JOURNAL OF COMPUTATIONAL PHYSICS 23, 53–62 (1977)

Comparison of Numerical Methods for Renal Network Flows

R. MEJIA,* R. B. KELLOGG,† AND J. L. STEPHENSON*

** Section on Theoretical Biophysics, National Heart and Lung Institute and Mathematical Research Branch, National Institute of Arthritis, Metabolism, and Digestive Diseases, National Institutes of Health, Bethesda, Maryland 20014, † Institute of Fluid Dynamics and Applied Mathematics, University of Maryland, College Park, Maryland 20742*

Received April 4, 1976

A comparison is made of the accuracy, iterative properties, storage, and cpu time of several procedures for solving the nonlinear differential equations which describe the renal concentrating mechanism. In two procedures, Newton's method is used to solve a nonlinear system of finite difference equations. The first employs a generalized linear equation solver, the second a specialized solver. In other procedures used, the unknowns of the problem are divided into two groups: The first group consists of the unknowns in each tube, and the second consists of the unknowns which link the tubes with each other. In conclusion, a set of guidelines for solving nonlinear fluid network problems is given.

1. INTRODUCTION

The operating mechanism of the kidney is often explained in terms of a flow network, in which a fluid containing several solutes is transported through a system of tubes [1–3]. Each tube may interchange fluid or solutes, along its length, with other tubes through a common interstitial space. This interchange is caused by osmotic and hydrostatic pressure and electrochemical driving forces acting through the tube walls. The model leads to a system of differential equations describing the network flows. It is important to construct accurate and efficient algorithms for the numerical solution of these equations. The purpose of this paper is to describe our experience with several numerical schemes for solving the kidney equations. At the end we state some general conclusions which, it is hoped, will be applicable to other network flow problems.

2. THE DIFFERENTIAL EQUATIONS

We consider a family of parallel tubes, each tube extending from $x = 0$ to $x = 1$. Some of the tubes are connected with one another at $x = 1$. A mixture of water and K solutes is flowing in each tube. The tubes are placed in a common interstitial water–solute mixture. Each tube may exchange water or solute with the interstitium, through the tube wall, due to an osmotic or hydrostatic pressure difference. The variables of

ISSN 0021-9991

the problem are $c_{ik}(x)$, the concentration of the kth solute in the ith tube; $F_{iv}(x)$, the total volume flow in the ith tube; $p_i(x)$, the hydrostatic pressure in the ith tube; $c_{0k}(x)$, the concentration of the kth solute in the interstitium; $p_0(x)$, the pressure in the interstitium. The amount of water, or of solute k, leaking from tube i into the interstitium is denoted by J_{iv} and J_{ik}, respectively. These quantities are known functions of the concentrations $c_{ik}(x)$, $c_{0k}(x)$ and pressures $p_i(x)$, $p_0(x)$ in the ith tube and in the interstitium.

With these notations, the differential equations describing the steady state flow along tube i are the incompressibility equation

$$dF_{iv}/dx = -J_{iv}, \qquad 1 \leqslant i \leqslant I, \tag{2.1}$$

the conservation of solute equations

$$d(F_{iv}c_{ik})/dx = -J_{ik} + s_{ik}, \qquad 1 \leqslant i \leqslant I, \qquad 1 \leqslant k \leqslant K, \tag{2.2}$$

and the pressure drop equation,

$$dp_i/dx = -R_iF_{iv}, \qquad 1 \leqslant i \leqslant I. \tag{2.3}$$

The function $s_{ik}(x)$ is an external source (or sink) of solute k in tube i. The quantity $R_i \geqslant 0$ is the flow resistance in tube i.

The differential equations are coupled together with the conservation laws in the interstitium. These conservation laws are

$$\sum_{i=1}^{I} J_{iv} = 0, \tag{2.4}$$

$$\sum_{i=1}^{I} [J_{ik} - s_{ik}] = s_{0k}, \qquad 1 \leqslant k \leqslant K, \tag{2.5}$$

where s_{0k} is an external source (or sink) of solute k in the interstitium.

The boundary conditions in tube i depend on whether the tube is attached to another tube at $x = 1$. If tube i is not attached at $x = 1$, $F_{iv}(0) > 0$, $p_i(0)$, and $c_{ik}(0) > 0$, $1 \leqslant k \leqslant K$, are specified. If tubes i and j are attached at $x = 1$ (see Fig. 1), fluid is considered as flowing into one of the tubes, say tube i, at $x = 0$. Then the boundary conditions for tube i are the conditions given above, and the boundary conditions for tube j are the continuity conditions $F_{jv}(1) = -F_{iv}(1)$, $p_j(1) = p_i(1)$, $c_{jk}(1) = c_{ik}(1)$, $1 \leqslant k \leqslant K$.

