

模拟半导体器件的一种稀疏 矩阵及其算法*

任建民 张义门
(西安电子科技大学,西安)

摘要 本文提出了一种用于半导体器件数值分析的新颖的稀疏矩阵技术及其算法实现。文中详述了该稀疏矩阵的存储方式及计算过程,并与现有的稀疏矩阵技术作了比较,说明该稀疏矩阵用于半导体器件模拟时可以大大减少存储量和节省运算时间,实施也非常方便。文中还给出用该稀疏矩阵技术完成的几个算法,并给出计算实例,以说明该稀疏矩阵技术所需的时空特性。

关键词 半导体器件;稀疏矩阵技术;算法;CAD

一、引言

器件模拟是进行 VLSI 器件设计的一种有效方法。器件模拟就是通过计算机,在给定制件结构、初始掺杂分布等参数的基础上,求解描述器件物理特性的一组偏微分方程的过程。对这组偏微分方程求解将导致求解大型稀疏线性方程组。对于二维问题,方程组的阶数可达数千,三维问题则达数万。进行器件模拟要求解一组椭圆偏微分方程:描述由于电荷分布形成电势分布的 Poisson 方程和描述半导体内载流子运输规律的电流连续性方程。以矩形域上椭圆偏微分方程的第一边界问题为例:

$$\frac{\partial}{\partial x}(P(x, y)u_x) + \frac{\partial}{\partial y}(Q(x, y)u_y) - \sigma(x, y)u(x, y) = f(x, y)$$

采用五点差分, x 方向分为 N_x 份 ($N_x + 1$ 个点), y 方向分为 N_y 份 ($N_y + 1$ 个点), 离散化形式为

$$D_{ij}u_{ij} - L_{ij}u_{i-1,j} - R_{ij}u_{i+1,j} - T_{ij}u_{i,j+1} - B_{ij}u_{i,j-1} = S_{ij}$$

其中 D_{ij} , L_{ij} , R_{ij} , T_{ij} 和 B_{ij} , S_{ij} 为系数^[1]。该方程的离散化一般形成 $N = (N_x - 1) * (N_y - 1)$ 个代数方程,如 MINIMOS 程序^[10]其方程个数多至 60×60 。写成矩阵形式:

$$AU = B$$

但系数矩阵 A 中每行只有五个非零元素,结构为带状,如图 1 所示。

一般求解系数矩阵为满的 n 阶线性代数方程组要求 $O(n^2)$ 个存储单元和 $O(n^3)$ 次乘法、加法运算。倘若矩阵的非零元素总数 $\lambda = O(n)$,则称之为稀疏矩阵。对稀疏矩阵,可以利用稀疏性,降低存储量和运算次数。一般可使存储量和运算量相应地降至 $O(n)$ 。

* 1987 年 4 月 9 日收到,1988 年 7 月 25 日修改定稿。

器件模拟中产生的系数矩阵是典型的稀疏矩阵。二维问题非零元素约为 $5n$, 三维问题约 $7n$ 。矩阵结构对称, 元素可对称也可不对称。对该问题用一般的满秩矩阵求法, 因问题过大, 难于求解; 用一般的稀疏矩阵技术也存在存储量较大、时间较长、程序实施困难等弱点。

带状稀疏矩阵的最简单存储是对角存储, 要存储带内的许多零元素, 对于二维及三维器件模拟, 带内这些零元素远大于非零元素总数。Knuth 1968 年提出的索引稀疏存储技术^[2]、Knuth-Rheinboldt-Mesztenyi^[3] 1973 年提出的 KRM 环索引技术以及 Duff 1977^[4] 年的修正 KRM 环索引技术, 都要数个矩阵存储许多索引信息, 并且程序完成非常复杂。Sherman 1975^[5] 年提出压缩存储技术, 完全避免零元素的存储, R. E. Bank, et al. 在 Sherman 存储技术的基础上, 用两个一维矩阵分别存储非零元素及索引信息^[6]。本文在 Sherman 和 Bank 的基础上, 针对器件模拟导出的系数矩阵形式, 提出一种新的稀疏矩阵技术, 使得存储空间明显减少, 加快了计算时间, 程序的处理过程也比较简单。

