UNITED STATES DEPARMENT OF THE INTERIOR

GEOLOGICAL SURVEY

DIFDEC2, a BASIC Program for Calculating Production, Diffusion, and Reaction of Two Components in a Tabular Layer

by

Richard F. Sanford[1]

Open-File Report 85-101

---

[1] U.S. Geological Survey, M.S. 916, Denver Federal Center, Denver, CO 80225

ABSTRACT

The BASIC computer program, DIFDEC2, is general enough to apply to many common geological occurrences involving two diffusing and reacting components. The program simulates the following situation: chemical component u is produced by a chemical reaction or radioactive decay in a tabular layer. It simultaneously diffuses out of the layer and reacts or decays to a second component, v, which itself diffuses away from the layer. It has been used to describe the production of ^{222}Rn from ^{238}U and ^{222}Rn's simultaneous diffusion and decay to a second diffusing component ^{206}Pb. The program uses the Crank-Nicholson finite-difference technique and consequently is stable for a wide range of values of diffusion coefficients and reaction constants. Optimum accuracy and computing time can be achieved by adjusting space and time increments. The applications are as numerous as the geological situations.

INTRODUCTION

Most geological processes involving diffusion and reaction are so complex that analytical solutions cannot represent them realistically. There are, for example, analytical solutions to simultaneous diffusion and first order reaction of one component for various idealized geometries and boundary conditions, but add another component or change the order of the reaction and one quickly faces an intractable problem. However, finite-difference methods can solve a wide range of complex problems if certain precautions are taken to assure numerical stability and accuracy.

The program described here solves the general problem of one chemical component (u) being produced in a reaction, diffusing away from its source and simultaneously reacting to a second chemical component (v) that also diffuses away. The name of the program is DIFDEC2, for DIFfusion and DECay of 2 components. The present program was used by K. R. Ludwig (personal communication) to determine the relative diffusion coefficients of Rn and Pb, because the radioactive decay constants are well known. In this example, ^{238}U in a strataform, tabular layer decays and produces ^{222}Rn, which simultaneously diffuses away from the source layer and decays to ^{206}Pb, which itself diffuses away from the layer. Alternatively, if diffusion coefficients are well known for two reacting species, then the reaction rate constants might be estimated. Another type of situation in which this program applies is to chemical reaction of species in solution. For example, a bed of limestone dissolves, producing Ca^{+2} in solution. The Ca^{+2} ions migrate and simultaneously cause a second dissolved species to precipitate. Oxidation-reduction reactions can also be modeled with this program. For example, Fe^{+3} is leached from an oxidized bed and diffuses away while also being reduced to Fe^{+2} by organic matter in an adjacent bed.

The following restrictions are assumed: 1) the two separate reactions (that produce u and v, respectively) are first order (like radioactive decay), i.e., the rate of reaction is proportional to the amount of reactant present; 2) the tabular layer is much wider than it is thick; 3) diffusion coefficients are constant with time and distance; 4) the reaction constants are constant with time and distance; and 5) there are, at most, 103 space grid points. The program can be modified to allow for variation of any of these conditions without any change in the basic method of solution. All that is required is a knowledge of BASIC programming. The only major restriction imposed by the method of solution is that the rate of diffusion not be very small compared to the rate of reaction. This point is explained below.

Advantages of the present program are: 1) capability to solve for two related components; 2) capacity to deal with two simultaneous reactions with coupled diffusion; 3) provision for either finite diffusion boundaries or virtually unlimited diffusion fields; 4) superior stability, speed, and efficiency of the second-order correct Crank-Nicholson method; 5) ease by which the program can be modified by anyone familiar with the BASIC language and; 6) flexibility allowed by provisions for changing the time and distance increments to achieve the desired accuracy in the shortest amount of computing time.

SAMPLE PROBLEM AND SOLUTION

To illustrate the use of this program, a sample run is given here (table 1). In this example we want to solve for the concentrations of ^{222}Rn and ^{206}Pb as a function of distance and time. Our diffusion field is the distance (x) from the center of the uranium-bearing layer (at x=0) to the center of the uranium-free layer (at x=Q). Thus the diffusion field has length Q. The half-thickness of the uranium-rich layer is P. As in any finite-difference solution, we solve for the desired values at discrete points in space and time. In this example, the number of space grid points, R, is equal to 13. Grid point 2 is at x=0, and grid point 12 is at x=Q. Points 1 and 13 are fictitious points outside of the physical diffusion field and are only used for computational purposes. Thus, points 2 through 12 represent the physical diffusion field, which is therefore divided into 10 increments having length $\Delta x = P/10$. The program automatically sets the size of the first time increment, but the user can cause the increment to increase at different rates by adjusting T1. At each new time step the time increment is multiplied by T1. For example, setting T1 = 1.5 makes the time increment increase by 50 percent each step. In this problem we wish to see the concentration profiles corresponding to 100 m.y. of diffusion and reaction. Therefore, we set the time limit T2 = 10^8 yrs. In other problems we might want to cut off the calculations after a certain number of time steps, T3, regardless of the total time elapsed.