Equations (2.1)–(2.5) together with the boundary conditions complete the mathematical specification of the problem. In these equations, it has been assumed that the diffusional flow of the solutes is small relative to the convective flow, and has therefore been neglected. In addition, we have not specified the pressure forced boundary conditions or the more complicated flow networks that arise in more complete kidney models [4, 5]. The computational procedures that are described here have been used to solve these more complicated renal network problems. We have not given here the

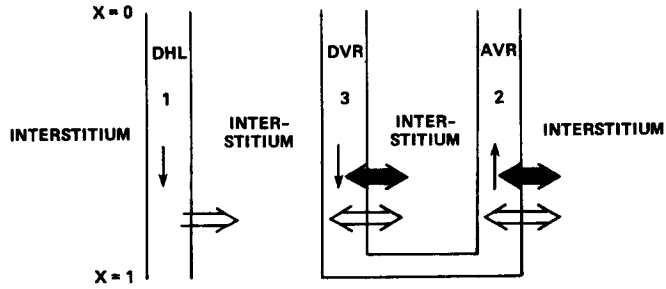

FIG. 1. Three tube model (→ indicates direction of flow, ⇒ indicates transmembrane water flux, and ⟶ indicates transmembrane salt flux).

time dependent form of the kidney equations, but we remark that the use of an implicit difference scheme in the time dependent equations leads to a nonlinear boundary value problem similar to the one given here.

3. The Computational Procedures

To describe the difference equations, we select a mesh spacing Δx and divide $[0, 1]$ into $J = (\Delta x)^{-1}$ equal subintervals. Letting $F_{iv}(j)$ denote the approximate value of $F_{iv}(j\Delta x)$, and similarly for the other unknowns, we use the difference equations

$$\frac{F_{iv}(j) - F_{iv}(j-1)}{\Delta x} = -\tfrac{1}{2}[J_{iv}(j) + J_{iv}(j-1)], \qquad 1 \leqslant j \leqslant J, \quad (3.1)$$

$$\frac{F_{iv}(j)\, c_{ik}(j) - F_{iv}(j-1)\, c_{ik}(j-1)}{\Delta x}$$

$$= -\tfrac{1}{2}[J_{ik}(j) + J_{ik}(j-1)] + \tfrac{1}{2}[s_{ik}(j) + s_{ik}(j-1)], \qquad 1 \leqslant j \leqslant J, \quad (3.2)$$

$$\frac{p_i(j) - p_i(j-1)}{\Delta x} = -\tfrac{1}{2}R_i[F_{iv}(j) + F_{iv}(j-1)], \qquad 1 \leqslant j \leqslant J. \quad (3.3)$$

The conservation laws in the interstitium will be evaluated at the mesh points to obtain

$$\sum_{i=1}^{I} J_{iv}(j) = 0, \qquad 0 \leqslant j \leqslant J, \quad (3.4)$$

$$\sum_{i=1}^{I} [J_{ik}(j) - s_{ik}] - s_{0k} = 0, \qquad 0 \leqslant j \leqslant J. \quad (3.5)$$

Equations (3.1)–(3.5), together with the boundary conditions, then give a system of nonlinear algebraic equations for the unknown concentrations, flows, and pressures. Our purpose is to describe four computational procedures, plus some numerical results, for solving this nonlinear system.

In two of our procedures, Newton's method is used to solve the nonlinear system of equations. A system of linear equations must be solved at each step of Newton's method. The matrix of the linear system may be large if there are many tubes or solutes in the model, and if there are many mesh intervals on each tube. Because the flow paths are either along a tube or into the interstitium, the matrix is sparse, and two sparse matrix routines have been used to solve the linear system. One routine is a general purpose program due to Rheinboldt [6]. The program uses a reordering of the unknowns based upon an arc-graph representation of the structure graph of the matrix. We shall refer to the use of Newton's method with the Rheinboldt matrix routine as procedure A. The second matrix routine, due to Tewarson *et al.* [7], involves a reordering of the unknowns which gives a matrix in bordered block triangular form (Table I). The use of Newton's method with the Tewarson routine will be called procedure B.