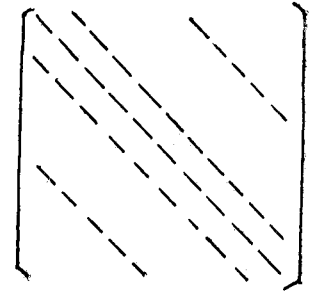


图 1 系数矩阵的带状结构

二、改进的稀疏矩阵技术

稀疏矩阵技术要求: 存储非零元素; 仅对非零元素操作; 稀疏矩阵的运算过程始终保持稀疏性。存储的元素要易于索引、运算。本文提出的稀疏矩阵技术是用一个一维数组 RSTORE 存储 A 的非零元素, 用另一个一维整数数组 IPOINT 指示 A 中元素的实际位置。一般对矩阵方程求解 (包括直接解法和迭代解法) 都归为将矩阵 A 作 $L + D + U$ 分解, 或 LDU 分解, 所以 RSTORE 的存储也分为三部。对 $N \times N$ 矩阵, RSTORE 的前 N 个元素存储 A 的对角部分。第 $N + 1$ 个元素作为判别条件用以判别矩阵的类型。若 $RSTORE(N + 1) < 0$, 为对角矩阵, 只需存储前 $N + 1$ 个元素; 若 $RSTORE(N + 1) = 0$, 为对称矩阵, 从第 $N + 2$ 个元素起按行存储上三角矩阵的非零元素; 若 $RSTORE(N + 1) = M$ (M 为上三角矩阵的非零元素个数), 矩阵 A 为非对称矩阵, 从第 $N + 2$ 个元素起仍按行存储上三角矩阵中的非零元素, 接着按列存储矩阵的下三角非零元素。

IPOINT 指示实际矩阵元素的行列位置。前 $N - 1$ 个元素用于指示 A 的上三角矩阵的起始元素在 RSTORE 中的位置, $IPOINT(N) = IPOINT(N - 1) + 1$ 。从第 $N + 1$ 个位置开始指示 A 中第一行非零元素的列坐标。

对矩阵 A (非对称矩阵):

$$A = \begin{bmatrix} 15 & 3 & 0 & 3 & 0 \\ 2 & 14 & 2 & 0 & 5 \\ 0 & 4 & 13 & 1 & 0 \\ 4 & 0 & 3 & 16 & 4 \\ 0 & 3 & 0 & 1 & 15 \end{bmatrix}$$

其一维存储为:

RSTORE = 15, 14, 13, 16, 15, 6, 3, 3, 2, 5, 1, 4, 2, 4, 4, 3, 3, 1

IPOINT = 7, 9, 11, 12, 13, 2, 4

在器件模拟中,一般对二维问题存储量为:RSTORE(\dots) $<5N-2$, IPOINT(\dots) $=N+2$, 总存储量小于 $6N$; 三维问题: RSTORE(\dots) $<7N-3$, IPOINT(\dots) $=N+3$, 总存储量小于 $8N$.

运用该稀疏矩阵结构,很容易完成下列稀疏算法: 对 $A = LDU$, 可完成 $AX = B$, $A^T X = B$ 及 $LX = B'$, $UX = B''$ 等算法. 完成 $AX = B$ 或 $A^T X = B$ 的算法如下:

N 为矩阵阶数; RSTORE, IPOINT 分别为 A 的一维非零元素存储和行列位置存储, $KL = 1$ 为对称矩阵, 否则为非对称矩阵.

Procedure (N , RSTORE, IPOINT, X , B , KL)

FOR $I = 1$ TO N DO

$B(I) = \text{RSTORE}(I) * X(I)$

 ISHIFT = RSTORE($N + 1$)

 IF (ISHIFT .GE. 0) THEN

 IF (KL , EQ, 1) THEN

 LSHIFT = ISHIFT

 MSHIFT = 0

 ELSE

 LSHIFT = 0

 MSHIFT = ISHIFT

 END IF

 FOR $I = 1$ TO $N - 1$ DO

 BEGIN

$IL = N$

 FOR $K = \text{IPOINT}(I)$, $\text{IPOINT}(I + 1) - 1$

 BEGIN

$IL = IL + 1$

$J = \text{IPOINT}(IL) + I - 1$

$B(I) = B(I) + \text{RSTORE}(K + \text{MSHIFT}) * X(J)$

$B(J) = B(J) + \text{RSTORE}(K + \text{LSHIFT}) * X(I)$

 END

 END

 ENDIF

END

由程序可见, 当 $\text{RSTORE}(N + 1) < 0$ 时, 只计算对角部分, 否则计算非对角元素. 当 $KL = 1$ 为对称矩阵时, 计算只应用上三角元素, 否则为非对称矩阵, 需存储所有非零元素. $\text{RSTORE}(N + 1)$ 在对称情况下为零, 在非对称情况下为上三角矩阵非零元素的个数, 使得上述算法变得非常方便. 并可见, 在同一循环内可完成二次矩阵元素乘积, 对

大型矩阵而言这大大减少了比较次数,节省了运算时间. 对 $A = L + D + U$, 在上述算法中除去 $B(J) = B(J) + RSTORE(K + LSHIFT) * X(I)$ 语句后, 就可完成上三角矩阵乘积 $(D + U)X = B (KL = 1)$ 或 $(L + D)X = B (KL = \text{其它})$.