The governing equation for ^{222}Rn is:

$$\frac{d}{dt}[^{222}\mathrm{Rn}] = D_{Rn}\frac{\partial^2[^{222}\mathrm{Rn}]}{\partial x^2} + [^{238}\mathrm{U}_o]\,\lambda_{238}e^{-\lambda_{238}t} - \lambda_{222}[^{222}\mathrm{Rn}] \qquad (1)$$

and that for ^{206}Pb is:

$$\frac{d}{dt}[^{206}\mathrm{Pb}] = D_{Pb}\frac{\partial^2[^{206}\mathrm{Pb}]}{\partial x^2} + \lambda_{222}[^{222}\mathrm{Rn}], \qquad (2)$$

where $[^{222}\mathrm{Rn}]$ and $[^{206}\mathrm{Pb}]$ are the concentrations to be solved for, D_{Rn} and D_{Pb}

are diffusion coefficients, $[^{238}U_0]$ is the initial concentration of ^{238}U, and λ_{238} and λ_{222} are decay constants of ^{238}U and ^{222}Rn, respectively. The following values were assigned:

$D_{Rn} = 100\ cm^2/yr$, $D_{Pb} = 10^{-8}\ cm^2/yr$, $\lambda_{238} = 0.155125 \times 10^{-9}/yr$,

$\lambda_{222} = 67/yr$, and $[^{238}U_0] = 10^{14}$.

Sample output is shown in table 1. Concentration profiles are shown for the first and last time steps. For the final time step, the number of the step and the total elapsed time is shown. Below this line are four columns showing the space grid-point number, the distance (x) from the origin (at the center of the uranium-rich layer), the concentration of component u (which here is $[^{222}Rn]$), and the concentration of component v, $[^{206}Pb]$.

This calculation was designed to estimate the relative effective diffusion rates of ^{222}Rn and ^{206}Pb, which we can do because we know the decay constants, λ_{238} and λ_{222}. Here we find that the diffusion coefficient of ^{222}Rn must be about ten orders of magnitude larger than the ^{206}Pb diffusion coefficient in order for the diffusion of ^{222}Rn to have any effect on the ^{206}Pb concentration profile. This suggests either that ^{206}Pb is quickly immobilized by a precipitation reaction or that diffusion of ^{206}Pb is by a much slower mechanism (perhaps solid-state diffusion) than that of ^{222}Rn. Further results of numerical experiments wil be discussed in detail by K. R. Ludwig (personal communication).

PROGRAM DESCRIPTION

The solution of equations (1) and (2) by the program DIFDEC2 is now described in detail. The program is listed in table 2. Equations (1) and (2) take the general form:

$$\frac{du}{dt} = D5\,\frac{\partial^2 u}{\partial x^2} + E(t) - L5\ u \tag{3}$$

and

$$\frac{dv}{dt} = D6\,\frac{\partial^2 v}{\partial x^2} + L5\ u, \tag{4}$$

where D5 and D6 are the diffusion coefficients of u and v, respectively, E(t) is the rate of the reaction that produces u, and L5 is the decay constant for the reaction of u to v. The terms in (3) and (4) are now replaced by second-order correct Crank-Nicholson finite-difference approximations (von Rosenberg, 1975, p. 22 ff.). These are, for u:

$$\left(\frac{du}{dt}\right)_i^{n+\frac{1}{2}} \simeq \frac{u_i^{n+1} - u_i^n}{\Delta t}, \tag{5}$$

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_i^{n+\frac{1}{2}} \simeq \frac{1}{2}\left[\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{(\Delta x)^2} + \frac{u_{i+1}^{n} - 2u_i^{n} + u_{i-1}^{n}}{(\Delta x)^2}\right], \tag{6}$$

$$E_i^{n+\frac{1}{2}} = \frac{U1}{\Delta t} \int L6e^{-L6t}dt$$

$$= \frac{U1}{\Delta t}\left(-e^{-L6t^{n+1}} + e^{-L6t^{n}}\right), \tag{7}$$

and

$$u_i^{n+\frac{1}{2}} \simeq \frac{1}{2}\left(u_i^{n+1} + u_i^{n}\right), \tag{8}$$

where i is the number of the current space grid-point, n is the number of the current time step and U1 is the initial concentration of the species that decays to u. Equations (5) to (8) for v are identical but with v instead of u.

Substituting (5), (6), (7), and (8) into (3) yields:

$$\frac{u_i^{n+1} - u_i^{n}}{\Delta t} = \frac{D5}{2}\left[\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{(\Delta x)^2} + \frac{u_{i+1}^{n} - 2u_i^{n} + u_{i-1}^{n}}{(\Delta x)^2}\right] + E_i^{n+\frac{1}{2}} - \frac{L5}{2}\left(u_i^{n+1} + u_i^{n}\right). \tag{9}$$

Multiplying equation 9 through by $2\Delta t(\Delta x)^2$ and rearranging so that all the unknown terms containing u^{n+1} are on the left, and all the known terms are on the right, yields:

$$\begin{aligned} -D5\,\Delta t\,u_{i-1}^{n+1} &+ [2(\Delta x)^2 + 2D5\Delta t + \Delta t(\Delta x)^2 L5]\,u_i^{n+1} - D5\Delta t u_{i+1}^{n+1} \\ &= D5\Delta t u_{i-1}^{n} + [2(\Delta x)^2 - 2D5\Delta t - \Delta t(\Delta x)^2 L5]\,u_i^{n} + D5\Delta t u_{i+1}^{n} \\ &\quad + 2\Delta t(\Delta x)^2 E_i^{n+\frac{1}{2}}. \end{aligned} \tag{10}$$