TABLE I

Residual, ϕ	Unknowns, γ			
	DHL	DVR	AVR	INT
DHL	×	·	·	×
DVR	·	×	·	×
AVR	·	×	×	×
INT	×	×	×	×

To describe the other computational procedures, it is convenient to divide the unknowns of the problem into two groups. The first group consists of the unknowns describing the flows along the tubes, and the second group consists of the unknowns which link the tubes with each other. The equations of the problem are, in a similar manner, divided into two groups. Thus, in our problem, the second group of unknowns consists of the interstitial concentrations, $c_{0k}(j)$, and pressures, $p_0(j)$, and the second group of equations consist of (3.4) and (3.5). We shall refer to this system as the reduced problem. The other variables of the problem are regarded as functions of the reduced variables. The functions are given implicitly by means of the tube equations (3.1)–(3.3).

The reduced system of equations is solved iteratively by Newton's method. These iterations will be called the outer iterations of the problem. During each outer iteration the values of the variables in each tube must be determined, possibly by using Newton's method. Thus, the computational procedure involves a set of outer iterations and, for each outer iteration, a set of tube calculations.

Two methods have been used to solve the tube equations. In one method, the set of tube equations is itself solved using Newton's method. The use of Newton's method to solve the reduced system, together with Newton's method to solve the tube equations will be called procedure C. In the second method, the tube equations are solved stepwise in the direction of flow. Thus, in a tube with flow in the positive x direction, if the interstitial variables are known and if $c_{ik}(j-1)$, $F_{iv}(j-1)$, $p_i(j-1)$ are

known, Equations (3.1)–(3.3) are solved for $c_{ik}(j)$, $F_{iv}(j)$, $p_i(j)$. The equations are solved by Newton's method, or in certain special cases, by direct elimination. For tubes with flows in the negative direction, the variables at level $j-1$ are evaluated analogously, given the values at level j. The use of Newton's method to solve the reduced system, together with the stepwise method for solving the tube equations, will be called procedure D.

To implement Newton's method in the above procedures requires the calculation of many derivatives. These derivatives are calculated numerically by means of difference quotients.

Equations (2.1)–(2.5) comprise a two point boundary value problem for a system of differential equations. It may be asked whether shooting methods would be appropriate for such problems. Such methods have been tried on simplified kidney equations, where the interstitium has been removed from the model [8]. In general, this approach requires a much finer space increment than finite differences and is impractical when more than one solute is considered. Furthermore, this approach requires solving some tubes by moving upstream, against the direction of flow, and this seems to create numerical instabilities. A form of shooting could be imagined as an alternative to procedures C and D. In this alternative, one uses a differential equation routine to solve the initial value problems that are involved in the tube calculations. This gives rise to different sets of mesh spacings in the tubes and in the interstitium, and errors can accumulate through the interpolation of interstitial concentrations that is required.

4. Numerical Results

Our numerical procedures have been applied to the solution of the simple three tube model shown in Fig. 1. The three tubes represent the descending Henle's limb, (DHL); the descending vasa recta, (DVR); and the ascending vasa recta, (AVR); these are numbered as shown.

The problems use two solutes, numbered $1 \equiv$ salt and $2 \equiv$ protein. In addition, there is assumed to be a source of salt in the interstitium. This salt source would come from active transport out of the ascending Henle's limb, which, for simplicity, has been removed from the problem. The following transmembrane flux law has been assumed for water.

$$J_{iv} = h_{iv}\left[\sum_k \sigma_{ik}(c_{0k} - c_{ik}) + p_i - p_0\right], \tag{4.1}$$

where h_{iv} is the hydraulic permeability of the ith tube, and σ_{ik} is the Staverman reflection coefficient of the kth solute in the ith tube.

The solute flux of the kth solute from the ith tube is taken as

$$J_{ik} = h_{ik}(c_{ik} - c_{0k}) + \frac{(1-\sigma_{ik})}{2} J_{iv}(c_{ik} + c_{0k}), \tag{4.2}$$

where h_{ik} is the membrane permeability for the kth solute.

It is expected that in the operation of this model, the salt source in the interstitium causes an increased interstitial concentration. The increased interstitial concentration draws fluid out of the DHL, and forces salt into the DVR and AVR. The fluid movement from the DHL to the interstitium causes an increased interstitial pressure, which forces fluid into the DVR and AVR. The resulting solution has an increased concentration and volume flow in the DVR and AVR. The parameters for this model are given in Table II. In addition, $s_{01} = 1$. The numerical conclusions that we have drawn from studies with this three tube model have been found to hold for more complicated models [4, 5].

TABLE II

Normalized Parameters

i	h_{iv}	h_{i1}	h_{i2}	σ_{i1}	σ_{i2}	R_i
1	100	0	0	1	1	0.38×10^{-3}
2	100	50	0	0	1	0.15×10^{-2}
3	100	50	0	0	1	0.19×10^{-2}

Tables III–V summarize the characteristics of each procedure. To measure the accuracy of each procedure the concentration ratio $r = c_{11}(1)/c_{11}(0)$ is shown, and the most accurate solution obtained, corresponding to $J = 1024$, is given in parenthesis.

