A Sparse Matrix Method for Renal Models

ABSTRACT

A sparse matrix method for the numerical solution of nonlinear differential equations arising in modeling of the renal concentrating mechanism is given. The method involves a symmetric permutation of the variables and equations such that the occurence matrix has a block bidiagonal structure and the blocks above and below the main diagonal have a known sparse structure. The method is faster and, for more than fifteen nodes per tube, requires less storage than the present methods.

1. MAIN RESULTS

The size and complexity of realistic models of renal concentrating mechanism make it necessary to use special algorithms for the numerical solution of differential equations describing these models. Model connectivity is used in these algorithms to decrease the storage, run time and truncation errors [1-7]. The structure of the Jacobian for the model problem [1, 5, 7] is shown in Fig. 1, where each * represents a 3x3 block (volume flow, salt & urea) and the x range is divided into ten equal parts (N = 10). Thus the matrix is of order 18N. It is evident from the figure that the Jacobian matrix is block bidiagonal except for the last 3N+3 rows and columns. Therefore, the solutions of the first 15N-3 equations and variables is relatively easy in terms of the last 3N+3 variables. The methods described in [1-7] make use of this fact to solve the whole problem as a function of the last 3N+3 variables. This requires the computation of the small 3N+3 Jacobian either by the repeated solution of the first 15N-3 equations [4, 6] or the use of Gaussian elimination on the whole Jacobian generated in small interacting parts [5, 6]. This in turn leads to methods which we will, respectively, call M1 and M2.

In Fig. 2 we have shown a permuted form of the Jacobian of Fig. 1. It was obtained by renumbering the variables and equations in each tube down the medulla. The Distal Nephron was divided into two equal parts and in each part the numbering was done starting from their respective junctions with the Ascending Henles Limb

