(/,,1X,10('**'),'PROGRAM TERMINATED BECAUSE STABILITY CONDITIONS WERE VIOLATED****',/,1H1)

RETURN

820 CONTINUE

RETURN

END
Input Data for MASSTRANS-I and MASSTRANS-II

Each line represents a new card unless single spaced.

1.
33, 11
0.25, 5.0
0.006944444, 2
0.0, 0
21, 1
17., 0.166667, 1, 1, 1
2, 2
33*0.0, 33*0.366667, 33*0.420, 33*0.440, 33*0.440, 33*0.433333,
33*0.389333, 33*0.27333, 33*0.233333, 33*0.233333, 33*0.0
363*450.
525., 568., 611., 654., 697
756., 814., 872., 931., 863., 796., 728., 661., 657., 654.,
650., 647., 633., 619., 605.,
591, 586., 582.
33*2.50
363*0.0
1
3.2, 3.2, 2.0, 1.7, 2.0, 3.0, 3.0, 1.8, 0.2, 0.0, 0.0
2
4.0, 4.0, 3.5, 2.0, 3.0, 4.0, 4.0, 2.1, 0.3, 0.0, 0.0
3
6.2, 6.2, 5.0, 4.0, 4.0, 5.0, 5.0, 2.3, 0.4, 0.0, 0.0
4
9.5, 9.5, 7.0, 6.0, 6.0, 7.0, 6.0, 2.5, 0.5, 0.0, 0.0
5
90.0, 90.0, 9.0, 9.0, 9.0, 10.0, 7.0, 2.7, 0.6, 0.0, 0.0
6
115.0, 115.0, 70.0, 40.0, 30.0, 25.0, 9.0, 3.0, 0.8, 0.0, 0.0
7
165., 165., 210., 290., 120., 55., 12., 3.5, 1.0, 0.0, 0.0
8
290., 290., 490., 440., 165., 70., 20., 4., 0.8, 0.0, 0.0
9
740., 740., 890., 320., 140., 55., 14., 3.3, 0.6, 0.0, 0.0
10
1990., 1990., 540., 215., 90., 30., 10., 2.6, 0.4, 0.0, 0.0
11
990., 990., 290., 120., 50., 15., 7., 2., 0.3, 0.0, 0.0
12
290., 290., 190., 75., 20., 8., 4., 1.6, 0.2, 0.0, 0.0
13
165., 165., 75., 25., 9., 5., 2., 1.2, 0.1, 0.0, 0.0
14
90., 90., 10., 9., 5., 2., 1.5, 0.8, 0.0, 0.0, 0.0
15
20., 20., 8., 7., 2., 1.7, 1.0, 0.4, 0.0, 0.0, 0.0
16
9., 9., 7., 5., 1.8, 1.5, 0.7, 4*0.0
17
7., 7., 6., 4., 1.6, 1.2., 0.4, 4*0.0
18
5., 5., 5., 3., 1.4, 1., 0.2, 4*0.0
19
4., 4., 4., 2., 1.3, 0.7, 5*0.0
20
3., 3., 3., 1.8, 1.2, 0.5, 5*0.0
21
2.5, 2.5, 2.0, 1.6, 1.1, 0.2, 5*0.0
363*0.0
33*0.0033333, 33*0.0033333, 33*0.00382, 33*0.004, 33*0.004,
33*0.00394.
33*0.00354, 33*0.002485, 33*0.00212, 33*0.00212, 33*0.00212
(/, /, 12X, '100(', '1'), /, 12X, '1', /, 12X,'1', 2X,
'CONCENTRATIONS AT TIME = ',',
F7.2, ' HOURS ( ', F6.4, ' DAYS')', /, 12X, '1', /, 12X,
100('1'))
12X, '1', /, 12X, '1', 42X, 'DEPTH', /, 12X, '1', /, 12X,
'1', 3X, 'IMAGE', 4X,
9(12, ' FEET '), 'IMAGE', /, 1X, 111('1'), /, 12X, '1')
(1X, 'MILE', F5.2, 1X, '1', 2X, 11(F6.1, 4X))

// END OF DATA FOR MASTRANS-1 //
33*0.0, 33*0.366667, 33*0.420, 33*0.440, 33*0.440, 33*0.433333,
33*0.389333, 33*0.273333, 33*0.233333, 33*0.233333, 33*0.0
363*450.
363*0.0
33*0.00333333, 33*0.00333333, 33*0.00382, 33*0.004, 33*0.004,
33*0.00394,
33*0.00354, 33*0.002485, 33*0.00212, 33*0.00212, 33*0.00212

// THIS PAGE OF DATA USED FOR MASSTRANS-II ONLY //
COMPUTER PROGRAM FOR

MASSTRANS-II

Object Code = 67,944 bytes
Array Area = 62,924 bytes
Total = 130,868 bytes
MASSTRAIN-II

ESTUARY MASS TRANSPORT MODEL

TWO DIMENSIONAL NUMERICAL ANALYSIS - IMPLICIT METHOD

LOGIC AND PROGRAMMING BY P. JONATHAN YOUNG
C/O ENVIRONMENTAL ENGINEERING DIVISION
CIVIL ENGINEERING DEPARTMENT
TEXAS A&M UNIVERSITY
COLLEGE STATION, TEXAS 77840

THIS COMPUTER PROGRAM WAS DESIGNED PRIMARILY TO ANALYZE THE MASS TRANSPORT OF DISSOLVED MATERIALS IN A PARTIALLY STRATIFIED ESTUARY. THESE MATERIALS MAY BE CONSERVATIVE OR NON-CONSERVATIVE. A CRANK-NICOLSON IMPLICIT FINITE DIFFERENCE APPROXIMATION TO THE ESTUARY MASS TRANSPORT EQUATION FORMS THE BASIS OF THE COMPUTATIONS.
THIS PROGRAM WAS DEVELOPED TO ANALYZE ESTUARIES WHOSE
CHARACTERISTICS DO NOT VARY SIGNIFICANTLY WITH WIDTH. THE WIDTH
OF THE ESTUARY MAY BE VARIED THROUGHOUT BUT THE CONCENTRA-
TIONS OF DISSOLVED MATERIALS AT EACH CROSS SECTION ARE CONSIDERED
TO BE UNCHANGING IN THE LATERAL DIRECTION. ALL PHYSICAL AND
HYDRODYNAMIC CHARACTERISTICS MAY VARY WITH TIME IN THE LONGITU-
DINAL AND VERTICAL DIRECTIONS. THE PROGRAM CAN BE APPLIED WITH
EQUAL EASE TO THE TWO HORIZONTAL DIRECTIONS, ALLOWING DEPTH TO
VARY RATHER THAN WIDTH.

CONCENTRATION PROFILES CAN BE CALCULATED BY THIS PROGRAM FOR
CONTINUOUS OR INSTANTANEOUS RELEASES. INPUT DATA MAY INCLUDE GRID
DIMENSIONS, DISTANCE INCREMENTS, TIME INCREMENTS, WIDTHS, LOADING
PARAMETERS, VELOCITIES, DISPERSION COEFFICIENTS, DECAY RATES,
BENTHAL DEMANDS, AND OTHER SOURCE OR SINK TERMS.

THE TIME INCREMENTS CAN BE INCREASED OR DECREASED AT ANY TIME
DURING THE CALCULATION OF THE CONCENTRATION PROFILE. THE DIS-
TANCES BETWEEN GRID POINTS CAN BE INCREASED AT ANY TIME, AND A
ROUTINE WITHIN THE PROGRAM WILL CHOOSE THE APPROPRIATE VALUES FROM
THE PREVIOUSLY CALCULATED PROFILE AND WILL PLACE THESE VALUES IN
THE DESIRED LOCATIONS IN THE NEW GRID SYSTEM. THE NUMBER OF GRID
POINTS MAY BE INCREASED OR DECREASED AT ANY TIME. LIKewise, AT
ANY TIME DURING THE CALCULATION, A NEW SET OF DATA FOR PHYSICAL
AND HYDRODYNAMIC CONDITIONS MAY BE READ INTO THE PROGRAM.
A user of this program must be familiar with the limitations on accuracy and stability inherent in the type of numerical procedure used in these calculations. A subroutine within the program prints out the proper increments which will insure stability. Another subroutine extrapolates the concentrations at the boundaries--several methods can be used for these extrapolations depending on the type of profile being analyzed and the choice of the user. A subroutine is also included which prints out error messages and terminates the program if certain inconsistencies occur in the input data.

This program was developed primarily to analyze partially stratified estuaries which have been dredged out to a fairly constant depth at the centerline of the channel--these estuaries are common in the Gulf Coast region. Application of this program to partially stratified estuaries with variable depths would require moderate revisions to the program and would make the program estuary-dependent.

This computer program can also be applied to estuaries which are well-mixed in the vertical direction. This option allows for varying width or varying depth and uses most of the routines available to the two-dimensional analysis.

Questions regarding this program may be referred to Jonathan Young at Hydrosience, Inc., 363 Old Hook Road, Westwood, New Jersey 07675 Phone 201 / 666-2600.
***INPUT VARIABLES***

**SET** = DATA SET NUMBER
--CAN BE EQUAL TO 1 IF NO NEW GRID DIMENSIONS ARE TO BE READ IN--MUST BE GREATER THAN 1 IF GRID DIMENSIONS ARE TO BE CHANGED

**********NOTE--OPTIONAL GRID ADJUSTMENT PARAMETERS ARE INSERTED HERE. THESE ARE EXPLAINED BELOW**********

**NX** = NUMBER OF GRID POINTS IN THE X DIRECTION

**NZ** = NUMBER OF GRID POINTS IN THE Z DIRECTION

**DELMX** = X INCREMENT (MILES)

**DELTZ** = Z INCREMENT (FEET)

**DELT1** = FIRST TIME INCREMENT (DAYS)

**ITER1** = NUMBER OF ITERATIONS USING DELT1

**DELT2** = SECOND TIME INCREMENT (DAYS)

**ITER2** = NUMBER OF ITERATIONS USING DELT2

**NC** = NUMBER OF GRID POINTS IN THE X DIRECTION WHERE CONCENTRATIONS ARE ADDED WITH THE DATA SET

**NS** = NUMBER OF POINT SOURCES

**XSTART** = MILEAGE AT UPPERMOST GRID POINT

**TSTART** = TIME TO BE ADDED TO THE TOTAL TIME AT THE FIRST ITERATION OF THE DATA SET BEING PROCESSED (DAYS)

**XPRINT** = PRINTING INTERVAL FOR X DIRECTION (INTEGER)

**TPRINT** = PRINTING INTERVAL FOR TIME (INTEGER)

**NPPAGE** = NUMBER OF GRIDS PRINTED PER PAGE OF OUTPUT
**TYPE** = TYPE OF LOADING (INSTANTANEOUS OR CONTINUOUS)
1 MEANS CONSTANT CONCENTRATION AT OUTFALL
   (THIS CAN BE USED FOR STEADY-STATE PROFILES)
2 MEANS AN INSTANTANEOUS RELEASE
3 MEANS CONTINUOUS LOADING

**OPTION** = OPTION FOR BOUNDARY CONDITIONS
1 MEANS NO TRANSFER ACROSS Z BOUNDARIES AND
   EXPONENTIAL EXTRAPOLATION IN X DIRECTION
2 MEANS NO TRANSFER ACROSS Z BOUNDARIES AND
   CONSTANT SLOPE EXTRAPOLATION IN X DIRECTION
3 MEANS A SYMMETRICAL DISTRIBUTION WITH
   EXPONENTIAL EXTRAPOLATION
4 MEANS A CONSTANT SLOPE EXTRAPOLATION IN
   BOTH DIRECTIONS
5 MEANS A SPECIAL CASE DEFINED BY THE USER

**VX(X,Z)** = HORIZONTAL VELOCITY (FT/SEC)
**EX(X,Z)** = HORIZONTAL DISPERSION (FT**2/SEC)
**W(X)** = WIDTH (FEET) FOR THE TWO DIMENSIONAL CASE OR
   CROSS SECTİONAL AREA (FT**2) FOR THE ONE
   DIMENSIONAL CASE
**KDOAYS(X)** = DECAY RATE (PER DAY)
**DEMAND(X,Z)** = OTHER SOURCES AND SINKS (CONCENTRATION UNITS)
**INSECT(I)** = SECTİCKS INTO WHICH ARE ADDED LOADS
   OR CONCENTRATIONS
**CONCIN(I,J)** = INITIAL CONCENTRATIONS ASSOCIATED WITH INSECT(I)

***************NOTE—INITIAL CONCENTRATIONS MAY ALSO BE INPUT
   THROUGH THE BLOCK DATA SUBPROGRAM***************
***********NOTE--VALUES FOR VZ AND EZ ARE TO BE INCLUDED
WITH THE DATA ONLY IF NZ IS GREATER THAN 1

VZ(X,Z) = VERTICAL VELOCITIES (FT/SEC)
EZ(X,Z) = VERTICAL DISPERSION (FT**2/SEC)

VXNEXT(X,Z) = HORIZONTAL VELOCITY,NEXT TIME STEP (FT/SEC)
EXNEXT(X,Z) = HORIZONTAL DISPERSION,NEXT TIME STEP (FT**2/SEC)
VZNEXT(X,Z) = VERTICAL VELOCITY,NEXT TIME STEP (FT/SEC)
EZNEXT(X,Z) = VERTICAL DISPERSION,NEXT TIME STEP (FT**2/SEC)

FMT1(I) = FORMAT FOR HEADING FOR COMPUTED PROFILES
FMT2(I) = FORMAT FOR DEPTH NOTATION FOR PROFILES
FMT3(I) = FORMAT FOR OUTPUT OF COMPUTED PROFILES

*** GRID ADJUSTMENT PARAMETERS ***

NXSKIP = NUMBER OF X INTERVALS SKIPPED IN OLD GRID
NZSKIP = NUMBER OF Z INTERVALS SKIPPED IN OLD GRID
NXTAKE = LOWEST GRID NUMBER IN X DIRECTION IN THE OLD GRID
FROM WHICH A CONCENTRATION IS TAKEN
NZTAKE = LOWEST GRID NUMBER IN Z DIRECTION IN THE OLD GRID
FROM WHICH A CONCENTRATION IS TAKEN
NXPUT = LOWEST GRID NUMBER IN THE X DIRECTION IN THE NEW
GRID INTO WHICH A CONCENTRATION IS PLACED
NZPUT = LOWEST GRID NUMBER IN THE Z DIRECTION IN THE NEW
GRID INTO WHICH A CONCENTRATION IS PLACED
***************
***** MAIN PROGRAM *****
***************

THE MAIN PROGRAM CALLS FOR THE DATA TO BE READ IN, THEN CALLS FOR THE STABILITY CONDITIONS TO BE EVALUATED, AND NEXT CALLS FOR THE PRINTING OF A SUMMARY OF INPUT DATA. THE NUMERICAL ANALYSIS SUBROUTINE IS THEN CALLED FOR EITHER THE ONE-DIMENSIONAL OR TWO-DIMENSIONAL CASE. IF THE TIME INCREMENT IS TO BE CHANGED DURING THIS PART OF THE PROGRAM, THE APPROPRIATE PARAMETERS ARE ADJUSTED. STABILITY IS AGAIN CHECKED AND THE NUMERICAL ANALYSIS IS CONTINUED. NEXT, A NEW DATA SET MAY BE READ TO CONTINUE THE ANALYSIS. ANY NUMBER OF DATA SETS MAY BE PROCESSED. WHEN NO MORE DATA SETS ARE AVAILABLE, THE PROGRAM IS TERMINATED.

THE INPUT AND OUTPUT OF DATA CAN BE CONTROLLED IN THREE WAYS-- THE INPUT DATA ITSELF, THE BLOCK DATA SUBROUTINE, AND THE SUBROUTINE PRINT2. NO OTHER PART OF THE PROGRAM SHOULD REQUIRE CHANGING.

THE FORTRAN USED IN THIS PROGRAM USES SEVERAL OF THE OPTIONS AVAILABLE IN WATFOR AS DESCRIBED IN THE FOLLOWING TEXT**FORTRAN IV WITH WATFOR AND WATFIV**BY CRESS, DIRKSEN, & GRAHAM, PRENTICE-HALL, INC., 1970. THE PROGRAM CAN BE RUN ON A COMPUTER WITH A WATFOR OR WATFIV COMPILER. THE SYNTAX HAS BEEN KEPT COMPATIBLE WITH FORTRAN G EXCEPT FOR THE UNFORMATTED READ AND PRINT STATEMENTS.
INTEGER X,Z,T,TNEXT,PRINT,PRINT1,COUNT,OPTION
REAL K0,KDAMAX,KDMIN,KDAYS,LOWER
COMMON/ARRAYS/ C(51,21,2),VX(51,21),VZ(51,21),EX(51,21),EZ(51,21),
1 W(51),CONCIN(51,21),INSECT(51),KDAYS(51),KD(51),
2 DEMAND(51,21),FMT1(440),FMT2(440),FMT3(20)
COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,WMAX,WMIN,
1 KMAX,KDMIN,ITER2,ITER1,DELT1,DELT2,TNEXT,
2 TPRINT,PRINT1,STABLX,STABLZ,STABLT,R1,R2,VZMAX,SET,
3 VZMIN,EZMAX,EZMIN,DEMAX,DEMIN,NPAGE,DELTAX,DELTZ,
4 DELTAT,XSTART,XSF,TSTART,COUNT,IPAGE,UPPER,LOWER,
5 MSU,MSL,TIME,NX1,NX2,NZ1,NZ2,ZSF,LTYPE,OPTION
COMMON/IMPLCT/EXNEXT(51,21),EZNEXT(51,21),CSTAR(51,21),
1 VXNEXT(51,21),VZNEXT(51,21),A(51),B(51),
2 G(51),D(51),V(51)

10 CALL IDATA(1000)
11 CALL STABL1
12 CALL PRINT1
13 CALL PRINT2
14 CALL ERROR1(1000)
15 IF (NZ.EQ.1) GO TO 310
16 CALL TWODIM
17 GO TO 315
18 CALL ONEDIM
19 IF (ITER2.EQ.0) GO TO 10
20 ITER=ITER2
21 ITER2=0
22 DELTAT=DELT2*86400.
23 GO TO 25
24 1000 CONTINUE
25 STOP
26 END
BLOCK DATA

********************
***** BLOCK DATA *****
********************

THIS ROUTINE INITIALIZES SELECTED VARIABLES IN THE COMMON BLOCKS. THIS SUBPROGRAM MAY BE USED TO ESTABLISH INITIAL CONCENTRATION PROFILES IF THE USER DESIRES.

```
994 INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,COUNT,OPTION

996 REAL KD,KMAX,KMIN,KDAYS,LOWER

997 COMMON/ARRAYS/ C(51,21,2), VX(51,21), VZ(51,21), EX(51,21), EZ(51,21),
1 W(51), CCNIN(51,21), INSECT(51), KDAYS(51), KD(51),
2 DEMAND(51,21), FMT1(40), FMT2(40), FMT3(20)

999 COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,WMAX,WMIN,
1 KMAX,KMIN,ITER,ITER1,ITER2,DELT1,DELT2,TNEXT,
2 TPRINT,XPRINT,STABLEX,STABLEZ,STABLET,R1,R2,VZMAX,SET,
3 VZMIN,EZMAX,EZMIN,DEM0,DEMIN,NPPAGE,DELTAZ,DELTAT,
4 DELTA,TSTART,XSF,TSTART,COUNT,PAGE,UPPER,LOWER,
5 MSU,MSL,TIME,NXM1,NXM2,NZM1,NZM2,ZSF,LTYP0,OPTION

999 COMMON/IMPLICIT/EXNEXT(51,21), EZNEXT(51,21), CSTAR(51,21),
1 VXNEXT(51,21), VZNEXT(51,21), A(51), B(51),
2 G(51), D(51), V(51)
```
C
C ESTABLISH INITIAL CONCENTRATION IF DESIRED
C
1000   DATA C/2142*0.0/
1001   DATA VX,VZ,EX,EZ,CONCIN,DEMAND/6426*0.0/
1002   DATA W,KDAYS,KD/153*0.0/
1003   DATA EXMAX,EZMAX,VXMAX,VZMAX,WMAX,KDMAX,DEMAX,TIME/8*0.0/
1004   DATA EXMIN,EZMIN,VXMIN,VZMIN,WMIN,KDMIN,DEMIN/7*1000000.0/
1005   DATA INSECT/51*0/
1006   DATA IPAGE/1/
1007   DATA T/1/,TNEXT/2/
1008   DATA VXNEXT,VZNEXT,EXNEXT,EZNEXT,CSTAR/5355*0.0/
1009   DATA A,B,G,D,V/255*0.0/
1010   END
**SUBROUTINE IDATA(*)**

****************************************************
** ***** SUBROUTINE IDATA *****
** **************************************************

THIS SUBROUTINE READS IN THE APPROPRIATE DATA SET AND ADJUSTS
THE GRID SIZE IF NECESSARY. THE LOCATIONS OF THE UPPERMOST AND
LOWERMOST PEAKS ARE DETERMINED, AND THE MINIMUM AND MAXIMUM VALUES
FOR THE INPUT PARAMETERS ARE CALCULATED. ALL OF THE INPUT DATA IS
PRINTED OUT UNFORMATTED. THE FORMAT IS READ IN FOR THE PRINTING
OUT OF THE CALCULATED CONCENTRATIONS.

**---------------------------------------------------
** INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,CCOUNT,OPTION
** REAL KD,KDMAX,KDMIN,KDAYS,LOWER
** COMMON/ARRAYS/ C(51,21,2),VX(51,21),VZ(51,21),EX(51,21),EZ(51,21),
** 1 W(51),CGNCIN(51,21),INSECT(51),KDAYS(51),KD(51),
** 2 DEMAND(51,21),FMT1(40),FMT2(40),FMT3(20)
** COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,WMAX,WMIN,
** 1 KDMAX,KDMIN,ITER,ITER1,ITER2,DELT1,DELT2,TNEXT,
** 2 TPRINT,XPRINT,STABLX,STABLZ,STABLRT1,R2,VZMAX,SET,
** 3 VZMIN,EZMAX,EZMIN,DEMIX,DEMIN,NPPAGE,DELTAX,DELTAZ,
** 4 DELTAT,XSTART,XSF,TSTART,COUNT,IPAGE,UPPER,LOWER,
** 5 MSU,MSL,TIME,NX1,NX2,NZ1,NZ2,ZSF,LTYPE,OPTION
** COMMON/IMPLCT/EXNEXT(51,21),EZNEXT(51,21),CSTAR(51,21),
** 1 VXNEXT(51,21),VZNEXT(51,21),A(51),B(51),
** 2 G(51),D(51),V(51)
READ(5,12,END=13) SET
12 FORMAT(F10.0)
GO TO 14
13 CONTINUE
RETURN
14 CONTINUE
IF (SET.EQ.1.) GO TO 400

C
CCHOOSE CONCENTRATIONS FROM THE OLD GRID TO PUT INTO THE
C  APPROPRIATE PLACES IN THE NEW GRID. THE GRID CAN ONLY BE
C   CHANGED IF LTYPE EQUALS 2. OTHERWISE, THE DATA CARD IS READ
C   BUT IS NOT IMPLEMENTED.
C
READ ,NXSKIP,NZSKIP,NXTAKE,NZTAKE,NXPUT,NZPUT
IF(LTYPE.NE.2) GO TO 400
NXSTOP=(NX-1)/NXSKIP+1
NZSTOP=(NZ-1)/NZSKIP+1
IF(NZ.EQ.1) NZSTOP=1
DO 350 I=1,NXSTOP
X=NXPUT-1+I
II=NXTAKE+NXSKIP*(I-1)
DO 350 J=1,NZSTOP
Z=NZPUT-1+J
JJ=NXTAKE+NZSKIP*(J-1)
C(X,Z,TNEXT)=C(II,JJ,T)
350 CONTINUE
400 CONTINUE
C----------------------------------------
C       READ AND PRINT INPUT DATA FOR HORIZONTAL DIRECTION
C----------------------------------------
   1038      READ,NX,NZ
   1039      READ,DELMX,DELTAZ
   1040      READ,DELT1,ITER1
   1041      READ,DELT2,ITER2
   1042      READ,NC,NS
   1043      READ,XSTART,TSTART,XPRINT,TPRINT,NPPAGE
   1044      READ,LTYPE,OPTION
   1045      PRINT 1001
   1046  1001 FORMAT( 1I1,10X,'INPUT DATA',/,,
   1047            1X,NX,NZ, DELMX,DELTAZ, DELT1,ITER1*)
   1048      PRINT,NX,NZ,DELMX,DELTAZ,DELT1,ITER1
   1049      PRINT 1002
   1050  1002 FORMAT(/,,1X,'DELT2,ITER2, NC,NS, XSTART,TSTART*)
   1051      PRINT, DELT2,ITER2,NC,NS,XSTART,TSTART
   1052      PRINT 1003
   1053  1003 FORMAT(/,,1X,'XPRINT,TPRINT,NPPAGE, LTYPE,OPTION*)
   1054      PRINT, XPRINT,TPRINT,NPPAGE,LTYPE,OPTION
   1055      IF(SET.EQ.1.) GO TO 500
   1056     DO 375 Z=1,NZ
   1057     DO 375 X=1,NX
   1058     C(X,Z,T)=C(X,Z,TNEXT)
   1059     CONTINUE
   1060     375 CONTINUE
   1061     500 CONTINUE
   1062      NXM1=NX-1
   1063      NXM2=NX-2
   1064      NZM1=NZ-1
NZ = NZ - 2
DELTAT = DELT1 * 86400.
DELTAX = DELMX * 5280.
XSF = XSTART * 5280.
ITER = ITER1
TIME = TIME + TSTART * 86400.
COUNT = (TSTART / DELTAT + 0.01)
COUNT = COUNT - (COUNT / TPRINT) * TPRINT
READ, ((VX(X,Z), X=1,NX), Z=1,NZ)
PRINT 2
2 FORMAT(/, 1X, *((VX(X,Z), X=1,NX), Z=1,NZ)*)
PRINT, ((VX(X,Z), X=1,NX), Z=1,NZ)
READ, ((EX(X,Z), X=1,NX), Z=1,NZ)
PRINT 3
3 FORMAT(/, 1X, *((EX(X,Z), X=1,NX), Z=1,NZ)*)
PRINT, ((EX(X,Z), X=1,NX), Z=1,NZ)
READ, (W(X), X=1,NX)
PRINT 4
4 FORMAT(/, 1X, *((W(X), X=1,NX)*)
PRINT, (W(X), X=1,NX)
READ, (KAYS(X), X=1,NX)
PRINT 5
5 FORMAT(/, 1X, *((KAYS(X), X=1,NX)*)
PRINT, (KAYS(X), X=1,NX)
READ, ((DEMAND(X,Z), X=1,NX), Z=1,NZ)
PRINT 6
6 FORMAT(/, 1X, *((DEMAND(X,Z), X=1,NX), Z=1,NZ)*)
PRINT, ((DEMAND(X,Z), X=1,NX), Z=1,NZ)
IF(NC.EQ.0) GO TO 24
DO 10 I = 1, NC
1094       READ, INSECT(I)
1095       PRINT 7
1096       7 FORMAT(//, 1X, 'INSECT(I)')
1097       PRINT, INSECT(I)
1098       READ, (CONCIN(I,J), J=1,NZ)
1099       PRINT 8
1100       8 FORMAT(//, 1X, '(CONCIN(I,J), J=1,NZ)')
1101       PRINT, (CONCIN(I,J), J=1,NZ)
1102      10 CONTINUE
1103      DO 15 I=1, NC
1104      DO 15 Z=1, NZ
1105      C(INSECT(I),Z,T)=CONCIN(I,Z)+C(INSECT(I),Z,T)
1106     15 CONTINUE