TABLE III

$J = 5$

	Procedure A[a]	Procedure B	Procedure C	Procedure D
r (3.7609)	3.74657	3.74658	3.74659	3.74652
cpu time	13.38	3.96	29.90	3.90
Iterations	5	5	6	6
Expected tube/space iterations (DVR, AVR, DHL)	—	—	12, 12, 12	40, 60
Actual tube/space iterations	—	—	15, 16, 22	40, 60
Matrix size	942	780	114	144
Arrays	3	2	3	1
$\Delta\gamma$	1E − 4	1E − 4	1E − 3	1E − 4
ϵ	1E − 5	1E − 5	1E − 6	1E − 6
ϵ_I	—	—	1E − 5	1E − 5

[a] Procedure A is Newton's method with Rheinboldt's matrix routine; procedure B uses Tewarson's routine; procedure C uses Newton's method to solve the tube equations; procedure D solves the tubes stepwise.

TABLE IV

$J = 10$

	Procedure A	Procedure B	Procedure D[a]
r (3.7609)	3.75702	3.75700	3.75702
cpu time	85.99	16.39	10.13
Iterations	5	5	6
Expected tube/space iterations (DVR, AVR)	—	—	130, 220
Actual tube/space iterations	—	—	116, 159
Matrix size	3316	3060	484
Arrays	3	2	1
$\Delta\gamma$	1E − 4	1E − 4	1E − 4
ϵ	1E − 5	1E − 5	1E − 6
ϵ_I	—	—	1E − 5

[a] Procedure C is prohibitive.

TABLE V

	$J = 20$		$J = 40$
	Procedure B[a]	Procedure D	Procedure D[b]
r(3.7609)	3.75992	3.76000	3.76031
cpu time	99.76	44.45	47.27
Iterations	6	8	2[c]
Expected tube/space iterations (DVR, AVR)	—	460, 840	1720, 3280
Actual tube/space iterations	—	285, 480	1071, 1865
Matrix size	12120	1764	6724
Arrays	2	1	1
$\Delta\gamma$	1E − 4	1E − 4	1E − 4
ϵ	1E − 5	1E − 6	1E − 6
ϵ_I	—	1E − 5	1E − 5

[a] Procedures A and C are prohibitive for $J = 20$.
[b] Procedure B is also prohibitive for $J = 40$.
[c] Initial guess used is the result with $J = 20$.

The total central processing unit (cpu) time required by procedures B and D are comparable for $J = 5$, but, even for $J = 10$, procedure D is significantly faster and becomes more so as J increases. A reason for the poor showing of procedure C, when compared to procedure D, is that the former method requires many more iterations. Although the number of outer (interstitial) iterations is generally the same for procedures C and D, the number of inner iterations is usually larger in procedure C. On the other hand, in procedure D, the number of pointwise iterations is close to a minimum in a certain sense. That is, as shown in Table III, with $J = 5$ the actual number of iterations in the DVR and AVR required to compute the Jacobian matrix for the interstitium is 40 and 60, respectively. This is about what one might expect from the connectivity of the system if one assumes that a perturbation at any position in the interstitium will require one iteration at each tubal segment downstream. However, as shown in Tables IV and V, as J increases, the actual number of iterations becomes much less than predicted since, for example,

$$\lim_{\Delta x \to 0} \{F_{iv}(j+1) - F_{iv}(j) + (\Delta x/2)[J_{iv}(j+1) + J_{iv}(j)]\} = 0, \tag{4.3}$$

so that a perturbation decays more rapidly as the spatial resolution increases. On the other hand, in the Newton methods the ratio of nonzero to total number of elements in the Jacobian matrix remains constant at about 30% for $J = 5$, 10, and 20.

The computer storage required is an important feature of a method. Procedure D requires the least amount of storage in that the largest linear system solved is for the interstitium and is of size $2J + 2$. The tubes are solved either exactly or as systems of $K + 1$ equations, where K is the number of solutes in the tube. Solution by Gaussian elimination with partial pivoting requires only one array of size $(2J + 2)^2$ for the interstitium. Procedure C requires, in addition, that a system of $K + 1$ equations be solved for each tube. Since such a system is sparse, use of the general sparse algorithm requires three arrays of size M, where $M << [(K + 1)J]^2$ is the number of nonzero elements in the Jacobian. Procedure B is competitive in its use of storage for up to moderately large J, since it requires but two arrays of size $3J(10J + 2)$. Procedure A requires three arrays of size M where $M << (10J + 2)^2$.