This is in ditridiagonal form. Comparison with equations in von Rosenberg (1975, p. 116), reproduced here as table 3, shows that

$$\begin{aligned} a_i^{(1)} &= -D5\,\Delta t & a_i^{(2)} &= 0 \\ b_i^{(1)} &= 2(\Delta x)^2 + 2D5\Delta t + \Delta t(\Delta x)^2 L5 & b_i^{(2)} &= 0 \\ c_i^{(1)} &= -D5\Delta t & c_i^{(2)} &= 0 \end{aligned} \tag{11}$$

$$d_i^{(1)} = D5\Delta t u_{i-1}^n + \left[2(\Delta x)^2 - 2D5\Delta t - \Delta t(\Delta x)^2 L5\right]u_i^n + D5\Delta t u_{i+1}^n + 2\Delta t(\Delta x)^2 E_i^{n+\frac{1}{2}}.$$

Now, substituting (5), (6), (7), and (8), with u replaced by v, into (4), we have:

$$\frac{v_i^{n+1} - v_i^n}{\Delta t} = \frac{D6}{2}\left[\frac{v_{i+1}^{n+1} - 2v_i^{n+1} + v_{i-1}^{n+1}}{(\Delta x)^2} + \frac{v_{i+1}^n - 2v_i^n + v_{i-1}^n}{(\Delta x)^2}\right] + \frac{L5}{2}\left(u_i^{n+1} + u_i^n\right). \qquad (12)$$

Again, multiplying equation (12) through by $2\Delta t(\Delta x)^2$ and rearranging so that v^{n+1} and u^{n+1} terms are on the left and known terms on the right, like we did for equation 9, yields:

$$-D6\Delta t v_{i-1}^{n+1} - \Delta t(\Delta x)^2 L5 u_i^{n+1} + \left[2(\Delta x)^2 + 2D6\Delta t\right]v_i^{n+1} - D6\Delta t v_{i+1}^{n+1}$$

$$= D6\Delta t v_{i-1}^n + \Delta t(\Delta x)^2 L5 u_i^n + \left[2(\Delta x)^2 - 2D6\Delta t\right]v_i^n + D6\Delta t v_{i+1}^n. \qquad (13)$$

Comparison with von Rosenberg (table 3) shows that:

$$\begin{array}{llll}
a_i^{(3)} = 0 & & a_i^{(4)} = -D6\Delta t & \\
b_i^{(3)} = -\Delta t(\Delta x)^2 L5 & & b_i^{(4)} = 2(\Delta x)^2 + 2D6\Delta t & (14) \\
c_i^{(3)} = 0 & & c_i^{(4)} = -D6\Delta t & \\
\multicolumn{4}{l}{d_i^{(2)} = D6\Delta t v_{i-1}^n + \Delta t(\Delta x)^2 L5 u_i^n + \left[2(\Delta x)^2 - 2D6\Delta t\right]v_i^n + D6\Delta t v_{i+1}^n.}
\end{array}$$

So far we have put equations (3) and (4) into ditridiagonal form. Now we incorporate the boundary conditions into the ditridiagonal matrix. Equations (3) and (4) are evaluated at the space grid points 2 through R-1, where R is the total number of space grid points. The boundary conditions are represented by equations corresponding to points 1 and R, which lie outside the physical limits of the diffusion field and are only for the purpose of calculating the concentration gradients at the physical end points, 2 and R-1. The condition of no flow across the boundaries is represented by:

$$\frac{\partial u}{\partial x} = 0 \quad \text{and} \quad \frac{\partial v}{\partial x} = 0 \quad \text{at } x=0 \text{ and } x=Q. \qquad (15)$$

Substituting the second-order correct Crank-Nicholson approximations,

$$\left(\frac{\partial u}{\partial x}\right)_i \approx \frac{u_{i+1} - u_{i-1}}{2\Delta x} \quad \text{and} \quad \left(\frac{\partial v}{\partial x}\right)_i \approx \frac{v_{i+1} - v_{i-1}}{2\Delta x},$$

for i=2 and i=R-1, yields:

$$u_1 = u_3, \quad v_1 = v_3. \qquad (16a)$$

$$u_{R-2} = u_R, \quad \text{and} \quad v_{R-2} = v_R. \qquad (16b)$$

These are not in ditridiagonal form. To fix them we eliminate u_3 and v_3 from (16a) by linear combination with von Rosenberg's first and second equations (table 3) for i=2 and i=R-1. For i=2, the first equation becomes:

$$a_2^{(1)}u_1 + a_2^{(2)}v_1 + b_2^{(1)}u_2 + b_2^{(2)}v_2 + c_2^{(1)}u_3 + c_2^{(2)}v_3 = d_2^{(1)}. \qquad (17)$$

Substituting (16a) and eliminating u_3 and v_3 yields:

$$[a_2^{(1)} + c_2^{(1)}]u_1 + [a_2^{(2)} + c_2^{(2)}]v_1 + b_2^{(1)}u_2 + b_2^{(2)}v_2 = d_2^{(1)}. \qquad (18)$$

Now this is in ditridiagonal form, *viz.*,

$$a_1^{(1)}=0,\ \ a_1^{(2)}=0,\ \ b_1^{(1)}=a_2^{(1)}+c_2^{(1)},\ \ b_1^{(2)}=a_2^{(2)}+c_2^{(2)},$$
$$c_1^{(1)}=b_2^{(1)},\ \ c_1^{(2)}=b_2^{(2)},\ \text{and}\ d_1^{(1)}=d_2^{(1)}. \qquad (19)$$