C-----------------------------------------------------------------------
C DETERMINE LOCATIONS OF UPPERMOST AND LOWERMOST
C PEAK CONCENTRATIONS
C-----------------------------------------------------------------------

1107      AMAX1=0.0
1108      PEAKU=0.0
1109      AMAX2=0.0
1110      PEAKL=0.0
1111      UPPER=XS
1112      LOWER=XS
1113      MSU=1
1114      MSL=NX
1115      DO 17 I=1, NC
1116      DO 16 J=1, NZ
1117      IF(CONCIN(I,J).LT.PEAKU) GO TO 16
1118      PEAKU=CONCIN(I,J)
1119     16 CONTINUE
1120      IF(PEAKU.LE.AMAX1) GO TO 20
1121      AMAX1=PEAKU
1122      MSU=INSECT(I)
1123      UPPER=(MSU-1)*DELTAX+XSF
1124      17 CONTINUE
1125      20 CONTINUE
1126      DO 22 II=1,NC
1127          I=NC+1-II
1128      DO 21 J=1,NZ
1129      IF(CONCIN(I,J).LT.PEAKL) GO TO 21
1130      PEAKL=CONCIN(I,J)
1131      21 CONTINUE
1132      IF(PEAKL.LE.AMAX2) GO TO 24
1133      AMAX2=PEAKL
1134      MSL=INSECT(I)
1135      LOWER=(MSL-1)*DELTAX+XSF
1136      22 CONTINUE
1137      24 CONTINUE
C
C-----------------------------------------------
C DETERMINE MAXIMUM AND MINIMUM VALUES FOR INPUT DATA
C-----------------------------------------------
1138      DO 25 X=1,NX
1139          KD(X)=KDAYS(X)/86400.
1140      IF(W(X).GT.WMAX)WMAX=W(X)
1141      IF(W(X).LT.WMIN)WMIN=W(X)
1142      IF(KD(X).GT.KDMAX)KDMAX=KD(X)
1143      IF(KD(X).LT.KDMIN)KDMIN=KD(X)
1144      DO 25 Z=1,NZ
ABS\text{VX} = \text{ABS}(\text{VX}(x,z))

\text{IF}(\text{ABS\text{VX}.GT.\text{VXMAX}}) \text{VXMAX} = \text{ABS\text{VX}}

\text{IF}(\text{ABS\text{VX}.LT.\text{VXMIN}}) \text{VXMIN} = \text{ABS\text{VX}}

\text{IF}(\text{EX}(x,z).GT.\text{EXMAX}) \text{EXMAX} = \text{EX}(x,z)

\text{IF}(\text{EX}(x,z).LT.\text{EXMIN}) \text{EXMIN} = \text{EX}(x,z)

\text{IF}(\text{DEMAND}(x,z).GT.\text{DEMAX}) \text{DEMAX} = \text{DEMAND}(x,z)

\text{IF}(\text{DEMAND}(x,z).LT.\text{DEMIN}) \text{DEMIN} = \text{DEMAND}(x,z)

25 \text{CONTINUE}

\text{IF}(\text{NZ}.EQ.1) \text{GO TO 200}

\begin{verbatim}
C------------------------
C READ AND PRINT DATA FOR VERTICAL DIRECTION
C------------------------
1154 \text{READ}(),((\text{VZ}(x,z),x=1,NX),z=1,NZ)
1155 \text{PRINT} 30
1156 30 \text{FORMAT}(/,,1X,'(\text{VZ}(x,z),x=1,NX),z=1,NZ)'
1157 \text{PRINT}(),((\text{VZ}(x,z),x=1,NX),z=1,NZ)
1158 \text{READ}(),((\text{EZ}(x,z),x=1,NX),z=1,NZ)
1159 \text{PRINT} 32
1160 32 \text{FORMAT}(/,,1X,'(\text{EZ}(x,z),x=1,NX),z=1,NZ)'
1161 \text{PRINT}(),((\text{EZ}(x,z),x=1,NX),z=1,NZ)
\end{verbatim}

\begin{verbatim}
C------------------------
C DETERMINE MAXIMUM AND MINIMUM VALUES FOR INPUT DATA
C------------------------
1162 \text{DO} 120 x=1,NX
1163 \text{DO} 120 z=1,NZ
1164 \text{ABSVZ} = \text{ABS}(\text{VZ}(x,z))
1165 \text{IF}(\text{ABSVZ}.GT.\text{VZMAX}) \text{VZMAX} = \text{ABSVZ}
1166 \text{IF}(\text{ABSVZ}.LT.\text{VZMIN}) \text{VZMIN} = \text{ABSVZ}
\end{verbatim}
IF(EZ(X,Z) > EZMAX) EZMAX = EZ(X,Z)
IF(EZ(X,Z) < EZMIN) EZMIN = EZ(X,Z)
120 CONTINUE
200 CONTINUE

C -----------------------------------------------
C READ AND PRINT FORMATS FOR OUTPUT
C -----------------------------------------------
READ 205, (FMT1(I), I=1,40)
PRINT 202
202 FORMAT(//,1X, 'FORMAT FOR OUTPUT')
PRINT 205, (FMT1(I), I=1,40)
205 FORMAT (20A4,/,20A4)
READ 205,(FMT2(I), I=1,40)
PRINT 205, (FMT2(I), I=1,40)
READ 207,(FMT3(I), I=1,20)
PRINT 207, (FMT3(I), I=1,20)
207 FORMAT(20A4)

C -----------------------------------------------
C READ AND PRINT VELOCITIES AND DISPERSION COEFFICIENTS FOR
C NEXT TIME STEP
C -----------------------------------------------
READ, ((VXNEXT(X,Z), X=1,NX), Z=1,NZ)
PRINT 210
210 FORMAT(//,1X,'((VXNEXT(X,Z), X=1,NX), Z=1,NZ)')
PRINT, ((VXNEXT(X,Z), X=1,NX), Z=1,NZ)
READ, ((EXNEXT(X,Z), X=1,NX), Z=1,NZ)
PRINT 215
215 FORMAT(//,1X,'((EXNEXT(X,Z), X=1,NX), Z=1,NZ)')
1188 PRINT, ((EXNEXT(X,Z), X=1,NX), Z=1,NZ)
1189 IF(NZ.EQ.1) GO TO 250
1190 READ, ((VZNEXT(X,Z), X=1,NX), Z=1,NZ)
1191 PRINT 220
1192 220 FORMAT(/,,1X,'((VZNEXT(X,Z), X=1,NX), Z=1,NZ)*')
1193 PRINT, ((VZNEXT(X,Z), X=1,NX), Z=1,NZ)
1194 READ, ((EZNEXT(X,Z), X=1,NX), Z=1,NZ)
1195 PRINT 225
1196 225 FORMAT(/,,1X,'((EZNEXT(X,Z), X=1,NX), Z=1,NZ)*')
1197 PRINT, ((EZNEXT(X,Z), X=1,NX), Z=1,NZ)
1198 250 CONTINUE
1199 PRINT 259
1200 259 FORMAT(1H1)
1201 RETURN
1202 END
**SUBROUTINE STABLI**

THE RELATIONSHIP BETWEEN DISTANCE INCREMENTS, TIME INCREMENTS, DISPERSION COEFFICIENTS, AND VELOCITIES IS NEEDED TO DETERMINE THE STABILITY OF THE IMPLICIT FINITE-DIFFERENCE PROCEDURE. THIS SUBROUTINE CALCULATES THE MAXIMUM ALLOWABLE INCREMENTS FOR TIME AND DISTANCE WHICH GENERALLY GUARANTEE STABILITY.

```
1203 SUBROUTINE STABLI

1204 INTEGER X, T, TNEXT, TPRINT, XPRINT, COUNT, OPTION

1205 REAL KD, KMAX, KMIN, KDAYS, LOWER

1206 COMMON/ARRAYS/ C(51, 21, 2), VX(51, 21), VZ(51, 21), EX(51, 21), EZ(51, 21),
1     W(51), CCNCIN(51, 21), INSECT(51), KDAYS(51), KD(51),
2     DEMAND(51, 21), FMT1(40), FMT2(40), FMT3(20)

1207 COMMON/NAMES/ X, T, NX, NZ, NC, TS, ES, EXMAX, EXMIN, VXMAX, VXMIN, WMAX, WMIN,
1     KD, KMAX, KMIN, ITER, ITER1, ITER2, DELT1, DELT2, TNEXT,
2     TPRINT, XPRINT, STABLX, STABLZ, STABLT, R1, R2, VZMAX, SET,
3     VZMIN, EZMAX, EZMIN, DEMAX, DEMIN, NPAGA, DELTAX, DELTAZ,
4     DELTAT, XSTART, XSF, TSTART, COUNT, IPAGE, UPPER, LOWER,
5     MSU, MSL, TIME, NXMI, NXM2, NZM1, NZM2, ZSF, LTYPE, OPTION

1208 COMMON/IMPLCT/EXNEXT(51, 21), EZNEXT(51, 21), CSTAR(51, 21),
1     VXNEXT(51, 21), VZNEXT(51, 21), A(51), B(51),
2     G(51), D(51), V(51)

1209 STABLX = DELTAX
1210 STABLZ = DELTAZ
1211 STABLT = DELTAT
```
C
C-----------------------------------
C CALCULATE STABILITY CRITERIA FOR ONE DIMENSIONAL CASE
C-----------------------------------

1212 IF(NZ.NE.1) GO TO 500
1213 IF(EXMIN.GT.0.0.AND.VXMAX.GT.0.0) STABLX=2.*EXMIN/VXMAX
1214 TERM=2.*EXMAX+DELTAX**2.*KMAX
1215 IF(TERM.NE.0.0) STABLT=(DELTAX**2.)/TERM*2.
1216 R1=DELTA/(DELTAX**2.)
1217 R2=0.
1218 GO TO 800
1219 500 CONTINUE

C
C-----------------------------------
C CALCULATE STABILITY CRITERIA FOR TWO DIMENSIONAL CASE
C-----------------------------------

1220 IF(EXMIN.GT.0.0.AND.VXMAX.GT.0.0) STABLX=2.*EXMIN/VXMAX
1221 IF(EZMIN.GT.0.0.AND.VZMAX.GT.0.0) STABLZ=2.*EZMIN/VZMAX
1222 TERM=2.*(|EXMAX*DELTAZ**2.*EZMAX*DELTAX**2.)*KMAX*(DELTAX**2.)*
$ (DELTAZ**2)
1223 IF(TERM.NE.0.0) STABLT=(DELTAZ**2.)*(DELTAX**2.)/TERM*2.
1224 R1=DELTA/(DELTAX**2.)
1225 R2=DELTA/(DELTAZ**2.)
1226 800 CONTINUE
1227 RETURN
1228 END
**SUBROUTINE ONEDIM**

**---------**

*** *** SUBROUTINE ONEDIM *** ***

**---------**

THIS SUBPROGRAM CALLS FOR THE APPROPRIATE ROUTINES TO SOLVE FOR THE CONCENTRATION PROFILE IN AN ESTUARY WHERE ONE-DIMENSIONAL BEHAVIOR CAN BE ASSUMED. IN ADDITION, WIDTH OF THE ESTUARY CAN VARY THROUGHOUT.

THE SOLUTION MATRIX IS SET UP ACCORDING TO A CRANK-NICOLSON IMPLICIT SCHEME IN SUBROUTINE ARRAY3 AND THIS TRIDIAGONAL MATRIX IS SOLVED BY SUBROUTINE TRIDAG.

**---------**

```plaintext
INTEGER X, Z, T, TNEXT, TPRINT, XPRINT, COUNT, OPTION
REAL KD, KDMAX, KDMIN, KDAYS, LOWER
COMMON/ ARRAYS/ C(51, 21, 2), VX(51, 21), VZ(51, 21), EX(51, 21), EZ(51, 21),
1 W(51), CONCIN(51, 21), INSECT(51), KDAYS(51), KD(51),
2 DEMAND(51, 21), FMT1(40), FMT2(40), FMT3(20)
COMMON/ NAMES/ X, Z, T, NX, NZ, NC, NS, EXMAX, EXMIN, VXMAX, VXMIN, WMAX, WMIN,
1 KD, KD, KDMIN, ITER, ITER1, ITER2, DELT1, DELT2, TNEXT,
2 TPRINT, XPRINT, STABXL, STABLZ, STABLT, R1, R2, VXMAX, SET,
3 VZMIN, EZMAX, EZMIN, DEMAX, DEMIN, NPAGE, DELTAX, DELTAZ,
4 DELTAT, XSTART, XSF, TSTART, COUNT, IPAGE, UPPER, LOWER,
5 MSU, MSL, TIME, NXM, NXM2, NZML, NZML2, ZSF, LTYPE, OPTION
COMMON/ IMPLCT/ EXNEXT(51, 21), EZNEXT(51, 21), CSTAR(51, 21),
1 VXNEXT(51, 21), VZNEXT(51, 21), A(51), B(51),
2 G(51), D(51), V(51)
```
DO 300 IT=1,ITER
TIME=TIME+DELTAT
CALL ARRAY3
CALL IBOUND
Z=1
DO 250 X=1,NX
IF(C(X,Z,TNEXT).LT.0.0) C(X,Z,TNEXT)=0.0
C(X,Z,T)=C(X,Z,TNEXT)
C(X,Z,TNEXT)=0.0
250 CONTINUE
COUNT=COUNT+1
IF(COUNT.GE.TPRINT) GO TO 292
GO TO 300

C-----------------------------------------------
C PRINT CONCENTRATIONS AT APPROPRIATE INTERVALS
C-----------------------------------------------

292 CONTINUE
COUNT=0
IPAGE=IPAGE+1
IF(IPAGE.GE.NPPAGE) GO TO 295
GO TO 297
295 PRINT 296
296 FORMAT(1H1)
IPAGE = 0
297 CALL PRINT2
300 CONTINUE
RETURN

END
SUBROUTINE TWODIM

**********************************************************************
**********************************************************************
**********************************************************************

THIS SUBPROGRAM CALLS FOR THE APPROPRIATE ROUTINES TO SOLVE
FOR THE CONCENTRATION PROFILE IN AN ESTUARY WHERE TWO DIMENSIONAL
BEHAVIOR CAN BE ASSUMED. IN ADDITION, WIDTH OF THE ESTUARY CAN
BE VARIED THROUGHOUT.

THE SOLUTION MATRIX IS SET UP ACCORDING TO A CRANK-NICOLSON
IMPLICIT SCHEME WHICH IS SOLVED BY AN IMPLICIT,ALTERNATING
DIRECTION METHOD. SUBROUTINE ARRAY1 SETS UP THE MATRIX FOR THE
FIRST HALF OF THE SOLUTION AND SUBROUTINE ARRAY2 SETS UP THE
MATRIX FOR THE SECOND HALF. SUBROUTINE TRIDAG SOLVES BOTH OF
THESE MATRICES.

INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,COUNT,OPTION
REAL KD,KMAX,KDMIN,KDAYS,LOWER
COMMON/ARRAYS/ C(51,21,2),VX(51,21),VZ(51,21),EX(51,21),EZ(51,21),
 1 W(51),CONCIN(51,21),INSECT(51),KDAYS(51),KD(51),
 2 DEMAND(51,21),FMT1(40),FMT2(40),FMT3(20)
COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,WMAX,WMIN,
 1 KMAX,KDMIN,ITER,ITER1,ITER2,DELT1,DELT2,TNEXT,
 2 TPRINT,XPRINT,STBLX,STBLZ,STBLT,RL,R2,VZMAX,SET,
 3 VZMIN,EXZMAX,EXZMIN,DEMAM,DEMIN,NPAGE,DELTAX,DELTAZ,
 4 DELTAT,XSTART,XSF,TSTART,COUNT,IPAGE,UPPER,LOWER,
 5 MSU,MSL,TIME,NXMI,NXM2,NZMI,NZM2,ZSF,LTYPE,OPTION
COMMON/IMPLCT/EXNEXT(51,21),EZNEXT(51,21),CSTAR(51,21),
VXNEXT(51,21),VZNEXT(51,21),A(51),B(51),
G(51),D(51),V(51)

DO 300 IT=1,ITER
TIME=TIME+DELTAT
CALL ARRAY1
CALL ARRAY2
CALL IBOUND
DO 250 Z=1,NZ
DO 250 X=1,NX
IF(C(X,Z,TNEXT).LT.0.0) C(X,Z,TNEXT)=0.0
C(X,Z,T)=C(X,Z,TNEXT)
C(X,Z,TNEXT)=0.0
250 CONTINUE
COUNT=COUNT+1
IF(COUNT.GE.TPRINT) GO TO 292
GO TO 300

C
PRINT CONCENTRATIONS AT APPROPRIATE INTERVALS
C
CONTINUE
COUNT=0
IPAGE=IPAGE+1
IF(IPAGE.GE.NPAGE) GO TO 295
GO TO 297
PRINT 296
FORMAT(1H1)
1287  IPAGE = 0
1288  297 CALL PRINT2
1289  300 CONTINUE
1290  RETURN
1291  END
**SUBROUTINE ARRAY1**

```c
1292
1293 INTEGER X, Z, T, TTEXT, TPRINT, XPRINT, COUNT, OPTION
1294 REAL KD, KDMAX, KDMIN, KDAYS, LOWER
1295 COMMON/ARRAYS/ C(51, 21, 2), VX(51, 21), VZ(51, 21), EX(51, 21), EZ(51, 21),
1 W(51), CCNCIN(51, 21), INSECT(51), KDAYS(51), KD(51),
2 DEMAND(51, 21), FMT1(40), FMT2(40), FMT3(20)
1296 COMMON/NAMES/X, Z, T, NX, NZ, NS, EXMAX, EXMIN, VXMAX, VXMIN, WXMAX, WMIN,
1 KDMAX, KDMIN, ITER, ITER1, ITER2, DELT1, DELT2, TNEXT,
2 TPRINT, XPRINT, STABLX, STABLY, STABLZ, STABLT, R1, R2, VZMAX, SET,
3 VZMIN, EZMAX, EZMIN, DEMAX, DEMIN, NPAGE, DELTAX, DELTAY, DELTAZ,
4 DELTAT, XSTART, XSF, TSTART, COUNT, IPAGE, UPER, LOWER,
5 MSU, MSL, TIME, NXM1, NXM2, NZM1, NZM2, ZSF, LTYPE, OPTION
1297 COMMON/IMPLCT/EXNEXT(51, 21), EZNEXT(51, 21), CSTAR(51, 21),
1 VXNEXT(51, 21), VZNEXT(51, 21), A(51), B(51),
2 G(51), D(51), V(51)
```

C

1298 ZUP = UPER
1299 ZLOW = LOWER
1300 ZSF = XSF
1301 FOURX = 4.*DELTAX
1302 FORXSQ = 4.*DELTAX**2
1303 FOURZ = 4.*DELTAY
1304 FORZSQ = 4.*DELTAY**2
C

SET UP THE ARRAY ACCORDING TO A CRANK–NICOLSON
C
IMPLICIT SCHEME
C

1305   DO 1005 Z=2,NZM1
1306   DO 100 X=2,NXM1
1307   A(X)=(DELTAT/W(X))*(-W(X)*VXNEXT(X,Z)/FOURX-(W(X)*EXNEXT(X,Z)
       $   +W(X-1)*EXNEXT(X-1,Z))/FORXSQ)
1308   B(X)=1.*((DELTAT/W(X))*(W(X+1)*EXNEXT(X+1,Z)+Z.*W(X)*EXNEXT(X,Z)
       $   +W(X-1)*EXNEXT(X-1,Z))/FORXSQ)
1309   G(X)=(DELTAT/W(X))*(W(X)*VXNEXT(X,Z)/FOURX-2.*(W(X+1)*EXNEXT(X+1,Z)
       $   +W(X)*EXNEXT(X,Z))/FORXSQ)
1310   D(X)=C(X,Z,T)+(DELTAT/h(X))*(-W(X)*VX(X,Z)/FOURX)*C(X+1,Z,T)
       1   -C(X-1,Z,T)+W(X+1)*EX(X+1,Z)+W(X)*EX(X,Z))/FORXSQ)
       2   *(C(X+1,Z,T)-C(X,Z,T))-(W(X)*EX(X,Z)+W(X-1)*EX(X-1,Z))
       3   /FORXSQ)*C(X,Z,T)-C(X-1,Z,T))
1311   D(X)=D(X)+(DELTAT/W(X))*(-2.*W(X)*VZ(X,Z)/FOURZ)*C(X,Z+1,T)
       1   -C(X,Z-1,T)+2.*(W(X)*EZ(X,Z+1)+W(X)*EZ(X,Z))/FORSZQ)
       2   *(C(X,Z+1,T)-C(X,Z,T))-2.*(W(X)*EZ(X,Z)+W(X)*EZ(X,Z-1))
       3   /FORSZQ)*(C(X,Z,T)-C(X,Z-1,T))
1312   100 CONTINUE
1313   NFAG=1
C

C

EVALUATE BOUNDARY VALUES BY EXPONENTIAL EXTRAPOLATION OR BY
C
CONSTANT SLOPE EXTRAPOLATION DEPENDING ON VALUE OF OPTION
C

297
GO TO (101, 2C2, 101, 202), OPTION
101 CONTINUE
VXI=VXNEXT(1, Z)
XX=UPPER+DELTAX+XSF+VX1*TSTART*86400.
EXI=EXNEXT(1, Z)
C2=4.*EX1*TIME
EXPON=(2.*XX*DELTAX-(DELTAX*DELTAX)-2.*DELTAX*VX1*TIME)/C2
P=EXP(EXPON)
B(2)=B(2)+A(2)*P
CSTAR(1, Z)=CSTAR(2, Z)*P
150 CONTINUE
VXNX=VXNEXT(NX, Z)
XX=(NX-2)*DELTAX-LOWER+XSF+VXNX*TSTART*86400.
EXNX=EXNEXT(NX, Z)
C2=4.*EXNX*TIME
EXPON=(-2.*XX*DELTAX-(DELTAX*DELTAX)+2.*DELTAX*VXNX*TIME)/C2
P=EXP(EXPON)
B(NXM1)=B(NXM1)+G(NXM1)*P
CSTAR(NX, Z)=CSTAR(NXM1, Z)*P
GO TO 1000
202 CONTINUE
B(2)=B(2)+2.*A(2)
G(2)=G(2)-A(2)
A(NX-1)=A(NX-1)-G(NX-1)
B(NX-1)=B(NX-1)+2.*A(NX-1)
CSTAR(1, Z)=2.*CSTAR(2, Z)-CSTAR(3, Z)
IF(CSTAR(1, Z).LT.0.) CSTAR(1, Z)=0.
CSTAR(NX, Z)=2.*CSTAR(NXM1, Z)-CSTAR(NXM2, Z)
IF(CSTAR(NX, Z).LT.0.) CSTAR(NX, Z)=0.
1000 CONTINUE
1344  IF(NFLAG.GT.1) GO TO 1005
1345    NFLAG=2
1346    CALL TRIDAG(2,NXM1)
1347    DO 1004 X=2,NXM1
1348       CSTAR(X,Z)=V(X)
1349  1004 CONTINUE
1350    GO TO (101,202,101,202),CPTION
1351  1005 CONTINUE
1352    RETURN
1353    END
SUBROUTINE ARRAY2

INTEGER X, Z, T, TNEXT, TPRINT, XPRINT, CCOUNT, OPTION
REAL KD, KD_MAX, KDMIN, KDAYS, LOWER

COMMON/ARRAYS/ C(51,21,2), VX(51,21), VZ(51,21), EX(51,21), EZ(51,21),
1 W(51), CONCIN(51,21), INSECT(51), KDAYS(51), KD(51),
2 DEMAND(51,21), FMT1(40), FMT2(40), FMT3(20)

COMMON/NAMES/X, Z, T, NX, NZ, NC, NS, EXMAX, EXMIN, VXMAX, VXMIN, WMAX, WMIN,
1 KD_MAX, KDMIN, ITER, ITER1, ITER2, DELT1, DELT2, TNEXT,
2 TPRINT, XPRINT, STABLX, STABLZ, STABLT, R1, R2, VZMAX, SET,
3 VZMIN, EZMAX, EZMIN, DEMAX, DEMIN, NPPAGE, DELTAX, DELTAY,
4 DELTAT, XSTART, XSF, TSTART, COUNT, IPAGE, UPPER, LOWER,
5 MSU, MSL, TIME, NXM1, NXM2, NZM1, NZM2, ZSF, LTYPE, OPTION

COMMON/IMPLCT/EXNEXT(51,21), EZNEXT(51,21), CSTAR(51,21),
1 VXNEXT(51,21), VZNEXT(51,21), A(51), B(51),
2 G(51), O(51), V(51)

ZUP=UPPER
ZLOW=LOWER
ZSF=XSF
FOURX=4.*DELTAX
FORXSQ=4.*DELTAX**2
FOURZ=4.*DELTAT
FORZSQ=4.*DELTAT**2
SET UP THE ARRAY ACCORDING TO A CRANK–NICOLSON IMPLICIT SCHEME

DO 100 X=2,NXM1
DO 100 Z=2,NZM1

A(Z)=(DELTAT/W(X))*(-W(X)*ZNEXT(X,Z)/FORZ-(W(X)*ZNEXT(X,Z))
$+W(X)*ZNEXT(X,Z-1))/FORZSQ)

