ULTRACENTRIFUGE SIMULATION USING CUBIC COLLOCATION

by

RICHARD S. DALE

B.S., Kansas State University, 1977

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1980

Approved:

Richard F. Sincovec
Major Professor

ACKNOWLEDGMENTS

I would like to thank Dr. David J. Cox and Dr. Richard F. Sincovec for their support and guidance through the course of this work. In addition I would like to thank my wife Debbie, without whose understanding and forbearance, this work would never have come to fruition.

This work was supported by a grant 6M-22243 from the National Institute of Health and by the College of Arts and Sciences, Kansas State University.

TABLE OF CONTENTS

INTRODUCTION

During the last fifteen years a number of methods for simulating the behavior of a sedimenting solute or system of solutes have been developed and presented in the literature. These include the distorted grid model of Cox(1-8), the countercurrent analogue of Bethune and Kegeles(9-10), the finite difference models of Dishon et. al. and Cann and Goad(11-13), and the finite element model of Claverie(14-16).

The method presented here offers three significant improvements over earlier models. It has automatic control over temporal integration errors. All terms of the continuity equation(s), including nonlinear coupling terms, are addressed simultaneously. The method has been implemented as an option in a sophisticated software package which is available for general distribution. For these reasons it deserves careful consideration.

Automatic control of temporal errors is an important feature of this method. It gives the user a great deal of control over errors which are committed in the course of the simulation. The time step is taken to be as large as is prudent in order to keep temporal integration errors below a specified maximum.

The fact that all terms of the continuity equation are addressed together makes the method extremely flexible. Models presented previously have generally required one or more intermediate steps in each step of time integration. Cox's model, for example, uses separate rounds of sedimentation and diffusion for each step of time integration. Claverie's model,

on the other hand, employs a sedimentation-diffusion operator but requires an intervening perturbation on the concentrations in order to accomplish relaxation to chemical equilibrium. Since all terms of the continuity equation are addressed together, the model presented here can account for virtually any physical effect that can be incorporated into the continuity equation(s).

The method presented here is available as an option in the code, PDECOL, which has been developed by Madsen and Sincovec(17). This code is available for a nominal distribution charge from the Association for Computing Machinery. PDECOL is currently in use in several hundred installations and no errors have been discovered in it since its release. Some penalty must, of course, be paid in overhead because of the generality of the methods employed in PDECOL. The user is relieved of such an extensive programming burden, however, that the overhead cost is nominal by comparison.

The method presented here is applicable to a wide range of transport experiments. It is presented in the context of centrifugation because this is the area in which the author has the most experience. It is anticipated that in the future this method will play an important role in research involving solutions of the continuity equations which govern transport phenomena.

CONTINUITY EQUATIONS

The continuity equation for the ultracentrifuge is a partial differential equation which governs the behavior of a sedimenting solute. In a fairly general form it is given by

$$c_{t_k} = \frac{1}{r}\frac{\partial}{\partial r}(-rJ_k) + f_k \ , \qquad 1 \le k \le K \ , \ t \ge 0 \ , \ r_m < r < r_b \ , \tag{1}$$

$$J_k = s_k\omega^2 r c_k - Dc_{r_k} \ . \tag{2}$$

Or, using a more general notation, by

$$c_{t_k} = L_k(r, t, \vec{c}, \vec{c}_r, \vec{c}_{rr}), \ 1 \le k \le K \ , \ t \ge 0, \ r_m < r < r_b \ , \tag{3}$$

where

r is the spatial variable,

t is the temporal variable,

r_m is the value of r at the meniscus,

r_b is the value of r at the cell bottom,

c_{t_k} denotes $\dfrac{\partial c_k}{\partial t}$

c_{r_k} denotes $\dfrac{\partial c_k}{\partial r}$

c_{rr_k} denotes $\dfrac{\partial^2 c_k}{\partial r^2}$

$\vec{c}$ denotes $(c_1, c_2, \ldots, c_K)$

$\vec{c}_r$ denotes $\left(\dfrac{\partial c_1}{\partial r}, \dfrac{\partial c_2}{\partial r}, \ldots, \dfrac{\partial c_K}{\partial r}\right)$

$\vec{c}_{rr}$ denotes $\left(\frac{\partial^2 c_1}{\partial r}, \frac{\partial^2 c_2}{\partial r}, \ldots, \frac{\partial^2 c_K}{\partial r}\right)$

J represents the flux,

K denotes the number of species present,

ω is the angular velocity and is a function of time t,

s_k and D_k are the sedimentation and diffusion coefficients respectively and are usually functions of $\vec{c}$,

f_k is a chemical coupling term and is a function of $\vec{c}$.

MATHEMATICAL MODEL

The method presented here uses a piecewise cubic collocation procedure for the discretization of the spatial variable, r. The collocation procedure reduces the original partial differential equation to a system of ordinary differential equations. The ODE system is an initial value system, which depends only on the temporal variable, t. Abstractions of this sort are generally referred to as the method of lines(18, 19, 20). The resulting system of ODE's is then integrated using fairly standard techniques discussed below.

One begins by partitioning the interval $[r_m, r_b]$ into N subintervals. These intervals need not be of uniform length. One chooses r_i, $i = 1, \ldots, N + 1$, so that any fine detail of the problem can be resolved. As a matter of notational convenience H_3 is defined to be the space of all functions which are continuous, continuously differentiable, and which are cubic on each of the subintervals $[r_i, r_{i+1}]$. This method uses members of H_3 to approximate the solutions, $c_k(r, t)$, to the continuity equations at any point in time.

A significant result from approximation theory(21) states that an arbitrary function, z, having four continuous derivatives can be approximated as a member of H within a maximum error of $(1/384)h^4 z^{(iv)}(\xi)$ where h is the maximum value of $(r_{i+1} - r_i)$, and $r_1 \leq \xi \leq r_{N+1}$.