Similarly for von Rosenberg's second equation,

$$a_1^{(3)}=0,\ \ a_1^{(4)}=0,\ \ b_1^{(3)}=a_2^{(3)}+c_2^{(3)},\ \ b_1^{(4)}=a_2^{(4)}+c_2^{(4)},$$
$$c_1^{(3)}=b_2^{(3)},\ \ c_1^{(4)}=b_2^{(4)},\ \text{and}\ d_1^{(2)}=d_2^{(2)}. \qquad (20)$$

Now for i=R-1, the first equation (table 3) becomes:

$$a_{R-1}^{(1)}u_{R-2} + a_{R-1}^{(2)}v_{R-2} + b_{R-1}^{(1)}u_{R-1} + b_{R-1}^{(2)}v_{R-1} + c_{R-1}^{(1)}u_R + c_{R-1}^{(2)}v_R = d_{R-1}^{(1)}. \qquad (21)$$

Substitute (16b) and eliminate u_{R-2} and v_{R-2}:

$$b_{R-1}^{(1)}u_{R-1} + b_{R-1}^{(2)}v_{R-1} + [a_{R-1}^{(1)} + c_{R-1}^{(1)}]u_R + [a_{R-1}^{(2)} + c_{R-1}^{(2)}]v_R = d_{R-1}^{(1)}. \qquad (22)$$

This is in ditridiagonal form, *viz.*,

$$a_R^{(1)} = b_{R-1}^{(1)},\ \ a_R^{(2)} = b_{R-1}^{(2)},\ \ b_R^{(1)} = a_{R-1}^{(1)} + c_{R-1}^{(1)},\ \ b_R^{(2)} = a_{R-1}^{(2)} + c_{R-1}^{(2)},$$
$$c_R^{(1)} = 0,\ \ c_R^{(2)} = 0,\ \text{and}\ d_R^{(1)} = d_{R-1}^{(1)}. \qquad (23)$$

Similarly for i=R-1 in von Rosenberg's second equation:

$$a_R^{(3)} = b_{R-1}^{(3)},\ \ a_R^{(4)} = b_{R-1}^{(4)},\ \ b_R^{(3)} = a_{R-1}^{(3)} + c_{R-1}^{(3)},\ \ b_R^{(4)} = a_{R-1}^{(4)} + c_{R-1}^{(4)},$$
$$c_R^{(3)} = 0,\ \ c_R^{(4)} = 0,\ \text{and}\ d_R^{(2)} = d_{R-1}^{(2)}. \qquad (24)$$

For a virtually infinite diffusion field, instead of finite no-flow boundaries, simply set the length of the diffusion field large enough so that the last few grid points at the ends are unaffected by diffusion in the time of the simulation.

Now we have everything in ditridiagonal form and we can solve using the algorithm of von Rosenberg (table 3).

For initial conditions, we set all u_i and v_i equal to zero. The initial concentration of the u-producing component is U1 for $x<P$, 0.5U1 for $x=P$, and 0 for $x>P$. Values of U1 at each grid point, i, can be assigned using the SGN function (for $x<0$, $SGN(x)=-1$; for $x=0$, $SGN(x)=0$; and for $x>0$, $SGN(x)=1$),

$$U1_i = U1[1-SGN(x_i-P)]/2.$$

It may be noted in the program that a factor of 10^{-9}U1 is added to this term. This number is negligible compared to the actual concentrations, but it helps to avoid truncation errors that result in numerical underflow and apparent negative concentrations.

The number of grid points input by the user can be adjusted to yield the desired accuracy. The closer the grid points, the better the approximations. However, this factor must be balanced by the increased computing time and the possibility of roundoff errors or numerical underflow.

The output shows the maximum roundoff error for the current time step as an aid in judging this contribution to the total error. This roundoff error is computed by plugging the calculated values of u_i^n and v_i^n back into the starting equations (9) and (12). The deviation from equality is the roundoff error. This is calculated for each space grid point, but only the largest value is printed out. Thus the user can decide for himself if the roundoff error is acceptably small relative to the computed values of u or v. Many BASIC compilers indicate underflow by an error message. On some computers using particular combinations of input parameters, roundoff errors may result in apparent negative concentrations. Because this is a clear indication of a numerical problem, the program is designed to terminate computing if a negative concentration appears. The Data General Eclipse computer used by the author carries 14 significant digits; a computer carrying fewer significant digits may have larger roundoff errors. (Use of this trade name is for identification purposes only and is not to imply any endorsement by the U.S. Geological Survey).

In practice, between 10 and 100 space grid points usually give acceptable results. It is convenient to make the total number of grid points equal to a nice round number (e.g., 10 or 100) plus 3 so that Δx is a nice fraction (e.g., 1/10 or 1/100) of the total length (Q). If more than 103 grid points are used, this number in the DIMENSION statements must be changed to the appropriate number.

The initial time step is computed by:

$$\Delta t = \frac{(\Delta x)^2}{2\ D7},$$

where D7 is the larger of the two diffusion coefficients, D5 or D6. This is the relationship that ensures stability for forward- and backward-finite-difference methods. Although unnecessary to ensure stabilty for the Crank-Nicholson method, which is stable for all combinations of Δx and Δt, it is a convenient size for the initial time step. The increase in this increment is determined by T1, input by the user. As with space grid points, smaller increments yield greater accuracy but longer computing time and possible numerical errors that the user must be aware of. In practice, a value of T1 that yields about 50 time steps gives more than sufficient accuracy in a reasonable length of computing time.

On a Data General Eclipse minicomputer, computing time was about 0.16 sec for one space grid-point per time step.