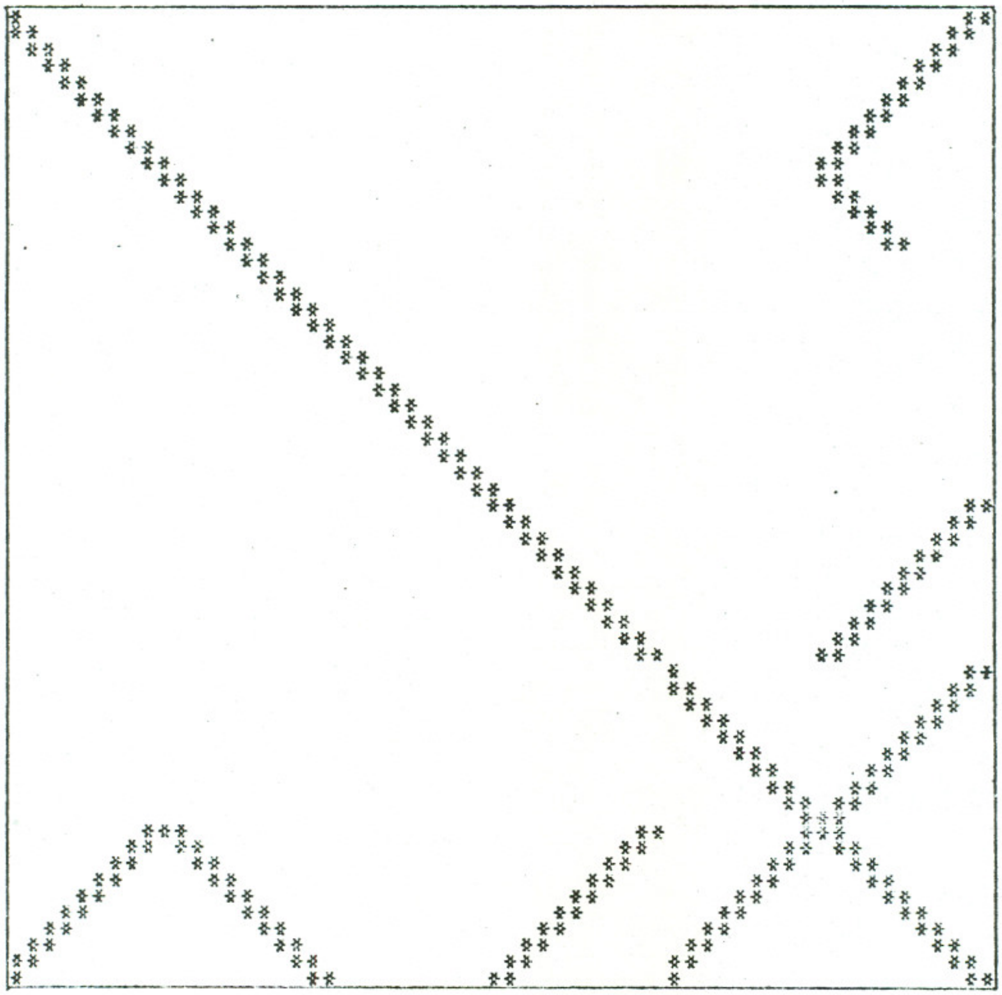

Fig. 1. The non-zeros in the Jacobian

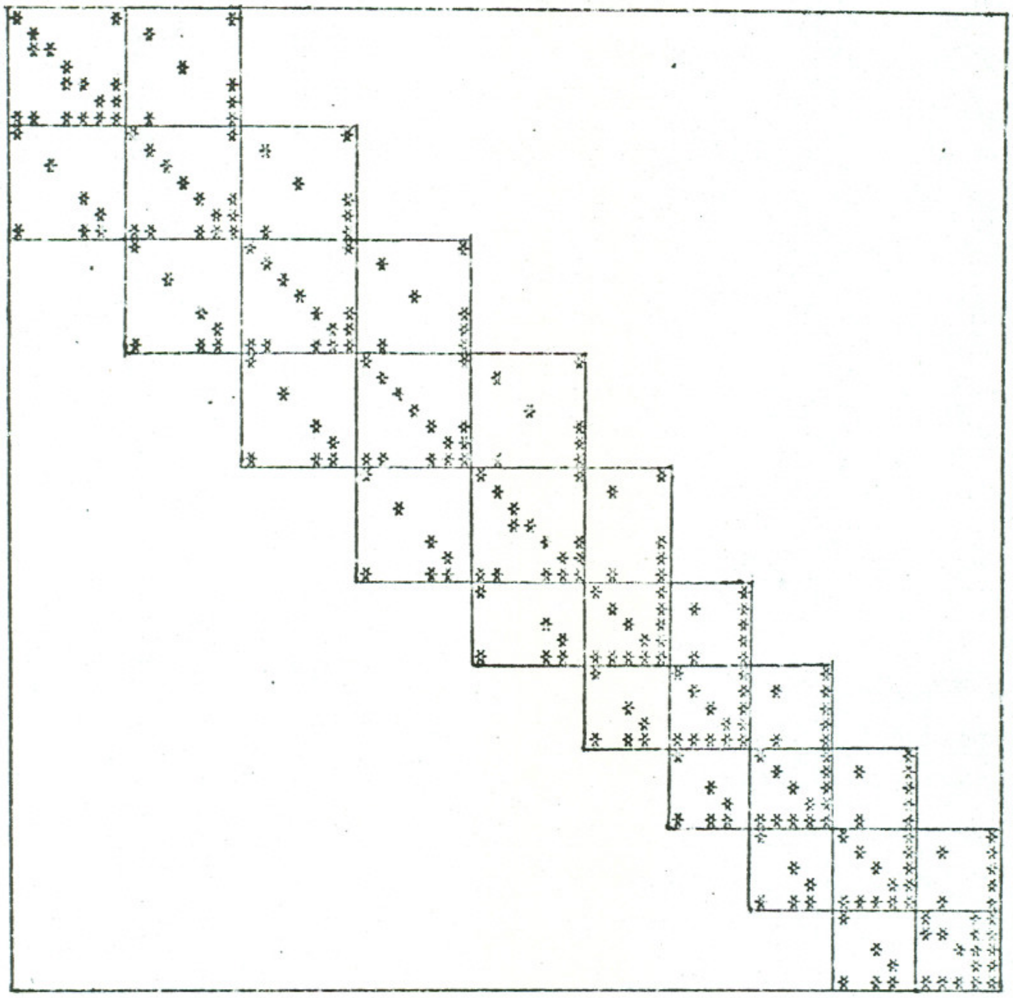

Fig. 2. The non-zeros in the permuted Jacobian

and the Collecting Duct. In Fig. 2, the non-zero elements in the super-diagonal blocks are due to the tube whose flow direction is reverse to that of our numbering. Uniform chop size was used in all the tubes - this makes each part of the Distal Nephron of length 0.5. The three nonzero columns in the first N/2 - 1 super-diagonal blocks are due to the Ascending Henles Limb, Ascending Vasa Recta and the part of Distal Nephron flowing against the numbering. The rest of the super-diagonal blocks have two non-zero columns since only half of the Distal Nephron is flowing against the numbering.

Let the blocks in Fig.2 be called A_{ij}. Clearly $A_{ij} = 0$, $|i-j| > 1$. At each Newton step, the system of equations having the coefficient matrix of Fig. 2 and B as the right hand side are solved as follows:

Algorithm M3

Generate the right hand side B. Let B_i be the part of B that corresponds to A_{ii}. Generate A_{11}. For i = 1,2, ..., N-1 do steps 1 to 3.