For a relatively smooth function this is an excellent approximation. The high order of accuracy of this approximation is the key to achieving good accuracy using a relatively coarse grid.

A basis for H_3 consists of $2N + 2$ linearly independent basis functions, g_i . These basis functions are sometimes referred to as elements. Each of the $2N - 2$ interior basis functions is nonzero on precisely one double interval, $[r_{i-1}, r_{i+1}]$. That is, g_{2i-1} and g_{2i}, $i = 2, \ldots, N$, are nonzero on the interval $[r_{i-1}, r_{i+1}]$ and zero elsewhere. g_1 and g_2 are nonzero on the interval $[r_1, r_2]$. g_{2N+1} and g_{2N+2} are nonzero on the interval $[r_N, r_{N+1}]$.

The principal assumption of this method is that at any time t , each component of the solution to equation (1) can be approximated by a member of H_3 . Thus the approximate solution is given by

$$c_k(r, t) = \sum_{i=1}^{2N+2} y_{i,k}(t) g_i(r) , \tag{4}$$

where the $y_{i,k}$ are scalars which depend only on time t , and the g_i depend only on r .

The ordinary differential equations are obtained by requiring the approximate c_k in equation (4) to satisfy the continuity equation(s), (1), at a set of $2N$ collocation points p_j . The way in which the p_j are chosen is very important. As might be expected $p_1 = r_m$ and $p_{aN+2} = r_b$. The choice of the interior collocation points is considerably more subtle.

Theory(22) predicts that a dramatic increase in the accuracy of the solution can be obtained by collocating at the roots of the second degree Legendre polynomial in each subinterval $[r_i, r_{i+1}]$. These points are given by

$$p_{2i} = r_i + (r_{i+1} - r_i)\left(\frac{1}{2} - \frac{1}{2\sqrt{3}}\right) ,$$

$$p_{2i+1} = r_i + (r_{i+1} - r_i)\left(\frac{1}{2} + \frac{1}{2\sqrt{3}}\right) . \tag{5}$$

Using equations (5) and substituting equation (4) into equation (3), and requiring equation (3) to be valid at the interior collocation points gives

$$\sum_{i=1}^{2N+2} g_i(p_j) \frac{dy_{i,k}}{dt} = L_k(t, p_j, c(t, p_j), c_r(t, p_j), c_{rr}(t, p_j)) . \qquad (6)$$

To determine equations corresponding to $j = 1$ and $j = 2N + 2$ the boundary conditions are imposed. At the left boundary the usual zero flux condition is used. For a fairly general case the boundary condition for the kth component is given by

$$s_k \omega^2 r_m c_k - D_k c_{r_k} = 0 , \qquad (7)$$

where s_k or D_k or both are usually functions of c, and ω is considered to be a function of t whose initial value is zero. Acceleration must be simulated because of the way that the boundary conditions are introduced. No provision is made for boundary conditions that are inconsistent with initial conditions. The initial distribution of mass in a centrifuge cell is generally uniform. Each c_{r_k} is therefore initially zero everywhere. In particular it is zero at the meniscus. It follows that equation (7) can only be satisfied initially if the first term is initially zero. This is accomplished by introducing acceleration into the simulation. Rearranging (7) gives

$$\frac{D_k c_{r_k}}{s_k c_k r_m} = \omega^2 \qquad (8)$$

For notational expedience we define

$$B_k(c, c_r) = \frac{D_k c_{r_k}}{s_k c_k r_m} \qquad (9)$$

Using this notation and employing the chain rule to differentiate both sides of equation (8) with respect to t gives

$$\sum_{j=1}^{K} \left\{ \frac{\partial B_k}{\partial c_j} \frac{\partial c_j}{\partial t} + \frac{\partial B_k}{\partial c_{r_j}} \frac{\partial c_{r_j}}{\partial t} \right\} = \frac{d(\omega^2)}{dt} \tag{10}$$

The basis functions g_1 and g_2 are the only basis functions which have nonzero derivatives at r_m. In addition to this only g_1 has a nonzero value at r_m. Using these facts the appropriate ODE is easily seen to be

$$\sum_{j=1}^{K} \left\{ \frac{\partial B_k}{\partial c_j} g_1(r_m) + \frac{\partial B_k}{\partial c_{r_j}} g_1'(r_m) \right\} \frac{dy_{1,j}}{dt} + \sum_{j=1}^{K} \left\{ \frac{\partial B_k}{\partial c_{r_j}} g_2'(r_m) \right\} \frac{dy_{2,j}}{dt}$$

$$= \frac{d(\omega^2)}{dt} . \tag{11}$$

It seems quite reasonable to employ an analogous method to construct the ODE corresponding to the boundary condition at the cell bottom, r_b. This is, however, not recommended. The reasons for this will be discussed in a subsequent section. Instead the so called "free end" condition is imposed. This is accomplished by collocating the differential equations in the usual manner at r_b.

Combining the system of equations (6) with the boundary condition information yields a system of $K(2N + 2)$ initial value ordinary differential equations. This system of ODE's is dependent only on the temporal variable, t. The system of equations has the form

$$A\frac{\vec{dy}}{dt} = \vec{g}(y, t) \tag{12}$$

The matrix A is banded, having maximum bandwidth of $6K - 1$. This is because of the very local nature of the basis functions used. The banded nature of A expedites the solution of the system of equations (12). With the exception of the rows corresponding to boundary condition information at the meniscus, the entries of A are simply basis function values at collocation points.

ODE INTEGRATION

Having fully discretized the original PDE problem with respect to the spatial variable, r, it is necessary to integrate the resulting system of ODE's. This is accomplished by using the stiffly stable methods of Gear(23). (Some authors refer to these methods as backward difference formulas.) Gear's methods are indicated because of their good stability properties when one or more terms of an ODE problem is decaying to an asymptotic value. They are based on the predictor formula

$$y_n = \sum_{i=1}^{m} \alpha_i y_{n-i} + \eta_1 \Delta t y'_{n-1} \quad , \quad 1 \leq m \leq 5 \quad , \tag{13}$$

and the corrector formula

$$y_n = \sum_{i=1}^{m} \alpha_i^* y_{n-i} + \eta_0 \Delta t y'_n \quad , \quad 1 \leq m \leq 5 \quad . \tag{14}$$

The integer m is the order of the method. The α's and η's are coefficients which define the method.

