

Numerical Solution of Multinephron Kidney Equations

R. MEJIA AND J. L. STEPHENSON

Section on Theoretical Biophysics, National Heart, Lung and Blood Institute, and Mathematical Research Branch, National Institute of Arthritis, Metabolism, and Digestive Diseases, National Institutes of Health, Bethesda, Maryland 20014

Received August 4, 1978; revised October 17, 1978

A description is given of an algorithm used to solve the time dependent kidney equations for many nephrons. The unknowns of the problem have been divided into two groups: the first consists of the unknowns in the tubes that comprise each nephron, and the second consists of the unknowns that link the tubes and nephrons with each other. A comparison of the accuracy, storage and *CPU* time is made for a code utilizing a strictly numerical algorithm and codes using a combination of analytic and numerical calculations.

1. INTRODUCTION

The mammalian kidney consists of some 10^6 nephrons that extend to varying depths within the renal medulla and function in concert to regulate urinary concentration. Its operating mechanism has been explained in terms of a flow network in which a fluid containing several solutes traverses through a system of tubes, or nephron segments [1-3]. Each nephron segment may exchange fluid or solutes, along its length, with other nephrons through a common interstitial space. This interchange is caused by osmotic, hydrostatic and electrochemical driving forces acting across the tube walls. Different nephron depths permit individual solute gradients to be generated and are significant in the overall concentrating mechanism.

The kidney model is formulated as a multi-point boundary value problem as follows: Given (1) hydrostatic pressure of blood entering and leaving the kidney, (2) the concentration of solutes in the entering blood flow, and (3) the bladder pressure due to urine formation by the kidney, we wish to calculate the pressure, flow and solute concentrations along each nephron [4]. This leads to a system of differential equations describing the time dependent network flows. This paper describes an algorithm for solution of these equations and compares several computational schemes used to solve them.

2. THE DIFFERENTIAL EQUATIONS

Consider a kidney consisting of many nephrons and separated into an inner (medullary) and an outer (cortical) region (Fig. 1). The nephrons extending into the inner region generate a family of parallel flow tubes of varying lengths, which are

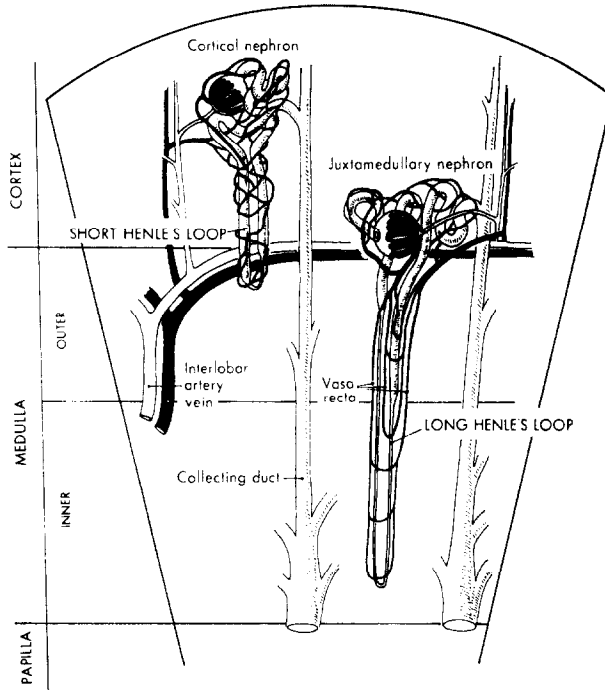


FIG. 1. Schematic of a sagittal section of the mammalian kidney (adapted from [11]).

separated by an interstitial space that is considered to be well mixed in each plane perpendicular to the tubes (Fig. 2). In the outer region, the tubes need not be parallel and are imbedded in a single well mixed interstitium.