Problems may arise if the diffusion coefficients are so small or the reaction rates so large that the terms in equations (3) and (4) containing second derivatives are much smaller than the reaction or decay terms. Because the concentrations in such cases are essentially only dependent on time and not distance, equations without the second derivative term are more appropriate. For example, Wasserburg (1954) and Tilton (1960) give analytical solutions to this problem.

Variables used in the program are defined in tables 4 and 5.

ACKNOWLEDGEMENTS

I thank K. R. Ludwig for suggesting the problem. K. R. Ludwig and R. P. Christian promptly reviewed the manuscript.

REFERENCES

Tilton, G.R., 1960, Volume diffusion as a mechamism for discordant lead ages: Journal of Geophysical Research, v. 65, p. 2933-2945.

von Rosenberg, D.U., 1975, Methods for the numerical solution of partial differential equations: New York, Elsevier, 128p.

Wasserberg, G.J., 1954, in Nuclear Geology, Faul, H., ed., New York, John Wiley and Sons, p. 343.

Table 1. Sample run of DIFDEC2.

```
* RUN
DO YOU WANT A LIST OF VARIABLES ? YES
D5   DIFFUSION COEFICIENT OF U
D6   DIFFUSION COEFFICIENT OF V
U1   INITIAL CONCENTRATION OF COMPONENT REACTING TO U
L5   DECAY CONSTANT OF U
L6   DECAY CONSTANT OF REACTION PRODUCING U
P    1/2 WIDTH OF REACTING LAYER
Q    TOTAL LENGTH OF DIFFUSION FIELD
R    TOTAL NUMBER OF SPACE GRID POINTS
T1   TIME INCREMENT MULTIPIED BY THIS AMOUNT EACH STEP
T2   TOTAL TIME ALLOWED FOR SIMULATION
T3   TOTAL NUMBER OF TIME STEPS ALLOWED

DEFAULT VALUES

   D5            D6            U1              L5             L6
 10000        .00000001     1E+12          66.6242       1.55125E-10

 P    Q    R   T1   T2     T3
 50  100  13  1.5  1E+08  100

NAME OF PARAMETER TO CHANGE ? T3
NEW VALUE OF T3 ? 1000
NAME OF PARAMETER TO CHANGE ?
time step =  1              time =  .005
i            x              u              v
 1          -10             .66469279      .11027527
 2            0             .66482519      .11073362
 3           10             .66469279      .11071156
 4           20             .6636778       .11054251
 5           30             .65704405      .10943759
 6           40             .61383685      .10224097
 7           50             .33244097      5.5371534E-02
 8           60             5.1045079E-02                 8.5020939E-03
 9           70             7.8378813E-03                 1.3054814E-03
 10          80             1.204131E-03   2.0056066E-04
 11          90             1.8914427E-04                 3.1503964E-05
 12          100            5.6747361E-05                 9.4518688E-06
 13          110            1.8914427E-04                 3.1502296E-05
MAX ROUNDOFF ERROR IN U= 0  IN V= 2.4424907E-15
time step =  57             time =  1E+08
i            x              u              v
 1          -10             2.2351459      1.500654E+10
 2            0             2.2495003      1.5103062E+10
 3           10             2.2351459      1.500654E+10
 4           20             2.1826267      1.4652667E+10
 5           30             2.0568906      1.3805676E+10
 6           40             1.7742087      1.1901356E+10
 7           50             1.1462363      7.6964067E+09
 8           60             .51826398      3.4914579E+09
 9           70             .2355821       1.587137E+09
 10          80             .10984595      7.4014679E+08
 11          90             5.7326715E-02                 3.8627321E+08
 12          100            4.2972365E-02                 2.8975105E+08
 13          110            5.7326715E-02                 3.8627321E+08
MAX ROUNDOFF ERROR IN U= 2.4414063E-04  IN V= 2.4424907E-15

END AT 2020
*
```

Table 2. Listing of DIFDEC2.

```
* LIST
0100 PRINT "DO YOU WANT A LIST OF VARIABLES";
0110 INPUT L$
0120 IF L$<>"YES" THEN GOTO 0250
0130 PRINT "D5  DIFFUSION COEFICIENT OF U"
0140 PRINT "D6  DIFFUSION COEFFICIENT OF V"
0150 PRINT "U1  INITIAL CONCENTRATION OF COMPONENT REACTING TO U"
0160 PRINT "L5  DECAY CONSTANT OF U"
0170 PRINT "L6  DECAY CONSTANT OF REACTION PRODUCING U"
0180 PRINT "P   1/2 WIDTH OF REACTING LAYER"
0190 PRINT "Q   TOTAL LENGTH OF DIFFUSION FIELD"
0200 PRINT "R   TOTAL NUMBER OF SPACE GRID POINTS"
0210 PRINT "T1  TIME INCREMENT MULTIPIED BY THIS AMOUNT EACH STEP"
0220 PRINT "T2  TOTAL TIME ALLOWED FOR SIMULATION"
0230 PRINT "T3  TOTAL NUMBER OF TIME STEPS ALLOWED"
0240 PRINT
0250 DIM U(103),V(103),U1(103),A1(103),A2(103),A3(103),A4(103),B1(103)
0260 DIM B2(103),B3(103),B4(103),C1(103),C2(103),C3(103),C4(103),D1(103)
0270 DIM D2(103),L1(103),L2(103),L3(103),L4(103),G1(103),G2(103)
0280 DIM G3(103),G4(104)
0290 DIM U9(103),V9(103)
0300 REM
0310 REM INITIALIZATION
0320 REM
0330 READ D5,D6,U1,L5,L6,P,Q,R,T1,T2,T3
0340 DATA 10000,.00000001,1000000000000,66.6242,1.55125E-10,50
0350 DATA 100,13,1.5,100000000,100
0360 PRINT "DEFAULT VALUES"
0365 PRINT
0370 PRINT "   D5","   D6","   U1","   L5","   L6"
0380 PRINT D5,D6,U1,L5,L6
0385 PRINT
0390 PRINT " P   ";" Q   ";" R  ";" T1  ";" T2   ";" T3   "
0400 PRINT P;Q;R;T1;T2;T3
0405 PRINT
0410 PRINT "NAME OF PARAMETER TO CHANGE";
0420 INPUT N$
0430 IF N$="" THEN GOTO 0570
0440 PRINT "NEW VALUE OF ";N$;
0450 IF N$="D5" THEN INPUT D5
0460 IF N$="D6" THEN INPUT D6
0470 IF N$="U1" THEN INPUT U1
0480 IF N$="L5" THEN INPUT L5
0490 IF N$="L6" THEN INPUT L6
0500 IF N$="P" THEN INPUT P
0510 IF N$="Q" THEN INPUT Q
0520 IF N$="R" THEN INPUT R
0530 IF N$="T1" THEN INPUT T1
0540 IF N$="T2" THEN INPUT T2
0550 IF N$="T3" THEN INPUT T3
0560 GOTO 0410
0570 LET X0=Q/(R-3)
0580 FOR I=1 TO R
0590   LET U(I)=0
0600   LET V(I)=0
0610   LET X=(I-2)*X0
```