Predictor-corrector methods have several important properties. The error committed in each step can be easily estimated. The stepsize, Δt, can be adjusted accordingly. And, when indicated, the order of the method can be easily changed. For these reasons ODE integrators using these methods give excellent performance while maintaining the integration errors committed at each step below a pre-specified level.

A bound, EPS, is chosen as the relative error tolerance for the integration. EPS is used in three different places. It is used to test convergence of the corrector equation. It is a bound on the estimated error calculated

after each step. And it is employed as a parameter for determining what change of order and stepsize is appropriate for the subsequent step.

The corrector equation is implicit. It includes both the value of the function y and its derivative y', at the new time, t. The corrector equation cannot, therefore, be solved explicitly. It must instead be solved by an iterative technique. A modified Newton's method is used for this iteration(24). Following each iteration the L_2 norm of the weighted differences in the values of y' are compared with a scaled value of EPS. (The L_2 norm is sometimes called the root-mean-square norm.) EPS is scaled to account for the number of ODE's and the current order of the integration method. Each difference is weighted by dividing by the maximum absolute value which the corresponding y has previously attained. If the value of the norm is smaller than the scaled value of EPS the iteration is considered to have converged.

After the corrector has converged, an estimate is made of the error in each of the ODE's. This estimate is obtained by taking the weighted difference in the predicted and corrected values of each component. The L_2 norm of these estimates is compared with a different but similarly scaled value of EPS. If the norm is smaller than EPS the step is accepted. If it is greater than EPS the step is rejected, and the stepsize Δt is reduced.

Following each step EPS is employed as a parameter in selecting the stepsize and order to be used on the subsequent step. Estimates are made for the current order m and, when appropriate, for orders $m + 1$ and $m - 1$ (23). A choice is then made regarding whether the order and/or stepsize should be increased, decreased, or retained.

The formulas, (13) and (14), are not used explicitly. The formulas used are called the "normal form" of Gear's methods(23, 26). The values of

y and y' in equations (13) and (14) define a polynomial. These polynomials can be equally well represented by the terms of a Taylor series. The normal form of Gear's methods uses the Taylor series representation of the extrapolating polynomial. This simplifies both the prediction process and changes of stepsize.

The time integration techniques described above are quite efficient. The automatic choice of stepsize and order are extremely useful features. The use of these techniques frees the user from the burden of determining the Δt which is appropriate for the particular problem at hand. Moreover, choosing and changing Δt dynamically keeps the integration from being bounded by a stepsize that is suitable for one phase of the problem, but represents unnecessary and inefficient overkill for another phase. The user must of course choose a suitable value of EPS for his problem.

IMPLEMENTATION

The procedures outlined above are available as options in the FORTRAN IV code PDECOL which has been developed by N. K. Madsen and R. F. Sincovec(17). This is an extremely powerful package which implements collocation techniques for a large class of PDE problems. The user is allowed to specify the order of the polynomial space used, the number of continuity conditions to be satisfied by the space, and a temporal integration technique.

The user is required to provide subroutines which specify the initial conditions, the boundary conditions, and the value of c_t given r, t, $\vec{c}$, $\vec{c}_r$, $\vec{c}_{rr}$. In addition a driving routine is required. The driving routine has several responsibilities. It must set the values of parameters used by PDECOL. It must call the package. It should incorporate logic for handling any error returns from PDECOL. It is responsible for the output of the results.

The manner in which boundary conditions are implemented in PDECOL requires that the initial and boundary conditions agree. For this reason, it is necessary to provide some means of interpolating ω during the interval between the initial time t_0 and the time t_1 when the centrifuge rotor achieves its terminal velocity. The recommended procedure is the construction of the cubic spline interpolate of $\omega(t)$ between the times t_0 and t_1. This is a very smooth interpolating function. The derivative, ω', is taken to be zero at the times t_0 and t_1. The interpolate has two continuous derivatives. Use of this interpolation method allows the user to incorporate data recorded during the acceleration phase of a centrifuge experiment.

Three subroutines are given in Conte and de Boor(21) which construct and evaluate the cubic spline interpolate. These can be used with only minor modification.

During early phases of testing this method some problems arose. It was discovered that the numerical solution could become unstable in the regions corresponding to the meniscus and the cell bottom. The reasons for the development of the instabilities are quite different. The instability at the bottom of the cell is a consequence of the spatial descretization. Near the bottom of the centrifuge cell the buildup of mass corresponds to an abrupt and severe rise in the values of the concentration and its spatial derivatives. As a consequence, the curvature of the cubic function approximating the solution over this interval is forced to become large. As is well known, if the curvature of a cubic is very large, it will have a maximum or a minimum nearby. When these maxima and minima appear they result in the development of violent oscillations in the numerical solution in the region near the bottom of the simulated cell. Since accurate answers in this region are generally of little tangible benefit, the usual zero flux condition at r_b has been abandoned in favor of the null boundary condition. This corresponds physically to a semi-infinite cell. PDECOL allows the user to specify the null boundary condition at one or both boundaries. When this option is invoked, PDECOL collocates the differential equation in the usual manner at the point involved.

The problems encountered at the boundary corresponding to the meniscus are much more subtle. They arise as a consequence of the error committed during the early steps of the integration. The zero flux condition at the boundary is given by equation (7). The errors committed initially are kept very small by the error control mechanisms in the temporal integration method.

