Optimum Scaling of Non-Symmetric Jacobian Matrices for Threshold Pivoting Preconditioners

C. Fischer and S. Selberherr
Institute for Microelectronics, Technical University of Vienna, A-1040 Vienna, Austria

Abstract

A new scaling method for non-symmetric matrices is presented. The scaling is based on purely mathematical considerations. It is highly adapted to the demands of incomplete LU factorization preconditioners with numerical dropping strategy. These preconditioners are the best state-of-the-art instruments for solving ill-conditioned linear systems with very large numbers of equations. The new scaling method causes a considerable improvement in the capabilities of these preconditioners, both in execution speed and robustness.

1 The Scaling Problem

In process and device simulation, one generally faces the situation of mixed physical quantities in the solution vector $\mathbf{x}$ and the right hand side vector $\mathbf{b}$ of the linearized coupled problem

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b}. \tag{1}$$

Mixed physical quantities present a problem as to the choice of an optimum scaling strategy. Scaling the equations is by far not obvious, and various scaling strategies have been developed and applied in semiconductor process and device simulation. The numbers appearing in the final Jacobian matrix of the system depend heavily on the physical units which have been chosen when representing physical quantities by dimensionless numbers. All established scaling strategies have one disadvantage in common: they require a lot of *a priori* knowledge about the solution components, and they do not guarantee improvement in the condition of the problem.

The problem of scaling an arbitrary matrix has been well treated in connection with pivoting in Gaussian elimination of full matrices. There, the accuracy to which the system of equations can be solved depends strongly on the sequence of pivot elements which are chosen. Row and column equilibration methods are frequently used to scale the matrix elements before pivoting. The scaling has no direct effect on the elimination process once the pivoting sequence is fixed.

A very similar view on the matter can be taken in preconditioning by incomplete LU factorization with threshold dropping [1, 2]. Here, the problem is not to find an optimum elimination sequence, but rather to decide which elements in a row/column are to be dropped. Again, once this decision is fixed the scaling has no further influence on the elimination process.

TH0640-3/94/0000-0123 $3.00

2 The Scaling Method

The crucial point in the dropping strategy is to judge the relative weight of the off-diagonals correctly. The relative contribution of an off-diagonal a_{ij} in the later elimination process of row j will depend on the product of the off-diagonal a_{ij} and the off-diagonal a_{ji}:

$$\begin{pmatrix} & \vdots & & \vdots & \\ \cdots & a_{ii} & \cdots & a_{ij} & \cdots \\ & \vdots & & \vdots & \\ \cdots & a_{ji} & \cdots & a_{jj} & \cdots \\ & \vdots & & \vdots & \end{pmatrix} \xrightarrow{\text{elimination of row } j} \begin{pmatrix} & \vdots & & \vdots & \\ \cdots & a_{ii} & \cdots & a_{ij} & \cdots \\ & \vdots & & \vdots & \\ \cdots & 0 & \cdots & a_{jj} \cdot (1 - \frac{a_{ij}}{a_{ii}} \cdot \frac{a_{ji}}{a_{jj}}) & \cdots \\ & \vdots & & \vdots & \end{pmatrix} \tag{2}$$

As can be seen from the matrix substructures in (2), the relative weight $(a_{ij} \cdot a_{ji})/(a_{ii} \cdot a_{jj})$ of the additional contribution is not influenced by row or column scaling. The scaling, however, influences the dropping strategy during the elimination of line i, since the part a_{ij}/a_{ii} of the fraction can be given any value by proper column scaling. The question now is to find a scaling which allows to decide which off-diagonal elements of row i will be most important in the further elimination process, and to decide this by the relative size in row i only.

To be able to compare the off-diagonals more easily, the diagonal elements (a_{ii}, a_{jj}, ...) are scaled to unity in a first phase. This is achieved by a set of r_i, c_i fulfilling Equation 3.

$$r_i \cdot a_{ii} \cdot c_i = 1 \tag{3}$$

Since this is only the first phase, the choice of the scaling method is not a critical point. Depending on the problem, one could preset the r_i or the c_i with unity or begin with any other predefined scaling for one of these sets. We use symmetric scaling vectors ($r_i = c_i$). The effect of phase 1 is that the contributions are determined only by the products $a_{ij} \cdot a_{ji}$.

In the second phase, the row scaling vector $\mathbf{r}$ and the column scaling vector $\mathbf{c}$ are modified. The basic idea is to make a_{ij} the same size as a_{ji}. This can be seen as a solution to the problem of minimizing the sum of $|a_{ij}|$ and $|a_{ji}|$ while keeping their product constant. In a more general formulation, and taking also into account the contributions to off-diagonals, the problem is to minimize

$$\sum_i \sum_{j \neq i} |r_i| \cdot |a_{ij}| \cdot |c_j| \tag{4}$$

under the constraint of Equation 3.

The problem is nonlinear, and several strategies could be applied to achieve a solution. For instance, iterations can be done over a linearization like $r_i = r_{i,o} \cdot (1 + \rho_i)$, $c_i = c_{i,o}/(1 + \rho_i) \approx c_{i,o} \cdot (1 - \rho_i)$, which brings the problem into a quadratic form of the ρ_i. This problem can be solved by well-known methods.

We propose a simple relaxation strategy which has proven quite satisfactory. A relaxation loop over pairs of row i and column i is used to modify the scaling. For each pair r_i and c_i, the sum of the off-diagonals in row i and column i is minimized while keeping the product $r_i \cdot c_i$ constant according to Equations 5 and 6.