三、运用该稀疏矩阵技术的几个算法及计算实例

在器件模拟中, 常常要用 CG 法(共轭梯度法)、BCG 法(双共轭梯度法)^[7] 以及 ICCG 法(不完全 Cholesky 分解——共轭梯度法)^[8] 来求解大型线性代数方程组. 利用本文提出的稀疏矩阵结构很容易完成上述算法过程.

- CG 法 (1) 选择解的初值 X_0 ,
 (2) 计算 $r_0 = B - AX_0$, $P_0 = r_0$,
 (3) 迭代求解:

$$\alpha_k = \frac{r_k^T r_k}{P_k^T A P_k}, \quad X_{k+1} = X_k + \alpha_k P_k$$

$$r_{k+1} = r_k - \alpha_k A P_k, \quad \beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$$

$$P_{k+1} = r_{k+1} + \beta_k P_k$$

- BCG 法 (1) 选择解的初值 X_0 ,
 (2) 计算 $r_0 = B - AX_0$, 且 $P_0^* = r_0^* = P_0 = r_0$,
 (3) 迭代求解:

$$\alpha_k = (r_k, r_k^*) / (A P_k, P_k^*)$$

$$X_{k+1} = X_k + \alpha_k P_k$$

$$r_{k+1} = r_k - \alpha_k A P_k$$

$$r_{k+1}^* = r_k^* - \alpha_k A^T P_k^*$$

$$\beta_k = (r_{k+1}, r_{k+1}^*) / (r_k, r_k^*)$$

$$P_{k+1} = r_{k+1} + \beta_k P_k$$

$$P_{k+1}^* = r_{k+1}^* + \beta_k P_k^*$$

由上述算法可见, CG 或 BCG 算法中对稀疏系数矩阵的运算只有 AP 或 $A^T P$, 利用前面的 Procedure 即可完成.

ICCG 法对 A 进行不完全 Cholesky 分解 $A = L^T L$, 保持原系数矩阵 A 的稀疏性, 再用 CG 法加速收敛. 算法如下:

- (1) 选择解的初值 X_0 ,
 (2) 计算 $r_0 = B - AX_0$, $P_0 = (LL^T)^{-1} r_0$,
 (3) 迭代求解:

$$\alpha_k = (r_k, [LL^T]^{-1} r_k) / (P_k, A P_k)$$

$$X_{k+1} = X_k + \alpha_k P_k$$

$$r_{k+1} = r_k - \alpha_k A P_k$$

$$\beta_k = (r_{k+1}, [LL^T]^{-1} r_{k+1}) / (r_k, [LL^T]^{-1} r_k)$$

$$P_{k+1} = [LL^T]^{-1}r_{k+1} + \beta_k P_k$$

其中要计算 $[LL^T]^{-1}r_0 = L^{-T}L^{-1}r_0$, 利用本文提出的稀疏存储结构很容易得到计算三角矩阵的逆与某一矢量的乘积的稀疏算法。

利用本文提出的稀疏矩阵技术求解前面给出的椭圆偏微分方程。其中 $P(x, y) = Q(x, y) = 1, \gamma(x, y) = 0$, 程序在 VAX-11/750 机上运行。

当 $\sigma(x, y) = 0$ 时, 为拉普拉斯方程, $\sigma(x, y) \neq 0$, 为泊松方程, 本文取 $\sigma(x, y) = 1$, 利用 CG 法和 BCG 法求解, 收敛标准为 $EPS = 10^{-15}$ ($r = B - AX$), 初值为 $U_{ij} = 10^4$, 解为 $U_{ij} = 0$ ($i, j = 1, 2, \dots, N$)。拉普拉斯方程和泊松方程的系数矩阵对称时, 计算所需的时空特性如表 1 所示。