There are, however, errors committed. The resulting instability can be visualized if one interprets the error physically. Rewriting equation (7) to account for the error gives

$$S\omega^2 rc - Dc_r = d \tag{15}$$

(The balance of this article is confined to the case of a single species.) This corresponds to a small source (if d is positive) or sink (if d is negative) of mass at the meniscus. The size of the discrepancy, d, is initially very small. However if d is positive the resulting influx of mass into the system will tend to make c slightly larger and c_r slightly smaller than the correct values in the neighborhood of r_m. As a consequence of these effects d will tend to be slightly larger on the following time step.

The case of a negative d results in a similar "widening gap" between the first and second terms of equation (15). The outflow of mass through the top of the simulated solution column results in a tendancy to increase c_r and decrease c. Once again the discrepancy tends to become larger on the subsequent step. As the simulation proceeds $|d|$ becomes larger than can be accounted for by the accumulation of truncation errors. When the smaller of the two left hand terms of equation (15) is close to zero the larger can still have a significant positive value. Because of this, the smaller term goes straight through zero and takes up residence at some negative value.

Things begin to get bad. With opposite signs on the two terms of the left side of equation (15), the growth of the discrepancy is accelerated. Serious artifacts begin to appear. In the case where c has gone through zero, negative concentrations appear and are propogated along the radial

axis by the sedimentation term in the continuity equation. If the sedimentation term includes a correction for concentration dependent sedimentation, the negative concentrations will be propogated straight into the sedimentation boundary. The already suspect solution is ruined. In the case where the derivative c_r goes through zero, the concentration, c, becomes extremely large at r_m and this enormous excess of mass is propogated along the solution column.

Several schemes for eliminating these artifacts from the simulation were tested before settling on one that works best. The ODE's involved in the solution at the meniscus were given greater weight in the error estimates used in PDECOL. An attempt was made to attenuate the residual error d in the subroutine that defines the boundary conditions. When the concentration had reached some threshold value, the boundary was released, and the null boundary condition was imposed. None of these schemes was completely successful.

The error estimates in PDECOL are all weighted by dividing the estimated error for each y_i by the largest absolute value which y_i has previously attained. The place in PDECOL where these weights are set is well documented and quite easy to change. The weights on y_1 and y_2 were changed to $.01$. This resulted in a reduction in the size of the artifacts and postponed the development of severe instabilities. It did not, however, eliminate the artifacts entirely. The effectiveness of this scheme was, moreover, not very general. Although nearly acceptable results were seen in some cases, other cases would persist in showing artifacts that became quite large. This scheme also degraded the performance of the model.

The attempt to attenuate d in the subroutine BNDRY which is responsible for conveying boundary condition information to PDECOL was completely

unsuccessful. The differential equation defining the behavior of the solute at the meniscus was taken to be

$$\frac{d}{dt}\left(s\omega^2 rc - Dc_r\right) = -\frac{1}{a}\left(s\omega^2 rc - Dc_r\right) \tag{16}$$

where a was the last Δt used successfully. In this case PDECOL was unable to achieve convergence in the corrector part of the temporal integration.

Another scheme involved releasing the boundary after the concentration had fallen below a prespecified threshold. The basis of this scheme was to impose the null boundary condition when the concentration had fallen to a relatively insignificant value. The idea was that, freed from the constraint of the boundary condition, the values of c and c_r would fall to zero together. This did not prove to be true. Although this scheme did stabilize the solution, it did so erroneously.

After a period of considerable vexation an effective solution was discovered. The best scheme for stabilizing the solution at the meniscus is to force the boundary condition to be satisfied. This is accomplished by modifying the core integrator of PDECOL so that the discrepancy, d, is forced to be no larger than the roundoff error of the machine used. After the corrector iteration has converged, a correction is made on the values of the ODE's that are involved with the boundary conditions. As a precaution the correction is incorporated into the error control mechanism responsible for accepting or rejecting a step.

The correction is based on a knowledge of g_1 and g_2 at the meniscus. For $i \geq 3$, g_i and g'_i are identically zero at r_m. The values of g_1 and g_2 at r_m are 1 and 0 respectively. The values of g'_1 and g'_2 are $-3/(r_2 - r_1)$ and $3/(r_2 - r_1)$ respectively. Armed with this information

and the value of y_1 it is a relatively easy matter to calculate y_2 so that the discrepancy, d, will be zero within the precision of the computer. The correction is made on y because this greatly simplifies the algebra involved. It is not known whether a given correction pushes the value of the numerical solution closer to or further from the true solution. Because of this it is important to incorporate the correction into the error control used by PDECOL.

As might be expected, the largest time integration errors occur in the region corresponding to the sedimentation boundary. During the early phases of the integration, this region corresponds to the region near r_m. The magnitude of the correction, and of the estimated error on y_1, y_2, y_3, and y_4 are quite comparable during this phase. If $s\omega^2/D$ is sufficiently large, c, c_r, and the error estimates fall to virtually negligible levels at the later stages. The size of the correction in the later stages is generally several orders of magnitude smaller than that of the estimated errors. The modification to PDECOL implementing this correction is considered to be of considerable importance. For this reason an example of this correction for a fairly general case is given verbatim in the appendix. It is accompanied by enough of the original code to allow the interested reader to employ it in his own research without inordinate attention to the internal workings of the package.

A suitable choice of grid spacing and EPS are required for effective use of PDECOL. Although PDECOL controls time integration errors, it has no control over errors resulting from the choice of grid spacing. The grid spacing must be chosen, therefore, with some care. For cases where $s\omega^2/D$ is much larger than about 80, a non-uniform grid is recommended. This grid should have a higher density of points in the region near r_m. This facilitates the approximation of the steeper, narrower sedimentation boundary in

this region. The gradient profile corresponding to larger values of $s\omega^2/D$ is narrower and steeper everywhere than for smaller values of this parameter. The density of the grid spacing should reflect this distinction if reliable results are to be obtained.