The variables in the i th tube are the axial volume flow, F_{iv} ; the hydrostatic pressure, p_i ; and the concentration of the k th solute, c_{ik} . The water and solute transmural flux into the interstitium are denoted by J_{iv} and J_{ik} respectively. With these notations, the time dependent equations describing flow along the i th tube are the incompressibility equations

$$\frac{\partial F_{iv}}{\partial x} = -J_{iv}, \quad 1 \leq i \leq I, \quad (2.1)$$

the solute conservation equations

$$\frac{\partial}{\partial x} (c_{ik} F_{iv}) = -J_{ik} - \frac{A_i}{\partial t} \frac{\partial c_{ik}}{\partial t}, \quad 1 \leq i \leq I, \quad 1 \leq k \leq K, \quad (2.2)$$

and the pressure drop equations

$$\frac{\partial p_i}{\partial x} = -R_i F_{iv}, \quad 1 \leq i \leq I, \quad (2.3)$$

where $R_i \geq 0$ is the resistance to flow, and A_i is the cross sectional area in tube i .

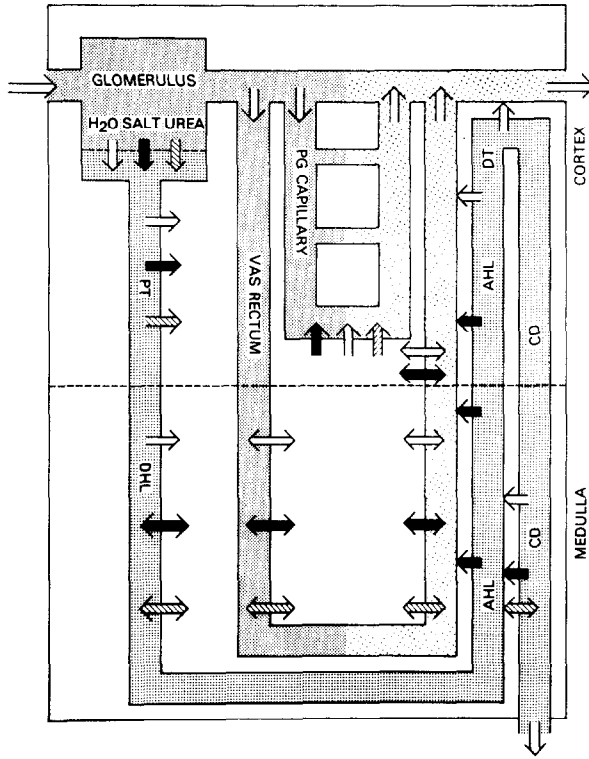


FIG. 2. Solute and water movement in a nephrovascular unit.

For tubes that flow in a positive direction, from $x = 0$ to $x = L_i \leq 1$, the boundary conditions are

$$F_{iv}(0) = F_{iv0}, \quad (2.4)$$

$$p_i(0) = p_{i0}, \quad (2.5)$$

$$c_{ik}(0) = c_{ik0}, \quad 1 \leq k \leq K. \quad (2.6)$$

If tube i is connected to tube $i - 1$ and the flow in each is positive,

$$F_{iv0} = F_{i-1,v}(L_{i-1}), \quad (2.7)$$

$$p_{i0} = p_{i-1}(L_{i-1}), \quad (2.8)$$

$$c_{ik0} = c_{i-1,k}(L_{i-1}), \quad 1 \leq k \leq K. \quad (2.9)$$

If either tube has flow in the negative direction, analogous conditions are prescribed at the boundary.

The variables in the interstitial space, which are defined analogously to the tube unknowns, are F_{mv} , p_m , c_{mk} and F_{cv} , p_c and c_{ck} in the medullary and cortical space respectively. The water and solute flux are similarly defined as J_{mv} , J_{cv} , J_{mk} and J_{ck} .

With these notations the incompressibility and pressure drop equations in the medullary interstitium are similar to the tubal equations, namely:

$$\frac{\partial F_{mv}}{\partial x} = -J_{mv}, \quad (2.10)$$

and

$$\frac{\partial p_m}{\partial x} = -R_m F_{mv}. \quad (2.11)$$

The solute flow equations are written to account for bulk flow and diffusion, so that

$$F_{mk} = F_{mv} c_{mk} - D_k \frac{\partial c_{mk}}{\partial x}, \quad 1 \leq k \leq K, \quad (2.12)$$

and

$$\frac{\partial F_{mk}}{\partial x} = -J_{mk} - \frac{A_m}{\partial t} \frac{\partial c_{ik}}{\partial t}, \quad 1 \leq k \leq K. \quad (2.13)$$

