Tu LHR1 12

Exploring the Use of SPIKE-based Solvers on Large Electromagnetic Modeling

S. Rodriguez Bernabeu* (Barcelona Supercomputing Center), V. Puzyrev (Barcelona Supercomputing Center), M. Hanzich (Barcelona Supercomputing Center) & S. Fernández (Repsol Technology Center)

SUMMARY

Frequency-domain seismic and electromagnetic modeling requires solving the linear systems resulting from the discretization of the corresponding time-harmonic equations. Geophysical inversion is typically performed using several discrete frequencies and multiple (up to tens of thousands) source/receiver combinations. Limitations of classical direct and iterative sparse linear solvers have caused the development of the so-called hybrid methods that can be viewed as an intermediate approach between the direct and iterative methods. We present an efficient parallel solver based on the SPIKE algorithm. Several examples in frequency domain electromagnetic modeling illustrate the computational efficiency of the developed method in terms of memory demand and floating-point operations. Multiple sources can be efficiently handled by employing sparse direct solvers in the factorization of diagonal blocks of the system matrix. Based on the divide and conquer idea, this kind of algorithms exposes different parallelism levels, being suitable to take advantage of multiple accelerator devices. The SPIKE solver partially overcomes the fill-in problem of direct solvers, allowing to solve much larger domains on the same system.
Introduction

Sparse linear systems arise in many areas of geophysics and engineering. For example, frequency-domain seismic and electromagnetic modelling requires solving the linear systems resulting from the discretization of the corresponding time-harmonic equations. Geophysical inversion is typically performed using several discrete frequencies and multiple (up to tens of thousands) source/receiver combinations. Limitations of classical direct and iterative sparse linear solvers have caused the development of the so-called hybrid methods that can be viewed as an intermediate approach between the direct and iterative methods (e.g. (Sourbier et al., 2011)). In this paper, we present an efficient parallel solver based on the SPIKE algorithm and show examples of its application to electromagnetic modeling and other challenging problems from sparse matrix benchmarks. Different iterative solvers are tested and compared with state of the art direct solvers and our implementation of the SPIKE algorithm. Multisource modelling can be efficiently performed by employing parallel sparse direct solvers in the factorization of diagonal blocks of the system matrix. Some examples illustrate the computational efficiency of the proposed solver in terms of memory demand and floating-point operations.

Theory

In many science and engineering applications, fast iterative solvers, such as Krylov subspace methods are the common approach to solve the large linear systems that appear in most applications of scientific computing. Krylov solvers allows to solve large linear systems with low memory requirement and can be implemented easily on shared and distributed memory architectures. However, their convergence depends on the condition number of the coefficient matrix. For highly non-linear problems, such as geophysical inversion, an effective preconditioning is mandatory in order to achieve the needed tolerance level. Determination of a suitable preconditioner is not a trivial task since it depends on the problem at hand, and in some cases, its calculation can be computationally more expensive than the solving of the original system itself.

As mentioned before, in order to increase the quality of the inversion, it is needed to solve the problem repeatedly for a large number of sources/receivers. Thus, a numerical technique that allows us to solve a linear system with multiple right-hand sides (RHS) at once can be extremely efficient. Most Krylov space solvers can be generalized to block Krylov space solvers in order to handle multiple RHS of the system, but their stability requires additional effort. While these methods are computationally beneficial for the problems with several tens of RHS (Puzyrev and Cela, 2015), difficulties arise when the number of RHS is of hundreds or thousands.

Direct methods for solving general linear systems are commonly based on the LU decomposition that represents the coefficient matrix $A$ as the product of lower and upper triangular matrices, $A = LU$. Consequently, solving the linear system can be achieved by solving two triangular systems. When the coefficient matrix is sparse, the triangular factors $L$ and $U$ typically have non-zero entries in many more locations than $A$ does. This is known as fill-in, and results in a superlinear grow in the memory and time requirements to solve a sparse system with respect to the size of the system. Despite a high memory requirement, direct methods are nowadays used in many real applications due to their robustness. State of the art direct solvers usually involve four phases: ordering, symbolic factorization, factorization and triangular solving. A fifth phase of iterative refinement is sometimes used after the solution phase to improve the accuracy of the solution. In applications requiring the solution of several linear system with the same coefficient matrix and multiple RHS vectors, the factorization of $A$ can be amortized over several inexpensive triangular solves. However, the use of direct solvers becomes too expensive in terms of memory beyond a certain problem size.