The choice of EPS is also quite important. PDECOL uses the L_2 norm of the estimated error in each component of the ODE system for calculations involving error estimates. Since time integration errors are much larger in the region corresponding to the sedimentation boundary, errors much larger than EPS are allowed to occur in this region. This effect becomes pronounced when the breadth of the sedimentation boundary is limited. Furthermore, the finer grid demanded by larger values of $s\omega^2/D$ places a smaller upper bound on Δt . This results in more time steps being required to "push" the sedimentation boundary along the grid. For example, when $s\omega^2/D$ is about 10 , excellent results can be obtained using 1.E-4 for EPS and an evenly spaced grid consisting of 50 subintervals. When $s\omega^2/D$ is about 80 a grid of 100 subintervals and an EPS of 1.E-6 are more appropriate. These values of EPS are sufficiently small to allow the errors in the solutions to be dominated by the errors due to the spatial discretization. It should be pointed out that the simulation in the second case will take three to four times longer to execute.

TEST OF VALIDITY

For the simple case where only one species is present and the values of s and D are constant, the flux, J, is given by

$$J = s\omega^2 rc - Dc_r \ .$$

The validity of the method is determined by considering the data generated during the course of the simulation to be "experimental" data. Observed sedimentation and diffusion coefficients are calculated using the equations

$$\ln(r_*) = q + s_*\omega^2 t \ , \text{ and}$$

$$(\frac{A}{H})^2 = D_*(\frac{2\pi}{s_*\omega^2})[\exp(2s_*\omega^2 t - 1)] \qquad (17)$$

r_* is the point where the gradient achieves its maximum.

q is a constant.

s_* is the calculated sedimentation coefficient.

ω is the angular velocity of the rotor.

t is the temporal variable.

A is the area under the gradient and is equivalent to the plateau concentration.

H is the maximum height of the sedimentation boundary gradient.

D_* is the calculated diffusion coefficient. (25)

Values for s_* and D_* are calculated using the method of linear least squares. The values of s and D used for input into the simulation correspond

roughly to a macromolecule having a molecular weight of 75,000 .

s = 5.25E-13

D = 6.3E-7

N = 50

The results of these calculations are given in Table 1. The accuracy of these findings is quite striking. The results regarding the diffusion coefficients in particular are an order of magnitude better than Claverie has reported for his model(14). The computation time is of course a function of the machine used. For purposes of comparison a Cox model on the computer used executes in about 20 seconds whereas a Claverie model requires several minutes.

TABLE 1

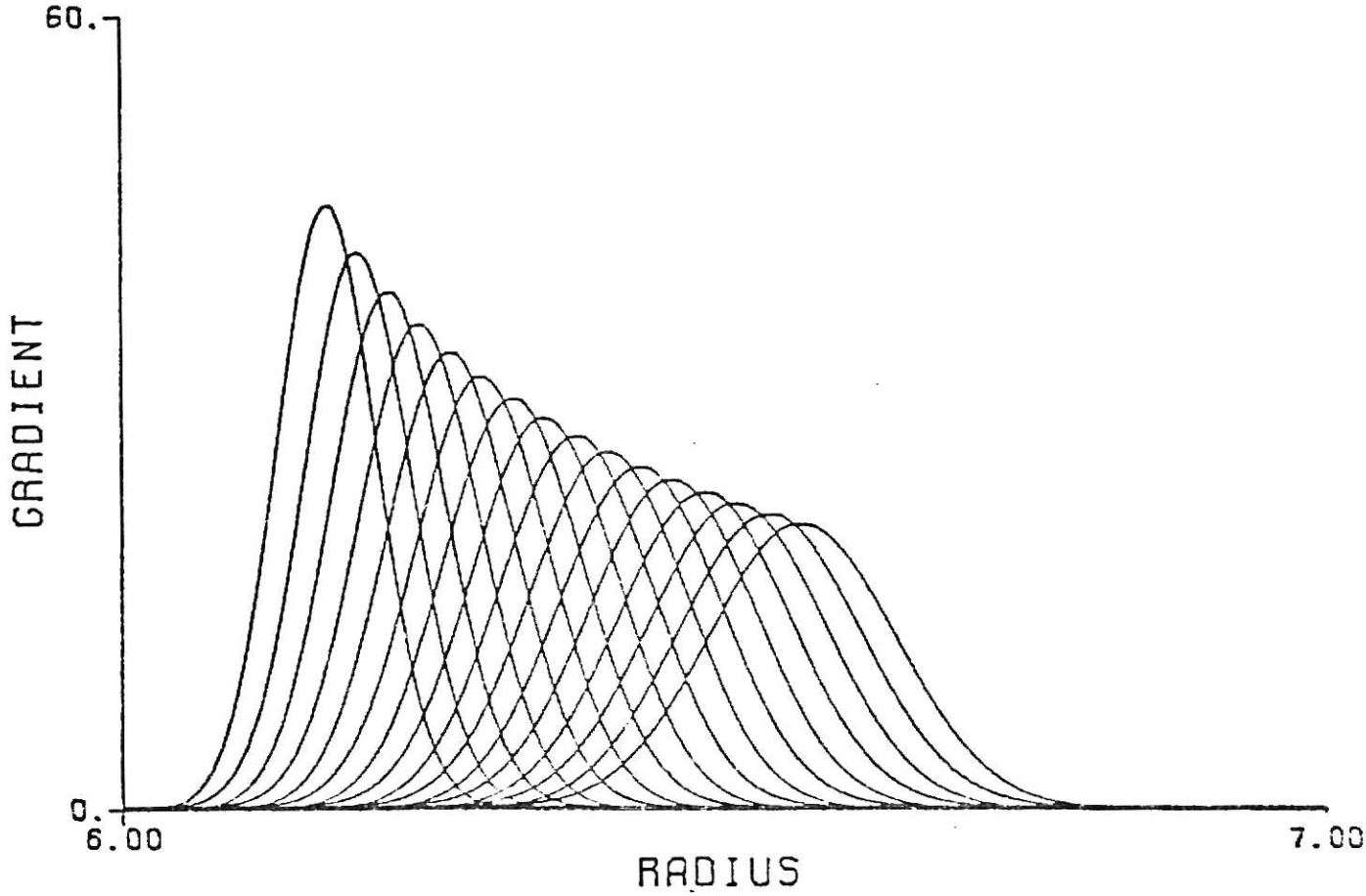
EPS	s	$\lvert s-s_* \rvert$ (%)	D	$\lvert D-D_* \rvert$ (%)	Computation time
1.E-4	5.262E-13	0.23%	6.264E-7	0.58%	37 seconds
1.E-6	5.248E-13	0.05%	6.273E-7	0.42%	53 seconds