In the cortical interstitium with fixed volume, V_c , fluid balance requires that

$$J_{cv} = -F_{mv}(0), \quad (2.14)$$

and solute balance that

$$V_c \frac{\partial c_{ck}}{\partial t} = -F_{mk}(0) - J_{ck}, \quad 1 \leq k \leq K. \quad (2.15)$$

The boundary conditions to be satisfied by the interstitial equations are

$$c_{mk}(0) = c_{ck}, \quad 1 \leq k \leq K, \quad (2.16)$$

$$p_m(0) = p_c, \quad (2.17)$$

and

$$F_{mv}(1) = F_{mk}(1) = 0, \quad 1 \leq k \leq K. \quad (2.18)$$

A boundary condition often used instead of matching the concentration at the corticomodullary border as in (2.16) is

$$c_{mk}(1) = \frac{J_{mk}(1)}{J_{mv}(1)}, \quad 1 \leq k \leq K, \quad (2.19)$$

which states that the solute concentration at the closed end of the interstitial space is the concentration of the transmural flux.

Transmural flux laws for all tubes except the proximal tubule are given by

$$J_{iv} = h_{iv} \left[\sum_k RT(c_{mk} - c_{ik}) \sigma_{ik} + p_i - p_m \right], \quad (2.20)$$

and

$$J_{ik} = h_{ik}(c_{ik} - c_{mk}) + (1 - \sigma_{ik}) J_{iv}(c_{ik} + c_{mk})/2 + \mathcal{J}_{ik}, \quad 1 \leq k \leq K, \quad (2.21)$$

where h_{iv} is the hydraulic permeability coefficient of the i th tube, R is the gas constant, T is the absolute temperature, σ_{ik} is the Staverman reflection coefficient of the wall of the i th tube for the k th solute, h_{ik} is its passive permeability for the k th solute, and \mathcal{J}_{ik} is the metabolically driven transport out of the i th tube.

The metabolically driven transport is assumed to obey Michaelis-Menten kinetics, namely

$$\mathcal{J}_{ik} = \frac{a_{ik}}{1 + b_{ik}/c_{ik}}, \quad 1 \leq k \leq K, \tag{2.22}$$

where a_{ik} is the maximum rate of transport and b_{ik} is the Michaelis constant.

For the proximal tubule, transport is assumed isotonic, so that

$$J_{iv} = \alpha F_{iv}(0), \tag{2.23}$$

and

$$J_{ik} = J_{iv} c_{ik}, \quad 1 \leq k \leq K, \tag{2.24}$$

where α is an arbitrary constant.

In the medulla, water and mass conservation require that

$$J_{mv}(x) = - \sum_i J_{iv}(x), \quad 0 \leq x \leq 1, \tag{2.25}$$

and

$$J_{mk}(x) = - \sum_i J_{ik}(x), \quad 0 \leq x \leq 1, \quad 1 \leq k \leq K. \tag{2.26}$$

The cortical interstitial fluxes are defined in a similar manner.

3. THE COMPUTATIONAL PROCEDURE

For solution, the system of differential equations (2.1)–(2.3) and (2.10)–(2.15) is replaced with a system of finite difference equations. To describe these equations we select a mesh spacing Δx and divide $[0, L_i]$ into $J = L_i/\Delta x$ subintervals, where $1 \geq L_i$, the length of the i th tube. A time increment Δt is chosen so that $t_n = n \Delta t$ for $n = 0, 1, \dots, J$, and writing the other unknowns similarly, we use the difference equations

$$\frac{F_{iv}^n(j) - F_{iv}^n(j-1)}{\Delta x} = - \frac{1}{2} [J_{iv}^n(j) + J_{iv}^n(j-1)] \tag{3.1}$$

$1 \leq i \leq I, \quad 1 \leq j \leq J,$

$$\frac{p_i^n(j) - p_i^n(j-1)}{\Delta x} = - \frac{1}{2} R_i [F_{iv}^n(j) + F_{iv}^n(j-1)] \tag{3.2}$$