B(Z)=1.*(DELTAT/W(X))*(W(X)*ZNEXT(X,Z+1)+Z.*W(X)*ZNEXT(X,Z)
$+W(X)*ZNEXT(X,Z-1))/FORZSQ

G(Z)=(DELTAT/W(X))*(W(X)*ZNEXT(X,Z)/FORZ-(W(X)*ZNEXT(X,Z+1)
$+W(X)*ZNEXT(X,Z))/FORZSQ)

D(Z)=C(X,Z,T)+(DELTAT/W(X))*(-(W(X)*VXNEXT(X,Z)/FOURX)*(CSTAR(1
X+1,Z)-CSTAR(X-1,Z))-W(X)*PX(X,Z)/FOURX)*(C(X+1,Z,T)
-2*C(X-1,Z,T)-(W(X)*VZ(X,Z)/FOURZ)*(C(X,Z+1,T)-C(X,Z-1,T))
3*W(X)*ZNEXT(X,Z)+W(X)*ZNEXT(X,Z))/FORZSQ)+(CSTAR(4
X+1,Z)-CSTAR(X,Z))+(W(X)*EX(X,Z))
$+W(X)*ZNEXT(X,Z))/FORZSQ)

D(Z)=D(Z)+(DELTAT/W(X))*-(W(X)*EXNEXT(X,Z)+W(X-1)*EXNEXT(X-1,Z))
/FORZSQ)*(CSTAR(X,Z)-CSTAR(X-1,Z))-(W(X)*EX(X,Z)+W(X-1)
*EX(X-1,Z))/FORZSQ)*(C(X,Z,T)-C(X-1,Z,T))+(W(X)*EZ(X,Z+1)
3*W(X)*EZ(X,Z))/FORZSQ)*(C(X,Z+1,T)-C(X,Z,T)-(W(X)*EZ(X,Z)
4*W(X)*EZ(X,Z-1))/FORZSQ)*(C(X,Z,T)-C(X,Z-1,T))

100 CONTINUE

EVALUATE BOUNDARY VALUES BY EXPONENTIAL EXTRAPOLATION OR BY
CONSTANT SLOPE EXTRAPOLATION DEPENDING ON VALUE OF OPTION
GO TO (303,303,101,202), CPTION
101 CONTINUE
VZ1=VZNEXT(X,1)
ZZ=-ZUP*DELTAZ+ZSF+VZ1*TSTART*86400.
EZ1=EZNEXT(X,1)
C2=4.*EZ1*TIME
EXPON=(-2.*ZZ*DELTAZ-(DELTAZ*DELTAZ)-2.*DELTAZ*VZ1*TIME)/C2
B(2)=B(2)+A(2)*EXP(EXPON)
150 CONTINUE
VZNZ=VZNEXT(X,NZ)
ZZ=(NZ-2)*DELTAZ-ZLCH+ZSF+VZNZ*TSTART*86400.
EZNZ=EZNEXT(X,NZ)
C2=4.*EZNZ*TIME
EXPON=(-2.*ZZ*DELTAZ-(DELTAZ*DELTAZ)+2.*DELTAZ*VZNZ*TIME)/C2
B(NZ-1)=B(NZ-1)+G(NZ-1)*EXP(EXPON)
GO TO 1000
202 CONTINUE
B(2)=B(2)+2.*A(2)
G(2)=G(2)-A(2)
A(NZ-1)=A(NZ-1)-G(NZ-1)
B(NZ-1)=B(NZ-1)+2.*A(NZ-1)
GO TO 1000
303 B(2)=B(2)+A(2)
B(NZM1)=B(NZM1)+G(NZM1)
1000 CONTINUE
C
C------------------------------------------------------------------------
C   SOLVE TRIDIAGONAL MATRIX
C------------------------------------------------------------------------
1400 CALL TRIDAG(2,NZM1)
C------------------------------------------------------------------------
C   SUBTRACT DECAY AND OTHER DEMANDS AND
C   PLACE FINAL SOLUTION IN C(X,Z,T+1)
C------------------------------------------------------------------------
1401 DO 1004 Z=2,NZM1
1402   C(X,Z,T+1)=V(Z)-DELTAT*(KD(X)/2.*(V(Z)+C(X,Z,T)))-DEMAND(X,Z)
1403 1004 CONTINUE
1404 1005 CONTINUE
1405 RETURN
1406 END
SUBROUTINE ARRAY3

INTEGER X, Z, T, TNEXT, TPRINT, XPRINT, COUNT, OPTION
REAL KD, KDMAX, KDMIN, KDAYS, LOWER
COMMON/ARRAYS/ C(51, 21, 2), VX(51, 21), VZ(51, 21), EX(51, 21), EZ(51, 21),
1 W(51), CONSIN(51, 21), INSECT(51), KDAYS(51), KD(51),
2 DEMAND(51, 21), FMT1(40), FMT2(40), FMT3(20)
COMMON/NAMES/X, Z, T, NX, NZ, NC, NS, EXMAX, EXMIN, VXMAX, VXMIN, WMX, WMIN,
1 KD, KDMIN, ITER1, ITER2, DELT1, DELT2, TNEXT,
2 TPRINT, XPRINT, STABLX, STABLY, STALT1, R1, R2, VXMAX, SET,
3 VZMIN, EZMAX, EZMIN, DEMAX, DEMIN, NPAGE, DELTAX, DELTAX,
4 DELTAT, XSTART, XSF, TSTART, COUNT, IPAGE, UPPER, LOWER,
5 MSU, MSL, TIME, NXML, NXM2, NZ1, NZ2, ZSF, LTYPE, OPTION
COMMON/IMPLCT/EXNEXT(51, 21), EZNEXT(51, 21), CSTAR(51, 21),
1 VXNEXT(51, 21), VZNEXT(51, 21), A(51), B(51),
2 G(51), D(51), V(51)

FOURX = 4, *DELTAX
FORXSQ = 4, *DELTAX**2

Z = 1

SET UP THE ARRAY ACCORDING TO A CRANK-NICOLSON
IMPLICIT SCHEME

DO 100 X=2, NXM1
A(X) = (DELTAT/W(X))*(-W(X)*VXNEXT(X, Z)/FOURX-W(X)*EXNEXT(X, Z)
S + W(X-1)*EXNEXT(X-1, Z))/FORXSQ
100 CONTINUE
B(X)=1.+(DELTAT/W(X))*((W(X+1)*EXNEXT(X+1,Z)+2.*W(X)*EXNEXT(X,Z))
$+W(X-1)*EXNEXT(X-1,Z))/FORXSQ+KD(X)/2.*W(X))
G(X)=((DELTAT/W(X))*((W(X)*VXNEXT(X,Z))/FOURX-((W(X+1)*EXNEXT(X+1,Z))
$+W(X)*EXNEXT(X,Z))/FORXSQ))
D(X)=C(X,Z,T)+((DELTAT/W(X))*(-((W(X)*VX(X,Z))/FOURX)*C(X+1,Z,T))
1-C(X-1,Z,T))+(W(X+1)*EX(X+1,Z)+W(X)*EX(X,Z))/FORXSQ)
2-((C(X+1,Z,T)-C(X,Z,T))-((W(X)*EX(X,Z)+W(X-1)*EX(X-1,Z)))
3/IFORXSQ)*(C(X,Z,T)-C(X-1,Z,T))-(KD(X)/2.*C(X,Z,T)*W(X))

100 CONTINUE

--- EVALUATE BOUNDARY VALUES BY EXPONENTIAL EXTRAPOLATION OR BY
--- CONSTANT SLOPE EXTRAPOLATION DEPENDING ON VALUE OF OPTION

GO TO (101,202,101,202), OPTION
101 CONTINUE
VX1=VXNEXT(1,Z)
X=-UPPER+DELTAX*XSF+VX1*TSTART*86400.
EX1=EXNEXT(1,Z)
C2=4.*EX1*TIME
EXPON=(2.*X*DELTAX-(DELTAX*DELTAX)-2.*DELTAX*VX1*TIME)/C2
B(2)=B(2)+A(2)*EXP(EXPCN)
150 CONTINUE
VXNX=VXNEXT(NX,Z)
1432 1433 1434 1435 1436 1437 1438 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449

\[
x = nx - 2 * \text{DELTA} \times \text{LOWER \times SF \times VN} \times \text{TIME} \times \text{START} \times 8 \times 64 \times 0.0
dn = n \times \text{NEXT} \times \text{TIME}
C2 = 4 \times \text{EXP} \times \text{TIME}
B(n-1) = B(n) \times \text{DELTA} \times (n) \times \text{DELTA} \times C2
\]

SOLVE TRIANGULAR MATRIX

CALL TRIDAG2 \text{NX} \times \text{NX}

SUBTRACT OTHER DEMANDS AND PLACE FINAL SOLUTION IN \text{C}(x, z, t+1)

DO 1004 x = 2, \text{NX} + 1
C(x, z, t+1) = v(x) - \text{DEMAND}(x, z)
1004 CONTINUE

RETURN
END
SUBROUTINE TRIDAG(IF,L)

***************
***** SUBROUTINE TRIDAG ****
***************

THIS SUBROUTINE SOLVES A SYSTEM OF LINEAR SIMULTANEOUS
EQUATIONS HAVING A TRIDIAGONAL COEFFICIENT MATRIX.
THE EQUATIONS ARE NUMBERED FROM IF THROUGH L, AND THEIR
SUB-DIAGONAL, DIAGONAL, AND SUPER-DIAGONAL COEFFICIENTS
ARE STORED IN THE ARRAYS A, B, AND G. THE COMPUTED
SOLUTION VECTOR V(IF) TO V(L) IS STORED IN THE ARRAY V.

THIS SUBPROGRAM IS A MODIFIED VERSION OF A PROGRAM FOUND IN THE
FOLLOWING TEXT**APPLIED NUMERICAL METHODS**BY CARNAHAN,LUTHER,
AND WILKES, JOHN WILEY&SONS,INC.,1969,P446.

INTEGER X,Z,T,TNEW,TPRINT,XPRINT,COUNT,OPTION
REAL KD,KMAX,KMIN,KDAYS,LOWER
COMMON/ARRAYS/ C(51,21,2),VX(51,21),VZ(51,21),EX(51,21),EZ(51,21),
1 W(51),CONCIN(51,21),INSECT(51),KDAYS(51),KD(51),
2 DEMAND(51,21),FMT1(40),FMT2(40),FMT3(20)
COMMON/NAMES/X,Z,T,NK,NZ,NS,EXMAX,EXMIN,VXMAX,VXMIN,WMAX,WMIN,
1 KMAX,KMIN,ITER,ITER1,ITER2,DELT1,DELT2,TNEW,
2 TPRINT,XPRINT,STABLY,STABZ,STABT,R1,R2,VZMAX,VZMIN,
3 EZMAX,EZMIN,DEMAX,DEMIN,NPPAGE,DELTAX,DELTAY,
4 DELTAT,XSTART,TSF,STSTART,COUNT,IPAGE,UPPER,LOWER,
5 MSU,MSL,TIME,NXM1,NXM2,NZM1,NZM2,ZSF,LTYPE,O_PION
COMMON/IMPLCT/EXNEXT(51,21),EZNEXT(51,21),CSTAR(51,21), VXNEXT(51,21),VZNEXT(51,21),A(51),B(51),
G(51),D(51),V(51)
DIMENSION BETA(51),GAMMA(51)

C

---

COMPUTE THE INTERMEDIATE ARRAYS BETA AND GAMMA

---

BETA(IF)=B(IF)
GAMMA(IF)=D(IF)/BETA(IF)
IFP1=IF+1
DO 1 I=IFP1,L
BETA(I)=B(I)-A(I)*G(I-1)/BETA(I-1)
1 GAMMA(I)=(D(I)-A(I)*GAMMA(I-1))/BETA(I)

---

COMPUTE FINAL SOLUTION VECTOR V

---

VL=GAMMA(L)
LAST=L-IF
DO 2 K=1,LAST
I=L-K
2 V(I)=GAMMA(I)-G(I)*V(I+1)/BETA(I)
RETURN
END
SUBROUTINE IBOUND

******************************************************************************************
***** SUBROUTINE IBOUND ****
******************************************************************************************

THIS FORM OF SUBROUTINE IBOUND EXTRAPOLATES THE CONCENTRATIONS
AT THE BOUNDARIES BY USING AN EXPONENTIAL EQUATION APPROPRIATE TO
CONCENTRATION PROFILES IN AN ESTUARY OR BY A CONSTANT SLOPE
EXTRAPOLATION. THE TYPE OF EXTRAPOLATION AT THE BOUNDARIES
DEPENDS ON THE VALUE FOR THE VARIABLE OPTION.

INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,CCOUNT,OPT
REAL KD,KDAX,KDMIN,KDAYS,LOWER

COMMON/ARRAYS/ C(51,21,2),VX(51,21),VZ(51,21),EX(51,21),EZ(51,21),
  W(51),CONCIN(51,21),INSECT(51),KDAYS(51),KD(51),
  DEMAND(51,21),FMT1(40),FMT2(40),FMT3(20)

COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,VMAX,VMIN,
  KMAX,KDMIN,ITER,ITER1,ITER2,DELT1,DELT2,TNEXT,
  TPRINT,XPRINT,STABLX,STABLY,STABLT,RL,RZ,VZMAX,SET,
  VMIN,EMAX,EMIN,DMAX,DMIN,NPAG,DELTAX,DELTAZ,
  DELTAT,XSTART,XSF,TSTART,COUNT,IPAGE,UPPER,LOWER,
  MSU,MSL,TIME,NXM1,NXM2,NZM1,NZM2,ZSF,LTYP,OPT

COMMON/IMPLCT/EXNEXT(51,21),EZNEXT(51,21),CSTAR(51,21),
  VXNEXT(51,21),VZNEXT(51,21),A(51),B(51),
  G(51),D(51),V(51)
1476        IF(NZ.EQ.1) NZM1=1
1477        ZUP=UPPER
1478        ZLOW=LOWER
1479        ZSF=XSF
1480        GO TO (1001,10C2,10C3,1004,1005,1002,1004), OPTION

C
C-----------------------------------------------
C        EXPONENTIAL EXTRAPOLATION
C-----------------------------------------------

1481     1003 CONTINUE
1482     IF(NZ.EQ.1) GO TO 1001
1483     DO 140 X=2,NXM1
1484     IF(C(X,2,T+1).LE.0.0) GO TO 140
1485     VZ1=VZNEXT(X,1)
1486     ZZ=-ZUP+DELTAZ+ZSF+VZ1*TSTART*86400.
1487     EZ1=EZNEXT(X,1)
1488     C2=4.*EZ1*TIME
1489     EXPON=(-2.*ZZ*DELTAZ-(DELTAZ*DELTAZ)-2.*DELTAZ*VZ1*TIME)/C2
1490     C(X,1,T+1)=C(X,2,T+1)*EXP(EXPON)
1491     140 CONTINUE
1492     DO 190 X=2,NXM1
1493     IF(C(X,NXM1,T+1).LE.0.0) GO TO 190
1494     VZNZ=VZNEXT(X,NZ)
1495     ZZ=(NZ-2)*DELTAZ-ZLOW+ZSF+VZNZ*TSTART*86400.
1496     EZNZ=EZNEXT(X,NZ)
1497     C2=4.*EZNZ*TIME
1498     EXPON=(-2.*ZZ*DELTAZ-(DELTAZ*DELTAZ)+2.*DELTAZ*VZNZ*TIME)/C2
1499     C(X,NZ,T+1)=C(X,NXM1,T+1)*EXP(EXPON)
1500     190 CONTINUE
1501     1001 CONTINUE
1502     IF(LTYPE.EQ.1.AND.MSU.EQ.1) GO TO 350
DO 340 Z=1,NZ
IF(C(Z,Z,T+1) .LE. 0.0) GO TO 340
VX1=VXNEXT(1,Z)
XX=UPPER+DELTAX*XSF+VX1*TSTART*86400.
EX1=EXNEXT(1,Z)
C2=4.*EX1*TIME
EXPON=(-2.*XX*DELTAX-(DELTAX*DELTAX)-2.*DELTAX*VX1*TIME)/C2
C(1,Z,T+1)=C(2,Z,T+1)*EXP(EXPON)
1340 CONTINUE
350 CONTINUE
IF(LTYPE.EQ.1.AND.MSL.EQ.NZ) GO TO 400
DO 390 Z=1,NZ
IF(C(NX-1,Z,T+1) .LE. 0.0) GO TO 390
VXNX=VXNEXT(NX,Z)
XX=(NX-2)*DELTAX-LOWER*XSF+VXNX*TSTART*86400.
EXNX=EXNEXT(NX,Z)
C2=4.*EXNX*TIME
EXPON=(-2.*XX*DELTAX-(DELTAX*DELTAX)+2.*DELTAX*VXNX*TIME)/C2
C(NX,Z,T+1)=C(NX-1,Z,T+1)*EXP(EXPON)
390 CONTINUE
400 CONTINUE
GO TO 2000
C
C---------------------------------------------------------------------------------
C CONSTANT SLOPE EXTRAPOLATION
C---------------------------------------------------------------------------------
1004 CONTINUE
IF(NZ.EQ.1) GO TO 1002
DO 540 X=2,NXM1
C(X,1,T+1)=2.*C(X,2,T+1)-C(X,3,T+1)
IF(C(X,1,T+1).LT.0.0) C(X,1,T+1)=0.0
C(X,NZ,T+1)=2.*C(X,NZM1,T+1)-C(X,NZM2,T+1)
IF(C(X,NZ,T+1).LT.0.0) C(X,NZ,T+1)=0.0
540 CONTINUE
1002 CONTINUE
DO 590 Z=1,NZ
C(1,Z,T+1)=2.*C(2,Z,T+1)-C(3,Z,T+1)
IF(C(1,Z,T+1).LT.0.0) C(1,Z,T+1)=0.0
C(NX,Z,T+1)=2.*C(NXM1,Z,T+1)-C(NXM2,Z,T+1)
IF(C(NX,Z,T+1).LT.0.0) C(NX,Z,T+1)=0.0
590 CONTINUE
GO TO 2000
C
C THIS OPTION ALLOWS THE PROGRAM USER TO SUBSTITUTE HIS OWN
C EXTRAPOLATION ROUTINE. THE USER MUST BE CAREFUL TO CHANGE
C THE APPROPRIATE ARRAY VALUES.
C
1005 CONTINUE
OPTION=2
GO TO 1002
2000 CONTINUE
IF(NZ.EQ.1) GO TO 2005
IF(OPTION.EQ.3.OR.OPTION.EQ.4.OR.OPTION.EQ.5) GO TO 2005
DO 700 X=1,NX
C(X,1,T+1)=C(X,2,T+1)
C(X,NZ,T+1)=C(X,NZM1,T+1)
700 CONTINUE
2005 CONTINUE
C
C-----------------------------------------------
C IF LTYPE EQUALS 1, SET CONCENTRATIONS AT SOURCE POINTS.
C IF LTYPE EQUALS 3, THE USER CAN INCLUDE A SPECIAL SUBROUTINE
C OR SET OF CALCULATIONS IN THIS PART OF THE PROGRAM.
C-----------------------------------------------
1552 IF(LTYPE.EQ.1.AND.NS.LE.2) GO TO 275
1553 IF(LTYPE.EQ.1.AND.NS.GT.2) GO TO 281
1554 GO TO 290
1555 275 CONTINUE
1556 DO 280 Z=1,NZ
1557 C(MSU,Z,T+1)=C(MSU,Z,T)
1558 C(MSL,Z,T+1)=C(MSL,Z,T)
1559 280 CONTINUE
1560 GO TO 290
1561 281 CONTINUE
1562 DO 285 I=1,NC
1563 DO 285 Z=1,NZ
1564 C(INSECT(I),Z,T+1)=CONCIN(I,Z)
1565 285 CONTINUE
1566 290 CONTINUE
1567 RETURN
1568 END
SUBROUTINE PRINT1

***************************************************************************

***** SUBROUTINE PRINT1 *****
***************************************************************************

THIS SUBROUTINE CALCULATES THE CONVERSION VALUES FOR MANY OF
THE INPUT PARAMETERS AND PRINTS OUT A SUMMARY OF THE INPUT DATA

------------------------------------------------------------------------

INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,COUNT,OPTION
REAL KD,KDAMX,KDMIN,KDAYS,LOWER
COMMON/ARRAYS/ C(51,21,2),VX(51,21),VZ(51,21),EX(51,21),EZ(51,21),
  1 W(51),CONCINT(51,21),INSECT(51),KDAYS(51),KD(51),
  2 DEMAND(51,21),FMT1(40),FMT2(40),FMT3(20)
COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,WMAX,WMIN,
  1 KDAMX,KDMIN,ITER,ITER1,ITER2,DELT1,DELT2,TNEXT,
  2 TPRINT,XPRINT,STABLX,STABLY,STABLX,STABL,RI,R2,VZMAX,STET,
  3 VZMIN,EZMAX,EZMIN,DEMAX,DEMIN,NPAGE,DELTAX,DELTAXZ,
  4 DELTAT,XSTART,XSF,TSTART,COUNT,IPAGE,UPPER,LOWER,
  5 MSU,MSL,TIME,NX1,NX2,NZ1,NZ2,ZSF,LTYP,OPTION
COMMON/IMPLCT/EXNEXT(51,21),EZNEXT(51,21),CSTAR(51,21),
  1 VXNEXT(51,21),VZNEXT(51,21),A(51),B(51),
  2 G(51),D(51),V(51)

------------------------------------------------------------------------

CALCULATE CONVERSION UNITS

------------------------------------------------------------------------
1575 UNITS1 = 86400./5280.
1576 UNITS2 = 86400. /(5280.*5280.).
1577 DELDAY = DELTAT/86400.
1578 SECS1 = DELT1*86400.
1579 SECS2 = DELT2*86400.
1580 DELMX = DELTAX/5280.
1581 DELM2 = DELTAZ/5280.
1582 TOTALX = (NX-1)*DELTAX
1583 TOTMX = TOTALX/5280.
1584 TOTALZ = (NZ-1)*DELTAZ
1585 TOTMZ = TOTALZ/5280.
1586 VX1 = VXMAX*UNITS1
1587 VX2 = VXMIN*UNITS1
1588 VZ1 = VZMAX*UNITS1
1589 IF(VZMIN.EQ.1000000.) VZMIN=0.
1590 VZ2 = VXMIN*UNITS1
1591 EX1 = EXMAX*UNITS2
1592 EX2 = EXMIN*UNITS2
1593 EZ1 = EZMAX*UNITS2
1594 IF(EZMIN.EQ.1000000.) EZMIN=0.
1595 EZ2 = EZMIN*UNITS2
1596 XKD1 = KDMAX*86400.
1597 XKD2 = KDMIN*86400.
1598 SXM = STABLX/5280.
1599 SXM = STABLZ/5280.
1600 STD = STABLT/86400.