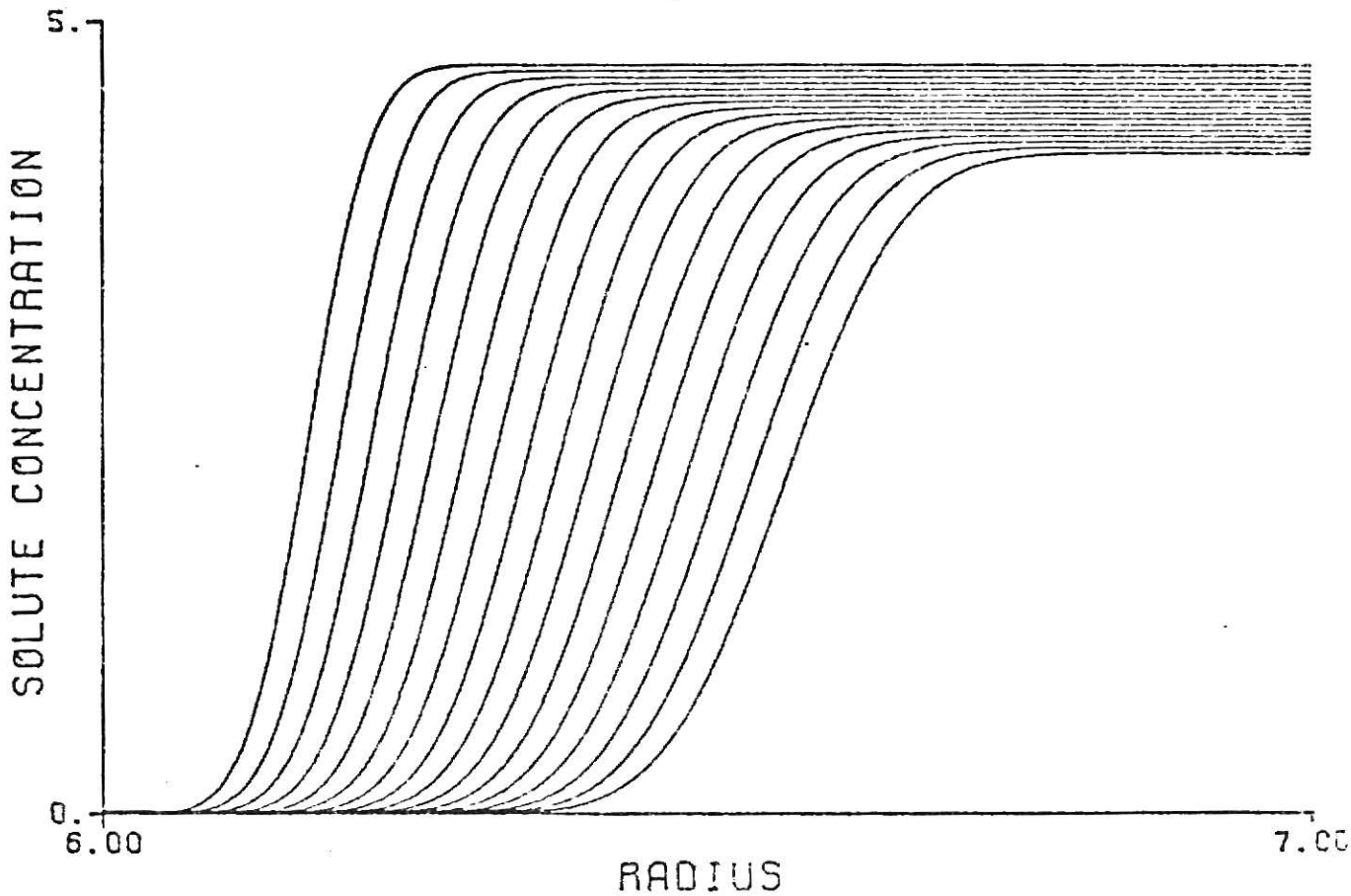
CONCLUSIONS

Collocation represents a considerable advance in the techniques available for the simulation of velocity sedimentation experiments. The extreme flexibility of the method makes it suitable for cases where the sedimentation is accompanied by chemical interaction of the sedimenting species. The high degree of accuracy that can be obtained at relatively low cost for low to moderate values of $s\omega^2/D$ makes it the method of choice in the case of small macromolecules where the shape of the sedimentation boundary is affected by the meniscus.

PDECOL is a useful tool for implementing collocation methods. Using the modification given in the appendix (or a similar modification) one can construct simulations for a variety of ultracentrifuge experiments with remarkably little effort.

FIGURE 1


60.
GRADIENT
0.
6.00
7.00
RADIUS


5.
SOLUTE CONCENTRATION
0.
6.00
7.00
RADIUS

LEGEND FOR FIGURE 1

Figure 1. Simulation of a boundary sedimentation experiment. The parameters s and D are the same as in Table 1. Rotor speed is given by a cubic accelerating from 0 to 60,000 RPM in 90 seconds. r_m and r_b are 6.00 cm and 7.00 cm respectively. These curves show the distribution of the concentration, c , and its gradient, $\frac{\partial c}{\partial r}$, at successive times. The first curve corresponds to a time of 1400 seconds. The interval between observations is 200 seconds.