$1 \leq i \leq I, \quad 1 \leq j \leq J,$

$$\frac{F_{iv}^n(j) c_{ik}^n(j) - F_{iv}^n(j-1) c_{ik}^n(j-1)}{\Delta x} = - \frac{1}{2} \{ [J_{ik}^n(j) + J_{ik}^n(j-1)] \tag{3.3}$$

$$+ \frac{c_{ik}^n(j) - c_{ik}^{n-1}(j) + c_{ik}^n(j-1) - c_{ik}^{n-1}(j-1)}{\Delta t} \}$$

$1 \leq i \leq I, \quad 1 \leq j \leq J, \quad 1 \leq k \leq K.$

This difference scheme, which is centered in space and is backward in time, has been shown to be stable and accurate in approximating solutions of the kidney equations [5].

If we denote by γ^n the vector of unknown flows, pressures and concentrations in the interstitial space and all nephrons at time t_n and the system of difference and conservation equations by ϕ , we seek a solution of the system

$$\phi(\gamma^n, \gamma^{n-1}) = 0 \quad n = 1, 2, \dots, \quad (3.4)$$

where γ^0 is a vector of initial values, and for $n > 1$ γ^{n-1} is obtained from the previous time iteration.

For example, in a three nephron model with two small filtered and one nonfiltered solute and with nephron lengths proportional to $J = 10, 6$ and 3 respectively, eq. (3.4) is a system of 855 equations in as many unknowns. If Newton's method is applied to solve this system, less than 10% of the entries in the Jacobian matrix, M , are non-zero, thus a sparse matrix inversion algorithm might be used to obtain a solution to the linear system

$$\phi(\gamma_i^n, \gamma^{n-1}) - M \Delta \gamma_i^n = 0, \quad (3.5)$$

where γ_i^n is the i th Newton iterate and

$$\gamma_{i+1}^n = \gamma_i^n - \Delta \gamma_i^n. \quad (3.6)$$

However, the advantage of solving a sequence of smaller problems has been demonstrated in [6] for a system of parallel flow tubes exchanging water and solute through an interstitial space. This technique is extended here to permit many nephrons coupled together through a common interstitial space.

As in [6], the problem is partitioned by dividing the unknowns into two groups. The first consists of the unknowns describing the flows along the tubes, which are denoted by γ_i for the i th tube. The second consists of the unknowns that link the tubes in all nephrons together and are denoted by γ_G . These include the interstitial variables: volume flow, pressure and solute concentrations, as well as the three unknowns per nephron that are associated with the exit boundary conditions, venous and bladder pressure.

The equations of the problem are, in a similar manner, divided into two groups, namely

$$\phi_i(\gamma_i^n, \gamma_i^{n-1}, \gamma_G^n) = 0, \quad i = 1, 2, \dots \quad (3.7)$$

and

$$\phi_G(\gamma_1^n, \gamma_2^n, \dots, \gamma_l^n, \gamma_G^{n-1}, \gamma_G^n) = 0. \quad (3.8)$$

Given an estimate of γ_G^n we solve (3.7) for each i . We do so stepwise and in the direction of flow [6]. Having γ_i^n for all i , we proceed to solve (3.8) and then obtain a

new estimate for γ_G^n . This process is iterated until γ_G^n is obtained to the desired accuracy. Then time is stepped forward and the procedure is repeated.

In fact, equation (3.7) can be solved explicitly for γ_i^n if γ_G^n is assumed, since γ_i^{n-1} is known. Newton's method is used, and the analytic expression of the inverse of the Jacobian at each step is calculated. With the transmembrane flux laws given by equations (2.20) through (2.26), the algebra to calculate the inverse is especially tedious for four or more unknowns (when $K \geq 2$), but the use of an algebraic manipulator [7] has facilitated it.

Equations (3.8) must be solved simultaneously, however, since the γ_G^n depend on flows in both the positive and negative space directions. Its Jacobian, which is often more than half full, is calculated numerically by means of difference quotients because the number of unknowns is large—in the order of $U(J+1)+B$, where $U=K+2$, K is the number of solutes, J is the number of space intervals, and B is the number of exit boundary conditions to be satisfied.