C -----------------------------------
C PRINT DATA SUMMARY
C -----------------------------------
1631   PRINT 572, VMAX, V1
1632   PRINT 573, VMIN, V2
1633   PRINT 575, EXMAX, EX1
1634   PRINT 580, EXMIN, EX2
1635   PRINT 585, EZMAX, EZ1
1636   PRINT 590, EZMIN, EZ2
1637   PRINT 600, WMAX
1638   PRINT 605, WMIN
1639   PRINT 610, XKD1, KMAX
1640   PRINT 615, XKD2, KMIN
1641   PRINT 620, DEMAX
1642   PRINT 630, NC

C--------------------------------------------------
C   XUPEAK AND XLPEAK ARE CALCULATED ASSUMING DISTANCES ARE
C   DECREASING IN THE DOWNSTREAM DIRECTION
C--------------------------------------------------
1643   XUPEAK = X START - (MSU-1)*DELMX
1644   XLPEAK = X START - (MSL-1)*DELMX
1645   PRINT 635, XUPEAK, MSU
1646   PRINT 637, XLPEAK, MSL
1647   PRINT 500
1648   PRINT 505
1649   PRINT 502
1650   PRINT 640
1651   PRINT 645, STABLX, SXM
1652   PRINT 650, STABLZ, SZM
1653   PRINT 655, STABLT, STD
1654   PRINT 660, R1
1655   PRINT 665, R2
1656      PRINT 505
1657      IF(NC.EQ.0) GO TO 100
1658      PRINT 500
1659      PRINT 700
1660      IF(NZ.EQ.1) GO TO 50
1661      DO 25 I=1,NC

C---------------------------------------------------------
C     WHERE IS CALCULATED ASSUMING DISTANCES ARE DECREASING IN
C     THE DOWNSTREAM DIRECTION
C---------------------------------------------------------
1662      WHERE=XSTART-(INSECT(I)-1)*DELNX
1663      PRINT 710, INSECT(I),WHERE,DELTAZ
1664      DO 25 Z=1,NZ
1665      PRINT 715,INSECT(I),Z,CONCIN(I,Z)
1666      25 CONTINUE
1667      GO TO 100
1668      50 DO 75 I=1,NC
1669      WHERE=XSTART-(INSECT(I)-1)*DELNX
1670      PRINT 705, INSECT(I), WHERE, CONCIN(I,1)
1671      PRINT 502
1672      75 CONTINUE
1673      100 CONTINUE
1674      PRINT 500
1675      500 FORMAT(1HL)
1676      502 FORMAT(/)
1677      505 FORMAT(T45, 3Bl''**'')
1678      507 FORMAT(T45,**',T82,**')
1679      510 FORMAT(T45,**',T56,'ESTUARY SIMULATION', T82, **')
1680      515 FORMAT(T45, **', T48, 'ONE DIMENSIONAL IMPLICIT METHOD', T82,**')
520 FORMAT(T45, '***', T48, 'TWO DIMENSIONAL IMPLICIT METHOD', T82, '***')
523 FORMAT(T45, '***', T53, 'CONSTANT CONCENTRATION', T82, '***')
525 FORMAT(T45, '***', T54, 'INSTANTANEOUS RELEASE', T82, '***')
530 FORMAT(T45, '***', T56, 'CONTINUOUS LOADING', T82, '***')
535 FORMAT(T45, '***', T51, 'PROGRAMMER -- JONATHAN YOUNG', T82, '***')
540 FORMAT(T53, 'X INCREMENT = ', F7.0, ' FEET (', F8.5, ' MILES $)/')
545 FORMAT(T53, 'Z INCREMENT = ', F7.0, ' FEET (', E10.3, ' MILES )//')
550 FORMAT(T42, 'INITIAL TIME INCREMENT = ', F7.0, ' SECONDS (', F6.4, ' DAYS )//')
551 FORMAT(T44, 'NUMBER OF ITERATIONS = ', I5, ')//')
552 FORMAT(T42, 'REVISED TIME INCREMENT = ', F7.0, ' SECONDS (', F6.4, ' DAYS )//')
553 FORMAT(T44, 'NUMBER OF ITERATIONS = ', I5, ')//')
555 FORMAT(T37, 'NUMBER OF HORIZONTAL POINTS = ', I4, ')//')
560 FORMAT(T39, 'NUMBER OF VERTICAL POINTS = ', I4, ')//')
565 FORMAT(T37, 'MAXIMUM HORIZONTAL VELOCITY = ', F5.2, ' FEET/SECOND ('$, F6.2, ' MILES/DAY )//')
570 FORMAT(T37, 'MINIMUM HORIZONTAL VELOCITY = ', F5.2, ' FEET/SECOND ('$, F6.2, ' MILES/DAY )//')
572 FORMAT(T39, 'MAXIMUM VERTICAL VELOCITY = ', E10.3, ' FEET/SECOND (', $, E10.3, ' MILES/DAY )//')
573 FORMAT(T39, 'MINIMUM VERTICAL VELOCITY = ', E10.3, ' FEET/SECOND (', $, E10.3, ' MILES/DAY )//')
575 FORMAT(T35, 'MAXIMUM HORIZONTAL DISPERSION = ', F6.0, ' FEET SQUARE $ED / SECOND (', '', F5.2, ' MILES SQUARED / DAY )//')
580 FORMAT(T35, 'MINIMUM HORIZONTAL DISPERSION = ', F6.0, ' FEET SQUARE $ED / SECOND (', '', F5.2, ' MILES SQUARED / DAY )//')
585 FORMAT(T37, 'MAXIMUM VERTICAL DISPERSION = ', F9.3, ' FEET SQUARED $ / SECOND (', '', E10.3, ' MILES SQUARED / DAY )//')
590 FORMAT(T37, 'MINIMUM VERTICAL DISPERSION = ', F9.3, ' FEET SQUARED  
$ / SECOND ( ', E10.3, ' MILES SQUARED / DAY )', 
591 600 FORMAT(T51, 'MAXIMUM WIDTH = ', F5.0, ' FEET', )  
592 605 FORMAT(T51, 'MINIMUM WIDTH = ', F5.0, ' FEET', )  
593 610 FORMAT(T46, 'MAXIMUM DECAY RATE = ', F5.3, ' PER DAY ( ', E10.3, 
$ ' PER SECOND )', )  
594 615 FORMAT(T46, 'MINIMUM DECAY RATE = ', F5.3, ' PER DAY ( ', E10.3, 
$ ' PER SECOND )', )  
595 620 FORMAT(T42, 'OTHER DEMANDS, MAXIMUM = ', F5.2, )  
596 630 FORMAT(T26, 'NUMBER OF INITIAL CONCENTRATION VALUES = ', I2, )  
597 635 FORMAT(T16, 'INITIAL LOCATION OF UPPERMOST PEAK CONCENTRATION = ',  
$ F6.2, ' MILES (SECTION NUMBER ' , I2, ' )', )  
598 637 FORMAT(T16, 'INITIAL LOCATION OF LOWERMOST PEAK CONCENTRATION = ',  
$ F6.2, ' MILES (SECTION NUMBER ' , I2, ' )', )  
599 640 FORMAT(T56, 'STABILITY CRITERIA', )  
600 645 FORMAT(T35, 'MAXIMUM ALLOWABLE X INCREMENT = ', F7.0, ' FEET ( ', 
$ F8.5, 'MILES )', )  
601 650 FORMAT(T35, 'MAXIMUM ALLOWABLE Z INCREMENT = ', F7.0, ' FEET ( ', 
$ F8.5, 'MILES )', )  
602 655 FORMAT(T32, 'MAXIMUM ALLOWABLE TIME INCREMENT = ', F7.0, ' SECONDS 
( ', E10.3, ' DAYS )', )  
603 660 FORMAT(T38, 'ACTUAL DELTAX/DELTA**2. = ', E10.3, )  
604 665 FORMAT(T38, 'ACTUAL DELTAX/DELTA**2. = ', E10.3, )  
605 700 FORMAT(T25, 'LOCATIONS OF INITIAL CONCENTRATIONS', )  
606 705 FORMAT( 'A WASTE SOURCE IS LOCATED AT STATION ', I2, ' MILE ', 
607 710 FORMAT( 1X,'AN INITIAL CONCENTRATION IS FOUND AT STATION ', I2,  
1 ' MILE ', F6.3, '_portal ').', ', THE CONCENTRATIONS AT ', F7.1,  
2 ' FOOT INTERVALS WITH DEPTH ARE', )  
608 715 FORMAT(T15, 'C(',I2,I2,I1) = ', F9.2, ' PPM')  
610 RETURN  
611 END
**SUBROUTINE PRINT2**

***SUBROUTINE PRINT2***

**THIS SUBROUTINE PRINTS OUT THE TIME AND THE CONCENTRATION PROFILE ACCORDING TO A FORMAT PREVIOUSLY READ INTO THE PROGRAM. THIS SUBROUTINE CAN BE CHANGED ACCORDING TO THE NEEDS OF THE USER.***

**WARNING—DO NOT CHANGE ANY VARIABLE OCCURRING IN A COMMON BLOCK.**

```fortran
INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,COUNT,OPTION
REAL KD,KDMAX,KDMIN,KDAYS,LOWER
COMMON/ARRAYS/ C(51,21,2), VX(51,21), VZ(51,21), EX(51,21), EZ(51,21),
1 W(51), CONCIN(51,21), INSECT(51), KDAYS(51), KD(51),
2 DEMAND(51,21), FMT1(40), FMT2(40), FMT3(20)
COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,VZMAX,VZMIN,
1 KDMAX,KDMIN,ITER,ITER1,ITER2,DELT1,DELT2,TNEXT,
2 TPRINT,XPRINT,STABLY,STABLZ,STABLT,R1,R2,VZMAX,SET,
3 VZMIN,EZMAX,EZMIN,DEMAY,DEMIN,NPAGE,DELTAX,DELTAZ,
4 DELTAT,XSTART,XSF,TSTART,COUNT,IPAGE,UPPER,LOWER,
5 MSU,MSL,TIME, NXM1,NXM2,NZM1,NZM2,ZSF,LTYP,OPTION
COMMON/IMPLCT/EXNEXT(51,21), EZNEXT(51,21), CSTAR(51,21),
1 VXNEXT(51,21), VZNEXT(51,21), A(51), B(51),
2 G(51), D(51), V(51)
DIMENSION NZREV(20)
DIMENSION R(51,21)
```
DAYS = TIME/86400.
HOURS=DAYS*24.
WRITE(6,FMT1) HOURS, DAYS
DO 100 IZ = 1, NZ
NZREV(IZ) = (FIX(DELTAVZ*(IZ-2))
100 CONTINUE
WRITE(6,FMT2) (NZREV(I), I=2, NZ+1)

C
C---------------------------------------------------------------
C
C ALLOW FOR A BACKGROUND CONCENTRATION OF 10.0
C---------------------------------------------------------------

DO 150 X=1,NX
DO 150 Z=1,NZ
R(X,Z) = C(X,Z,T) + 10.0
150 CONTINUE
XFIRST = XSTART - (NX-1)*DELTA*5280.
DO 200 X=1,NX, XPRINT
M = NX+1-X
XMILE = XFIRST + (X-1)*DELTA*XPRINT)/5280.
WRITE(6,FMT3) XMILE, (R(M,IZ), IZ=1,NZ)
200 CONTINUE
RETURN
END
SUBROUTINE ERRORI(*)

***************************
**** SUBROUTINE ERRORI ****
***************************

THIS SUBROUTINE CORRECTS SOME OF THE MOST COMMON ERRORS
IN THE INPUT DATA AND PRINTS OUT APPROPRIATE ERROR MESSAGES.

INTEGER X, Z, T, TNEXT, TPRINT, XPRINT, COUNT, OPTION
REAL KD, KDMAX, KDMIN, KDAYS, LOWER
COMMON/ARRAYS/ C(51, 21, 2), VX(51, 21), VZ(51, 21), EX(51, 21), EZ(51, 21),
1 W(51), CONCIN(51, 21), INSECT(51), KDAYS(51), KD(51),
2 DEMAND(51, 21), FMT1(40), FMT2(40), FMT3(20)
COMMON/NAMES/X, Z, T, TX, NX, NZ, NC, NS, EXMAX, EXMIN, VXMAX, VXMIN, WMAX, WMIN,
1 KDMAX, KDMIN, ITER, ITER1, ITER2, DELT1, DELT2, TNEXT,
2 TPRINT, XPRINT, STABLX, STABLZ, STABLT, R1, R2, VZMAX, SET,
3 VZMIN, EZMAX, EZMIN, DEMAX, DEMIN, NPAGE, DELTAX, DELTAY,
4 DELTAT, XSTART, XSF, TSTART, COUNT, IPAGE, UPPER, LOWER,
5 MSU, MSL, TIME, NXM1, NXM2, NZM1, NZM2, ZSF, LTYPE, OPTION
COMMON/IMPLCT/XNEXT(51, 21), EZNEXT(51, 21), CSTAR(51, 21),
1 VXNEXT(51, 21), VZNEXT(51, 21), A(51), B(51),
2 G(51), D(51), V(51)
IF(OPTION.GT.7) GO TO 180
IF(OPTION.EQ.7) GO TO 140
IF(OPTION.EQ.6) GO TO 120
GO TO 200
1759 120 CONTINUE
1760 OPTION=2
1761 PRINT 130
1762 130 FORMAT(///,/ , '****OPTION IS SET EQUAL TO 2. OPTION WAS EQUAL TO
1763 $6 ****',/,'1H1)
1764 IPAGE = 0
1765 GO TO 200
1766 140 CONTINUE
1767 OPTION=4
1768 PRINT 150
1769 150 FORMAT(///,/ , '**** OPTION IS SET EQUAL TO 4. OPTION WAS EQUAL TO
1770 $7 ****',/,'1H1)
1771 IPAGE=0
1772 GO TO 200
1773 180 CONTINUE
1774 PRINT 181
1775 181 FORMAT(///,/ , '**** PROGRAM TERMINATED BECAUSE OPTION WAS GREATER
1776 $THAN 7 ****',/,'1H1)
1777 RETURN
1778 200 IF(LTYPE.LT.3) GO TO 300
1779 PRINT 201
1780 201 FORMAT(///,/ , '**** PROGRAM TERMINATED BECAUSE LTYPE IS GREATER THAN OR EQUAL TO 3.**** IF LTYPE EQUALS 3, THE USER MUST SUPPLY $A SUBROUTINE FOR THE LOADING CONDITIONS.****',/,'1H1)
1781 RETURN
1782 300 CONTINUE
1783 RETURN
1784 END
Input Data for MASSTRANS-II

See Input Data for MASSTRANS-I.
COMPUTER PROGRAM FOR
OXTRANS-I

Object Code = 66,312 bytes
Array Area = 58,372 bytes
Total = 124,684 bytes
Oxtrans-I

Estuary Mass Transport and Oxygen Transport Model

Two Dimensional Numerical Analysis - Explicit Method

Logic and Programming by

P. Jonathan Young
C/O Environmental Engineering Division
Civil Engineering Department
Texas A&M University
College Station, Texas 77840

This program analyzes the relationship between a primary pollutant and a dissolved gas in a partially stratified estuary. The principle application of Oxtrans-I is to the interaction of biochemical oxygen demand (BOD) and dissolved oxygen (DO).
OXTRANS-I PERFORMS TWO BASIC CALCULATIONS DURING EACH TIME STEP. THE FIRST CALCULATION IS TO DETERMINE THE PROFILE FOR THE PRIMARY POLLUTANT, SUCH AS BOD, BY AN EXPLICIT FINITE DIFFERENCE APPROXIMATION TO THE MASS TRANSPORT EQUATION. THE SECOND CALCULATION USES INFORMATION FROM THE FIRST CALCULATION TO DETERMINE THE VALUES FOR THE DISSOLVED GAS. THIS PROGRAM HAS OPTIONS FOR ANAEROBIC CONDITIONS AND FOR MECHANICAL REAERATION. DURING A CALCULATION FOR BOD AND DD, THE PROGRAM AUTOMATICALLY SWITCHES TO ANAEROBIC DECAY RATES WHEREVER THE CONCENTRATION OF OXYGEN IS ZERO. REAERATION TERMS CAN BE APPLIED TO ANY POINT ON THE GRID AND, THUS, THE EFFECTS OF MECHANICAL AERATION AT ANY POINT OR POINTS CAN BE INVESTIGATED. AT THE USERS OPTION, THE PROFILES FOR THE PRIMARY POLLUTANTS ALONE CAN BE CALCULATED, AND THE CALCULATIONS FOR THE SECONDARY PARAMETERS CAN BE BY-PASSED.

THIS PROGRAM WAS DEVELOPED TO ANALYZE ESTUARIES WHOSE CHARACTERISTICS DO NOT VARY SIGNIFICANTLY WITH WIDTH. THE WIDTH OF THE ESTUARY MAY BE VARIED THROUGHOUT BUT THE CONCENTRATIONS OF DISSOLVED MATERIALS AT EACH CROSS SECTION ARE CONSIDERED TO BE UNCHANGING IN THE LATERAL DIRECTION. ALL PHYSICAL AND HYDRODYNAMIC CHARACTERISTICS MAY VARY WITH TIME IN THE LONGITUDINAL AND VERTICAL DIRECTIONS. THE PROGRAM CAN BE APPLIED WITH EQUAL EASE TO THE TWO HORIZONTAL DIRECTIONS, ALLOWING DEPTH TO VARY RATHER THAN WIDTH.

CONCENTRATION PROFILES CAN BE CALCULATED BY THIS PROGRAM FOR CONTINUOUS OR INSTANTANEOUS RELEASES. INPUT DATA MAY INCLUDE GRID DIMENSIONS, DISTANCE INCREMENTS, TIME INCREMENTS, WIDTHS, LOADING PARAMETERS, VELOCITIES, DISPERSION COEFFICIENTS, DECAY RATES, BENTHAL DEMANDS, REAERATION COEFFICIENTS, AEROBIC DECAY RATES, ANAEROBIC DECAY RATES, AND OTHER OXYGEN DEMANDS OR SOURCES.
THE TIME INCREMENTS CAN BE INCREASED OR DECREASED AT ANY TIME DURING THE CALCULATION OF THE CONCENTRATION PROFILE. THE DISTANCES BETWEEN GRID POINTS CAN BE INCREASED AT ANY TIME, AND A ROUTINE WITHIN THE PROGRAM WILL CHOOSE THE APPROPRIATE VALUES FROM THE PREVIOUSLY CALCULATED PROFILE AND WILL PLACE THESE VALUES IN THE DESIRED LOCATIONS IN THE NEW GRID SYSTEM. THE NUMBER OF GRID POINTS MAY BE INCREASED OR DECREASED AT ANY TIME. LIKewise, AT ANY TIME DURING THE CALCULATION, A NEW SET OF DATA FOR PHYSICAL AND HYDRODYNAMIC CONDITIONS MAY BE READ INTO THE PROGRAM.

A USER OF THIS PROGRAM MUST BE FAMILIAR WITH THE LIMITATIONS ON ACCURACY AND STABILITY INHERENT IN THE TYPE OF NUMERICAL PROCEDURE USED IN THESE CALCULATIONS. A SUBROUTINE WITHIN THE PROGRAM PRINTS OUT THE PROPER INCREMENTS TO INSURE STABILITY AND TERMINATES THE PROGRAM IF THIS CRITERIA IS VIOLATED BY THE INPUT PARAMETERS. ANOTHER SUBROUTINE EXTRAPOLATES CONCENTRATIONS AT THE BOUNDARIES—SEVERAL METHODS CAN BE USED FOR THESE EXTRAPOLATIONS DEPENDING ON THE TYPE OF PROFILE BEING ANALYZED AND THE CHOICE OF THE USER. A SUBROUTINE IS ALSO INCLUDED WHICH PRINTS OUT ERROR MESSAGES AND TERMINATES THE PROGRAM IF CERTAIN INCONSISTENCIES OCCUR IN THE INPUT DATA.
THIS PROGRAM WAS DEVELOPED PRIMARILY TO ANALYZE PARTIALLY
STRATIFIED ESTUARIES WHICH HAVE BEEN DREDGED OUT TO A FAIRLY
CONSTANT DEPTH AT THE CENTERLINE OF THE CHANNEL—THOSE ESTUARIES
ARE COMMON IN THE GULF COAST REGION. APPLICATION OF THIS PROGRAM
TO PARTIALLY STRATIFIED ESTUARIES WITH VARIABLE DEPTHS WOULD
REQUIRE MODERATE REVISIONS TO THE PROGRAM AND WOULD MAKE THE
PROGRAM ESTUARY-DEPENDENT.

THIS COMPUTER PROGRAM CAN ALSO BE APPLIED TO ESTUARIES WHICH
ARE WELL-MIXED IN THE VERTICAL DIRECTION. THIS OPTION ALLOWS FOR
VARYING WIDTH OR VARYING DEPTH AND USES MOST OF THE ROUTINES
AVAILABLE TO THE TWO-DIMENSIONAL ANALYSIS.

QUESTIONS REGARDING THIS PROGRAM MAY BE REFERRED TO
JONATHAN YOUNG AT HYDROSCIENCE, INC, 363 OLD HOOK ROAD,
WESTWOOD, NEW JERSEY 07675  PHONE 201 / 666-2600
***INPUT VARIABLES***

SET = DATA SET NUMBER
    --CAN BE EQUAL TO 1 IF NO NEW GRID DIMENSIONS
    ARE TO BE READ IN--MUST BE GREATER THAN 1 IF
    GRID DIMENSIONS ARE TO BE CHANGED
SECOND = .TRUE. FOR CALCULATION OF SECONDARY PARAMETERS
         = .FALSE. FOR CALCULATING PRIMARY POLLUTANTS ALONE

***************NOTE--OPTIONAL GRID ADJUSTMENT PARAMETERS ARE
         INSERTED HERE. THESE ARE EXPLAINED BELOW

NX = NUMBER OF GRID POINTS IN THE X DIRECTION
NZ = NUMBER OF GRID POINTS IN THE Z DIRECTION
DELMX = X INCREMENT ( MILES )
DELTZ = Z INCREMENT ( FEET )
DELT1 = FIRST TIME INCREMENT ( DAYS )
ITER1 = NUMBER OF ITERATIONS USING DELT1
DELT2 = SECOND TIME INCREMENT ( DAYS )
ITER2 = NUMBER OF ITERATIONS USING DELT2
NC = NUMBER OF GRID POINTS IN THE X DIRECTION WHERE
     CONCENTRATIONS ARE ADDED WITH THE DATA SET
NS = NUMBER OF POINT SOURCES
XSTART = MILEAGE AT UPPERMOST GRID POINT
TSTART = TIME TO BE ADDED TO THE TOTAL TIME AT THE FIRST
         ITERATION OF THE DATA SET BEING PROCESSED ( DAYS )
XPRINT = PRINTING INTERVAL FOR X DIRECTION ( INTEGER )
TPRINT = PRINTING INTERVAL FOR TIME ( INTEGER )
NPPAGE = NUMBER OF GRIDS PRINTED PER PAGE OF OUTPUT
LTYPE = TYPE OF LOADING (INSTANTANEOUS OR CONTINUOUS)
1 MEANS CONSTANT CONCENTRATION AT OUTFALL
   (THIS CAN BE USED FOR STEADY-STATE PROFILES)
2 MEANS AN INSTANTANEOUS RELEASE
3 MEANS CONTINUOUS LOADING

OPTION = OPTION FOR BOUNDARY CONDITIONS
1 MEANS NO TRANSFER ACROSS Z BOUNDARIES AND
   EXPONENTIAL EXTRAPOLATION IN X DIRECTION
2 MEANS NO TRANSFER ACROSS Z BOUNDARIES AND
   CONSTANT SLOPE EXTRAPOLATION IN X DIRECTION
3 MEANS A SYMMETRICAL DISTRIBUTION WITH
   EXPONENTIAL EXTRAPOLATION
4 MEANS A CONSTANT SLOPE EXTRAPOLATION IN
   BOTH DIRECTIONS
5 MEANS A SPECIAL CASE DEFINED BY THE USER
6 MEANS NO TRANSFER ACROSS Z BOUNDARIES AND
   EXTRAPOLATION IN X DIRECTION BY INVERTED
   DIFFERENCES (ACFI)
7 MEANS EXTRAPOLATION IN BOTH DIRECTIONS BY
   INVERTED DIFFERENCES (ACFI)

VX(X,Z) = HORIZONTAL VELOCITY (FT/SEC)
EX(X,Z) = HORIZONTAL DISPERSION (FT**2/SEC)
W(X) = WIDTH (FEET) FOR THE TWO DIMENSIONAL CASE OR
   CROSS SECTIONAL AREA (FT**2) FOR THE ONE
   DIMENSIONAL CASE
KDAYS(X) = DECAY RATE (PER DAY)
DEMAND(X,Z) = OTHER SOURCES AND SINKS (CONCENTRATION UNITS)
INSECT(I) = SECTIONS INTO WHICH ARE ADDED LOADS
   OR CONCENTRATIONS
CONCIN(I,J) = INITIAL CONCENTRATIONS ASSOCIATED WITH INSECT(I)
*************** NOTE -- VALUES FOR VZ AND EZ ARE TO BE INCLUDED
WITH THE DATA ONLY IF NZ IS GREATER THAN 1

VZ(X,Z) = VERTICAL VELOCITIES ( FT/SEC )
EZ(X,Z) = VERTICAL DISPERSION ( FT**2/SEC )

FMT1(I) = FORMAT FOR HEADING FOR PRIMARY POLLUTANT PROFILES
FMT2(I) = FORMAT FOR DEPTH NOTATION FOR PRIMARY POLLUTANTS
FMT3(I) = FORMAT FOR PRIMARY POLLUTANT PROFILES

*************** NOTE -- THE FOLLOWING VALUES ARE TO BE INCLUDED
IN THE INPUT DATA ONLY IF SECOND = .TRUE.

FMT4(I) = FORMAT FOR HEADING FOR SECONDARY PARAMETERS
FMT5(I) = FORMAT FOR DEPTH NOTATION FOR SECONDARY PROFILES
FMT6(I) = FORMAT FOR SECONDARY PARAMETER PROFILES
RDAYS(X,Z) = REAERATION COEFFICIENTS ( /DAY )
AKDAYS(X) = ANAEROBIC DECAY RATE ( /DAY )
OXSAT(X) = OXYGEN SATURATION VALUES (CONCENTRATION UNITS )
OXOUT(X,Z) = OXYGEN SINKS (CONCENTRATION UNITS )

*************** NOTE -- INITIAL CONCENTRATIONS MAY ALSO BE INPUT
THROUGH THE BLOCK DATA SUBPROGRAM
*** GRID ADJUSTMENT PARAMETERS ***

NXSKIP = NUMBER OF X INTERVALS SKIPPED IN OLD GRID
NZSKIP = NUMBER OF Z INTERVALS SKIPPED IN OLD GRID
NXTAKE = LOWEST GRID NUMBER IN X DIRECTION IN THE OLD GRID FROM WHICH A CONCENTRATION IS TAKEN
NZTAKE = LOWEST GRID NUMBER IN Z DIRECTION IN THE OLD GRID FROM WHICH A CONCENTRATION IS TAKEN
NXPUT = LOWEST GRID NUMBER IN THE X DIRECTION IN THE NEW GRID INTO WHICH A CONCENTRATION IS PLACED
NZPUT = LOWEST GRID NUMBER IN THE Z DIRECTION IN THE NEW GRID INTO WHICH A CONCENTRATION IS PLACED
THE MAIN PROGRAM CALLS FOR THE DATA TO BE READ IN, THEN CALLS FOR THE STABILITY CONDITIONS TO BE EVALUATED, AND NEXT CALLS FOR THE PRINTING OF A SUMMARY OF INPUT DATA. THE NUMERICAL ANALYSIS SUBROUTINE IS THEN CALLED FOR EITHER THE ONE-DIMENSIONAL OR TWO-DIMENSIONAL CASE. IF THE TIME INCREMENT IS TO BE CHANGED DURING THIS PART OF THE PROGRAM, THE APPROPRIATE PARAMETERS ARE ADJUSTED. STABILITY IS AGAIN CHECKED AND THE NUMERICAL ANALYSIS IS CONTINUED. NEXT, A NEW DATA SET MAY BE READ TO CONTINUE THE ANALYSIS. ANY NUMBER OF DATA SETS MAY BE PROCESSED. WHEN NO MORE DATA SETS ARE AVAILABLE, THE PROGRAM IS TERMINATED.


THE FORTRAN USED IN THIS PROGRAM USES SEVERAL OF THE OPTIONS AVAILABLE IN WATFOR AS DESCRIBED IN THE FOLLOWING TEXT* FORTRAN IV WITH WATFOR AND WATFIV** BY CRESS, DIRKSEN, & GRAHAM, PRENTICE-HALL, INC., 1970. THE PROGRAM CAN BE RUN ON A COMPUTER WITH A WATFOR OR WATFIV COMPILER. THE SYNTAX HAS BEEN KEPT COMPATIBLE WITH FORTRAN G EXCEPT FOR THE UNFORMATTED READ AND PRINT STATEMENTS.
1977          LOGICAL SECOND
1978          INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,CCOUNT,OPTION
1979          REAL KD,KMAX,KMIN,KDAYS,LOWER
1980          COMMON/ARRAYS/ C(51,21,4), VX(51,21), VZ(51,21), EX(51,21), EZ(51,21),
1                   W(51), CONCIN(51,21), INSECT(51), KDAYS(51), KD(51),
2                   DEMAND(51,21), FMT1(40), FMT2(40), FMT3(20)
1981          COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,WMAX,WMIN,
1                   KMAX,KMIN,ITER,ITER1,ITER2,DELT1,DELT2,TNEXT,
2                   TPRINT,XPRINT,STABLX,STABLY,STABLT,R1,R2,VZMAX,SET,
3                   VXMIN,EZMAX,EZMIN,DEMIX,DEMIN,NNPAGE,DELTAX,DELTAS,
4                   DELTAT,XSTART,XSF,TSTART,COUNT,IPAGE,UPPER,LOWER,
5                   MSU,MSL,TIME,NXM1,NXM2,NZM1,NZM2,ZSF,LYTE,OPTION
1982          COMMON/OXYGEN/OXOUT(51,21), R(51,21), R DAYS(51,21), O XSAT(51), AK(51),
1                   AKDAYS(51), FMT4(40), FMT5(40), FMT6(20),
2                   SECOND, AKMAX, RMAX, OUTMAX, AKMIN, RMIN, OUTMIN
1983           10 CALL DATA(61000)
1984           CALL STABLE
1985           CALL PRINT1
1986           CALL PRINT2
1987           T=3
1988           PRINT 15
1989           15 FORMAT(1H1)
1990           CALL PRINT2
1991           CALL ERROR(61000)
1992           25 IF(NZ.EQ.1) GC TO 310
1993           CALL TWODEX
1994           GO TO 315
1995           310 CALL ONEDEX
1996           315 IF(ITER2.EQ.0) GC TO 10
1997  ITER=ITER2
1998  ITER2=0
1999  TPRINT=TPRINT*(DELT1/DELT2+.00001)
2000  DELTAT=DELT2*86400.
2001  CALL STABLE
2002  IF(ITER.EQ.0) CALL ERROR(&1000)
2003  GO TO 25
2004  1000 CONTINUE
2005  STOP
2006  END
** BLOCK DATA

***********************
***** BLOCK DATA *****
***********************

THIS ROUTINE INITIALIZES SELECTED VARIABLES IN THE COMMON BLOCKS. THIS SUBPROGRAM MAY BE USED TO ESTABLISH INITIAL CONCENTRATION PROFILES IF THE USER DESIRES.