REFERENCES

1. D. J. Cox, Arch. Biochem. Biophys. 112, 249 (1965)
2. D. J. Cox, Arch. Biochem. Biophys. 112, 259 (1965)
3. D. J. Cox, Arch. Biochem. Biophys. 119, 230 (1967)
4. D. J. Cox, Arch. Biochem. Biophys. 129, 106 (1969)
5. D. J. Cox, Arch. Biochem. Biophys. 142, 512 (1971)
6. D. J. Cox, Arch. Biochem. Biophys. 146, 181 (1971)
7. R. R. Holloway and D. J. Cox, Arch. Biochem. Biophys. 160, 595 (1974)
8. D. J. Cox, Meth. Enzymol., 48, 212 (1978)
9. J. L. Bethune and G. Kegeles, J. Phys. Chem. 65, 1761 (1961)
10. G. Kegeles and M. L. Johnson, Biochemistry 6, 796 (1967)
11. J. R. Cann and W. B. Goad, J. Biol. Chem. 240, 1948 (1965)
12. J. R. Cann, "Interacting Molecules", Academic Press, New York, N.Y. (1970)
13. M. Dishon, G. H. Weiss, and D. A. Yphantis, Biopolymers 4, 449 (1966)
14. J.-M. Claverie, H. Dreux, and R. Cohen, Biopolymers 14, 1685 (1975)
15. R. Cohen and J.-M. Claverie Biopolymers, 14, 1701 (1975)
16. J.-M. Claverie, Biopolymers 15, 843 (1976)
17. N. K. Madsen and R. F. Sincovec, ACM Trans. on Math. Software 5, 326 (1979)
18. N. K. Madsen and R. F. Sincovec, "Computational Methods in Nonlinear Mechanics", J. T. Oden et al., Eds., Texas Inst. for Computational Mechanics, Austin, Texas (1975)
19. N. K. Madsen and R. F. Sincovec, "Proceedings of the 1975 Summer Computer Simulation Conference", San Francisco, California (1975)
20. R. F. Sincovec and N. K. Madsen, ACM Trans. on Math. Software 1, 261 (1975)
21. S. D. Conte and C. de Boor, "Elementary Numerical Analysis", McGraw Hill, (1972)

22. C. de Boor and B. Swartz, SIAM J. Num. Anal. 10, 582 (1973)

23. C. W. Gear, "Numerical Initial Value Problems in Ordinary Differential Equations", Prentice Hall, Englewood Cliffs, N. J. (1971)