Various procedures for the solution of equations (3.7) and (3.8) are compared. In each case, given values for the global variables, γ_G , the solution in the tubes is obtained stepwise in the direction of flow. In one case, the flow equations (2.1) and (2.10) are integrated explicitly, thus reducing the number of unknowns in the Newton iterations for the tubes and for the interstitium; in another, only the tube flows are integrated out. In one instance, all Jacobians are approximated using difference quotients and are inverted numerically; in another, the pointwise Jacobian in the tubes is obtained analytically; in yet another, the pointwise Jacobian is inverted analytically, and the FORTRAN code for evaluation of the inverse generated by the algebraic manipulator, REDUCE.

Procedure A consists of the solution of equations (2.1)–(2.3) and (2.10)–(2.15), with the inverse of the pointwise Jacobian computed analytically, and is described in detail in [6]. Procedure B is like A except that the volume flow in each tube is integrated explicitly. Procedure C is also like A except that the pointwise Jacobian is computed analytically and inverted numerically. In procedure D all flow equations are integrated out, and all Jacobians are computed numerically. This was the method that we first used to solve a model of the medulla [8] and of a complete single nephron [4].

Figures 3–6 show profiles of salt and urea concentrations and of volume flow for a model with a distribution of cortical and juxtamedullary nephrons. The input parameters used are given in Table I, and output of the model for a ratio of two cortical to one juxtamedullary nephron is shown in Table II. The concentration of salt and urea in the nephron is shown in Fig. 3 and 4, and normalized volume flow in the vasculature and the nephron is shown in Figs. 5 and 6. These show that a mesh with ten intervals yields a relatively smooth profile that is adequate to describe quantitative and qualitative features of the model such as requirements in order to concentrate with passive transport in the inner medulla [10]. A refinement of the mesh by a factor of two to twenty intervals, for example, shows about a 1% error in total urine concentration.

Table III shows for each procedure used the total urine concentration, C_{urine} , the PDP-10 cpu time in seconds per interstitial iteration, and the total number of

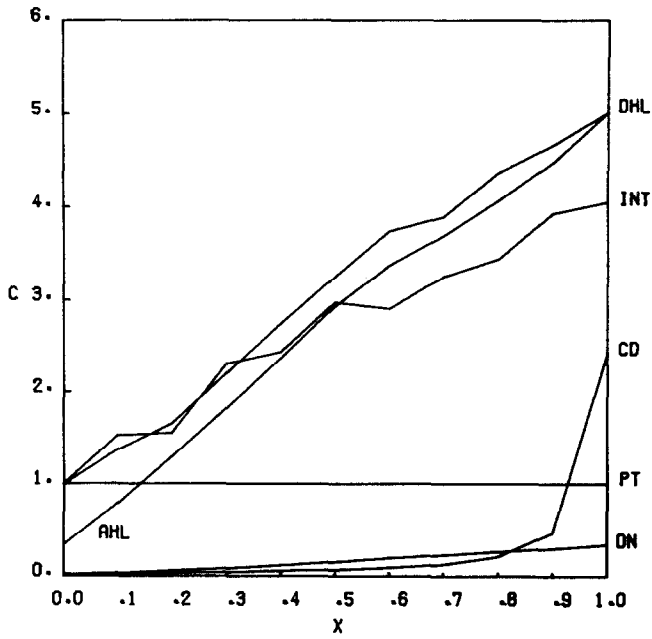


FIG. 3. Salt concentration in a juxtamedullary nephron and the interstitium.