LOGICAL SECOND

INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,CCOUNT,OPTION

REAL KD,KDMAX,KDMIN,KDAYS,LOWER

COMMON/ARRAYS/ C(51,21,4),VX(51,21),VZ(51,21),EX(51,21),EZ(51,21),
1 W(51),CCINCIN(51,21),INSECT(51),KDAYS(51),KO(51),
2 DEMAND(51,21),FMT1(40),FMT2(40),FMT3(20)

COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,VZMAX,VZMIN,
1 KDMAX,KDMIN,ITER,ITER1,ITER2,DELT1,DELT2,TNEXT,
2 TPRINT,XPRINT,STABLX,STABLY,STABLB,R1,R2,VZMAX,SET,
3 VZMIN,EZMAX,EZMIN,DEMAX,DEMIN,NPNS,DN,DELT1,DELT2,
4 DELT2,XSTART,XT,STSTART,COUNT,IPAGE,UPPER,LOWER,
5 MSU,MSL,TIME,NXM1,NXM2,NZM1,NZM2,ZSF,LTYPE,OPTICN

COMMON/OXYGEN/OXOUT(51,21),R(51,21),RDAYS(51,21),OXSAT(51),AK(51),
1 AKDAYS(51),FMT4(40),FMT5(40),FMT6(20),
2 SECOND,AKMAX,AKMAX,OUTMAX,AKMIN,RMIN,OUTMIN
ESTABLISH INITIAL CONCENTRATION IF DESIRED

2014
DATA C / 9*13., 42*0., 9*13., 42*0., 9*13., 42*0., 9*13., 42*0.,
1 9*13., 42*0., 9*13., 42*0., 9*13., 42*0., 1785*0.0,
2 9*0., 42*8., 9*0., 42*8., 9*0., 42*8., 9*0., 42*8.,
3 9*0., 42*8., 9*0., 42*8., 9*0., 42*8., 1785*8. /

2015
DATA VX, VZ, EX, EZ, CONCIN, DEMAND/6426*0.0/

2016
DATA W, KDAYS, KD/153*0.0/

2017
DATA EXMAX, EZMAX, VXMAX, VZMAX, WMAX, KDMAX, DEMAX, TIME/8*0.0/

2018
DATA EXMIN, EZMIN, VXMIN, VZMIN, WMIN, KDMIN, DEMIN/7*1000000. /

2019
DATA INSECT/51*0/

2020
DATA IPAGE/1/

2021
DATA T/1/, TNEXT/2/

2022
DATA AKMAX, RMAX, OUTMAX/3*0.0/

2023
DATA AKMIN, RMIN, OUTMIN/3*1000000. /

2024
END
SUBROUTINE DATA(*)

******************************************************************************
****  SUBROUTINE DATA  ****
******************************************************************************

THIS SUBROUTINE READS IN THE APPROPRIATE DATA SET AND ADJUSTS
THE GRID SIZE IF NECESSARY. THE LOCATIONS OF THE UPPERMOST AND
LOWERMOST PEAKS ARE DETERMINED, AND THE MINIMUM AND MAXIMUM VALUES
FOR THE INPUT PARAMETERS ARE CALCULATED. ALL OF THE INPUT DATA IS
PRINTED OUT UNFORMATTED. THE FORMAT IS READ IN FOR THE PRINTING
OUT OF THE CALCULATED CONCENTRATIONS.

LOGICAL SECOND
INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,CCOUNT,OPTION
REAL KD,KMAX,KMIN,KDAYS,LOWER
COMMON/ARRAYS/ C(51,21,4), VX(51,21), VZ(51,21), EX(51,21), EZ(51,21),
1  W(51), CONCIN(51,21), INSECT(51), KDAYS(51), KD(51),
2  DEMAND(51,21), FMT1(40), FMT2(40), FMT3(20)
COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EZMAX,EZMIN,EXMAX,EXMIN,VXMAX,VXMIN,WMAX,WMIN,
1  KDMAX,KDMIN, ITER, ITER1, ITER2, DELT1, DELT2, TNEXT,
2  TPRINT, XPRINT, STABLX, STABLZ, STABLT, RL, R2, VZMAX, SET,
3  VZMIN, EZMAX, EZMIN, DEMAX, DEMIN, NPPAGE, DELTAX, DELTAZ,
4  DELTAT, XSTART, XSF, TSTART, COUNT, IPAGE, UPPER, LOWER,
5  MSU, MSL, TIME, NXML, NXM2, NZML, NZM2, ZSF, LTYPE, OPTION
COMMON/OXYGEN/OXOUT(51,21), R(51,21), RDAYS(51,21), OXSA(51), AK(51),
1  AKDAYS(51), FMT4(40), FMT5(40), FMT6(20),
2  SECOND, AKMAX, RMAX, OUTMAX, AKMIN, RMIN, OUTMIN
READ(5,12,END=13) SET
12 FORMAT(F10.0)
GO TO 14
13 CONTINUE
RETURN
14 CONTINUE
READ,SECOND
IF(SET.EQ.1.) GO TO 400
C
C---------------------------------------------------------------------
C
C CHOOSE CONCENTRATIONS FROM THE OLD GRID TO PUT INTO THE
C APPROPRIATE PLACES IN THE NEW GRID. THE GRID CAN ONLY BE
C CHANGED IF LTYPE EQUALS 2. OTHERWISE, THE DATA CARD IS READ
C BUT IS NOT IMPLEMENTED.
C---------------------------------------------------------------------

READ ,NXSKIP,NZSKIP,NXTAKE,NZTAKE,NXPUT,NZPUT
 IF(LTYPE.NE.2) GO TO 400
 NXSTOP=(NX-1)/NXSKIP+1
 NZSTOP=(NZ-1)/NZSKIP+1
 IF(NZ.EQ.1) NZSTOP=1
 DO 350 I=1,NXSTOP
   X=NXPUT-1+I
   II=NXTAKE+NXSKIP*(I-1)
 350 CONTINUE
 DO 400 J=1,NZSTOP
   Z=NZPUT-1+J
   JJ=NZTAKE+NZSKIP*(J-1)
   C(X,Z,TNEXT)=C(II,JJ,T)
 400 CONTINUE

350 CONTINUE
400 CONTINUE
READ,NX,NZ
READ,DELMX,DELTAZ
READ,DELT1,ITER1
READ,DELT2,ITER2
READ,NC,NS
READ,XSTART,TSTART,XPRINT,TPRINT,NP_PAGE
READ,LTYPE,OPTION
PRINT 1001
1001 FORMAT(1H1,10X,'INPUT DATA',/,,  
1,LX,*NX,NZ, DELMX,DELTAZ, DELT1,ITER1*)
PRINT,NX,NZ,DELMX,DELTAZ,DELT1,ITER1
PRINT 1010
1010 FORMAT(//,1X,'SECOND')
PRINT,SECOND
PRINT 1002
1002 FORMAT(//,1X,'DELT2,ITER2, NC,NS, XSTART,TSTART')
PRINT, DELT2,ITER2,NC,NS,XSTART,TSTART
PRINT 1003
1003 FORMAT(//,1X,'XPRINT,TPRINT,NP_PAGE, LTYPE,OPTION')
PRINT, XPRINT,TPRINT,NP_PAGE,LTYPE,OPTION
IF (SET.EQ.1.) GC TO 500
DO 375 Z=1,NZ
375 CONTINUE
DO 375 X=1,NX
C(X,Z,T)=C(X,Z,TNEXT)
C(X,Z,TNEXT)=0.0
375 CONTINUE
500 CONTINUE
NXM1=NX-1
NXM2=NX-2
NZM1=NZ-1
NZM2=NZ-2
DELTAT=DELT1*86400.
DELTAX=DELMAX*5280.
XSF=START*5280.
ITER=ITER1
TIME=TIME+TSTART*86400.
COUNT=(TSTART/DELTAT+0.01)
COUNT=COUNT-(COUNT/TPRINT)*TPRINT
IF(SEX,EQ,2) RETURN
READ,((VX(X,Z),X=1,NX),Z=1,NZ)
PRINT 2
2 FORMAT(/, 1X, ((((VX(X,Z),X=1,NX),Z=1,NZ)*)
PRINT,((VX(X,Z),X=1,NX),Z=1,NZ)
READ,((EX(X,Z),X=1,NX),Z=1,NZ)
PRINT 3
3 FORMAT(/, 1X, (((EX(X,Z),X=1,NX),Z=1,NZ)*)
PRINT,((EX(X,Z),X=1,NX),Z=1,NZ)
READ,(W(X),X=1,NX)
PRINT 4
4 FORMAT(/, 1X, ((W(X),X=1,NX))
PRINT, (W(X),X=1,NX)
READ,(KAYS(X),X=1,NX)
PRINT 5
5 FORMAT(/, 1X, ((KAYS(X),X=1,NX))
PRINT, (KAYS(X),X=1,NX)
READ,((DEMAND(X,Z),X=1,NX),Z=1,NZ)
PRINT 6
6 FORMAT(/,1X,"((DEMAND(X,Z),X=1,NX),Z=1,NZ)"
PRINT,((DEMAND(X,Z),X=1,NX),Z=1,NZ)
IF(NC.EQ.0) GO TO 24
DO 10 I=1,NC
10 READ,INSECT(I)
PRINT 7
7 FORMAT(/,1X,"INSECT(I)"
PRINT,INSECT(I)
READ,(CONCIN(I,J),J=1,NZ)
PRINT 8
8 FORMAT(/,1X,"(CONCIN(I,J),J=1,NZ)"
PRINT,(CONCIN(I,J),J=1,NZ)
1C CONTINUE
DO 15 I=1,NC
15 CONTINUE
C---------------------------------------------------------------------
C DETERMINE LOCATIONS OF UPPERMOST AND LOWERMOST
C PEAK CONCENTRATIONS
C---------------------------------------------------------------------
AMAX1=0.0
PEAKU=0.0
AMAX2=0.0
PEAKL=0.0
UPPER=XSF
LOWER=XSF
MSU=1
MSL=NX
DO 17 I=1,NC
DO 16 J=1,NZ
IF(CONCIN(I,J).LT.PEAKU) GO TO 16
PEAKU=CONCIN(I,J)
16 CONTINUE
IF(PEAKU.LE.AMAX1) GO TO 20
AMAX1=PEAKU
MSU=INSECT(I)
UPPER=(MSU-1)*DELTAX+XSF
17 CONTINUE
20 CONTINUE
DO 22 II=1,NC
I=NC+1-II
DO 21 J=1,NZ
IF(CONCIN(I,J).LT.PEAKL) GO TO 21
PEAKL=CONCIN(I,J)
21 CONTINUE
IF(PEAKL.LE.AMAX2) GO TO 24
AMAX2=PEAKL
MSL=INSECT(I)
LOWER=(MSL-1)*DELTAX+XSF
22 CONTINUE
24 CONTINUE
DO 25 X=1,NX
2159  KD(X) = KDAYS(X) / 86400.
2160  IF (W(X) .GT. WMAX) WMAX = W(X)
2161  IF (W(X) .LT. WMIN) WMIN = W(X)
2162  IF (K(X) .GT. KMAX) KMAX = K(X)
2163  IF (K(X) .LT. KMIN) KMIN = K(X)
2164  DO 25 Z = 1, NZ
2165     ACSVX = ABS(VX(X, Z))
2166     IF (ACSVX .GT. VXMAX) VXMAX = ACSVX
2167     IF (ACSVX .LT. VXMIN) VXMIN = ACSVX
2168     IF (EX(X, Z) .GT. EXMAX) EXMAX = EX(X, Z)
2169     IF (EX(X, Z) .LT. EXMIN) EXMIN = EX(X, Z)
2170     IF (DEMAND(X, Z) .GT. DEMAX) DEMAX = DEMAND(X, Z)
2171     IF (DEMAND(X, Z) .LT. DEMIN) DEMIN = DEMAND(X, Z)
2172  25 CONTINUE
2173  IF (NZ .EQ. 1) GO TO 200

C-----------------------------------------------
C-----------------------------READ AND PRINT DATA FOR VERTICAL DIRECTION-----------------------------
C-----------------------------------------------

2174  READ, ((VZ(X, Z), X=1, NX), Z=1, NZ)
2175  PRINT 30
2176  30 FORMAT (/,, 1X, '((VZ((X,Z)),X=1,NX),Z=1,NZ)')
2177  PRINT, ((VZ(X, Z), X=1, NX), Z=1, NZ)
2178  READ, ((EZ(X, Z), X=1, NX), Z=1, NZ)
2179  PRINT 32
2180  32 FORMAT (/,, 1X, '((EZ((X,Z)),X=1,NX),Z=1,NZ)')
2181  PRINT, ((EZ(X, Z), X=1, NX), Z=1, NZ)

C-----------------------------------------------
C---------------------DETERMINE MAXIMUM AND MINIMUM VALUES FOR INPUT DATA---------------------
C-----------------------------------------------

DO 120 X=1,NX
DO 120 Z=1,NZ
ABS VZ=ABS(VZ(X,Z))
IF(ABS VZ.GT.VMAX)VMAX=ABS VZ
IF(ABS VZ.LT.VMIN)VMIN=ABS VZ
IF(EZ(X,Z).GT.EZMAX)EZMAX=EZ(X,Z)
IF(EZ(X,Z).LT.EZMIN)EZMIN=EZ(X,Z)
120 CONTINUE
200 CONTINUE

C
C-------------------------------------------------------------
C READ AND PRINT FORMATS FOR OUTPUT
C-------------------------------------------------------------

2191 READ 205, (FMT1(I), I=1,40)
2192 PRINT 202
2193 202 FORMAT(/,1X, "FORMAT FOR OUTPUT")
2194 PRINT 205, (FMT1(I), I=1,40)
2195 205 FORMAT (20A4,/,20A4)
2196 READ 205,(FMT2(I), I=1,40)
2197 PRINT 205, (FMT2(I), I=1,40)
2198 READ 207,(FMT3(I), I=1,20)
2199 PRINT 207, (FMT3(I), I=1,20)
2200 207 FORMAT(20A4)
2201 IF(.NOT.SECOND) GO TO 250
2202 READ 205, (FMT4(I), I=1,40)
2203 PRINT 205, (FMT4(I), I=1,40)
2204 READ 205, (FMT5(I), I=1,40)
2205 PRINT 205, (FMT5(I), I=1,40)
2206 READ 207, (FMT6(I), I=1,20)
2207 PRINT 207, (FMT6(I),I=1,20)
2208 READ, ((RDAYS(X,Z),X=1,NX),Z=1,NZ)
2209 PRINT 210
2210 FORMAT(//'1X,((RDAYS(X,Z),X=1,NX),Z=1,NZ)')
2211 PRINT, ((RDAYS(X,Z),X=1,NX),Z=1,NZ)
2212 READ, (AKDAYS(X),X=1,NX)
2213 PRINT 215
2214 FORMAT(//'1X,(AKDAYS(X),X=1,NX)'),
2215 PRINT, (AKDAYS(X),X=1,NX)
2216 READ, (OXSAT(X),X=1,NX)
2217 PRINT 220
2218 FORMAT(//'1X,(OXSAT(X),X=1,NX)'),
2219 PRINT, (OXSAT(X),X=1,NX)
2220 READ, ((OXOUT(X,Z),X=1,NX),Z=1,NZ)
2221 PRINT 222
2222 FORMAT(//'1X,((OXOUT(X,Z),X=1,NX),Z=1,NZ)'),
2223 PRINT, ((OXOUT(X,Z),X=1,NX),Z=1,NZ)
2224 DO 225 X=1,NX
2225 AK(X)=AKDAYS(X)/86400.
2226 IF(AK(X).GE.AKMAX) AKMAX=AK(X)
2227 IF(AK(X).LT.AKMIN) AKMIN=AK(X)
2228 DO 225 Z=1,NZ
2229 R(X,Z)=RDAYS(X,Z)/86400.
2230 IF(R(X,Z).GE.RMAX) RMAX=R(X,Z)
2231 IF(R(X,Z).LT.RMIN) RMIN=R(X,Z)
2232 IF(OXOUT(X,Z).GE.OUTMAX) OUTMAX=OXOUT(X,Z)
2233 IF(OXOUT(X,Z).LT.OUTMIN) OUTMIN=CXCUT(X,Z)
2234 225 CONTINUE
2235 250 CONTINUE
2236 PRINT 259
2237 259 FORMAT(1H1)
2238 RETURN
2239 END
THE RELATIONSHIP BETWEEN DISTANCE INCREMENTS, TIME INCREMENTS, DISPERSION COEFFICIENTS, AND VELOCITIES IS NEEDED TO DETERMINE THE STABILITY OF THE EXPLICIT FINITE-DIFFERENCE PROCEDURE. THIS SUBROUTINE CALCULATES THE MAXIMUM ALLOWABLE INCREMENTS FOR TIME AND DISTANCE. THE PROGRAM IS TERMINATED IF THE INPUT PARAMETERS VIOLATE THIS CRITERIA.

SUBROUTINE STABLE

LOGICAL SECOND
INTEGER KMAX, KDAY, XMAX, NDAY, XN, NZ, NDAY, XMAX, EEHMIN, VMIN, DEMAND, KMIN, DAY, XSTAR, XPRINT, COUNT, OPTION
REAL KMAX, KDAY, XMAX, KMIN, NC, NDAY, XMAX, EEHMIN, VMIN, DEMAND, KMIN, DAY, XSTAR, XPRINT, COUNT, OPTION
STABLX = DELTAX
STABLZ = DELTAZ
STABLT = DELTAT

C
---------
C            CALCULATE STABILITY CRITERIA FOR ONE DIMENSIONAL CASE
C
---------

2250 IF(NZ .NE. 1) GO TO 500
2251 IF(EXMIN.GT.0.0.AND.VXMAX.GT.0.0) STABLX=2.*EXMIN/VXMAX
2252 IF(STABLX.LT.DELTAX) ITER=0
2253 TERM=2.*EXMAX+DELTAX**2.*KDMAX
2254 IF(ITER.NE.0.0) STABLT=(DELTAX**2.)/TERM
2255 IF(STABLT.LT.DELTAT) ITER=0
2256 R1=DELTAT/(DELTAX**2.)
2257 R2=0.
2258 GO TO 800
2259 500 CONTINUE

C
---------
C            CALCULATE STABILITY CRITERIA FOR TWO DIMENSIONAL CASE
C
---------

2260 IF(EXMIN.GT.0.0.AND.VXMAX.GT.0.0) STABLX=2.*EXMIN/VXMAX
2261 IF(STABLX.LT.DELTAX) ITER=0
2262 IF(EXMIN.GT.0.0.AND.VZMAX.GT.0.0) STABLZ=2.*EZMIN/VZMAX
2263 IF(STABLZ.LT.DELTAX) ITER=0
2264 TERM=2.*VXMAX*DELTAX**2.*EZMAX*DELTAX**2.*KDMAX*(DELTAX**2.)*
     (DELTAX**2.)*
2265 IF(ITER.NE.0.0) STABLT=(DELTAX**2.)*(DELTAX**2.)/TERM
2266 IF(STABLT.LT.DELTAT) ITER=0
2267 R1=DELTAT/(DELTAX**2.)
2268 R2=DELTAT/(DELTAX**2.)
2269 800 CONTINUE
2270 RETURN
2271 END
SUBROUTINE ONEDEX

*** SUBROUTINE ONEDEX ***

THIS SUBROUTINE CALCULATES THE ONE DIMENSIONAL CONCENTRATION PROFILE FOR THE ASSIGNED GRID AND THE ASSIGNED NUMBER OF ITERATIONS. A FINITE-DIFFERENCE, EXPLICIT SCHEME IS USED TO SOLVE THE PARTIAL DIFFERENTIAL EQUATION FOR MASS TRANSFER.

LOGICAL SECOND
INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,COUNT,OPTION
REAL KD,KDAX,KDMIN,KDAYS,LOWER
COMMON ARRAYS: C(51,21,4), VX(51,21), VZ(51,21), EX(51,21), EZ(51,21), W(51), CONCIN(51,21), INSECT(51), KDAYS(51), KD(51), DEMAND(51,21), FMT1(40), FMT2(40), FMT3(20)
COMMON NAMES: X,Z,T,NX,NZ,NS,EXMAX,EXMIN,VXMAX,VXMIN,WMAX,WMIN,
               KMAX,KDMIN,ITER,ITER1,ITER2,DELT1,DELT2,TNEXT,
               TPRINT,XPRINT,STABLX,STABLY,STABLZ,STABLX,R1,R2,VZMAX,SET,
               VZMIN,EZMAX,EZMIN,DEMEX,DEMIN,NPGE,DELTAX,DELTZ,
               DELTAT,XSTART,XF,F,TSTART,COUNT,TSPACE,UPPER,LOWER,
               MSU,MSL,TIME,NXM1,NXM2,NZM1,NZM2,MSF,MSF2,OPTION
COMMON/OXYGEN: OXOUT(51,21), R(51,21), RDAYS(51,21), OXSAT(51), AK(51),
               AKDAYS(51), FMT4(40), FMT5(40), FMT6(20),
               SECOND, AKMAX, RMAX, OMAX, AKMIN, RMIN, OMIN

Z=1
NXM1=NX-1
C
C---------------------------------------------------
C CALCULATE TERMS WHICH ARE USED REPEATEDLY
C---------------------------------------------------
2281  TWOXSQ=2.*DELTAX**2.
2282  TWDX=2.*DELTAX
C
C---------------------------------------------------
C CALCULATE NEW CONCENTRATION AT EACH POINT
C---------------------------------------------------
2283  DO 300 IT=1,ITER
2284     TIME=TIME+DELTAT
2285     T=1
2286     TNEXT=2
2287  10 DO 200 X=2,NXM1
2288     DECAY=KD(X)
2289     IF(.NOT.SECOND) GO TO 15
2290     IF(C(X,Z,3).LE.0.) DECAY=AK(X)
2291  15 CONTINUE
2292     TERM1A=(W(X)*VX(X,Z))/TWDX
2293     TERM1B=(W(X)*EX(X,Z)+W(X-1)*EX(X-1,Z))/TWOXSQ
2294     TERM2B = (W(X+1)*EX(X+1,Z) + W(X)*EX(X,Z)) / TWOXSQ
2295     TERM3A=(W(X)*VX(X,Z))/TWDX
2296     TERM3B=(W(X+1)*EX(X+1,Z)*W(X)*EX(X,Z))/TWOXSQ
2297     TERM1=C(X-1,Z,T)*(TERM1A+TERM1B)
2298     TERM2=-C(X,Z,T)*TERM2B
2299     TERM3=C(X+1,Z,T)*(TERM3A+TERM3B)
2300     IF(T.EQ.3) GO TO 50
C(X,Z,2) = C(X,Z,1) + (DELTAT/W(X)) *(TERM1+TERM2+TERM3)
   - DEMAND(X,Z) - DECAY*DELTAT*C(X,Z,1)
   $\frac{\partial C}{\partial t} = \frac{\partial}{\partial x} \left( D \frac{\partial C}{\partial x} \right) + \frac{\partial}{\partial z} \left( \frac{D}{\kappa} \frac{\partial C}{\partial z} \right) - \dot{Q}(X,Z)
   = \dot{Q}(X,Z) - \dot{Q}(X,Z) = 0

GO TO 75
50 CONTINUE
C(X,Z,4) = -DECAY*DELTAT*C(X,Z,4) - OXOUT(X,Z)
   + R(X,Z) *(OXSAT(X) - C(X,Z,3)) * DELTAT
C(X,Z,4) = C(X,Z,3) *(DELTAT/W(X)) *(TERM1+TERM2+TERM3) + C(X,Z,4)
75 IF(C(X,Z,TNEXT).LT.0.0) C(X,Z,TNEXT) = 0.0
200 CONTINUE

C-----------------------------------------------------
C CALCULATE BOUNDARY VALUES
C-----------------------------------------------------

CALL BOUND
Z = 1
DO 250 X = 1, NX
   IF(TNEXT .EQ. 2) C(X,Z,4) = C(X,Z,1)
   C(X,Z,T) = C(X,Z,TNEXT)
   IF(C(X,Z,3).GT.OXSAT(X)) C(X,Z,3) = OXSAT(X)
   C(X,Z,TNEXT) = 0.0
250 CONTINUE
IF(T.EQ.1) COUNT = COUNT + 1
IF(COUNT.GE.TPRINT) GO TO 292
298 CONTINUE

C-----------------------------------------------------
C PRINT CONCENTRATIONS AT APPROPRIATE INTERVALS
C-----------------------------------------------------

292 CONTINUE
IF(T.EQ.3.OR.(.NOT.SECOND)) COUNT = 0
2321 IPAGE=IPAGE+1
2322 IF(IPAGE.GE.NPPAGE) GO TO 295
2323 GO TO 297
2324 295 PRINT 296
2325 296 FORMAT(1H1)
2326 IPAGE = 0
2327 CALL PRINT2
2328 298 CONTINUE
2329 IF((.NOT.SECOND).OR.(T.EQ.3)) GO TO 300
2330 T=3
2331 TNEXT=4
2332 GO TO 10.
2333 300 CONTINUE
2334 RETURN
2335 END
SUBROUTINE TWODEX

THIS SUBROUTINE CALCULATES THE TWO-DIMENSIONAL CONCENTRATION PROFILE FOR THE ASSIGNED GRID AND THE ASSIGNED NUMBER OF ITERATIONS. A FINITE-DIFFERENCE, EXPLICIT SCHEME IS USED TO SOLVE THE PARTIAL DIFFERENTIAL EQUATION FOR MASS TRANSPORT.