$\Delta\gamma$ is the increment used in differencing numerically to calculate the Jacobian matrix. A value slightly larger than the convergence criterion has been found to be adequate. ϵ and ϵ_I are convergence criteria for the inner iterations and the outer (interstitial) iterations, respectively. In procedure D, it is advantageous to converge the tubes to a closer tolerance than is required for overall convergence, since this will generally lead to fewer outer iterations.

The region of attraction of solutions has been investigated for each of the difference schemes. As shown in Table VI, procedure D has the widest region of attraction when the DHL is solved exactly. Procedure C differs from procedure D only in the manner in which the tube equations are solved. This suggests that it is important to solve the tube equations quite accurately, perhaps using double precision, to take full advantage of the method of interstitial iterations. Since the choice of physical parameters, such as permeabilities, affects the region of attraction, it is possible to extend

TABLE VI

	Initial concentrations c_{i1} [a]						
	0.01	0.3	0.7	0.9	1.0	1.5	1.7
Procedure A	—	+	+	+	+	+	+
Procedure B	—	+	+	+	+	+	+
Procedure C	—	0	+	0	+	0	0
Procedure D (DHL linearized)	—	0	0	+	+	+	—
Procedure D (DHL exact)	+	+	+	+	+	+	+

[a] + indicates convergence to positive solution; 0 indicates no convergence to solution; — indicates convergence with flow reversal.

the region initially by a judicious choice of parameters. For example, as shown in Table VI, procedure D with the equations for the DHL solved by Newton's method will result in flow reversal when $\sigma_{11} = 1$. However, $\sigma_{11} = 0.9$ will yield a positive solution which will lead to convergence when used as an initial guess to the desired problem with $\sigma_{11} = 1$.

Procedure D was used to calculate the solution for a sequence of fine mesh spacings. The results are shown in Table VII, together with the results of Richardson extrapolation from these values. From these results it is seen that the error in the difference method is $O((\Delta x)^2)$, as would be expected. Richardson extrapolation does indeed give more accurate results, but, on the other hand, the 1 % error in r for $J = 4$ indicates that, for this problem, four mesh intervals is adequate for practical purposes.

TABLE VII

Concentration Ratio r

	Procedure D	Extrapolated
4	3.73750	
6	3.75005	
8	3.75480	
12	3.75821	
16	3.75938	3.76056
24	3.76015	
32	3.76051	
36	3.76058	3.76093
48	3.76064	
64	3.76079	3.76091
144		3.76088
256		3.76089
576		3.76087
1024		3.76088

5. Conclusions

Based on our experience in modelling renal behavior, the following seems to be a useful set of guidelines for solving nonlinear fluid network problems. (a) Formulate difference equations so as to preserve the conservation character of the original system of equations. For this it is convenient to write the differential equations in a conservation form and apply a centered Euler difference approximation. (b) Divide the unknowns of the problem into groups, each group corresponding to either unknowns along a physical flow path of the network, or to unknowns which link the flow paths. By eliminating the unknowns within a flow path, the network equations are reduced to a system of equations involving the linking variables. Use Newton's method to solve the reduced system. (c) Within each flow path, the equations should be solved, if possible, in a stepwise procedure proceeding in the direction of flow. The equations within each flow path must be solved with adequate accuracy, perhaps using double precision calculations.

In the renal problem, the flow paths are the tubes of the network, and the linking variables are the interstitial variables. The blockwise elimination of variables is discussed in more detail in Stephenson [9].

References

1. J. L. Stephenson, *Biophys. J.* **6** (1966), 539.
2. J. L. Stephenson, *Biophys. J.* **13** (1973), 512.
3. J. L. Stephenson, *Biophys. J.* **13** (1973), 546.
4. J. L. Stephenson, R. P. Tewarson, and R. Mejia, *Proc. Nat. Acad. Sci. USA* **71** (1974), 1618.
5. J. L. Stephenson, R. Mejia, and R. P. Tewarson, *Proc. Nat. Acad. Sci. USA* **73** (1976), 252.
6. W. C. Rheinboldt and C. K. Mesztenyi, Technical Report **TR-262** (1973), University of Maryland Computer Science Center.
7. R. P. Tewarson, J. L. Stephenson, A. Keydes, and R. Mejia, *Comput. Biomed. Res.*, 1976, in press.
8. R. Mejia and J. L. Stephenson, 1973, Summer Computer Simulation Conference, 806.
9. J. L. Stephenson, in preparation.