24. F. S. Acton, "Numerical Methods that Work", Harper and Row, New York, N.Y. (1970)

25. H. Fujita, "Mathematical Theory of Sedimentation Analysis", Acedemic Press, New York, N.Y., (1962)

26. C. W. Gear, Communications of the ACM 14, 176 (1971)

```
C-----------------------------------------------------------------------COL17550
C THE CORRECTOR HAS CONVERGED.  IWEVAL IS SET TO -1 TO SIGNAL           COL17560
C THAT PW MAY NEED UPDATING ON SUBSEQUENT STEPS.  THE ERROR TEST        COL17570
C IS MADE AND CONTROL PASSES TO STATEMENT 500 IF IT FAILS.              COL17580
C-----------------------------------------------------------------------COL17590
 450  IWEVAL = -1                                                       COL17600
      NFE = NFE + M                                                     COL17610
C*********************************************************************AMEND
C                                                                       AMEND
C     THE MODIFICATION TO PDECOL BEGINS HERE.                           AMEND
C     Y(I+(K-1)*NCPTS,J) CONTAINS THE (J-1)TH SCALED DERIVATIVE OF THE  AMEND
C     ODE GOVERNING THE ITH BASIS FUNCTION OF THE KTH SPECIES.          AMEND
C                                                                       AMEND
C     ERROR(I)*EL(1) IS THE DIFFERENCE BETWEEN THE PREDICTED AND        AMEND
C     CORRECTED VALUES OF Y(I,1).                                       AMEND
C                                                                       AMEND
C     THE CORRECTION THAT IS ILLUSTRATED HERE IS FOR THE CASE           AMEND
C     IN WHICH THERE ARE NPDE DIFFERENT SPECIES PRESENT.                AMEND
C     RMEN IS THE RADIUS OF THE MENISCUS.                               AMEND
C     DX1 IS THE LENGTH OF THE FIRST SUBINTERVAL.                       AMEND
C     DIFFUS(I) IS THE DIFFUSION COEFFICIENT FOR THE ITH SPECIES.       AMEND
C     THE SEDIMENTATION COEFFICIENT IS TAKEN TO BE A FUNCTION           AMEND
C     OF THE CONCENTRATIONS OF ALL THE SPECIES PRESENT.                 AMEND
C     THE NONLINEAR FORM OF THE CONCENTRATION DEPENDANCE IS USED.       AMEND
C                                                                       AMEND
C     THE FORM USED IS                                                  AMEND
C                                                                       AMEND
C     S(I)=S0(I)/(1. + SUM(HDK(I,J) * C(J))                            AMEND
C                                                                       AMEND
C     HDK(I,J) IS THE CONSTANT THAT REPRESENTS THE INFLUENCE OF         AMEND
C     THE CONCENTRATION OF THE JTH SPECIES ON THE SEDIMENTATION         AMEND
C     COEFFICIENT OF THE ITH SPECIES.                                   AMEND
C     SOMSQ0(I) IS THE UNCORRECTED (FOR CONCENTRATION DEPENDANCE)       AMEND
C     SEDIMENTATION COEFFICIENT OF THE ITH SPECIES MULTIPLIED BY        AMEND
C     THE SQUARE OF THE ANGULAR VELOCITY, OMEGA.                        AMEND
C     SOMSQ0, DIFFUS, HDK, DX1, AND RMEN HAVE BEEN PASSED TO THE        AMEND
C     SUBROUTINE STIFIB IN A COMMON AREA WHICH WAS CREATED FOR          AMEND
C     THAT PURPOSE.                                                     AMEND
C                                                                       AMEND
C     Y1 IS THE NEW (IF THE STEP IS ACCEPTED)  COEFFICIENT FOR          AMEND
C     THE FIRST BASIS FUNCTION.                                         AMEND
C     Y2 IS THE VALUE WHICH IS BEING FORCED ONTO THE COEFFICIENT FOR     AMEND
C     SECOND BASIS FUNCTION IN ORDER TO ASSURE THAT THE BOUNDARY        AMEND
C     CONDITION IS SATISFIED.                                           AMEND
C                                                                       AMEND
C*********************************************************************AMEND
      DO  455  I=1,NPDE                                                 AMEND
       I1=1+(I-1)*NCPTS                                                 AMEND
       I2=I1+1                                                          AMEND
       HSUM=1.D0                                                        AMEND
       DO 4550 J=1,NPDE                                                 AMEND
        J1=1+(J-1)*NCPTS                                                AMEND
```

```
 4550   HSUM=HSUM+HDK(I,J)*(Y(J1,1)+EL(1)*ERROR(J1))                    AMEND
       ERROLD=ERROR(I2)                                                 AMEND
       Y1=Y(I1)+EL(1)*ERROR(I1)                                         AMEND
       Y2=Y1*(SOMSQO(I)*RMEN/HSUM+3.D0*DIFFUS(I)/DX1)*DX1/3.D0/DIFFUS(I)AMEND
       ERROR(I2)=(Y(I2,1)-Y2)/EL(1)                                     AMEND
       CORSUM(I)=ERROR(I2)-ERROLD                                       AMEND
  455 CONTINUE                                                          AMEND
C*********************************************************************AMEND
C     ERROR(2+M*NCPTS)  M=0,...,NPDE-1   HAS BEEN CHANGED TO           AMEND
C     FORCE THE BOUNDARY CONDITIONS TO BE SATISFIED                     AMEND
C*********************************************************************AMEND
      D = 0.                                                            COL17620
      DO 460 I = 1,N                                                    COL17630
  460   D = D + (ERROR(I)/YMAX(I))**2                                   COL17640
C*********************************************************************AMEND
C     THE FOUR LINES OF CODE THAT FOLLOW INCORPORATE THE CHANGE         AMEND
C     MADE INTO THE ERROR CONTROL OF PDECOL                             AMEND
C*********************************************************************AMEND
      DO 465 I=1,NPDE                                                   AMEND
       I2=2+(I-1)*NCPTS                                                 AMEND
       D=D+(CORSUM(I)/YMAX(I2))**2                                      AMEND
  465 CONTINUE                                                          AMEND
      IF (D .GT. E) GO TO 500                                           COL17650
C---------------------------------------------------------------------COL17660
C AFTER A SUCCESSFUL STEP, UPDATE THE Y ARRAY.                          COL17670
C CONSIDER CHANGING H IF IDOUB = 1.  OTHERWISE DECREASE IDOUB BY 1.     COL17680
C IF IDOUB IS THEN 1 AND NQ .LT. MAXDER, THEN ERROR IS SAVED FOR        COL17690
C USE IN A POSSIBLE ORDER INCREASE ON THE NEXT STEP.                    COL17700
C IF A CHANGE IN H IS CONSIDERED, AN INCREASE OR DECREASE IN ORDER      COL17710
C BY ONE IS CONSIDERED ALSO.  A CHANGE IN H IS MADE ONLY IF IT IS BY A  COL17720
C FACTOR OF AT LEAST 1.1.  IF NOT, IDOUB IS SET TO 10 TO PREVENT        COL17730
C TESTING FOR THAT MANY STEPS.                                          COL17740
C---------------------------------------------------------------------COL17750
      KFLAG = 0                                                         COL17760
      IREDO = 0                                                         COL17770
      NSTEP = NSTEP + 1                                                 COL17780
      HUSED = H                                                         COL17790
      NQUSED = NQ                                                       COL17800
      DO 470 J = 1,L                                                    COL17810
        DO 470 I = 1,N                                                  COL17820
  470     Y(I,J) = Y(I,J) + EL(J)*ERROR(I)                              COL17830
      IF (IDOUB .EQ. 1) GO TO 520                                       COL17840
      IDOUB = IDOUB - 1                                                 COL17850
      IF (IDOUB .GT. 1) GO TO 700                                       COL17860
      IF (L .EQ. LMAX) GO TO 700                                        COL17870
      DO 490 I = 1,N                                                    COL17880
  490   Y(I,LMAX) = ERROR(I)                                            COL17890
      GO TO 700                                                         COL17900
C---------------------------------------------------------------------COL17910
C THE ERROR TEST FAILED.  KFLAG KEEPS TRACK OF MULTIPLE FAILURES.       COL17920
C RESTORE T AND THE Y ARRAY TO THEIR PREVIOUS VALUES, AND PREPARE       COL17930
C TO TRY THE STEP AGAIN.  COMPUTE THE OPTIMUM STEP SIZE FOR THIS OR     COL17940
C ONE LOWER ORDER.                                                      COL17950
C---------------------------------------------------------------------COL17960
```

ULTRACENTRIFUGE SIMULATION USING CUBIC COLLOCATION

by

RICHARD S. DALE

B.S., Kansas State University, 1977

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1980

ABSTRACT

This paper briefly describes the development of cubic collocation techniques for solving the continuity equations for the ultracentrifuge. An extremely flexible method is presented, taking into account cell geometry, the effect of the meniscus on the sedimentation boundary, chemical kinetics, etc. A powerful, publicly available software package is introduced and its utility in implementing the method described here is discussed. Results are given for a simple but typical case.