表 1

网格划分	矩阵阶数 N	对称五点拉普拉斯方程				对称五点泊松方程			
		存储量(单元)		CPU 时间 (s)		存储量(单元)		CPU 时间 (s)	
		实存储量	整存储量	CG 法	BCG 法	实存储量	整存储量	CG 法	BCG 法
10×10	100	290	102	2.79	4.19	290	102	4.12	3.77
20×20	400	1180	402	22.77	39.20	1180	402	27.42	39.08
30×30	900	2670	902	105.79	159.05	2670	902	105.78	158.02
40×40	1600	4760	1602	488.41	737.99	4760	1602	1481.71	2228.22

由表 1 可见, 该稀疏矩阵技术在对称情况下占用内存量很小。对 $N = 400$ 阶的矩阵, 需要的存储量为: 实存储为 1180 单元, 整存储量为 402 单元。表中还给出了用 CG 法和 BCG 法求解拉普拉斯方程和泊松方程的时间特性。可见当方程阶数增加时, 泊松方程的求解需要较多的 CPU 时间。

表 2 给出求解非对称情况下五点拉普拉斯时, 本文的稀疏矩阵与 Bank, et al. 的稀疏矩阵所需存储量的比较。

表 2

网格划分	矩阵阶数 N	非对称五点拉普拉斯方程			
		本文稀疏矩阵的存储量		Bank, et al. 稀疏矩阵的存储量	
		实存储量	整存储量	实存储量	整存储量
10×10	100	479	102	1215	572
20×20	400	1959	402	7427	2643
30×30	900	4439	902	20277	6266
40×40	1600	7919	1602	40695	11461

由表 2 可见, 对 $N = 400$ 阶矩阵, 本文的稀疏矩阵需实存储量 1959 单元, 整存储量 402 单元。Bank, et al. 的稀疏矩阵需实存储量 7427 单元, 整存储量 2643 单元。可见对半导体器件模拟, 本文提出的稀疏矩阵技术大大减少了存储空间, 并且由于 A 的存储已分为对角部分和上、下三角部分, 这很适于分别进入对角、上三角和下三角矩阵的各种迭代过程

参 考 文 献

- [1] R. S. Varga, *Matrix Iterative Analysis*, Englewood Cliff, NJ: Prentice-Hall, 1962, pp. 161—208.
- [2] D. E. Knuth, *The Art of Computer Programming*, Vol. 1, *Fundamental Algorithms*, Addison-Wesley: Reading, MA, pp. 14—65.
- [3] W. C. Rheinboldt, C. K. Mesztenyi, *Programs for the Solution of Large Sparse Matrix Problems Based on the Arc-Graph Structure*, Computer Science Center, University of Maryland, College Park, MD. Technical Report 'TR-262.
- [4] I. S. Duff, *Proc. IEEE* 65(1977), 500—535.
- [5] A. H. Sherman, *On the Efficient Solution of Sparse Systems of Linear and Nonlinear Equation*, Ph. D. Thesis, Department of Computer Science, Yale University, New Haven, CT. Report No. 46, pp. 31—50.
- [6] R. E. Bank, et al., *Computational Aspects of Semiconductor Device Simulation, Process and Device Modeling*, Elsevier Science Publishers B. V. (North-Holland), 1986, pp. 229—261.
- [7] Y. Saad, *SIAM J. Numer. Anal.*, 19(1982) 3, 485—506.
- [8] J. A. Meijerink, H. A. Vamder Vorst, *Math. of Computation*, 31(1977)137, 148—162.
- [9] P. Sergio, *Sparse Matrix Technology*, Academic Press (Harcourt Brace Jovanovich, Publishers), 1984, pp. 8—37, pp. 288—300.
- [10] S. Selberherr, A. Schütz, H. W. Pötl, *IEEE Trans. on ED*, ED-27(1980)8, 1540—1550.

A SPARSE MATRIX TECHNIQUE FOR SIMULATING SEMICONDUCTOR DEVICES AND ITS ALGORITHMS

Ren Jianmin Zhang Yimen

(Xidian University, Xi'an)

Abstract A novel sparse matrix technique for the numerical analysis of semiconductor devices and its algorithms are presented. Storage scheme and calculation procedure of the sparse matrix are described in details. The sparse matrix technique in the device simulation can decrease storage greatly with less CPU time and its implementation is very easy. Some algorithms and calculation examples to show the time and space characteristics of the sparse matrix are given.

Key words Semiconductor devices; Sparse matrix technique; algorithm; CAD