Table 2 con't.

```
0620   LET U1(I)=U1*(1-SGN(X-P))/2+.000000001*U1
0630 NEXT I
0640 LET X2=X0*X0
0650 LET D7=D5
0660 IF D6>D5 THEN LET D7=D6
0670 LET T=0
0680 LET T0=.5*X2/D7/T1
0690 LET S1=0
0700 REM
0710 REM BEGIN MAIN LOOP
0720 REM
0730 LET S1=S1+1
0740 LET T4=T
0750 LET T0=T0*T1
0760 LET T=T+T0
0770 IF T>T2 THEN LET T=T2
0780 IF T=T2 THEN LET T0=T-T4
0790 REM
0800 REM ASSIGN DITRIDIAGONAL COEFFICIENTS
0810 REM FIRST FOR I = 2 THROUGH R-1
0820 REM
0830 REM for i=2 through r-1
0840 FOR I=2 TO R-1
0850   LET A1(I)=-D5*T0
0860   LET A2(I)=0
0870   LET B1(I)=2*X2+2*D5*T0+T0*X2*L5
0880   LET B2(I)=0
0890   LET C1(I)=-D5*T0
0900   LET C2(I)=0
0910   LET E=U1(I)*L6*(EXP(-L6*T)+EXP(-L6*T4))/2
0920   LET D1(I)=D5*T0*U(I-1)+(2*X2-2*D5*T0-T0*X2*L5)*U(I)
0930   LET D1(I)=D1(I)+D5*T0*U(I+1)+2*T0*X2*E
0940   LET A3(I)=0
0950   LET A4(I)=-D6*T0
0960   LET B3(I)=-T0*X2*L5
0970   LET B4(I)=2*X2+2*D6*T0
0980   LET C3(I)=0
0990   LET C4(I)=-D6*T0
1000   LET D2(I)=D6*T0*V(I-1)+T0*X2*L5*U(I)+(2*X2-2*D6*T0)*V(I)+D6*T0*V(I+1)
1010 NEXT I
1020 REM
1030 REM NOW FOR I = 1
1040 REM
1050 LET A1(1)=0
1060 LET A2(1)=0
1070 LET B1(1)=A1(2)+C1(2)
1080 LET B2(1)=A2(2)+C2(2)
1090 LET C1(1)=B1(2)
1100 LET C2(1)=B2(2)
1110 LET D1(1)=D1(2)
1120 LET A3(1)=0
1130 LET A4(1)=0
1140 LET B3(1)=A3(2)+C3(2)
1150 LET B4(1)=A4(2)+C4(2)
1160 LET C3(1)=B3(2)
1170 LET C4(1)=B4(2)
```