LOGICAL SECOND
INTEGER X, Z, T, TNEXT, TPRINT, XPRINT, COUNT, OPTION
REAL KD, KMAX, KMIN, KDAYS, LOWER
COMMON/ARRAYS/C(51, 21, 4), VX(51, 21), VZ(51, 21), EX(51, 21), EZ(51, 21),
1 W(51), CONCIN(51, 21), INSECT(51), KDAYS(51), KD(51),
2 DEMAND(51, 21), FMT1(40), FMT2(40), FMT3(20)
COMMON/NAMES/X, Z, T, NX, NZ, NC, NS, EMAX, EXMIN, VXMAX, VXMIN, WMAX, WMIN,
1 KMAX, KMIN, ITER, ITER1, ITER2, DELT1, DELT2, TNEXT,
2 TPRINT, XPRINT, STABRXI, STABRXII, STABLX, R1, R2, VZMAX, SET,
3 VZMIN, EZMAX, EZMIN, DEMAX, DEMIN, NNPAGE, DELTAX, DELTAZ,
4 DELTAT, XSTART, XSF, TSTART, COUNT, IPAGE, UP, LOWER,
5 MSU, MSL, TIME, NXM1, NXM2, NZM1, NZM2, ZSF, LTYPE, OPTION
COMMON/OXYGEN/OXOUT(51, 21), R(51, 21), RDAYS(51, 21), OXSAT(51), AK(51),
1 AKDAYS(51), FMT4(40), FMT5(40), FMT6(20),
2 SECOND, AKMAX, RMAX, OUTMAX, AKMIN, RMIN, OUTMIN
NXM1 = NX - 1
NZM1 = NZ - 1
NZ1 = 2
C
C -----------------------------------------------
C CALCULATE TERMS WHICH ARE USED REPEATEDLY
C -----------------------------------------------
2346  TWOXSQ=2.*DELTAX**2.
2347   TWOX=2.*DELTAX
2348  TWOZSQ=2.*DELTAZ**2.
2349   TWOZ=2.*DELTAZ
C
C -----------------------------------------------
C CALCULATE NEW CONCENTRATION AT EACH POINT
C -----------------------------------------------
2350  DO 300 IT=1,ITER
2351    TIME=TIME+DELTAT
2352    T=1
2353   TNEXT=2
2354  10 DO 200 X=2,NXM1
2355   DO 100 Z=NZ1,NZM1
2356    DECAY=KD(X)
2357   IF(.NOT.SECOND) GO TO 15
2358   IF(C(X,Z,3).LE.0.) DECAY=AK(X)
2359  15 CONTINUE
2360   TERM1A=W(X)*VX(X,Z)/TWXC
2361   TERM1B=(W(X)*EX(X,Z)+W(X-1)*EX(X-1,Z))/TWOXSQ
2362   TERM2B = (W(X+1)*EX(X+1,Z) + W(X)*EX(X,Z)) / TWOXSQ
2363    +(W(X)*EX(X,Z) + W(X-1)*EX(X-1,Z)) / TWOXSQ
2364   TERM3A=(-W(X)*VX(X,Z))/TWOX
2365   TERM3B=(W(X+1)*EX(X+1,Z)+W(X)*EX(X,Z))/TWOXSQ
TERM4A = \(\frac{W(X) \cdot VZ(X,Z)}{TWGZ}\)

TERM4B = \(\frac{W(X) \cdot EZ(X,Z) + W(X) \cdot EZ(X,Z-1)}{TWOZSQ}\)

TERM5A = \(\frac{-W(X) \cdot VZ(X,Z)}{TWOZ}\)

TERM5B = \(\frac{W(X) \cdot EZ(X,Z+1) + W(X) \cdot EZ(X,Z)}{TWOZSQ}\)

TERM6B = \(\frac{W(X) \cdot EZ(X,Z+1) + W(X) \cdot EZ(X,Z)}{TWOZSQ}\)

\[ + (W(X) \cdot EZ(X,Z) + W(X) \cdot EZ(X,Z-1)) \div TWOZSQ \]

TERM1 = \(C(X-1,Z,T) \times (TERM1A + TERM1B)\)

TERM2 = \(-C(X,Z,T) \times (TERM2B + TERM6B)\)

TERM3 = \(C(X+1,Z,T) \times (TERM3A + TERM3B)\)

TERM4 = \(C(X,Z-1,T) \times (TERM4A + TERM4B)\)

TERM5 = \(C(X,Z+1,T) \times (TERM5A + TERM5B)\)

IF \((T.EQ.3)\) GO TO 50

\[ C(X,Z,2) = C(X,Z,1) + (DELTAT/W(X)) \times (TERM1 + TERM2 + TERM3 + TERM4 + TERM5) \]

\[ \times \quad -DEMAND(X,Z) - DECAY \times DELTAT \times C(X,Z,1) \]

GO TO 75

50 CONTINUE

\[ C(X,Z,4) = \quad -DECAY \times DELTAT \times C(X,Z,4) - OXOUT(X,Z) \]

\[ + R(X,Z)(OXSAT(X) - C(X,Z,3)) \times DELTAT \]

\[ C(X,Z,4) = C(X,Z,3) + (DELTAT/W(X)) \times (TERM1 + TERM2 + TERM3 + TERM4 + TERM5) \]

\[ + C(X,Z,4) \]

75 IF \((C(X,Z,TNEXT) . LT. 0.0)\) \(C(X,Z,TNEXT) = 0.0\)

100 CONTINUE

200 CONTINUE

C

C-------------------------------------------------------------

C CALCULATE BOUNDARY VALUES

C-------------------------------------------------------------

2384 CALL BOUND

2385 DO 250 Z=1,NZ

2386 DO 250 X=1,NX
IF(TNEXT.EQ.2) C(X,Z,4)=C(X,Z,1)
C(X,Z,T)=C(X,Z,TNEXT)
IF(C(X,Z,3).GT.OXSAT(X)) C(X,Z,3) = CXSAT(X)
C(X,Z,TNEXT)=0.0
2391 CONTINUE
2392 IF(T.EQ.1) COUNT=COUNT+1
2393 IF(COUNT.GE.TPRINT) GO TO 292
2394 GO TO 298

C-----------------------------------------------
C PRINT CONCENTRATIONS AT APPROPRIATE INTERVALS
C-----------------------------------------------
2395 CONTINUE
2396 IF(T.EQ.3.OR.(.NOT.SECOND)) COUNT=0
2397 IPAGE=IPAGE+1
2398 IF(IPAGE.GE.NPAGE) GO TO 295
2399 GO TO 297
2400 PRINT 296
2401 296 FORMAT(1H1)
2402 IPAGE = 0
2403 CALL PRINT2
2404 CONTINUE
2405 IF((.NOT.SECOND).OR.(T.EQ.3)) GO TO 300
2406 T=3
2407 TNEXT=4
2408 GO TO 10
2409 CONTINUE
2410 RETURN
2411 END
SUBROUTINE BOUND

*******************************************
***** SUBROUTINE BOUND *****
*******************************************

SUBROUTINE BOUND PROVIDES FOR EXTRAPOLATION OF CONCENTRATIONS AT THE BOUNDARIES. THE TYPE OF EXTRAPOLATION DEPENDS ON THE VALUE OF THE VARIABLE OPTION.

LOGICAL SECOND
INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,COUNT,OPTION
REAL KD,KMAX,KMIN,KDAYS,LOWER
COMMON/ARRAYS/ C(51,21,4),VX(51,21),VZ(51,21),EX(51,21),EZ(51,21),
  1 W(51),CONCIN(51,21),INSECT(51),KDAYS(51),KD(51),
  2 DEMAND(51,21),FMT1(40),FMT2(40),FMT3(20)
COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,WMAX,WMIN,
  1 KMAX,KMIN,ITER,ITER1,ITER2,DELT1,DELT2,TNEXT,
  2 TPRINT,XPRINT,STABLX,STABLZ,STABLT,R1,R2,VZMAX,SET,
  3 VZMIN,EZMAX,EZMIN,DEMAX,DEMIN,NPAGEN,DELTA,X,DELTAZ,
  4 DELTAT,XSTART,XSF,TSTART,COUNT,IPAGE,UPPER,LOWER,
  5 MSU,MSL,TIME,NXMI,NXM2,NZM1,NZM2,ZSF,LTYPE,OPTION
COMMON/OXYGEN/OXOUT(51,21),R(51,21),RDAYS(51,21),OXSAT(51),AK(51),
  1 AKDAYS(51),FMT4(40),FMT5(40),FMT6(20),
  2 SECOND,AKMAX,RMAX,OUTMAX,AKMIN,RMIN,OUTMIN
IF(NZ.EQ.1) NZM1=1
ZUP=UPPER
ZLOW=LOWER
ZSF=XSZ
GO TO (1001,1002,1003,1004,1005,1006), OPTION
C
C-------------------------------------------------------------
C EXPONENTIAL EXTRAPOLATION
C-------------------------------------------------------------
1003 CONTINUE
IF(NZ.EQ.1) GO TO 1001
DO 140 X=2,NXM1
IF(C(X,2,T+1),LE,0.0) GO TO 140
VZ1=VZ(X,1)
ZZ=-ZUP+DELTAZ+ZSF+VZ1*TSTART*86400.
EZ1=EZ(X,1)
C2=4.*EZ1*TIME
EXPON=(2.*ZZ*DELTAZ-(DELTAZ*DELTAZ)-2.*DELTAZ*VZ1*TIME)/C2
C(X,1,T+1)=C(X,2,T+1)*EXP(EXPON)
140 CONTINUE
DO 190 X=2,NXM1
IF(C(X,NZM1,T+1),LE,0.0) GO TO 190
VZNZ=VZ(X,NZ)
ZZ=(NZ-2)*DELTAZ-ZLOW+ZSF+VZNZ*TSTART*86400.
EZNZ=EZ(X,NZ)
C2=4.*EZNZ*TIME
EXPON=(-2.*ZZ*DELTAZ-(DELTAZ*DELTAZ)+2.*DELTAZ*VZNZ*TIME)/C2
C(X,NZ,T+1)=C(X,NZM1,T+1)*EXP(EXPON)
190 CONTINUE
1001 CONTINUE
IF(LTYPE.EQ.1.AND.MSU.EQ.1) GO TO 350
DO 340 Z=1,NZ
2447 IF(C(2,Z,T+1)*LE.0.0) GO TO 340
2448 VX1=VX(1,Z)
2449 XX=-UPPER+DELTAX+XSF+VX1*TSTART*86400.
2450 EX1=EX(1,Z)
2451 C2=4.*EX1*TIME
2452 EXPON=(2.*XX*DELTAX-(DELTAX*DELTAX)-2.*DELTAX*VX1*TIME)/C2
2453 C(1,Z,T+1)=C(2,Z,T+1)*EXP(EXPON)
2454 340 CONTINUE
2455 350 CONTINUE
2456 IF(LTYPE.EQ.1.AND.MSL.EQ.NZ) GO TO 400
2457 DO 390 Z=1,NZ
2458 IF(C(NX-1,Z,T+1)*LE.0.0) GO TO 390
2459 VXNX=VX(NX,Z)
2460 XX=(NX-2)*DELTAX-LOWER+XSF+VXNX*TSTART*86400.
2461 EXNX=EX(NX,Z)
2462 C2=4.*EXNX*TIME
2463 EXPON=(2.*XX*DELTAX-(DELTAX*DELTAX)+2.*DELTAX*VXNX*TIME)/C2
2464 C(NX,Z,T+1)=C(NX-1,Z,T+1)*EXP(EXPON)
2465 390 CONTINUE
2466 400 CONTINUE
2467 GO TO 2000
2468
C
C-------------------------------
C CONSTANT SLOPE EXTRAPOLATION
C
C-------------------------------
2469 1004 CONTINUE
2470 IF(NZ.EQ.1) GO TO 1002
2471 DO 540 X=2,NXM1
2472 C(X,1,T+1)=2.*C(X,2,T+1)-C(X,3,T+1)
2473 IF(C(X,1,T+1)*LT.0.0) C(X,1,T+1)=0.0
C(X,NZ,T+1) = 2.*C(X,NZM1,T+1) - C(X,NZM2,T+1)
IF(C(X,NZ,T+1) .LT. 0.0) C(X,NZ,T+1) = 0.0
540 CONTINUE
1002 CONTINUE
DO 590 Z = 1, NZ
  C(1, Z, T+1) = 2.*C(2, Z, T+1) - C(3, Z, T+1)
  IF(C(1, Z, T+1) .LT. 0.0) C(1, Z, T+1) = 0.0
  C(NX, Z, T+1) = 2.*C(NXM1, Z, T+1) - C(NXM2, Z, T+1)
  IF(C(NX, Z, T+1) .LT. 0.0) C(NX, Z, T+1) = 0.0
590 CONTINUE
GO TO 2000
1005 CONTINUE
C
C THIS OPTION ALLOWS THE PROGRAM USER TO SUBSTITUTE HIS OWN
C EXTRAPOLATION ROUTINE
C
Z = -40.
EZ1 = 0.025
VZ1 = 0.
C2 = 4.*EZ1*TIME
EXPON = (+2.*Z*DELTAZ - (DELTAZ**2.0) - 2.*DELTAZ*VZ1*TIME)/C2
DO 240 X = 1, NX
  C(X,1,T+1) = C(X,2,T+1) * EXP(EXPON)
  C(X,NZ,T+1) = C(X,1,T+1)
240 CONTINUE
GO TO 1001
C
C INVERTED DIFFERENCES EXTRAPOLATION (ACF1)
C

!----------------------------------------!
2495 1006 CALL EXTRAP
2496   GO TO 2000
2497 2000 CONTINUE
2498 IF(NZ.EQ.1) GO TO 2005
2499 IF(OPTION.EQ.3.OR.OPTION.EQ.4.OR.OPTION.EQ.5) GO TO 2005
2500 DO 700 X=1,NX
2501 C(X,1,T+1)=C(X,2,T+1)
2502 C(X,NZ,T+1)=C(X,NZ+1,T+1)
2503 700 CONTINUE
2504 2005 CONTINUE
C
C
C IF LTYPE EQUALS 1, SET CONCENTRATIONS AT SOURCE POINTS.
C IF LTYPE EQUALS 3, THE USER CAN INCLUDE A SPECIAL SUBROUTINE
C OR SET OF CALCULATIONS IN THIS PART OF THE PROGRAM.
C
C
2505 IF(T.EQ.3) GO TO 290
2506 IF(LTYPE.EQ.1.AND.NS.LE.2) GO TO 275
2507 IF(LTYPE.EQ.1.AND.NS.GT.2) GO TO 281
2508 GO TO 290
2509 275 CONTINUE
2510 DO 280 Z=1,NZ
2511 C(MSU,Z,T+1)=C(MSU,Z,T)
2512 C(MSL,Z,T+1)=C(MSL,Z,T)
2513 280 CONTINUE
2514 GO TO 290
2515 281 CONTINUE
2516 DO 285 I=1,NC
2517 DO 285 Z=1,NZ
2518 C(INSECT(I),Z,T+1)=CONCIN(I,Z)
2519 285 CONTINUE
2520 290 CONTINUE
2521 RETURN
2522 END
SUBROUTINE EXTRAP

***********************
***** SUBROUTINE EXTRAP *****
***********************

THIS SUBROUTINE EXTRAPOLATES THE CONCENTRATION PROFILE BY USING A CONTINUED FRACTIONS AND INVERTED DIFFERENCES SCHEME. IT IS A MODIFIED VERSION OF THE IBM SCIENTIFIC SUBPROGRAM ACFI.


LOGICAL SECOND

INTEGER X,Z,I,T,NEXT,PRINT,PRINT,CCOUNT,OPTION
REAL KD,KMAX,KDMIN,KDAYS,LOWER
COMMON/ARRAYS/C(51,21,4),VX(51,21),VZ(51,21),EX(51,21),EZ(51,21),
W(51),CCMIN(51,21),INSECT(51),KDAYS(51),KDI(51),
DEMAND(51,21),FMT1(40),FMT2(40),FMT3(20)
COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,WMAX,WMIN,
KMAX,KMIN,ITER,ITER,ITER,DELTA1,DELTA2,NEXT,
PRINT,PRINT,STABLN,STABLT,STABLT,R1,R2,VZMAX,SET,
VZMIN,IZMAX,IZMIN,IZMAX,IZMIN,NPPAGE,DELTA,DELTAZ,
DELTAT,XSTART,XSF,TSTART,COUNT,IPAGE,UPPER,LOWER,
MSU,MSL,TIME,NXMI,NXMX,NZMI,NZMX,2SF,LTYPE,OPTION
COMMON/OXYGEN/OXOUT(51,21),O(51,21),RDAY(51,21),OXSAT(51),AK(51),
RDAYS(51),FMT4(40),FMT5(40),FMT6(20),
SECOND,KMAX,RMAX,OUTMAX,AKMIN,ROIMIN,OUTMIN
2530  DIMENSION ARG(10), VALY(10)
2531  NZ1=2
2532  IF(NZ.EQ.1) NZ1=1
2533  IF(NZ.EQ.1) NZM1=1
2534  IF(MSU.EQ.1.AND.LTYPE.EQ.1) GO TO 228

C
C--------------------------------------------------------
C                EXTRAPOLATE THE PROFILE IN THE X DIRECTION
C--------------------------------------------------------
2535  DO 225 Z=NZ1,NZM1
2536  IF(C(2, Z, T+1) .LE. 0.00) GO TO 220

C
C--------------------------------------------------------
C                CHOOSE THE NUMBER OF POINTS (NDIM) TO BE USED IN THE
C                EXTRAPOLATION.
C--------------------------------------------------------
2537  NDIM=3
2538  NSTOP=NDIM+1

C
C--------------------------------------------------------
C                PLACE CONCENTRATIONS IN PROPER ORDER FOR SUBROUTINE ACFI
C--------------------------------------------------------
2539  DO 210 X=Z,NSTOP
2540  ARG(X - 1) = FLOAT(X)
2541  VALY(X-1) = C(X, Z, T+1)
2542  IF(VALY(X-1) .LE. 0.0) GO TO 205
2543       GO TO 210
2544       205  NDIM = X-1
2545       GO TO 211
2546       210  CONTINUE
2547       211  CONTINUE
2548       EPS = VALY(1)/1000.
2549       CALL ACF1 ( 1.0, ARG, VALY, Y, NDIM, EPS, IER, J )
2550       C(1, Z, T+1)=Y
2551       IF(Y.LE.0.0.OR.Y.GT.C(Z,Z,T+1)) C(1,Z,T+1)=C(Z,Z,T+1)*C(Z,Z,T+1)
2552       /C(3,Z,T+1)
2553       GO TO 225
2554       220  C(1, Z, T+1) = 0.0
2555       225  CONTINUE
2556       228  CONTINUE
2557       IF(LTYPE.EQ.1.AND.MSL.EQ.NX) GO TO 251
2558       DO 250 Z=N21,N2M1
2559       NDIM=3
2560       NSTOP=NDIM
2561       DO 235 I=1,NSTOP
2562       IBACK = NX - I
2563       ARG(I) = FLOAT(I+1)
2564       VALY(I) = C( IBACK, Z, T+1)
2565       IF( VALY(I) . LE. 0.00) GO TO 230
2566       GO TO 235
2567 230 NDIM = 1
2568 235 CONTINUE
2570 236 CONTINUE
2571 EPS = VALY(1)/1000.
2572 CALL ACFI ( 1., ARG, VALY, Y, NDIM, EPS, IER, J )
2573 C(NX, Z, T+1) = Y
2574 IF(Y.LE.0.0.OR.Y.GT.C(NX-1,Z,T+1))C(NX,Z,T+1)=C(NXM1,Z,T+1)*
2575 $ C(NXM1,Z,T+1)/C(NXM2,Z,T+1)
2576 245 C(NX, Z, T+1) = 0.,0
2577 250 CONTINUE
2578 251 CONTINUE
2579 IF(NZ.EQ.1.OR.OPTIGN.EQ.6) RETURN

C---------------------------------------------------------------------
C EXTRAPOLATE THE PROFILE IN THE Z DIRECTION
C---------------------------------------------------------------------
2580 DO 275 X = 1, NX
2581 IF(C(X, Z, T+1) . LE. 0.00) GO TO 270
2582 NDIM=3
2583 NSTOP=NDIM
2584 DO 260 Z = 1, NSTOP
2585 ARG(Z) = FLOAT(Z+1)
2586 VALY(Z) = C(X, Z+1, T+1)
2587 IF( VALY(Z) . LE. 0.00) GO TO 255
2588 GO TO 260
2589 255 NDIM = Z
2590 GO TO 261
2591 260 CONTINUE
2592 261 CONTINUE
2593     EPS = VALY(1)/1000.
2594     CALL ACFI ( 1.0, ARG, VALY, Y, NDIM, EPS, IER, J )
2595     C ( X, 1, T+1 ) = Y
2596     IF( (Y .LE. 0.0) OR. (Y .GT. C(X,2,T+1)) ) C(X,1,T+1) = C(X,2,T+1) * C(X,2,T+1)
2597          / C(X,3,T+1)
2598     IF( (C(X,1,T+1) .LT. 0.0) ) C(X,1,T+1) = 0.0
2599     GO TO 275
2600 270 C(X, 1, T+1) = 0.0
2601 275 CONTINUE
2602     DO 295 I = 1, NX
2603     IF ( (C(X, NZ-1, T+1) . LE. 0.00) ) GO TO 292
2604     NDIM = 3
2605     NSTOP = NDIM
2606     DO 285 I = 1, NSTOP
2607     ARG(I) = FLOAT(I+1)
2608     IBACK = NZ - I
2609     VALY(I) = C(X, IBACK, T+1)
2610     IF ( VALY(I) . LE. 0.00 ) GO TO 280
2611     GO TO 285
2612 280 NDIM = I
2613     GO TO 286
2614 285 CONTINUE
2615 286 CONTINUE
2616     EPS = VALY(1)/1000.
2617     CALL ACFI ( 1.0, ARG, VALY, Y, NDIM, EPS, IER, J )
2618     C(X, NZ, T+1) = Y
2619     IF ( (Y .LE. 0.0) OR. (Y .GT. C(X,NZ-1,T+1)) ) C(X,NZ,T+1) = C(X,NZM1,T+1) *
2620          C(X,NZM1,T+1) / C(X,NZM2,T+1)
2621     GO TO 295
2622 292 C(X, NZ, T+1) = 0.0
2623 295 CONTINUE
2624     RETURN
2625     END
SUBROUTINE ACFI ( X, ARG, VALY, Y, NDIM, EPS, IER, J )

********** SUBROUTINE ACFI *****
**********

PURPOSE
TO INTERPOLATE FUNCTION VALUE Y FOR A GIVEN ARGUMENT VALUE X USING A GIVEN TABLE (ARG,VAL) OF ARGUMENT AND FUNCTION VALUES.

DESCRIPTION OF PARAMETERS
X - THE ARGUMENT VALUE SPECIFIED BY INPUT.
ARG - THE INPUT VECTOR (DIMENSION NDIM) OF ARGUMENT VALUES OF THE TABLE (POSSIBLY DESTROYED).
VAL - THE INPUT VECTOR (DIMENSION NDIM) OF FUNCTION VALUES OF THE TABLE (DESTROYED).
Y - THE RESULTING INTERPOLATED FUNCTION VALUE.
NDIM - AN INPUT VALUE WHICH SPECIFIES THE NUMBER OF POINTS IN TABLE (ARG,VAL).
EPS - AN INPUT CONSTANT WHICH IS USED AS UPPER BOUND FOR THE ABSOLUTE ERROR.
IER - A RESULTING ERROR PARAMETER.
REMARKS

1. TABLE (ARG, VAL) SHOULD REPRESENT A SINGLE-VALUED FUNCTION AND SHOULD BE STORED IN SUCH A WAY, THAT THE DISTANCES ABS(ARG(I) - X) INCREASE WITH INCREASING SUBSCRIPT I. TO GENERATE THIS ORDER IN TABLE (ARG, VAL), SUBROUTINES ATSG, ATSM OR ATSE COULD BE USED IN A PREVIOUS STAGE.

2. NO ACTION BESIDES ERROR MESSAGE IN CASE NDIM LESS THAN 1.

3. INTERPOLATION IS TERMINATED EITHER IF THE DIFFERENCE BETWEEN TWO SUCCESSIVE INTERPOLATED VALUES IS ABSOLUTELY LESS THAN TOLERANCE EPS, OR IF THE ABSOLUTE VALUE OF THIS DIFFERENCE STOPS DIMINISHING, OR AFTER (NDIM - 1) STEPS (THE NUMBER OF POSSIBLE STEPS IS DIMINISHED IF AT ANY STAGE INFINITY ELEMENT APPEARS IN THE DOWNWARD DIAGONAL OF INVERTED-DIFFERENCES-SCHME AND IF IT IS IMPOSSIBLE TO ELIMINATE THIS INFINITY ELEMENT BY INTERCHANGING OF TABLE POINTS). FURTHER IT IS TERMINATED IF THE PROCEDURE DISCOVERS TWO ARGUMENT VALUES IN VECTOR ARG WHICH ARE IDENTICAL. DEPENDENT ON THESE FOUR CASES, ERROR PARAMETER IER IS CODED IN THE FOLLOWING FORM

IER=0 - IT WAS POSSIBLE TO REACH THE REQUIRED ACCURACY (NO ERROR).
IER=1 - IT WAS IMPOSSIBLE TO REACH THE REQUIRED ACCURACY BECAUSE OF ROUNING ERRORS.
IER=2 - IT WAS IMPOSSIBLE TO CHECK ACCURACY BECAUSE NDIM IS LESS THAN 2, OR THE REQUIRED ACCURACY COULD NOT BE REACHED BY MEANS OF THE GIVEN TABLE. NDIM SHOULD BE INCREASED.
IER=3 - THE PROCEDURE DISCOVERED TWO ARGUMENT VALUES IN VECTOR ARG WHICH ARE IDENTICAL.
METHOD

INTERPOLATION IS DONE BY CONTINUED FRACTIONS AND INVERTED-
DIFFERENCES-SCHEME. ON RETURN Y CONTAINS AN INTERPOLATED
FUNCTION VALUE AT POINT X, WHICH IS IN THE SENSE OF REMARK
(3) OPTIMAL WITH RESPECT TO GIVEN TABLE. FOR REFERENCE, SEE
F.B. HILDEBRAND, INTRODUCTION TO NUMERICAL ANALYSIS,
MCGRAW-HILL, NEW YORK/TORONTO/LONDON, 1956, PP. 395-406.