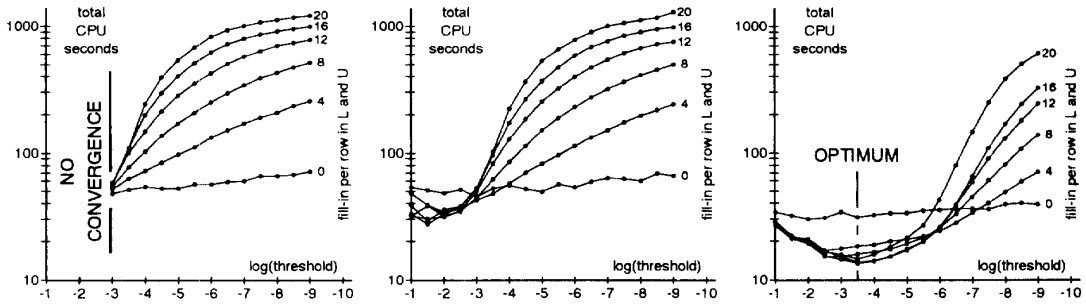
$$R_i = \sum_{j \neq i} |a_{ij}| \cdot |c_j| \qquad C_i = \sum_{j \neq i} |r_j| \cdot |a_{ji}| \tag{5}$$

$$r_{i,\mathrm{new}} = \sqrt{\frac{C_i}{R_i \cdot |a_{ii}|}} \qquad c_{i,\mathrm{new}} = \sqrt{\frac{R_i}{C_i \cdot |a_{ii}|}} \tag{6}$$

A loop of i over the whole matrix modifies all the scaling coefficients and decreases the total sum of off-diagonals. In the beginning, the maximum of the sum of off-diagonals in a row/column decreases very rapidly. Already a very few iterations result in a scaling which shows considerable improvement in the subsequent preconditioning and solving process. The whole relaxation scheme is very inexpensive in terms of CPU time consumption and coding effort.

3 Results

The main effect of the new method is a significant decrease in preconditioning time, since the new scaling enables the preconditioner to factorize a matrix with much higher drop thresholds and less fill-in than without scaling while maintaining the quality of the LU decomposition. This effect can be best observed when different physical quantities are used in the same solution vector and right hand side. But even for homogeneous systems of equations (like a single discretized continuity equation) speed-ups of up to 30 percent have been observed. The improvement of course depends on the original scaling; even for good initial states improvements show up (Fig. 1), whereas for poor initial scalings, the new method makes solution possible at all. It should be noted that the relaxation will not give any improvement if the matrix is already symmetric, since a symmetric scaling is already the optimum suppression of the off-diagonals.



a) No scaling.

b) Phase 1 scaling (diagonal unity).

c) Phase 1 + 2 scaling, 6 relaxation loops.

Figure 1: Total CPU time for scaling, preconditioning, and solving a system of 30207 equations with mixed quantities (potential, carrier concentrations) on a DEC ALPHA 7640. Parameter of the curves is the allowed preconditioner fill-in per matrix row; the dropping threshold in the range from 10^{-1} to 10^{-9} is plotted on the abscissa. A BiCGStab [3,4] algorithm was used to solve the system to the relative accuracy of 10^{-8}. Without scaling, the system could not be solved for thresholds above 10^{-3}. With scaling remarkable reduction and broadening in the overall minimum can be observed. Note the logarithmic scale on both axes.

4 Spin-offs

Total Decoupling
If the total system decouples completely into independent subsystems, the scaling within a subsystem will be as described above, while the relations of the scalings of the subsystems will maintain their original values. So decoupled systems should not be solved together.

Numerical Decoupling
If the total system contains values of a physical quantity which depends on but does not (heavily) influence the other quantities, this quantity will be suppressed by the scaling. That will lead to less exact solution of this quantity, but better stability. Some special iterations for this quantity might be desirable. Very small norms of the right hand side or solution vector for this quantity will indicate that the quantity might be solved separately.

$$\begin{pmatrix} \mathbf{A}_{\psi\psi} & \mathbf{A}_{\psi n} & \mathbf{A}_{\psi p} \\ \mathbf{A}_{n\psi} & \mathbf{A}_{nn} & \mathbf{A}_{np} \\ \mathbf{A}_{p\psi} & \mathbf{A}_{pn} & \mathbf{A}_{pp} \end{pmatrix} \cdot \begin{pmatrix} \delta\psi \\ \delta \mathrm{n} \\ \delta \mathrm{p} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_{\psi} \\ \mathbf{b}_{n} \\ \mathbf{b}_{p} \end{pmatrix} \tag{7}$$

For instance, in a device where holes have no influence on the electrical behavior, the values in the submatrices $\mathbf{A}_{\psi p}$ and $\mathbf{A}_{np}$ of the (simplified) system matrix in Equation 7 are comparatively small. As a consequence, after scaling the hole concentration updates show norms of several orders of magnitude below the norms of the other quantities.

5 Acknowledgements

This work has been supported by Siemens Corporation at Munich, Germany; and Digital Equipment Corporation at Hudson, U.S.A..

References

[1] O. Heinreichsberger et al., *Practical Use of a Hierarchical Linear Solver Concept for 3D MOS Device Simulation*, Proc. SISDEP, Vol. 5, pp. 85–88 (1993).

[2] C. Pommerell, *Solution of Large Unsymmetric Systems of Linear Equations*, PhD Thesis, Swiss Federal Institute of Technology, Zürich (1992).

[3] H. A. van der Vorst, *A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Systems*, SIAM J.Sci.Stat.Comp., Vol. 13, No. 2, pp. 631–644 (1992).

[4] H. A. van der Vorst et al., *Further Improvements in Nonsymmetric Hybrid Iterative Methods*, Proc. SISDEP, Vol. 5, pp. 85–88 (1993).