Table 2 con't.

```
1180 LET D2(1)=D2(2)
1190 REM
1200 REM AND NOW FOR I = R
1210 REM
1220 LET A1(R)=B1(R-1)
1230 LET A2(R)=B2(R-1)
1240 LET B1(R)=A1(R-1)+C1(R-1)
1250 LET B2(R)=A2(R-1)+C2(R-1)
1260 LET C1(R)=0
1270 LET C2(R)=0
1280 LET D1(R)=D1(R-1)
1290 LET A3(R)=B3(R-1)
1300 LET A4(R)=B4(R-1)
1310 LET B3(R)=A3(R-1)+C3(R-1)
1320 LET B4(R)=A4(R-1)+C4(R-1)
1330 LET C3(R)=0
1340 LET C4(R)=0
1350 LET D2(R)=D2(R-1)
1360 REM
1370 REM SOLVE USING ALGORITHM, FORWARD STEPS FIRST
1380 REM
1390 LET M1=B1(1)*B4(1)-B2(1)*B3(1)
1400 LET L1(1)=(B4(1)*C1(1)-B2(1)*C3(1))/M1
1410 LET L2(1)=(B4(1)*C2(1)-B2(1)*C4(1))/M1
1420 LET L3(1)=(B1(1)*C3(1)-B3(1)*C1(1))/M1
1430 LET L4(1)=(B1(1)*C4(1)-B3(1)*C2(1))/M1
1440 LET G1(1)=(B4(1)*D1(1)-B2(1)*D2(1))/M1
1450 LET G2(1)=(B1(1)*D2(1)-B3(1)*D1(1))/M1
1460 FOR I=2 TO R
1470   LET E1=B1(I)-A1(I)*L1(I-1)-A2(I)*L3(I-1)
1480   LET E2=B2(I)-A1(I)*L2(I-1)-A2(I)*L4(I-1)
1490   LET E3=B3(I)-A3(I)*L1(I-1)-A4(I)*L3(I-1)
1500   LET E4=B4(I)-A3(I)*L2(I-1)-A4(I)*L4(I-1)
1510   LET W1=D1(I)-A1(I)*G1(I-1)-A2(I)*G2(I-1)
1520   LET W2=D2(I)-A3(I)*G1(I-1)-A4(I)*G2(I-1)
1530   LET M1=E1*E4-E2*E3
1540   LET L1(I)=(E4*C1(I)-E2*C3(I))/M1
1550   LET L2(I)=(E4*C2(I)-E2*C4(I))/M1
1560   LET L3(I)=(E1*C3(I)-E3*C1(I))/M1
1570   LET L4(I)=(E1*C4(I)-E3*C2(I))/M1
1580   LET G1(I)=(E4*W1-E2*W2)/M1
1590   LET G2(I)=(E1*W2-E3*W1)/M1
1600 NEXT I
1610 REM
1620 REM BACKWARD STEPS TO GET SOLUTION
1630 REM
1640 LET U(R)=G1(R)
1650 LET V(R)=G2(R)
1660 IF U(R)<0 THEN GOTO 1810
1670 IF V(R)<0 THEN GOTO 1810
1680 FOR I=R-1 TO 1 STEP -1
1690   LET U(I)=G1(I)-L1(I)*U(I+1)-L2(I)*V(I+1)
1700   LET V(I)=G2(I)-L3(I)*U(I+1)-L4(I)*V(I+1)
1710   IF U(I)<0 THEN GOTO 1810
1720   IF V(I)<0 THEN GOTO 1810
1730 NEXT I
```

Table 2 con't.

```
1740 IF S1<=1 THEN GOTO 1830
1750 IF S1=T3 THEN GOTO 1830
1760 IF T>=T2 THEN GOTO 1830
1770 GOTO 1970
1780 REM
1790 REM PRINT RESULTS FOR SELECTED TIME STEPS
1800 REM
1810 PRINT "WARNING: CONCENTRATION < 0.  PROGRAM STOPPED."
1820 LET T3=S1
1830 PRINT "time step = ";S1,"time = ";T
1840 PRINT "i","x","u","v"
1850 FOR I=1 TO R
1860   PRINT I,(I-2)*X0,U(I),V(I)
1870 NEXT I
1880 FOR I=2 TO R-1
1890   LET Y1=A1(I)*U(I-1)+B1(I)*U(I)+C1(I)*U(I+1)-D1(I)
1900   LET Y1=Y1+A2(I)*V(I-1)+B2(I)*V(I)+C2(I)*V(I+1)
1910   LET Y2=A4(I)*V(I-1)+B4(I)*V(I)+C4(I)*V(I+1)-D2(I)
1920   LET Y2=Y2+A3(I)*U(I-1)+B3(I)*U(I)+C3(I)*U(I+1)
1930   IF Y3<Y1 THEN LET Y3=Y1
1940   IF Y4<Y2 THEN LET Y4=Y2
1950 NEXT I
1960 PRINT "MAX ROUNDOFF ERROR IN U=";Y3;"IN V=";Y4
1970 IF T>=T2 THEN GOTO 2000
1980 IF S1>=T3 THEN GOTO 2000
1990 GOTO 0710
2000 REM END
2010 CLOSE
2020 END
*
```

Table 3. Algorithm for solution of tridiogonal matrix (from von Rosenberg, 1975, p. 116.

The equations are

$$a_i^{(1)}u_{i-1} + a_i^{(2)}v_{i-1} + b_i^{(1)}u_i + b_i^{(2)}v_i + c_i^{(1)}u_{i+1} + c_i^{(2)}v_{i+1} = d_i^{(1)}$$

and

$$a_i^{(3)}u_{i-1} + a_i^{(4)}v_{i-1} + b_i^{(3)}u_i + b_i^{(4)}v_i + c_i^{(3)}u_{i+1} + c_i^{(4)}v_{i+1} = d_i^{(2)}$$

$$\text{for } 1 \leq i \leq R$$

$$\text{with } a_1^{(m)} = c_R^{(m)} = 0 \qquad \text{for } 1 \leq m \leq 4$$

The algorithm is as follows:
First compute

$$\begin{aligned}
\beta_i^{(1)} &= b_i^{(1)} - a_i^{(1)}\lambda_{i-1}^{(1)} - a_i^{(2)}\lambda_{i-1}^{(2)} \\
\beta_i^{(2)} &= b_i^{(2)} - a_i^{(1)}\lambda_{i-1}^{(2)} - a_i^{(2)}\lambda_{i-1}^{(4)} \\
\beta_i^{(3)} &= b_i^{(3)} - a_i^{(2)}\lambda_{i-1}^{(1)} - a_i^{(4)}\lambda_{i-1}^{(3)} \\
\beta_i^{(4)} &= b_i^{(4)} - a_i^{(3)}\lambda_{i-1}^{(2)} - a_i^{(4)}\lambda_{i-1}^{(4)}
\end{aligned}$$