---------------------------------------------------------------------

DIMENSION ARG(10), VALY(10), VAL(10)

DO 100 M = 1, NDIM

VAL(M) = VALY(M)

100 CONTINUE

IER=2

IF(NDIM)20, 20, 1

Y = VAL(1)

DELT2 = 0.

IF(NDIM-1)20, 20, 2

C

PREPARATIONS FOR INTERPOLATION LOOP

P2 = 1.

P3 = Y

Q2 = 0.

Q3 = 1.

C
C COMPUTATION OF VAL(I) IN CASE VAL(I) = VAL(J) AND J LESS THAN I-1
2662  8 VAL(I)=1.E75
2663  GOTO 10
C
C COMPUTATION OF VAL(I) IN CASE VAL(I) NOT EQUAL TO VAL(J)
2664  9 VAL(I)=(ARG(I)-ARG(J))/H
2665  10 CONTINUE
C INVERTED DIFFERENCES ARE COMPUTED
C
C COMPUTATION OF NEW Y
2666  P3=VAL(I)*P2+(X-ARG(I-1))*P1
2667  Q3=VAL(I)*Q2+(X-ARG(I-1))*Q1
2668  IF(Q3)11,12,11
2669  11 Y=P3/Q3
2670  GOTO 13
2671  12 Y=1.E75
2672  13 DELT2=ABS(Z-Y)
2673  IF(DELT2-EPS)19,19,14
2674  14 IF(I-8)16,15,15
2675  15 IF(DELT2-DELT1)16,18,18
2676  16 CONTINUE
C END OF INTERPOLATION LCOP
C
C
2677  RETURN
C
C THERE ARE TWO IDENTICAL ARGUMENT VALUES IN VECTOR ARG
2678  17 IER=3
RETURN
C TEST VALUE DELT2 STARTS OSCILLATING
18 Y=Z
IER=1
RETURN
C THERE IS SATISFACTORY ACCURACY WITHIN NDIM-1 STEPS
19 IER=0
20 RETURN
END
**SUBROUTINE PRINT1**

*-------------------------*
** SUBROUTINE PRINT1 ****
*-------------------------*

This subroutine calculates the conversion values for many of the input parameters and prints out a summary of the input data and stability criteria.

**LOGICAL SECOND**

**INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,COUNT,OPTION**

**REAL KD,KDMAX,KDMIN,KDAYS,LOWER**

**COMMON/ARRAYS/ C(51,21,4), VX(51,21), VZ(51,21), EX(51,21), EZ(51,21),
1 W(51), CONCIN(51,21), INSECT(51), KDAYS(51), K(D51),
2 DEMAND(51,21), FMT1(40), FMT2(40), FMT3(20)**

**COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,WMAX,WMIN,**
1 KMAX,KDMIN,ITER,ITER1,ITER2,DELT1,DELT2,TNEXT,
2 TPRINT,XPRINT,STABLX,STABLZ,STABLT,R1,R2,VZMAX,SET,
3 VZMIN, EMAX, EZMIN, DEMAX, DEMIN, NPPAGE, DELTAX, DELTAX,
4 DELTAT, TSTART, TSF, TSTART, COUNT, IPAGE, UPPER, LOWER,
5 MSU, MSL, TIME, NX1, NX2, NZ1, NZ2, ZSF, LTYPE, OPTION**

**COMMON/OXYGEN/OXOUT(51,21), R(51,21), RDAYS(51,21), OXSAT(51), AK(51),**
1 KDAYS(51), FMT4(40), FMT5(40), FMT6(20),
2 SECOND, AKMAX, RAMAX, OUTMAX, AKMIN, RMIN, OUTMIN

**UNIT1 = 86400./5280.**

**UNIT2 = 86400./{5280.*5280.}**

**DELDAY = DELTAT/86400.**
SECS1 = DELT1 * 86400.
SECS2 = DELT2 * 86400.
DELMX = DELTAX / 5280.
DELMZ = DELTAZ / 5280.
TOTALX = (NX - 1) * DELTAX
TOTMX = TOTALX / 5280.
TOTALZ = (NZ - 1) * DELTAZ
TOTMZ = TOTALZ / 5280.
VX1 = VXMAX * UNITS1
VX2 = VXMIN * UNITS1
VZ1 = VZMAX * UNITS1
IF (VZMIN.EQ.1000000.) VZMIN = 0.
VZ2 = VZMIN * UNITS1
EX1 = MAXMAX * UNITS2
EX2 = EXMIN * UNITS2
EZ1 = EZMAX * UNITS2
IF (EZXMIN.EQ.1000000.) EZXMIN = 0.
EZ2 = EZXMIN * UNITS2
XKD1 = KDMAX * 86400.
XKD2 = KDIN * 86400.
XAK1 = AKMAX * 86400.
XAK2 = AKNIN * 86400.
XR1 = RMAX * 86400.
XR2 = RMIN * 86400.
SXM = STBLX / 5280.
SZM = STBLZ / 5280.
STD = STBLT / 86400.
PRINT 500
PRINT 505
2725 PRINT 507
2726 PRINT 510
2727 PRINT 507
2728 IF(NZ.EQ.1) PRINT 515
2729 IF(NZ.NE.1) PRINT 520
2730 PRINT 507
2731 IF(LTYPE.EQ.1) GO TO 7
2732 IF(LTYPE.EQ.2) GO TO 5
2733 PRINT 530
2734 GO TO 10
2735 5 PRINT 525
2736 GO TO 10
2737 7 PRINT 523
2738 10 CONTINUE
2739 PRINT 507
2740 PRINT 535
2741 PRINT 507
2742 PRINT 505
2743 PRINT 540, DELTAX, DELMX
2744 PRINT 545, DELTAZ, DELMZ
2745 PRINT 550, SECS1, DELT1
2746 PRINT 551, ITER1
2747 PRINT 552, SECS2, DELT2
2748 PRINT 553, ITER2
2749 PRINT 555, NX
2750 PRINT 560, NZ
2751 PRINT 565, VXMAX, VX1
2752 PRINT 570, VXMIN, VX2
2753 PRINT 572, VZMAX, VZ1
2754 PRINT 573, VZMIN, VZ2
PRINT 575, EXMAX, EX1
PRINT 580, EXMIN, EX2
PRINT 585, EZMAX, EZ1
PRINT 590, EZMIN, EZ2
PRINT 600, WMAX
PRINT 605, WMIN
PRINT 610, XKD1, KMAX
PRINT 615, XKD2, KDMIN
PRINT 620, DEMAX
PRINT 630, NC

C ---
C XUPEAK AND XLPEAK ARE CALCULATED ASSUMING DISTANCES ARE
C DECREASING IN THE DOWNSTREAM DIRECTION
C ---

XUPEAK = X START - (MSU-1)*DELMX
XLPEAK = X START - (MSL-1)*DELMX
PRINT 635, XUPEAK, MSU
PRINT 637, XLPEAK, MSL
PRINT 500
IF(.NOT.SECOND) GO TO 20
PRINT 800
PRINT 810, XR1, RMAX
PRINT 815, XR2, RMIN
PRINT 820, XAK1, AKMAX
PRINT 825, XAK2, AKMIN
PRINT 830, OUMAX
20 CONTINUE
PRINT 505
PRINT 502
2780 PRINT 640
2781 PRINT 645, STBLX, SXM
2782 PRINT 650, STBLZ, SZM
2783 PRINT 655, STBLT, STD
2784 PRINT 660, R1
2785 PRINT 665, R2
2786 PRINT 505
2787 IF (NC.EQ.0) GO TO 100
2788 PRINT 500
2789 PRINT 700
2790 IF (NZ.EQ.1) GO TO 50
2791 DO 25 I=1, NC

C--------------------------------------------------
C WHERE IS CALCULATED ASSUMING DISTANCES ARE DECREASING IN
C THE DOWNSTREAM DIRECTION
C--------------------------------------------------

2792 WHERE=XSTART-(INSECT(I)-1)*DELMX
2793 PRINT 710, INSECT(I), WHERE, DELTAZ
2794 DO 25 Z=1, NZ
2795 PRINT 715, INSECT(I), Z, CONCIN(I, Z)
2796 25 CONTINUE
2797 GO TO 100
2798 50 DO 75 I=1, NC
2799 WHERE=XSTART-(INSECT(I)-1)*DELMX
2800 PRINT 705, INSECT(I), WHERE, CONCIN(I, 1)
2801 PRINT 502
2802 75 CONTINUE
2803 100 CONTINUE
2804 PRINT 500
575 FORMAT(T35, 'MAXIMUM HORIZONTAL DISPERSION = ', F6.0, ' FEET SQUARED / SECOND (', F5.2, ' MILES SQUARED / DAY )', /
580 FORMAT(T35, 'MINIMUM HORIZONTAL DISPERSION = ', F6.0, ' FEET SQUARED / SECOND (', F5.2, ' MILES SQUARED / DAY )', /
585 FORMAT(T37, 'MAXIMUM VERTICAL DISPERSION = ', F9.3, ' FEET SQUARED / SECOND (', E10.3, ' MILES SQUARED / DAY )', /
590 FORMAT(T37, 'MINIMUM VERTICAL DISPERSION = ', F9.3, ' FEET SQUARED / SECOND (', E10.3, ' MILES SQUARED / DAY )', /
600 FORMAT(T51, 'MAXIMUM WIDTH = ', F5.0, ' FEET', /
605 FORMAT(T51, 'MINIMUM WIDTH = ', F5.0, ' FEET', /
610 FORMAT(T46, 'MAXIMUM DECAY RATE = ', F5.3, ' PER DAY (', E10.3, ' PER SECOND )', /
615 FORMAT(T46, 'MINIMUM DECAY RATE = ', F5.3, ' PER DAY (', E10.3, ' PER SECOND )', /
620 FORMAT(T42, 'OTHER DEMANDS, MAXIMUM = ', F5.2, '/
630 FORMAT(T26, 'NUMBER OF INITIAL CONCENTRATION VALUES = ', I2, ')
635 FORMAT(T16, 'INITIAL LOCATION OF UPPER MOST PEAK CONCENTRATION = ', $ F5.2, ' MILES (SECTION NUMBER ', I2, ')', /
637 FORMAT(T16, 'INITIAL LOCATION OF LOWER MOST PEAK CONCENTRATION = ', $ F5.2, ' MILES (SECTION NUMBER ', I2, ')', /
640 FORMAT(T56, 'STABILITY CRITERIA', /
645 FORMAT(T35, 'MAXIMUM ALLOWABLE X INCREMENT = ', F7.0, ' FEET (', $ F8.5, ' MILES )', /
650 FORMAT(T35, 'MAXIMUM ALLOWABLE Z INCREMENT = ', F7.0, ' FEET (', $ F8.5, ' MILES )', /
655 FORMAT(T32, 'MAXIMUM ALLOWABLE TIME INCREMENT = ', F7.0, ' SECONDS (', E10.3, ' DAYS )', /
660 FORMAT(T38, 'ACTUAL DELTAT/(DELTAX**2.) = ', E10.3, ')
665 FORMAT(T38, 'ACTUAL DELTAT/(DELTAZ**2.) = ', E10.3, '/
700 FORMAT(T25,'LOCATIONS OF INITIAL CONCENTRATIONS',//)
705 FORMAT('A WASTE SOURCE IS LOCATED AT STATION ',I2,' (MILE ',
    $F6.3','). THE CONCENTRATION IS ',F10.2,' PPM'.)
710 FORMAT(1X,'AN INITIAL CONCENTRATION IS FOUND AT STATION ',I2,
    ' (MILE ',F6.3,').',//,' THE CONCENTRATIONS AT ',F7.1,
    ' FOOT INTERVALS WITH DEPTH ARE')
715 FORMAT(T15,'C(',I2,':I2,1) = ',F9.2,' PPM')
800 FORMAT(T45,'*** SECONDARY INPUT PARAMETERS ***',//)
810 FORMAT(T41,'MAXIMUM REAERATION RATE = ',
    $F5.3,' PER DAY ( ',E10.3,' PER SECOND )',//)
815 FORMAT(T41,'MINIMUM REAERATION RATE = ',
    $F5.3,' PER DAY ( ',E10.3,' PER SECOND )',//)
820 FORMAT(T41,'MAXIMUM ANAEROBIC DECAY = ',
    $F5.3,' PER DAY ( ',E10.3,' PER SECOND )',//)
825 FORMAT(T41,'MINIMUM ANAEROBIC DECAY = ',
    $F5.3,' PER DAY ( ',E10.3,' PER SECOND )',//)
830 FORMAT(T35,'OTHER OXYGEN DEMANDS, MAXIMUM = ',F6.3,////)
2856 RETURN
2857 END
SUBROUTINE PRINT2

***************
 **** SUBROUTINE PRINT2 *****
***************

THIS SUBROUTINE PRINTS OUT THE TIME AND THE CONCENTRATION
PROFILE ACCORDING TO A FORMAT PREVIOUSLY READ INTO THE PROGRAM.
THIS SUBROUTINE CAN BE CHANGED ACCORDING TO THE NEEDS OF THE USER.
***WARNING--DO NOT CHANGE ANY VARIABLE OCCURRING IN A COMMON BLOCK.

LOGICAL SECOND
INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,CCOUNT,OPTION
REAL KD,KDMAX,KDMIN,KDAYS,LOWER
DIMENSION NZREV(20)
COMMON/ARRAYS/ C(51,21,4), VX(51,21), VZ(51,21), EX(51,21), EZ(51,21),
1 W(51), CONCIN(51,21), INSECT(51), KDAYS(51), KD(51),
2 DEMAND(51,21), FMT1(40), FMT2(40), FMT3(20)
COMMON/NAMES/X,Z,T,NX,NZ,NS,EXMAX,EXMIN,VXMAX,VXMIN,WMAX,WMIN,
1 KD,KDMIN,KDMAX,ITER,ITER1,ITER2,DELT1,DELT2,TNEXT,
2 TPRINT,XPRINT,STABLX,STABLZ,STABLY,R1,R2,VZMAX,SET,
3 VZMIN,EZMAX,EZMIN,DEMAX,DEMIN,NPPAGE,DELTAX,DELTAZ,
4 DELTAT,XSTART,XSF,TSTART,COUNT,IPAGE,UPPER,LOWER,
5 MSU,MSL,TIME,NXM1,NXM2,NZM1,NZM2,LSF,LYTYPE,OPTION
COMMON/OXYGEN/OXOUT(51,21),R(51,21), RDAYS(51,21), OXSAT(51), AK(51),
1 AKDAYS(51), FMT4(40), FMT5(40), FMT6(20),
2 SECOND, AKMAX,RMAX,OUTMAX,AKMIN,RMIN,OUTMIN
DAYS = TIME/86400.
HOURS=DAYS*24.
XFIRST=XSTART-(NX-1)*DELTAX/5280.
DO 100 IZ = 1, NZ
   NZREV(IZ)=IFIX(DELTAZ*(IZ-2))
100 CONTINUE
IF(T.EQ.3) GO TO 300
WRITE(6,FMT1) HOURS,DAYS
WRITE(6,FMT2) (NZREV(I), I=2,NZM1)
DO 200 X=1,NX,XPRINT
   XMILE = XFIRST + ((X-1)*DELTAX*XPRINT)/5280.
   M=NX+1-X
   WRITE(6,FMT3) XMILE, (C(M,Z,T),Z=1,NZ)
200 CONTINUE
GO TO 600
300 CONTINUE
WRITE(6,FMT4) HCURS, DAYS
WRITE(6,FMT5) (NZREV(I), I=2,NZM1)
DO 500 X=1,NX,XPRINT
   XMILE = XFIRST + ((X-1)*DELTAX*XPRINT)/5280.
   M=NX+1-X
   WRITE(6,FMT6) XMILE, (C(M,Z,T),Z=1,NZ)
500 CONTINUE
600 CONTINUE
RETURN
END
C SUBROUTINE ERROR(*)
C
C
C *************************************************************
C  **** SUBROUTINE ERROR *****
C  *************************************************************
C
C THIS SUBROUTINE CORRECTS SOME OF THE MOST COMMON ERRORS
C IN THE INPUT DATA AND PRINTS OUT APPROPRIATE ERROR MESSAGES.
C
C
2892 Logical SECOND
2893 INTEGER X,Z,T,TNEXT,TPRINT,XPRINT,COUNT,OPTION
2894 REAL KD,KMAX,KMIN,KDAYS,LOWER
2895 COMMON/ARRAYS/ C(51,21,4),VX(51,21),VZ(51,21),EX(51,21),EZ(51,21),
2896  W(51),CONCIN(51,21),INSECT(51),KDAYS(51),KD(51),
2897  DEMAND(51,21),FMT1(40),FMT2(40),FMT3(20)
2898 COMMON/NAMES/X,Z,T,NX,NZ,NC,NS,EXMAX,EXMIN,VXMAX,VXMIN,WMAX,WMIN,
2899  KD,KMAX,KMIN,ITER,ITER1,ITER2,DEL1,DEL2,TNEXT,
2900  TPRINT,XPRINT,STABLY,STABLY,STABLY,RI2,R2,VZMAX,SET,
2901  VZMIN,EZMAX,EZMIN,DEMAX,DEMIN,NPPAGE,DELTA,DELTA,
2902  DELTAT,XSTART,XSF,TSTART,COUNT,IPAGE,UPPER,LOWER,
2903  MSU,MSL,TIME,NX1,NXM2,NZ1,NZM2,ZSF,LTYPE,OPTION
2904 COMMON/OXYGEN/OKOUT(51,21),RI(51,21),RDAYS(51,21),OXSAT(51),AK(51),
2905  AKDAYS(51),FMT4(40),FMT5(40),FMT6(20),
2906  SECOND,AKMAX,RAKMAX,AKMIN,AKMIN,OAKMIN
2907 IF(OPTION.LT.8) GO TO 200
2908 PRINT 181
2909 181 FORMAT(///,'***PROGRAM TERMINATED BECAUSE OPTION IS GREATER TH
2910  $AN 7****','/1H1)
2911 RETURN
2912 300 IF(LTYPE.LT.3) GO TO 300
2904 PRINT 201
2905 201 FORMAT(///1X,'***** WARNING -- LTYPE IS GREATER THAN OR EQUAL TO 3',/4X,'IF LTYPE EXISTS 3, THE USER MUST SUPPLY A SUBROUTINE FOR THE LOADING CONDITIONS','/4X,'IF LTYPE IS GREATER THAN 3, THE PROGRAM IS TERMINATED *****',/1H1)
2906 IF(LTYPE.GT.3) RETURN1
2907 RETURN
2908 300 CONTINUE
2909 IF((SECOND).AND.(OPTION.EQ.3.OR.OPTION.EQ.4.OR.OPTION.EQ.7))
2910 $ GO TO 350
2911 GO TO 400
2912 360 FORMAT(///1X,'***** PROGRAM TERMINATED BECAUSE WHEN SECOND EQUAL IS 2 THEN OPTION CANNOT EQUAL 3 OR 4 OR 7 *****')
2913 RETURN1
2914 400 CONTINUE
2915 IF(ITER.NE.0) GO TO 820

C---------------------------------------------------------------
C TERMINATE PROGRAM WITH PRINTED MESSAGE IF INPUT DATA VIOLATES STABILITY CRITERIA
C---------------------------------------------------------------

2916 PRINT 815
2917 815 FORMAT(///1X,'PROGRAM TERMINATED BECAUSE STABILITY CONDITIONS WERE VIOLATED *****',/1H1)
2918 RETURN1
2919 820 CONTINUE
2920 RETURN
2921 END
Input Data for OXTRANS-I

Each line represents a new card unless single spaced.

1.
.TURE.
33, 7
0.25, 10.,
0.006944444, 2
0.0, 0
1, 1
17., 4.0, 1, 1, 1
1, 6
33*0.0, 33*0.366667, 33*0.440, 33*0.433333, 33*0.273333,
33*0.233333, 33*0.0
231*450.
525., 568., 611., 654., 697.,
756., 814., 872., 931., 863., 796., 728., 661., 657., 654.,
650., 647., 633., 619., 605.,
591., 586., 582.
33*0.25
231*0.0
2
7*13.
231*0.0
66\times 0.00333333, 33\times 0.004, 33\times 0.00394, 33\times 0.002485, 66\times 0.00212

(///, 12X, 100(''), /, 12X, '\*', /, 12X, '\*', 2X,
\text{CONCENTRATIONS AT TIME = '},

F7.2, ' HOURS ( ', F6.4, ' DAYS )', /, 12X, '\*', /, 12X, 100('')

(12X, '\*', /, 12X, '\*', 42X, 'DEPTH', /, 12X, '\*', /, 12X, '\*', 3X, 'IMAGE', 4X,
5(12, 'FEET'), 'IMAGE', /, 1X, 111(''), /, 12X, '\')

(1X, 'MILE', F5.2, 1X, '\*', 2X, 7(F6.1, 4X))

(///, 12X, 100(''), /, 12X, '\*', /, 12X, '\*', 2X,
\text{OXYGEN AT TIME = '},

F7.2, 'HOURS ( ', F6.4, ' DAYS )', /, 12X, '\*', /, 12X, 100('')

(12X, '\*', /, 12X, '\*', 42X, 'DEPTH', /, 12X, '\*', /, 12X, '\*', 3X, 'IMAGE', 4X,
5(12, 'FEET'), 'IMAGE', /, 1X, 111(''), /, 12X, '\')

(1X, 'MILE', F5.2, 1X, '\*', 2X, 7(F6.1, 4X))

33\times 0.08
33\times 0.1, 165\times 0.0

231\times 0.
COMPUTER PROGRAM FOR
STABLE-I

Object Code = 928 bytes
Array Area = 0 bytes
Total = 928 bytes
STRAIGHT-1

THIS PROGRAM COMPUTES THE STABILITY CRITERIA FOR TIME
INCREMENT AND DISTANCE INCREMENTS FOR A LARGE NUMBER OF
COMBINATIONS OF DISPERSION COEFFICIENTS AND VELOCITIES.
THESE CRITERIA APPLY TO ONE DIMENSIONAL FINITE DIFFERENCE MODELS FOR
MASS TRANSPORT IN AN ESTUARY.

** Output Variables **

- **EF** = Dispersion Coefficient (Miles**2/**Day)
- **UF** = Dispersion Coefficient (Feet**2/**Second)
- **VF** = Velocity (Miles/**Day)
- **DF** = Allowable Distance Increment (Miles)
- **DT** = Allowable Time Increment (Days)
- **DELT** = Allowable Distance Increment (Seconds)
PRINT 10
10 FORMAT(I1H, T5, 'E', T20, 'EF', T35, 'U', T50, 'UF', T65, 'DELX',
$ T80, *DELFX*, T95, 'DELT', T110, 'DELTS', //)

3204   DELX=.25
3205   E=.333E-05
3206   DO 200 J=1,7
3207   UF=0.5E-05
3208   DO 100 I=1,7
3209   U=UF*86400.*/5280.
3210   IF(U.NE.0) DELX=2.*E/U
3211   DELT=(DELX**2.)/(2.*E)
3212   DELXF=5280.*DELX
3213   DELTS=86400.*DELT
3214   EF=(5280.*5280.*E)/86400.
3215   PRINT 75, E,EF,U,UF,DELX,DELXF,DELT,DELTS
3216   75 FORMAT(I1X, 8E15.5)
3217   UF=UF*2.
3218   100 CONTINUE
3219   E=2.*E
3220   200 CONTINUE
3221   STOP
3222   END
COMPUTER PROGRAM FOR

STABLE-II

Object Code = 4016 bytes
Array Area = ___0 bytes
Total = 4016 bytes
*--------------------------------------*
*                                     *
* STABLE-II                           *
*                                     *
* STABILITY CRITERIA FOR ONE- AND TWO-DIMENSIONAL *
* EXPLICIT FINITE DIFFERENCE SCHEMES    *
*                                     *
*--------------------------------------*

THIS COMPUTER PROGRAM COMPUTES THE MAXIMUM ALLOWABLE DISTANCE
AND TIME INCREMENTS FOR A SET OF ESTUARY DATA WHICH IS TO BE USED
IN A ONE DIMENSIONAL OR TWO DIMENSIONAL EXPLICIT FINITE DIFFERENCE
MODEL FOR MASS TRANSPORT.