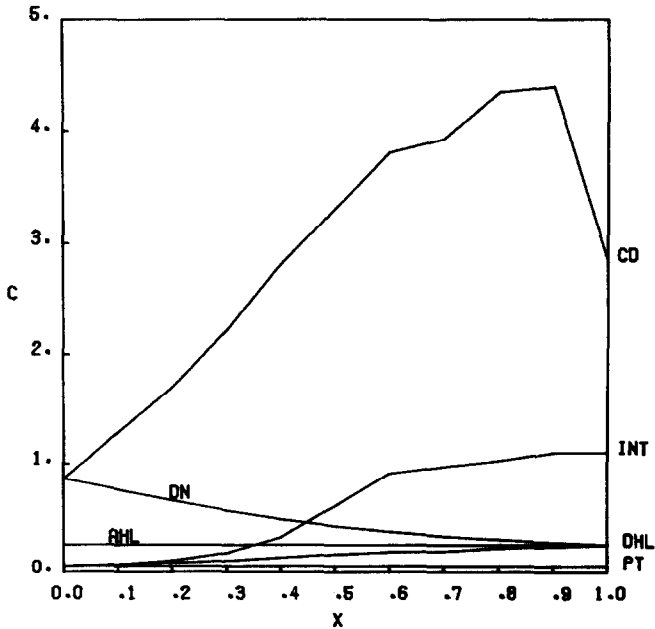


FIG. 4. Urea concentration in a juxtamedullary nephron and the interstitium.

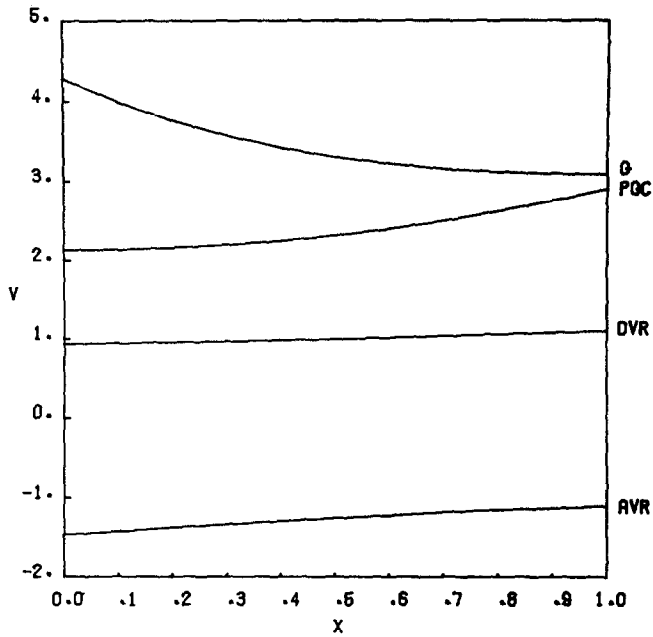


FIG. 5. Volume flow in the vasculature normalized with 1 unit ≈ 16.86 nl/min.

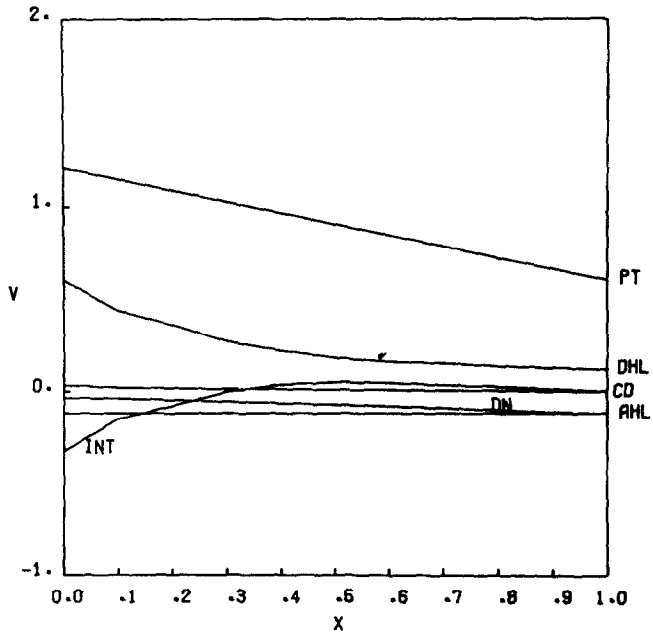


FIG. 6. Volume flow in a juxtamedullary nephron and the interstitium (1 unit ≈ 16.86 nl/min).