$$\text{with } \beta_1^{(m)} = b_1^{(m)} \qquad \text{for } 1 \leq m \leq 4$$

and

$$\begin{aligned}
\delta_i^{(1)} &= d_i^{(1)} - a_i^{(1)}\gamma_{i-1}^{(1)} - a_i^{(2)}\gamma_{i-1}^{(2)} \\
\delta_i^{(2)} &= d_i^{(2)} - a_i^{(3)}\gamma_{i-1}^{(1)} - a_i^{(4)}\gamma_{i-1}^{(2)}
\end{aligned}$$

$$\text{with } \delta_1^{(1)} = d_1^{(1)} \qquad \text{and} \qquad \delta_1^{(2)} = d_1^{(2)}$$

$$\text{and} \quad \mu_i = \beta_i^{(1)}\beta_i^{(4)} - \beta_i^{(2)}\beta_i^{(3)}$$

The $\beta_i^{(m)}$, $\delta_i^{(m)}$, and μ_i are computed to aid in the computation of the following functions and need not be stored after the computation of

$$\begin{aligned}
\lambda_i^{(1)} &= (\beta_i^{(4)}c_i^{(1)} - \beta_i^{(2)}c_i^{(3)})/\mu_i \\
\lambda_i^{(2)} &= (\beta_i^{(4)}c_i^{(2)} - \beta_i^{(2)}c_i^{(4)})/\mu_i \\
\lambda_i^{(3)} &= (\beta_i^{(1)}c_i^{(3)} - \beta_i^{(3)}c_i^{(1)})/\mu_i \\
\lambda_i^{(4)} &= (\beta_i^{(1)}c_i^{(4)} - \beta_i^{(3)}c_i^{(2)})/\mu_i
\end{aligned}$$

and

$$\begin{aligned}
\gamma_i^{(1)} &= (\beta_i^{(4)}\delta_i^{(1)} - \beta_i^{(2)}\delta_i^{(2)})/\mu_i \\
\gamma_i^{(2)} &= (\beta_i^{(1)}\delta_i^{(2)} - \beta_i^{(3)}\delta_i^{(1)})/\mu_i
\end{aligned}$$

The values of $\lambda_i^{(m)}$ and $\gamma_i^{(m)}$ must be stored, as they are used in the back solution. This is

$$\begin{aligned}
u_R &= \gamma_R^{(1)} \\
v_R &= \gamma_R^{(2)}
\end{aligned}$$

and

$$\begin{aligned}
u_i &= \gamma_i^{(1)} - \lambda_i^{(1)}u_{i+1} - \lambda_i^{(2)}v_{i+1} \\
v_i &= \gamma_i^{(2)} - \lambda_i^{(3)}u_{i+1} - \lambda_i^{(4)}v_{i+1}
\end{aligned}$$

$$\text{for } (R-1) \geq i \geq 1$$

Table 4. Definition of input parameters in DIFDEC2

Parameter	Example	Definition
D5	D_{Rn}	Diffusion coefficient of component u
D6	D_{Pb}	Diffusion coefficient of component v
U1	$[^{238}U_o]$	Initial concentration of component that produces u
L5	λ_{222}	Decay or reaction constant of u
L6	λ_{238}	Decay or reaction constant of reactant producing u
P	50	1/2 width of reacting layer
Q	100	Total length of diffusion field
R	13	Total number of space grid points (= number of space increments + 3)
T1	1.5	Time increment multiplied by this amount each step
T2	10^8	Total time allowed for simulation
T3	1000	Total number of time steps allowed

Table 5. Definition of internal variables in DIFDEC2

Program	Text	Definition
A1(I)	$a_j^{(1)}$	Coefficient in ditridiagonal equations
A2(I)	$a_j^{(2)}$	"
A3(I)	$a_j^{(3)}$	"
A4(I)	$a_j^{(4)}$	"
B1(I)	$b_j^{(1)}$	"
B2(I)	$b_j^{(2)}$	"
B3(I)	$b_j^{(3)}$	"
B4(I)	$b_j^{(4)}$	"
C1(I)	$c_j^{(1)}$	"
C2(I)	$c_j^{(2)}$	"
C3(I)	$c_j^{(3)}$	"
C4(I)	$c_j^{(4)}$	"
D1(I)	$c_1^{(1)}$	"
D2(I)	$c_j^{(2)}$	"
E1	$\beta_j^{(1)}$	Used in solution of ditridiagonal equations
E2	$\beta_j^{(2)}$	"
E3	$\beta_j^{(3)}$	"
E4	$\beta_j^{(4)}$	"
L1	$\lambda_j^{(1)}$	"
L2	$\lambda_j^{(2)}$	"
L3	$\lambda_j^{(3)}$	"
L4	$\lambda_j^{(4)}$	"
W1	$\delta_j^{(1)}$	Used in solution of ditridiagonal equations
W2	$\delta_j^{(2)}$	"
G1	$\gamma_j^{(1)}$	"

G2	$\gamma_i^{(2)}$	"
G3	$\gamma_i^{(3)}$	"
G4	$\gamma_i^{(4)}$	"
M1	μ_i	"
D7	D7	Larger of D5 and D6
E	E	Exponential term in equation (3)
S1	S1	Current step number
T	t	Elapsed time
T0	Δt	Time increment
T4	T4	Time at previous step
X0	Δx	Space increment
X2	$(\Delta x)^2$	Space increment squared