LOGIC AND PROGRAMMING--JONATHAN YOUNG, TEXAS A&M UNIVERSITY

*** INPUT VARIABLES ***

NSETS = NUMBER OF DATA SETS TO BE PROCESSED
DELMX = DISTANCE INCREMENT IN THE X DIRECTION THAT THE USER
       WISHES TO USE IN HIS MODEL (MILES)
DELTAZ = DISTANCE INCREMENT IN THE Z DIRECTION THAT THE USER
       WISHES TO USE IN HIS MODEL (FEET)
NZ = NUMBER OF POINTS IN THE Z DIRECTION (IF NZ = 1, THE
    MODEL IS ONE DIMENSIONAL. IF NZ IS GREATER THAN 1, THE
    MODEL IS CONSIDERED TO BE TWO DIMENSIONAL.)
C EXMIN = MINIMUM DISPERSION COEFFICIENT IN THE X DIRECTION (FEET**2/SECOND)
C EXMAX = MAXIMUM DISPERSION COEFFICIENT IN THE X DIRECTION (FEET**2/SECOND)
C EZMIN = MINIMUM DISPERSION COEFFICIENT IN THE Z DIRECTION (FEET**2/SECOND)
C EZMAX = MAXIMUM DISPERSION COEFFICIENT IN THE Z DIRECTION (FEET**2/SECOND)
C VXMAX = MAXIMUM VELOCITY IN THE X DIRECTION (FEET/SECOND)
C VZMAX = MAXIMUM VELOCITY IN THE Z DIRECTION (FEET/SECOND)
C KDAYS = MAXIMUM DECAY RATE (/DAY)
REAL KMAX
REAL KDAYS
READ, NSETS
PRINT, NSETS
DO 1000 II=1,NSETS
READ, DELMX, DELTAZ, NZ, EXMIN, EXMAX, EZMIN, EZMAX, VXMAX, VZMAX, KDAYS
PRINT, DELMX, DELTAZ, NZ, EXMIN, EXMAX, EZMIN, EZMAX, VXMAX, VZMAX, KDAYS
DELTAX=DELMX*5280.
XKDO1=KDAYS
KMAX=KDAYS/86400.
UNITS1 = 86400./5280.
UNITS2 = 86400./((5280.*5280.))
DELMAZ=DELTAZ/5280.
VX1=VXMAX*UNITS1
VZ1=VZMAX*UNITS1
EX1=EXMAX*UNITS2
EX2=EXMIN*UNITS2
EZ1=EZMAX*UNITS2
EZ2=EZMIN*UNITS2
STABLX = DELTAX
STABLZ = DELTAZ
IF(NZ.NE.1) GO TO 500
IF(EXMIN.GT.0.0.AND.VXMAX.GT.0.0) STABLX=2.*EXMIN/VXMAX
TERM=2.*EXMAX+DELTAX**2.*KMAX
IF(TERM.GT.0.0) STABLT=(DELTAX**2.)/TERM
GO TO 800
500 CONTINUE
IF(EXMIN.GT.0.0.AND.VXMAX.GT.0.0) STABLX=2.*EXMIN/VXMAX
IF(EZMIN.GT.0.0.AND.VZMAX.GT.0.0) STABLZ=2.*EZMIN/VZMAX
TERM = 2.0*(EXMAX*DELTAZ**2.+EZMAX*DELTAX**2.)*KDMAX*(DELTAX**2.)*
(DELTAZ**2)
IF (TERM.NE.0.0) STABL = (DELTAZ**2.)*(DELTAX**2.)/TERM
800 CONTINUE
SXM = STABLX/5280.
SZM = STABLZ/5280.
STD = STABL/T86400.
PRINT 100
PRINT 540, DELTAX, DELMX
PRINT 545, DELTAX, DELMZ
PRINT 560, NZ
PRINT 565, VMAX, VX1
PRINT 572, VZMAX, VZ1
PRINT 575, EXMAX, EX1
PRINT 580, EXMIN, EX2
PRINT 585, EZMAX, EZ1
PRINT 590, EZMIN, EZ2
PRINT 610, KDMI, KDMAX
PRINT 505
PRINT 502
PRINT 505
PRINT 640
PRINT 645, STABLX, SXM
PRINT 650, STABLZ, SZM
PRINT 655, STABL, STD
PRINT 505
100 FORMAT(1H1)
502 FORMAT(/)
505 FORMAT(T45, 38(**))
540 FORMAT(1H1, T53, 'X INCREMENT =', F7.0, ' FEET', F8.5, ' MILES $')/
545 FORMAT(T53, 'Z INCREMENT = ', F7.0, ' FEET (', E10.3, ' MILES )')
560 FORMAT(T39, 'NUMBER OF VERTICAL POINTS = ', I4, ')
565 FORMAT(T37, 'MAXIMUM HORIZONTAL VELOCITY = ', F5.2, ' FEET/SECOND $	imes$(', F6.2, ' MILES/DAY )')
572 FORMAT(T39, 'MAXIMUM VERTICAL VELOCITY = ', E10.3, ' FEET/SECOND ('$	imes$ E10.3, ' MILES/DAY )')
575 FORMAT(T35, 'MAXIMUM HORIZONTAL DISPERSION = ', F6.0, ' FEET SQUARED / SECOND ('$	imes$ F5.2, ' MILES SQUARED / DAY )')
580 FORMAT(T35, 'MINIMUM HORIZONTAL DISPERSION = ', F6.0, ' FEET SQUARED / SECOND (', F5.2, ' MILES SQUARED / DAY )')
585 FORMAT(T37, 'MAXIMUM VERTICAL DISPERSION = ', F6.3, ' FEET SQUARED / SECOND (', E10.3, ' MILES SQUARED / DAY )')
590 FORMAT(T37, 'MINIMUM VERTICAL DISPERSION = ', F6.0, ' FEET SQUARED / SECOND (', E10.3, ' MILES SQUARED / DAY )')
610 FORMAT(T46, 'MAXIMUM DECAY RATE = ', F5.3, ' PER DAY ('$	imes$ E10.3, '$\times$ PER SECOND )')
640 FORMAT(T56, 'STABILITY CRITERIA')
645 FORMAT(T35, 'MAXIMUM ALLOWABLE X INCREMENT = ', F7.0, ' FEET ('$	imes$ F8.5, ' MILES )')
650 FORMAT(T35, 'MAXIMUM ALLOWABLE Z INCREMENT = ', F7.0, ' FEET ('$	imes$ F8.5, ' MILES )')
655 FORMAT(T32, 'MAXIMUM ALLOWABLE TIME INCREMENT = ', F7.0, ' SECONDS (', E10.3, ' DAYS )')
1000 CONTINUE
STOP
END
Input Data for STABLE-II

Each line represents a new card unless single spaced.

1

0.2, 5., 11, 500., 500., 0.001, 0.005, 0.441, 0., 0.384
COMPUTER PROGRAM FOR

EXACT-I

Object Code = 5664 bytes
Array Area = 400 bytes
Total = 6064 bytes
THIS COMPUTER PROGRAM COMPUTES THE PROFILES FOR VARIOUS CLOSED-FORM SOLUTIONS TO THE ESTUARY MASS TRANSPORT EQUATION. THESE SOLUTIONS ARE OBTAINED BY ASSUMING CONSTANT COEFFICIENTS AND ESTUARIES OF INFINITE LENGTH. SOLUTIONS ARE CALCULATED FOR CONSTANT CONCENTRATIONS, INSTANTANEOUS RELEASES, AND CONTINUOUS DISCHARGES. THIS PROGRAM CAN BE USED TO CHECK THE ACCURACY OFFINITE DIFFERENCE FORMULATIONS.

LOGIC AND PROGRAMMING--JONATHAN YOUNG, TEXAS A&M UNIVERSITY
*** INPUT VARIABLES ***

DELTA X = DISTANCE INCREMENT (MILES OR FEET)
DELTA T = TIME INCREMENT (DAYS OR SECONDS)
U = VELOCITY (FEET/SECOND)
EX = DISPERSION COEFFICIENT (MILES**2/DAY OR FEET**2/SECOND)
CZERO = CONSTANT CONCENTRATION AT AN OUTFALL, OR MASS PER UNIT CROSS SECTIONAL AREA, OR MASS RATE PER UNIT AREA
ITEND = NUMBER OF ITERATIONS
IXEND = NUMBER OF X INCREMENTS
XBEGIN = INITIAL X VALUE (FEET)
LTYPE = TYPE OF LOADING
  1 MEANS CONSTANT CONCENTRATION AT THE OUTFALL
  2 MEANS AN INSTANTANEOUS RELEASE
  3 MEANS A CONTINUOUS DISCHARGE
XKO = DECAY RATE (/DAY) -- USED FOR LTYPE = 2 ONLY
DIMENSION C(100)
DATA C/100*0.0/, PI/ 3.141593/
READ , DELTAX, DELTAT, U, EX, CZERO, ITEND, IXEND, XBEGIN, LTYPE,
$ XKD
PRINT, DELTAX, DELTAT, U, EX, CZERC, ITEND, IXEND, XBEGIN, LTYPE,
$ XKD
DAYS=DELTAT
SECS=DELTAT
IF(DELTAT.LE.1.) SECS=DELTAT*86400.
IF(DELTAT.GT.1.) DAYS=DELTAT/86400.
DELF=DELTAX
DELM=DELTAX
IF(DELTAX.GT.2.0) DELM=DELTAX/5280.
IF(DELTAX.LE.2.0) DELF=DELTAX*5280.
UM=U*86400./5280.
D=EX
DM=EX
IF(EX.LE.25.0) D=EX*5280.*5280./86400.
IF(EX.GT.25.0) DM=EX*86400./(5280.*5280.)
XKDS=XKD/86400.
PRINT 5
5 FORMAT(1H1,1X,110('**'),//)
PRINT 7
7 FORMAT(40X, 42('**'))
PRINT 25
25 FORMAT(40X, '******* ONE DIMENSIONAL ANALYSIS *******')
GO TO (10, 20, 30, 40, 50), LTYPE
10 PRINT 11, CZERO
11 FORMAT(40X, '********* CONSTANT CONCENTRATION *********', //, 10X,
     $ ' THE FOLLOWING PARAMETERS ARE APPLIED TO A CONSTANT CONCENTRAT
     $ION OF ', F7.0, ' PPM MAINTAINED AT X = 0.0', //)
10 GO TO 60
11 20 PRINT 11, CZERO
12 GO TO 60
13 30 PRINT 31, CZERO
14 31 FORMAT(40X, '********* INSTANTANEOUS RELEASE *********', //, 10X,
     $ ' THE FOLLOWING PARAMETERS ARE APPLIED TO AN INSTANTANEOUS RELEAS
     $E OF ', F10.0, ' UNITS OF MASS PER UNIT AREA', //, 15X, ' AT X=0.0 A
     $ND TIME = 0.0', //)
15 GO TO 60
16 40 PRINT 31, CZERO
17 GO TO 60
18 50 PRINT 51, CZERO
19 51 FORMAT(40X, '********* CONTINUOUS DISCHARGE *********', //, 10X,
     $ ' THE FOLLOWING PARAMETERS ARE APPLIED TO A CONTINUOUS DISCHARG
     $E OF ', F9.0, ' UNITS OF MASS PER UNIT AREA PER UNIT TIME AT X = C
     $0', //)
20 GO TO 60
21 60 CONTINUE
22 PRINT 535, LTYPE
23 535 FORMAT(//, T59, 'LTYPE = ', I3, //)
24 PRINT 540, DELF, DELM
25 540 FORMAT( T53, 'X INCREMENT = ', F7.0, ' FEET (', F8.5, ' MILES
     $) ', //)
26 PRINT 550, SECS, DAYS
27 550 FORMAT(T50, 'TIME INCREMENT = ', F7.0, ' SECONDS (', E10.3, ' DAYS
     $) ')
1828 PRINT 565, U, U,W
1829 565 FORMAT(T37, 'HORIZONTAL VELOCITY = ', F5.2, ' FEET/SECOND
1830 $(', F6.2, ' MILES/DAY )',/)
1831 PRINT 575, D, DM
1832 575 FORMAT(T35, 'HORIZONTAL DISPERSION = ', F6.0, ' FEET SQUAR
1833 $ED / SECOND ( ', F5.2, ' MILES SQUARED / DAY )',/)
1834 PRINT 580, XKD, XKD
1835 580 FORMAT(T54, 'DECAY RATE = ', F6.3, ' PER DAY ( ', E10.3,
1836 $ ' PER SECOND )',/)
1837 PRINT 65
1838 65 FORMAT(//'/*',IX,110('**'),///)
1839 DO 1000 IT=1,ITEND
1840 T=SECS*IT
1841 TIME=T/86400.
1842 GOTO (100, 300, 500), LTYPE
1843 100 CONTINUE
1844 IF(U.NE.0.0) GO TO 20C
1845 DO 120 IX=1,IXEND
1846 X=(IX-1)*DELF*XBEGIN
1847 B3=X/(2.*SQR(D*T))
1848 C(IX)=CZERO*ERFC(B3)
1849 120 CONTINUE
1850 GO TO 800
1851 DO 220 IX=1,IXEND
1852 X=(IX-1)*DELF*XBEGIN
1853 A=U*X/D
1854 B1=(X+U*T)/(2.*SQR(D*T))
1855 B2=(X-U*T)/(2.*SQR(D*T))
1856 C(IX)=(CZERO/2.)*(EXP(A)*ERFC(B1)+ERFC(B2))
220 CONTINUE
221 GO TO 800
222 XM=CZERO
223 DECAY = -XKD*TIME
224 DO 350 IX=1,IXEND
225 X=(IX-1)*DELF+XBEGIN
226 C1=XM/SQRT(4.*PI*D*T)
227 C2=(X-U*T)**2/(4.*D*T)
228 C(IX)=C1*EXP(C2)
229 C(IX)=C(IX)*EXP(DCAY)
230 CONTINUE
231 GO TO 800
232 XM=CZERO
233 DO 520 IX=1,IXEND
234 X=(IX-1)*DELF+XBEGIN
235 E1=XM**0.5/SQRT(P*+D)
236 E2=X**2/(4.*D*T)
237 E3=XM*X/(2.*D)
238 E4=X/(2.*SQRT(D*T))
239 C(IX)=E1*EXP(E2)-E3*ERFC(E4)
240 CONTINUE
241 GO TO 800
242 800 CONTINUE
243 PRINT 825, T, TIME
244 825 FORMAT(///, ' TIME = ', F7.0, ' SECONDS (' , F8.5, ' DAYS )')
MSTOP=IXEND/2
DO 875 M=1,MSTOP
    P1=DEL*(M-1)
    P2=DELM*(M-1)
    P3=DEL*(M+MSTOP-1)
    P4=DELM*(M+MSTOP-1)
    M1=M+MSTOP
    PRINT 850, P1, P2, C(M), P3, P4, C(M1)
850 FORMAT(2(* C AT ', F7.0, ' FEET (*, F8.4, ' MILES ) = ', F10.2, $ 10X))
CONTINUE
CONTINUE
STOP
END
Input Data for EXACT-I

Each line represents a new card unless single spaced.

0.25, 0.1, 5., 10., 100., 2, 20, -5280., 1, 0.
COMPUTER PROGRAM FOR

EXACT-II

Object Code = 4,656 bytes
Array Area = 40,160 bytes
Total = 44,816 bytes
********** EXACT-II **********
***** TWO DIMENSIONAL CONVECTIVE DISPERSION *****
******************************

This computer program computes the concentration profile with
time in two dimensions for an instantaneous release of a mass M
per unit depth (or width). The calculations are based upon a
known, closed-form solution to the estuary mass transport equation
assuming constant coefficients and infinite boundaries. This
program can be used to check the accuracy of finite difference
formulations.

logic and programming--Jonathan Young, Texas A&M University
*** INPUT VARIABLES ***

NX = NUMBER OF POINTS IN THE X DIRECTION
NZ = NUMBER OF POINTS IN THE Z DIRECTION
DELTAX = DISTANCE INCREMENT IN THE X DIRECTION (MILES)
DELTAZ = DISTANCE INCREMENT IN THE Z DIRECTION (MILES)
DELTAT = TIME INCREMENT (DAYS)
XBEGIN = INITIAL X VALUE (MILES)
ZBEGIN = INITIAL Z VALUE (MILES)
ITER = NUMBER OF ITERATIONS
EX = DISPERSION COEFFICIENT IN THE X DIRECTION (MILES**2/DAY)
EZ = DISPERSION COEFFICIENT IN THE Z DIRECTION (MILES**2/DAY)
VXF = VELOCITY IN X DIRECTION (FEET/SECOND)
VZF = VELOCITY IN Z DIRECTION (FEET/SECOND)
XM = MASS PER UNIT DEPTH OR WIDTH
XKD = DECAY RATE (1/DAY)
FMT1 = FORMAT FOR HEADING
C

1893 DIMENSION C(100,100)
1894 DIMENSION FMT1(40)
1895 INTEGER COUNT
1896 DATA C/10000.0,0/
1897 READ, NX,NZ,DELTA, DELFZ,DELTAT,XBEGIN,ZBEGIN,ITER
1898 PRINT, NX,NZ,DELTA, DELFZ,DELTAT,XBEGIN,ZBEGIN,ITER
1899 READ, EXF,EZF,VXF,VZF, XM,XKD
1900 PRINT, EXF,EZF,VXF,VZF, XM,XKD
1901 READ 205, (FMT1(I), I=1,40)
1902 PRINT 202
1903 202 FORMAT('',1X,'FORMAT FOR OUTPUT')
1904 PRINT 205, (FMT1(I), I=1,40)
1905 205 FORMAT(20A4,/,20A4)
1906 DELTAT=1./24.
1907 COUNT=2
1908 PI=3.14159
1909 SECS=DELTAT*86400.
1910 DELFX=DELTAX*5280.
1911 DELTAZ=DELFZ/5280.
1912 VX=VXF*86400./5280.
1913 VZ=VZF*86400./5280.
1914 EX=EXF*86400./(5280.*5280.)
1915 EZ= EZF*86400./(5280.*5280.)
1916 XKDS = XKD/86400.
1917 PRINT 5
1918 5 FORMAT(1H1,1X,110('**'),//
1919 PRINT 7
1920 7 FORMAT(40X, 42('**'))
PRINT 25
25 FORMAT(40X, '******** TWC DIMENSIONAL ANALYSIS ********')
PRINT 31, XM
31 FORMAT(40X, '******** INSTANTANEOUS RELEASE ********', $
   40X, 42('**'), '/',
   40X, 'THE FOLLOWING PARAMETERS ARE APPLIED TO AN INSTANTANEOUS RELEASE', $
   40X, 'OF ', F10.0, ' PPM AT X = 0.0 AND TIME = 0.0', '.
PRINT 540, DELFX, DELTAX
540 FORMAT( T53, 'X INCREMENT = ', F7.0, ' FEET (', F8.5, ' MILES $
   545 FORMAT( T53, 'Z INCREMENT = ', F7.0, ' FEET (', F8.5, ' MILES $
   550 FORMAT(T50, 'TIME INCREMENT = ', F7.0, ' SECONDS (', E10.3, ' DAYS $
   565 FORMAT(T37, ' HORIZONTAL VELOCITY = ', F5.2, ' FEET/SECOND $
   570 FORMAT(T37, ' VERTICAL VELOCITY = ', F5.2, ' FEET/SECOND $
   575 FORMAT(T35, ' HORIZONTAL DISPERSION = ', F6.0, ' FEET SQUAR $
   580 FORMAT(T35, ' VERTICAL DISPERSION = ', F9.3, ' FEET SQUAR $
   585 FORMAT(T54, 'DECAY RATE = ', F6.3, ' PER DAY (', E10.3, ' PER SECCA $
   590 FORMAT(T54, 'PER DAY (', E10.3, ' PER SECCA$
   595 FORMAT(T54, ' PER SECCA$
   600 FORMAT(T54, ' PER SECCA$
1941   \texttt{XBF=XBEGIN*5280.}
1942   \texttt{PRINT 585, XBF, XBEGIN}
1943   \texttt{585 FORMAT(T49, 'INITIAL X VALUE =', F7.0, ' FEET (', F8.5, ' MILES $')/}
1944   \texttt{ZBF=ZBEGIN*5280.}
1945   \texttt{PRINT 590, ZBF, ZBEGIN}
1946   \texttt{590 FORMAT(T49, 'INITIAL Z VALUE =', F7.0, ' FEET (', F8.5, ' MILES $')/}
1947   \texttt{PRINT 595}
1948   \texttt{595 FORMAT( 1X, 110('**') )}
1949   \texttt{DO 300 IT=1,ITER}
1950   \texttt{T=IT*DELTAT}
1951   \texttt{DO 200 I=1,NX}
1952   \texttt{X=(I-1)*DELTAX+XBEGIN}
1953   \texttt{DO 100 J=1,NZ}
1954   \texttt{Z=(J-1)*DELTAZ+ZBEGIN}
1955   \texttt{TERM1=-(X-VX*T)**2/(4.*T*EX)}
1956   \texttt{IF(TERM1.LT.-50.) TERM1=-50.}
1957   \texttt{TERM2=SQR(T/(2.*T*EX))}
1958   \texttt{TERM3=-(Z-VZ*T)**2/(4.*T*EZ)}
1959   \texttt{IF(TERM3.LT.-50.) TERM3=-50.}
1960   \texttt{TERM4=SQR(T/(2.*T*EZ))}
1961   \texttt{C(I,J)=(XM/(2.*PI))*EXP(TERM1)*EXP(TERM3)/(TERM2*TERM4)}
1962   \texttt{XKDT=-XKDT*T}
1963   \texttt{C(I,J)=C(I,J)*EXP(XKDT)}
1964   \texttt{100 CONTINUE}
1965   \texttt{200 CONTINUE}
1966   \texttt{PRINT 275}
1967   \texttt{275 FORMAT(1H1)}}
1968    TIME = T * 86400.
1970    WRITE (6, FMT1) HOURS, T
1971    PRINT 225, ((C(I, J), J = 1, NZ), I = 1, NX)
1972    225 FORMAT (1 (1X, 11 (F6.0)))
1973    300 CONTINUE
1974    STOP
1975    END
Input Data for EXACT-II

Each line represents a new card unless single spaced.

28, 11, 0.2, 5., 0.04083333, -2.6, 0.0, 1
500., 0.010, 0., 0., 5.0, 0.

(//, 12X, 100('*'), /, 12X, '*', /, 12X, '*', 2X,
'CONCENTRATIONS AT TIME = ',
F7.2, ' HOURS (', F6.4, ' DAYS )', /, 12X, '*', /, 12X,
100('*') )
COMPUTER PROGRAM FOR

PROFILE-I

Object Code = 1864 bytes
Array Area = 00 bytes
Total = 1864 bytes
PROFILE - I

This program computes the steady-state concentration profile for a nonconservative substance, where the concentration is known at the source point.

*** INPUT VARIABLES ***

E = Dispersion coefficient (miles**2/day)
U = Velocity (miles/day)
K0 = Decay coefficient (/day)
CZERO = Concentration at outfall

1 READ(5,10)E,U,K0,CZERO
10 FORMAT(4F14.0)
2 WRITE(6,15)E
15 FORMAT(4X,24SDiffusion coefficient = ,F5.1, 21H square miles per day,
2Y,/) 
3 WRITE(6,20)U
20 FORMAT(4X,11HVelocity = ,F5.1, 14H miles per day,/) 
4 WRITE(6,23)CZERO
23 FORMAT(4X,'Concentration at outfall = ', F6.1, ' PPM,/') 
5 WRITE(6,25)K0
25 FORMAT(4X,20SDecay coefficient = ,F5.2, 39H per day,///1 
6 WRITE(6,30)
30 FORMAT(5X,1HX,13X,1HC,///)
ADONW=U/(2.*E)*(1.0-SQRT(1.0+4.*XXD/E/(U**2.)))
AUP= U/(2.*E)*(1.0+SQRT(1.0+4.*XXD/E/(U**2.)))
X=-11.0
DO 100 I=1,9
X=X+1.0
C=CZERO*EXP(AUP*X)
WRITE(6,50)X,C
50 FORMAT(F10.2,4X,F9.4,/)  
100 CONTINUE
DO 150 J=1,8
X=X+0.25
C=CZERO*EXP(AUP*X)
WRITE(6,50)X,C
150 CONTINUE
DO 200 K=1,2C
X=X+0.25
C=CZERO*EXP(ADONW*X)
WRITE(6,50)X,C
200 CONTINUE
DO 250 L=1,45
X=X+1.0
C=CZERO*EXP(ADONW*X)
WRITE(6,50)X,C
250 CONTINUE
STOP
END
Input Data for PROFILE-I

Each line represents a new card unless single spaced.

10.0  1.0  0.25  10.0
COMPUTER PROGRAM FOR

PROFILE-II

Object Code = 2,864 bytes
Array Area = 12,060 bytes
Total = 14,924 bytes
PROFILE-II

THIS COMPUTER PROGRAM CALCULATES THE STEADY-STATE PROFILES FOR BIOCHEMICAL OXYGEN DEMAND (BOD) AND DISSOLVED OXYGEN (DO) FOR A CONTINUOUS RELEASE OF A SUBSTANCE INTO AN IDEALIZED, ONE-DIMENSIONAL ESTUARY.

LOGIC AND PROGRAMMING--JONATHAN YOUNG, TEXAS A&M UNIVERSITY

*** INPUT VARIABLES ***

W = LOADING RATE (POUNDS OF BOD PER DAY)
Q = FLOWRATE (CFS)
U = VELOCITY (FEET/SECOND)
E = DISPERSION COEFFICIENT (FEET**2/SECOND)
K = DECAY RATE (/DAY, BASE E)
K2 = REAERATION COEFFICIENT (/DAY)
CSAT = OXYGEN SATURATION VALUE (PPM)
XUP = DISTANCE ABOVE OUTFALL (MILES)
XDOWN = DISTANCE BELOW OUTFALL (MILES)
DELMX = DISTANCE INCREMENT (MILES)
DI = INITIAL DEFICIT AT OUTFALL (PPM)
REAL L,K,K2,M1,M2,LZ
DIMENSION L(1005),D(1005),C(1005)
READ, W,Q,U,E,K,K2,CSAT,XUP,XDOWN,DELMX,DI
PRINT, W,Q,U,E,K,K2,CSAT,XUP,XDOWN,DELMX,DI
Q=Q*62.4
W=W/86400. * 1.E6
XXK=K
XXK2=K2
K=K/86400.
K2=K2/86400.
DELMX=DELMX*5280.
IF(XXK.EQ.XXK2) XUP=0.0
NDIM=((-XUP+XDOWN)/DELMX + 2
TERM1=1. + (4.*K*E)/(U*U)
M1=SQRT(TERM1)
TERM2=1. + (4.*K2*E)/(U*U)
M2=SQRT(TERM2)
LZ=W/(Q*M1)
U2E=U/(2.*E)
XJ=U2E*(1.-M1)
X=(XUP-DELMX)*5280.
PRINT 50
50 FORMAT(1HL,/,1X,'BOD AND OXYGEN PROFILES',/,$ T5, 'X', T15, 'L', T25, 'O', T32, 'OX',/)
SIGN=1.
TEST=ABS(XXX-XXK2)
IF(TEST.<.001) 250,25,25
CONTINUE
DZ=(K*W)/((K2-K)*Q)
DO 200 I=1,NDIM
X=X+DELTA X
IF(X.GE.0.0) SIGN=-1.
EXPON1=U2E*X*(1.+SIGN*M1)
EXPON2=U2E*X*(1.+SIGN*M2)
L(I)=LZ*EXP(EXPON1)
D(I)=DZ*((1./M1)*EXP(EXPON1)-(1./M2)*EXP(EXPON2))
D(I)=D(I) + DI*EXP(EXPON2)
C(I)=CSAT-D(I)
X=X/5280.
PRINT 100, XM,L(I),D(I),C(I)
100 FORMAT(1X,F7.2,3X,F8.3,3X,F5.2,3X,F5.2)
200 CONTINUE
GO TO 400
250 CONTINUE
X=-DELTA X
DO 300 I=1,NDIM
X=X+DELTA X
IF(X.GE.0.0) SIGN=-1.
EXPON=U2E*X*(1.+SIGN*M1)
L(I)=LZ*EXP(EXPON)
DZ=K*LZ*X/(U-2.*E*XJ)
XJX=XJ*X
D(I)=DZ*EXP(EXPON)+DI*EXP(XJX)
C(I)=CSAT-D(I)
300 CONTINUE
XM=X/5280.
PRINT 100, XM,L(I),D(I),C(I)
400 CONTINUE
STOP
END
Input Data for PROFILE-II

Each line represents a new card unless single spaced.

50000., 1000., .2, 400., .23, .10, 8., -10., 20., 0.5, 0.0