1. Generate and store the non-zero columns of $A_{i,i+1}$.
2. Perform the Gaussian elimination on the augmented matrix $(A_{ii}, A_{i,i+1}, B_i)$ to transform A_{ii} to the identity matrix I. Save the updated $A_{i,i+1}$.
3. Generate $A_{i+1,i+1}$ on top of I and the non-zero elements of $A_{i+1,i}$, then set

$$A_{i+1,i+1} = A_{i+1,i+1} - A_{i+1,i} \cdot A_{i,i+1}$$
$$B_{i+1} = B_{i+1} - A_{i+1,i} \cdot B_i$$

Perform Gaussian elimination on (A_{NN}, B_N) to transform A_{NN} to I. Back-solve for super-diagonal blocks:

$$B_{i-1} = B_{i-1} - A_{i-1,i} \cdot B_i, \quad i = N, N-1, \ldots\ldots, 2.$$

The solution is now in B.

1. COMPUTATIONAL RESULTS

Algorithm M3 was programmed for UNIVAC 1100 computer in FORTRAN and used on the model problem [1, 5, 7]. The Trapezoidal (TR) and Simpsons Rule (SR) [8] implementations were run for N = 10, 20, 40, 80 and 160. In each case, the

Table I

N Segments		40	40	80	80	160	160	6614
Integration Scheme		SR	TR	SR	TR	SR	TR	MS
Order of Truncation Error		3.9(-7)	6.3(-4)	2.4(-8)	1.6(-4)	1.5(-9)	3.9(-5)	1.0(-15)
DHL	v(1)	.3642397(+0)	.3641254(+0)	.3642428(+0)	.3642131(+0)	.3642430(+0)	.3642355(+0)	.3642430(+0)
DHL	$C_s(1)$	.2745445(+1)	.2746306(+1)	.2745422(+1)	.2745646(+1)	.2745420(+1)	.2745476(+1)	.2745420(+1)
DHL	$C_u(1)$	.1372722(+1)	.1373153(+1)	.1372711(+1)	.1372823(+1)	.1372710(+1)	.1372738(+1)	.1372710(+1)
AVR	v(0)	-.5670590(+1)	-.5670710(+1)	-.5670587(+1)	-.5670618(+1)	-.5670587(+1)	-.5670594(+1)	-.5670587(+1)
AVR	$C_s(0)$	.1049184(+1)	.1049165(+1)	.1049184(+1)	.1049180(+1)	.1049184(+1)	.1049183(+1)	.1049184(+1)
AVR	$C_u(0)$	.4639605(-1)	.4639582(-1)	.4639606(-1)	.4639600(-1)	.4639606(-1)	.4639604(-1)	.4639606(-1)
CD	v(1)	.1280192(-1)	.1279552(-1)	.1280208(-1)	.1280063(-1)	.1280209(-1)	.1280173(-1)	.1280209(-1)
CD	$C_s(1)$	.3000898(-3)	.2788139(-3)	.3000759(-3)	.2946770(-3)	.3000758(-3)	.2987224(-3)	.3000758(-3)
CD	$C_u(1)$	.2882929(+1)	.2884037(+1)	.2882901(+1)	.2883142(+1)	.2882899(+1)	.2882960(+1)	.2882899(+1)

Model Problem with Jump Discontinuity

Table II

N Segments		40	40	80	80	160	160	5071
Integration Scheme		SR	TR	SR	TR	SR	TR	MS
Order of Truncation Error		3.9(-7)	6.3(-4)	2.4(-8)	1.6(-4)	1.5(-9)	3.9(-5)	1.0(-15)
DHL	v(1)	.4283244(+0)	.4275207(+0)	.4283376(+0)	.4281342(+0)	.4283386(+0)	.4282876(+0)	.4283387(+0)
DHL	$C_s(1)$	.2334679(+1)	.2339068(+1)	.2334607(+1)	.2335716(+1)	.2334602(+1)	.2334880(+1)	.2334601(+1)
DHL	$C_u(1)$	.1167340(+0)	.1169534(+0)	.1167304(+0)	.1167858(+0)	.1167301(+0)	.1167440(+0)	.1167301(+0)
AVR	v(0)	-.5605556(+1)	-.5606309(+1)	-.5605543(+1)	-.5605723(+1)	-.5605542(+1)	-.5605590(+1)	-.5605542(+1)
AVR	$C_s(0)$	.1043763(+1)	.1043814(+1)	.1043765(+1)	.1043777(+1)	.1043765(+1)	.1043768(+1)	.1043765(+1)
AVR	$C_u(0)$	.4673648(-1)	.4673640(-1)	.4673651(-1)	.4673650(-1)	.4673652(-1)	.4673651(-1)	.4673652(-1)
CD	v(1)	.1579934(-1)	.1573760(-1)	.1580013(-1)	.1578443(-1)	.1580019(-1)	.1579625(-1)	.1580020(-1)
CD	$C_s(1)$	.4667617(-1)	.4403120(-1)	.4669398(-1)	.4601938(-1)	.4669526(-1)	.4652573(-1)	.4669535(-1)
CD	$C_u(1)$	.2406177(+1)	.2413412(+1)	.2406084(+1)	.2407921(+1)	.2406077(+1)	.2406538(+1)	.2406077(+1)

Model Problem with Ramp