The SPIKE solver

The SPIKE algorithm, introduced back in the late seventies (Sameh and Kuck, 1977, 1978), originally was targeted for very large sparse diagonally dominant narrow banded systems. Later on, new ideas adapted the solver to some special cases such as diagonally and non-diagonally dominant systems and
positive definite matrices (Polizzi and Sameh, 2007), increasing its overall performance and scalability.

The SPIKE algorithm is a flexible algorithm based on the divide and conquer idea. Parallelism is extracted by decoupling the blocks along the diagonal, solving them independently, and then reconstructing the system by solving a reduced system. This allows SPIKE to be efficient and scalable on large-scale distributed multicore systems.

Consider solving the linear system $Ax = f$, where $A$ is a non-singular $n \times n$ matrix, of bandwidth $b$ much less than $n$, and $f$ is the $n \times s$ matrix containing $s$ RHS. The decomposition phase segments $A$ into $p$ square sub-matrices along the diagonal 1a and the $B_l$ and $C_l$ blocks, containing the elements of $A$ along the diagonal that are outside of $A_l$.

The factorization stage consists of the factorization of $A$ into $D$ and $S$ matrices (Fig. 1b and Fig. 1c). The $V_l$ and $W_l$ matrices may be though of as the coupling between contiguous $A_l$ blocks, formed as:

$$A_lV_l = \begin{bmatrix} 0 \\ B_l \\ \end{bmatrix} \rightarrow A_l^{-1}\begin{bmatrix} 0 \\ B_l \\ \end{bmatrix} = V_l$$

$$A_lW_l = \begin{bmatrix} C_l \\ 0 \\ \end{bmatrix} \rightarrow A_l^{-1}\begin{bmatrix} C_l \\ 0 \\ \end{bmatrix} = W_l$$

After the factorization comes the solving stage. Being $Ax = DSx = f$, we have to solve $Dy = f$ and $Sx = y$. For the $Sx = y$ system, the top and bottom parts of the $x_l$ and $y_l$ are decoupled from their middle section. Thus, is is possible to extract a reduced system that will give us a the values of $x_{i,t}$ and $x_{i,b}$ sub-vectors. We can use this partial solution to retrieve the rest of the values of $x$.

$$x_{1,m} = y_{1,m} - V_{1,m}x_{2,t}$$

$$x_{i,m} = y_{i,m} - \left( V_{i,m}x_{i+1,t} + W_{i,m}x_{i-1,b} \right)$$

$$x_{p,m} = y_{p,m} - V_{p,m}x_{(p-1),b}$$

**Numerical experiments**

We consider four iterative methods that are widely used for this class of problems: the biconjugate gradient method (BiCG), the biconjugate gradient stabilized (BiCGSTAB), quasi-minimal residual (QMR)
Beyond the time reduction, the algorithm presents
complete solver implementation system (Castillo
been tested on matrices arising from electromagnetic
simulations in geophysics and other two challenging
problems from the Matrix Market Collection. The
numerical experiments show that the recursive
decomposition implementation of the SPIKE solver offers
large performance improvements respect to both the
iterative and direct solvers tested for all problems. On
small problems, SPIKE about the same time than
SuperLU and Pardiso. However, on large problems, our
implementation outperforms these state of the art direct
solvers in both memory and time requirements. On
such cases, direct solvers run out of memory and can
not find the solution due to the fill in problem. Also, for
this kind of problems, convergence of Krylov
solvers heavily depends on the preconditioner at hand.
Beyond the time reduction, the algorithm presents

Table 1 Execution time for tested solvers in seconds. Dual socket Intel Xeon E5-2670 with 64GB RAM. Drop tolerance for ILU 1x10^{-2}. Convergence criteria 1x10^{-8}.

