OPTIMIZING THE INEXACT NEWTON KRYLOV METHOD USING COMBINATORIAL APPROACHES

Marcelo Carrion, Brenno Lugon
Maria Cristina Rangel, Lucia Catabriga, Maria Claudia Silva Boeres
Universidade Federal do Espírito Santo - Brazil

{marcelo.tperc,brennolugon}@gmail.com,{crangel,luciac,boeres}@inf.ufes.br

1 Introduction

Combinatorial Scientific Computing is an important interdisciplinary field combining issues from Combinatorial Optimization to solve efficiently Scientific Computing problems. In this work, we solve a 3D nonlinear problem using a Newton-type method that requires, at each step, the evaluation of a Jacobian matrix and the solving of a linear system. The Jacobian evaluation is optimized through matrix partitioning and a matrix reordering scheme is used to accelerate the convergence of the preconditioned iterative GMRES solver, as shown in Figure 1.

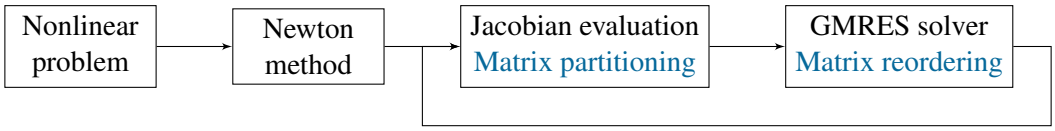

Figure 1: Problem flow and optimizations (in blue) scheme.

2 Inexact Newton Krylov Method

The Newton-type algorithm requires the solution of linear systems at each iteration. When iterative Krylov methods are used to solve these linearized systems, the resulting methods are known as Inexact Newton-Krylov methods. Inexact Newton methods are especially well suited for large-scale problems and have been used very successfully in many applications. Their success depends mainly on three factors: (i) quality of initial Newton step, (ii) robustness of Jacobian evaluation and (iii) robustness of the Krylov iterative method, the last two which we focus on.

3 Optimizations

Sparsity in the derivative matrices can be exploited to compute the nonzero entries efficiently. In finite difference approximations, efficiency can be achieved by partitioning the columns of a sparse Jacobian matrix into a few groups of structurally orthogonal ones. In each group, no two columns have a nonzero in a common row and all nonzeros can be estimated through one finite difference operation, improving the whole evaluation time of the matrix. This partitioning problem is generally modeled and solved as a graph coloring problem [1], but a different approach is used for the matrices treated here. Since they have a trivial sparsity pattern, we drew inspiration from [2] to obtain a set expression, presented in the next section, that defines a minimum partitioning.

Sparse matrix reordering schemes with the purpose of minimizing the bandwidth were also implemented in this work, as alternatives to speed up the convergence of the preconditioned GMRES method. The preconditioning technique used is based on incomplete LU factorization ILU(p), where p is the *level of fill* used to control the number of new elements generated during the process. The preconditioning calculations are optimized when we use a matrix reordering, once with the reduced bandwidth less floating-point operations are made. For the tests we adopted the Sloan reordering [3] for its good solutions achieved for the matrices generated in our specific problem.

4 Test Problem

For the numerical experiments we use a 3D heat transfer problem defined by the nonlinear differential equation:

$$-\nabla \cdot (K(u)\nabla u) = 0 \quad \text{in} \quad \Omega = (0,1) \times (0,1) \times (0,1) \tag{1}$$

where u is the temperature, the thermal conductivity is considered as $K(u) = 0.0000002u^2 + 0.00001u + 0.001$ and the boundary conditions are $u(x,y,0) = u(x,0,z) = u(1,y,z) = 10$ and $u(x,y,1) = u(x,1,z) = u(0,y,z) = 100$. Consider a discretization of Ω into an uniform grid with $N = n \times m \times l$ unknowns points, respectively, in the x, y, z directions. We approximate the derivatives by combining forward, backward and centered finite differences, arriving to the nonlinear system of equations $F(\boldsymbol{u}) = 0$, where $F : \mathbb{R}^N \to \mathbb{R}^N$ is a nonlinear vector function, $\boldsymbol{u} = (u_1, u_2, \ldots, u_N)^T$ is the unknown vector and each component of F depends only on the seven unknowns $u_{I-m\cdot n}$, u_{I-n}, u_{I-1}, u_I, u_{I+1}, u_{I+n} and $u_{I+m\cdot n}$, for $I = 1, 2, \ldots, N$. Each iteration of the Newton's method is given by $\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \boldsymbol{s}^k$, where $\boldsymbol{s}^k$ is calculated by the solution of the linear system $J(\boldsymbol{u}^k)\boldsymbol{s}^k = -F(\boldsymbol{u}^k)$. The Jacobian matrix

J represents the variation of F with respect of $\boldsymbol{u}$, that is considered as a forward finite difference approximation. We may terminate the iteration when the relative nonlinear residual $\|F(\boldsymbol{u}^k)\|/\|F(\boldsymbol{u}^0)\|$ is small. Given its heptadiagonal structure, J can be partitioned into seven groups of structurally orthogonal columns, determined by $G_p = \{i+n(j-1)+m\cdot n(k-1) \mid 1\leq i\leq n, 1\leq j\leq m, 1\leq k\leq l, (i+3j+2k) \bmod 7 = p\}$, for $p = 0,\ldots,6$.