TABLE I
Normalized Parameters

Tube ^a	h_v	R ($\times 10^4$)	σ	h_s	h_u	
G	1400	2	—	—	—	$R_A = 10.5 \times 10^{-4}$
PGC	300	28.5	0	4	4	$R_E = 0.1 \times 10^{-4}$
DVR 1	100	28.5	0	1,000	1,000	$R_m = 250. \times 10^{-4}$
DVR 2	100	2,000	0	1,000	1,000	
CAVR	100	4.9	0	10,000	10,000	
AVR 1	100	24.5	0	1,000	1,000	
AVR 2	100	2,000	0	1,000	1,000	
BC	0	0	1	0	0	
PT	—	7	—	—	—	$\alpha = 0.5$
DHL	20	10	1	0	0	
AHL 1	0	10	1	0.05, 0.85 ^b	0	$a = 0.7, 0. b = 0.1$
AHL 2	0	10	1	0.05	0	$a = 1.3 \quad b = 0.1$
DN 1	0.2	6	1	0	0	$a = 0.2 \quad b = 1.$
DN 2	0.2	6	1	0	0	$a = 0.45 \quad b = 1.$
CD	0.5	6	1	0	0., 0.02 ^c	

^aG - Glomerulus, PGC - postglomerular capillary, DVR 1 - descending vas rectum for first (juxta-medullary) nephrovascular unit, CAVR - cortical ascending vas rectum, AVR 2 - ascending vas rectum for second (cortical) nephrovascular unit, BC - Bowman's capsule, PT - proximal tubule, DHL - descending loop of Henle's limb, AHL - ascending loop of Henle's limb, DN - distal nephron, CD - collecting duct.

^bThe first value refers to the outer medulla where $0. \leq x \leq 0.5$; the second refers to the inner medulla where $0.6 \leq x \leq 1$. For $0.5 < x < 0.6$ the value varies linearly.

^cThe first value holds for $0. \leq x \leq 0.4$; the second holds for $0.5 \leq x \leq 1.$; and h_u varies linearly for $0.4 < x < 0.5$.

interstitial iterations required to reduce the residuals to less than 10^{-5} , which yields concentrations accurate to approximately three figures. After each tubal and interstitial iteration, if the residual has been reduced by an order of magnitude ($EPJB = 10$), the Jacobian has not been recalculated.

Note that procedures A and B require the fewest iterations while procedure B requires the least cpu time. All schemes converge to essentially the same solution as shown by C_{urine} after starting with the same initial guess.

Table IV contains results for a ratio of seven cortical to one juxtamedullary nephron. When a reduction of the maximum residual by a factor of 10 is required to permit reusing the current Jacobian, procedure B converges to a solution including negative concentrations. A more stringent criterion of 50 leads to the desired positive solution at a corresponding cost in computer time. Note that in this case the number of interstitial iterations remains at 5 while the number of tube iterations increases.

Inverting the Jacobian numerically, as in procedure C, not only requires additional computer time, but requires more iterations to obtain the same accuracy as procedures A and B. Procedure D requires more iterations than procedures B and C, indicating that approximation of the volume flow in the interstitium substantially reduces the

TABLE II

Results of Procedure A — 2/1 Cortical to Juxtamedullary Nephrons

Position ^a	Volume flow (nl/min)		[Na Cl] (mOsm/300)		[Urea] (mOsm/300)		Oncotic pressure (mm Hg)		Hydrostatic pressure (mm Hg)	
	long	short	long	short	long	short	long	short	long	short
Aff	72.19	62.89	1.000	1.000	0.050	0.050	22.61	22.61	77.35	77.35
G	72.19	62.89	1.000	1.000	0.050	0.050	22.61	22.61	50.63	54.03
Eff	51.83	39.99	1.000	1.000	0.050	0.050	31.48	35.58	46.53	50.75
PGC _a	35.90	39.70	1.000	1.000	0.050	0.050	31.48	35.58	46.35	50.63
PGC _e	48.82	53.87	1.009	1.009	0.048	0.048	23.15	26.18	5.95	5.95
DVR	15.94	0.29	1.000	1.000	0.050	0.050	31.48	35.58	46.35	50.63
AVR	31.39	4.97	0.946	1.019	0.048	0.046	16.01	2.08	5.95	5.95
CI	—	—	1.011	1.011	0.048	0.048	—	—	14.04	14.04
BC	20.35	22.90	1.000	1.000	0.050	0.050	—	—	14.82	15.17
PT	20.35	22.90	1.000	1.000	0.050	0.050	—	—	14.82	15.17
DHL	10.18	11.45	1.000	1.000	0.050	0.050	—	—	11.01	10.95
AHL	2.03	3.52	5.011	3.252	0.250	0.163	—	—	9.58	9.82
DN	2.03	3.52	0.338	0.389	0.250	0.163	—	—	8.87	9.16
CD	0.60	1.48	0.017	0.062	0.853	0.388	—	—	8.63	8.63
Urine	0.00	0.05	2.419	1.854	2.838	3.372	—	—	8.57	8.57
MI(0)	5.45	5.45	1.011	1.011	0.048	0.048	—	—	14.04	14.04
MI(1)	0.00	0.00	4.049	4.049	1.096	1.096	—	—	16.72	16.72