<table>
<thead>
<tr>
<th></th>
<th>Inlet</th>
<th>LFAT</th>
<th>EFEM (small)</th>
<th>EFEM (large)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Raw Prec</td>
<td>Raw Prec</td>
<td>Raw Prec</td>
<td>Raw Prec</td>
</tr>
<tr>
<td>BiCG</td>
<td>- 0.685</td>
<td>- -</td>
<td>41.712</td>
<td>- -</td>
</tr>
<tr>
<td>BiCGSTAB</td>
<td>- 0.535</td>
<td>- -</td>
<td>1.788</td>
<td>23.428</td>
</tr>
<tr>
<td>QMR</td>
<td>- -</td>
<td>- -</td>
<td>53.374</td>
<td>- -</td>
</tr>
<tr>
<td>GMRES(20)</td>
<td>- 0.582</td>
<td>- -</td>
<td>- -</td>
<td>- -</td>
</tr>
<tr>
<td>Pardiso</td>
<td>0.159</td>
<td>0.128</td>
<td>1.611</td>
<td>-</td>
</tr>
<tr>
<td>SuperLU</td>
<td>0.168</td>
<td>0.134</td>
<td>1.663</td>
<td>-</td>
</tr>
<tr>
<td>SPIKE</td>
<td>0.155</td>
<td>0.125</td>
<td>0.811</td>
<td>92.514</td>
</tr>
</tbody>
</table>

Figure 2 Sparsity pattern of EFEM matrices.

and the generalized minimal residual method (GMRES). Each of these methods was tested using incomplete LU decomposition (ILU) preconditioning on a set of large, sparse matrices problems. Two state of the art direct solver libraries SuperLU (Li, 2005) and Intel’s Pardiso (Schenk and Gärtner, 2004) have been tested as well. Our implementation of the SPIKE solver is compared against all of them. Empirical results are presented in Table 1 for each method.

EFEM matrices were extracted from an Edge Finite Element Method (EFEM) code for EM problems (Castillo et al., 2015), currently under development at the Barcelona Supercomputing Center. Figure 2 shows the sparsity pattern of a) the original matrix and b) the structure of the matrix after applying a reordering algorithm. For this study, we use an small matrix (61750 degrees of freedom) and a large system (3.8 millions degrees of freedom). We also choose other two challenging matrix problems -inlet and LFAT5000- from the Oberwolfach collection. These narrow-banded systems presents some features that we often find in other EM problems so we can stress the solvers on very different scenarios.

Conclusions

We have considered the use of the SPIKE poly-algorithm for the solution of large systems of linear equations arising from electromagnetic simulations in geophysics and other two challenging problems from the Matrix Market Collection. The numerical experiments show that the recursive decomposition implementation of the SPIKE solver offers large performance improvements respect to both the iterative and direct solvers tested for all problems. On small problems, SPIKE about the same time than SuperLU and Pardiso. However, on large problems, our implementation outperforms these state of the art direct solvers in both memory and time requirements. On such cases, direct solvers run out of memory and can not find the solution due to the fill in problem. Also, for this kind of problems, convergence of Krylov solvers heavily depends on the preconditioner at hand. Beyond the time reduction, the algorithm presents
other not so evident advantages with respect to more traditional algorithms: i) it handles multiple right-hand sides of the system as direct solvers ii) while reducing their memory demand iii) with an easy to code structure. Also, since a direct method is used on the factorization stage the SPIKE solver iv) does not suffer from convergence issues.

However, one of the greatest potential of the SPIKE solver lays on its ability to expose parallelism at different granularities. The inversion of diagonal blocks can be done asynchronously whether using different computational nodes or hardware accelerators. At a lower level, the factorization benefits from multi and many-core implementations of state of the art linear algebra libraries.

One of the drawbacks of the recursive SPIKE algorithm lays on the assembly of the factorized systems, which is a pure bandwidth-bound operation. However, being the sparsity pattern of those systems identical, an ad-hoc algorithm can be used to compute their inverse only from the dense $V$ and $W$ spikes, avoiding the construction of the system itself.

Finally, the implicit decomposition posed by the SPIKE algorithm makes possible the implementation of an out-of-core version of the solver for an arbitrary number of recursion steps to solve much bigger systems with a low implementation effort.

**Acknowledgements**

The present research was carried out at the Repsol-BSC Research Center. The authors acknowledge Repsol for the funding support. Numerical tests were performed on the MareNostrum supercomputer.

**References**