5 Experimental Results

For the described problem, we developed programs in the C language. All computational tests were run on an Intel Core i5-3570 3.40GHz $\times 4$ machine with 4GB of RAM under Ubuntu 12.04. To store the sparse matrices derived from the problem we use an optimized storage scheme called Compressed Sparse Row (*CSR*). Eight problem instances were considered and Table 1 shows, for each instance, its dimension N, the p parameter to the ILU(p) preconditioner and the CPU time, in seconds, for computing the solution in four cases: with no optimization (T_{NO}), using only the matrix partitioning (T_{MP}) and the matrix reordering (T_{MR}) techniques and finally with both optimizations (T_{OP}). The values in parentheses on columns T_{MP}, T_{MR}, T_{OP} indicate the percentage time reduction ($red\%$) from time T_{NO}. For all instances, the optimized Jacobian evaluation performed only seven finite difference operations, as opposed to N in the regular computation. Figure 2 shows the approximate solution obtained for a mesh with 100.000 unknowns.

Instance	N $(n\times m\times l)$	p	T_{NO}	T_{MP} $(red\%)$	T_{MR} $(red\%)$	T_{OP} $(red\%)$
1	10.000 (100×10×10)	5	23,8	12,5 (47,6%)	16,4 (31,1%)	5,1 (78,6%)
2	10.000 (100×10×10)	10	196,7	185,5 (5,7%)	42,7 (78,3%)	31,4 (84,0%)
3	50.000 (200×50×5)	5	340,6	84,7 (75,1%)	300,7 (11,7%)	45,6 (86,6%)
4	50.000 (200×50×5)	10	1.911,9	1.655,3 (13,4%)	354,9 (81,4%)	100,5 (94,7%)
5	100.000 (100×50×20)	5	1.784,0	255,6 (85,7%)	1.698,8 (4,8%)	178,7 (90,0%)
6	100.000 (100×50×20)	10	6.027,5	4.499,5 (25,4%)	2.255,8 (62,6%)	733,7 (87,8%)
7	300.000 (200×50×30)	5	15.991,3	1.600,4 (90,0%)	15.679,1 (2,0%)	1.344,8 (91,6%)
8	300.000 (200×50×30)	10	29.148,3	14.753,7 (49,4%)	17.522,7 (39,9%)	3.186,3 (89,1%)

Table 1: Set of chosen parameters and computational results.

6 Analysis and Conclusion

Examining tests 1 through 8, we observe a higher level of fill-in induces a higher contribution of the reordering strategy on the final time reduction, while on the lower level the matrix partitioning strategy is responsible for the biggest improvement. Furthermore, by increasing the dimension of the problem but maintaining the same level of fill-in, the reduction achieved by the Jacobian optimization increases while the reduction achieved by the reordering decreases. Nevertheless, the final time reduction is always higher than 78%, showing the strength in the use of the optimization strategies. It is worth mentioning that the final execution time is lower when using ILU(5). The preliminary results obtained in this work show that the optimization strategies imposed on the Jacobian evaluation and the GMRES solver significantly reduced the final execution time for the benchmark 3D nonlinear heat transfer problem.

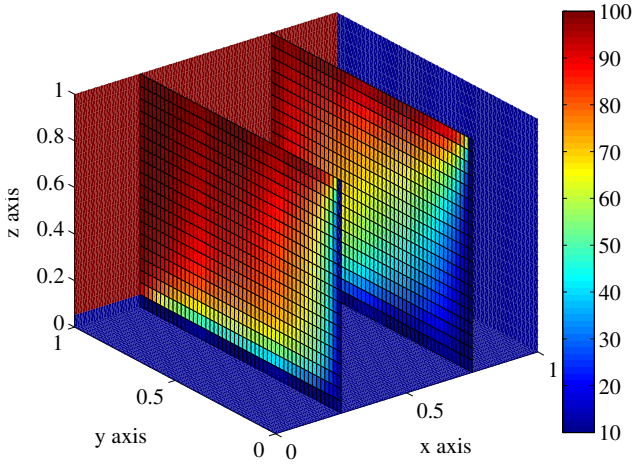

Figure 2: Temperature distribution for a 100.000 mesh.

Acknowledgement

This work was partly supported by FAPES n^o 48511579/2009, CNPq 552630/2011-0 and CNPq 307020/2012-6.

References

[1] A. H. Gebremedhin, F. Manne, and A. Pothen, "What color is your jacobian? graph coloring for computing derivatives," *SIAM REV*, vol. 47, pp. 629–705, 2005.

[2] D. Goldfarb and P. L. Toint, "Optimal estimation of jacobian and hessian matrices that arise in finite difference calculations," *Mathematics of Computation*, vol. 43, pp. 69–88, 1984.

[3] S. W. Sloan, "An algorithm for profile and wavefront reduction of sparse matrices," *Internacional Journal for Numerical Methods in Engineering*, vol. 23, pp. 239–251, 1986.