^a Aff – afferent to glomerulus, Eff – efferent to glomerulus, PGC_a – afferent to post glomerular capillaries, CI – cortical interstitium, MI(0) – medullary interstitium at corticomedullary border, MI(1) – medullary interstitium at the papilla.

TABLE III

2/1 Cortical to Juxtamedullary Nephrons ($J = 10$)^a

	Procedure A	Procedure B	Procedure C	Procedure D
C_{urine}	5.226	5.227	5.226	5.227
cpu time/iteration	38.87	33.61	39.53	48.52
Iterations	4	4	5	6
Storage	2500	2500	2500	1521

^a Procedure A solves the full system of equations with the inverse of the pointwise Jacobian computed analytically. Procedure B solves a reduced system where the tube volume flows are integrated explicitly. In Procedure C the pointwise Jacobian is inverted numerically. Procedure D integrates out all flows, and all Jacobians for the reduced system are computed numerically.

TABLE IV

7/1 Cortical to Juxtamedullary Nephrons ($J = 10$)

	Procedure A	Procedure B	Procedure C	Procedure D	
C_{urine}	5.896	5.923 ^a	5.896 ^b	5.896	5.896
cpu time/iteration	40.82	35.89	43.15	54.20	47.96
Iterations	5	6	5	6	9
Storage	2500	2500	2500	2500	1521
EPJB	10	10	50	10	10

^a Results in flow reversal within the collecting duct and corresponding negative concentrations.

^b All concentrations are positive.

accuracy. This is consistent with the results shown in Table 1, p. 63 of Farahzad and Tewarson [9].

The storage requirements for each procedure are in the order of $(U(J + 1) + 3V)^2$ where $U = K + 2$ for procedures A, B and C and $U = K + 1$ in procedure D; K is the number of solutes; J is the number of space intervals and V is the number of distinct nephron populations (two populations, one cortical and one juxtamedullary, in the data shown).

4. CONCLUSION

Procedure A has been shown to be an efficient method to solve the time dependent renal flow equations for a multinephron model. Through an example, it is shown to be accurate, stable and to have an adequate radius of convergence for solutions to complex kidney models.

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. MEJIA, AND R. P. TEWARSON, *Proc. Nat. Acad. Sci. USA* **73** (1976), 252.
5. R. MEJIA, J. L. STEPHENSON, AND R. B. KELLOGG, in "Proceedings of the 1976 Summer Computer Simulation Conference", p. 502.
6. R. MEJIA, R. B. KELLOGG, AND J. L. STEPHENSON, *J. Computational Phys.* **1** (1977), 53.
7. A. C. HEARN, "Reduce 2 User's Manual," 2nd ed., University of Utah, 1973.
8. J. L. STEPHENSON, R. P. TEWARSON, AND R. MEJIA, *Proc. Nat. Acad. Sci. USA* **71** (1974), 1618.
9. P. FARAHZAD AND R. P. TEWARSON, *Comput. Biol. Med.* **8** (1978), 57.
10. J. L. STEPHENSON AND R. MEJIA, Abstract 1606, Feder. Amer. Soc. Exptl. Biol., Bethesda, Md., 1978.
11. R. F. PITTS, "Physiology of the Kidney and Body Fluids," 3rd ed., Year Book Med. Pub., Chicago, 1974.