Parallel computer algorithms for the solution of volume integral equation models in marine electromagnetics

BRODD LEIF ANDERSSON, JAN-OVE HALL, KURT OTTO

FOI, Swedish Defence Research Agency, is a mainly assignment-funded agency under the Ministry of Defence. The core activities are research, method and technology development, as well as studies conducted in the interests of Swedish defence and the safety and security of society. The organisation employs approximately 1000 personnel of whom about 800 are scientists. This makes FOI Sweden's largest research institute. FOI gives its customers access to leading-edge expertise in a large number of fields such as security policy studies, defence and security related analyses, the assessment of various types of threat, systems for control and management of crises, protection against and management of hazardous substances, IT security and the potential offered by new sensors.

FOI
Defence Research Agency
Försvars- och säkerhetssystem
SE-164 90 Stockholm

Phone: +46 8 555 030 00
Fax: +46 8 555 031 00

www.foi.se

FOI-R-- 2326 --SE Technical report
ISSN 1650-1942 December 2007

Försvars- och säkerhetssystem

Brodd Leif Andersson, Jan-Ove Hall, Kurt Otto

Parallel computer algorithms for the solution of volume integral equation models in marine electromagnetics

Rapportnr/Report no	FOI-R--2326--SE
Rapporttyp Report Type	Teknisk rapport Technical report
Utgivningsår/Year	2007
Antal sidor/Pages	13 p
ISSN	ISSN 1650-1942
Kund/Customer	Försvarsmakten
Forskningsområde Programme area	4. Sensorer och signaturanpassning 4. Sensors and Low Observables
Delområde Subcategory	43 UV-teknik - sensorer
Projektnr/Project no	E20607
Godkänd av/Approved by	Matts Gustavsson

Totalförsvarets Forskningsinstitut FOI

Avdelningen för Försvars- och säkerhetssystem

164 90 Stockholm

Table of Contents

Sammanfattning

Volymintegralekvationer är ett viktigt hjälpmedel för att modellera och analysera elektromagnetisk vågutbredning i marin miljö, t.ex. för att bedöma detektionsavstånd. I denna rapport studeras algoritmer för att lösa volymintegralekvationer med hjälp av parallelldatorer. Speciellt har algoritmer för lösning av linjära ekvationssystem med fyllda matriser studerats. En implementation av en direktlösare för parallelldatorer har utvärderats genom att lösa ett endimensionellt integralekvationsproblem. Lösaren visar sig vara mycket effektiv. Iterativa lösare och prekonditionerare som är lämpliga för parallelldatorer diskuteras också. Dessa algoritmer kan implementeras i framtida parallelliserade versioner av EMrad, som är en kod för elektromagnetisk vågutbredning i marin miljö. Denna studie visar att kodens prestanda kan förbättras avsevärt genom att använda parallelldatoralgoritmer.

Nyckelord: integralekvationsmetod, marin elektromagnetik, tre dimensioner

Summary

Volume integral equations are a useful tool to model and analyze electromagnetic wave propagation in marine environments, e.g., to estimate detection ranges. In the present report, parallel computer algorithms for the solution of volume integral equations are examined. In particular, algorithms for the solution of dense systems of linear equations are considered. A parallel computer implementation of a direct solver has been evaluated by solving a one-dimensional integral equation. The solver shows high parallel efficiency. Iterative solvers and preconditioners appropriate for parallel computers are also discussed. These algorithms could be implemented in future parallelized versions of the marine electromagnetic wave propagation model EMrad. The present investigation shows that the performance of EMrad code can be significantly increased by using parallel computer algorithms.

Keyword: integral equation method, marine electromagnetics, three-dimensions

1 Introduction

The propagation of electromagnetic (EM) waves in marine environments is important in many military and civil applications. An example of a military application is EM Rapid Environmental Assessment (REA). EM REA aims at determining a model of the electrical properties in the sub-bottom structure. For example, this information is necessary in order to estimate detection ranges and vulnerability to detection for own assets. It was shown in Ref. [1] that detection ranges may be underestimated or overestimated by as much as 50% when using inappropriate models of the electrical properties. See Ref. [2] for an overview of recent EM REA research performed by the Swedish Defence Research Agency (FOI). Low frequency EM wave propagation is governed by the conductivity of the marine environment. The conductivity of the sub-bottom structure can be estimated by using Controlled Source Electromagnetic (CSEM) methods. The response from the sub-bottom is measured when a strong signal from a known source is transmitted and scattered off the bottom structure. The sub-bottom conductivity profile is then estimated by iteratively matching a forward wave propagation model to the experimental data. The iterative procedure requires many evaluations of the forward wave propagation model. Thus, the forward wave propagation model must be sufficiently fast. Many marine environments can be modeled using a horizontally stratified model. For this purpose, FOI has developed the numerical code NLAYER [3].

Some situations require a more detailed description where the three-dimensional nature of the environment must be taken into account, e.g., in harbors and other coastal areas. The subject of three-dimensional EM modeling and inversion of sub-bottom structures is an active area of research and has experienced a tremendous progress in the last decade, see Ref. [4] for a recent review. Many sub-bottom and bathymetry structures in marine electromagnetism can be modeled as a set of three-dimensional conductivity irregularities embedded in a horizontally stratified medium. This type of geometry is well suited for volume integral equation methods, where the time consuming computational part is limited to the conductivity irregularities. The impact of the horizontally stratified background medium can be absorbed into the kernel of the integral equation via a Green's function. This integral equation method has been implemented in the forward wave propagation model EMrad [5]. The numerical method is based on the method of collocation, which transforms the continuous integral equation problem into an approximate finite dimension matrix equation. The resulting matrix equation can be solved by using either an iterative or a direct method.

Further developments and improvements of the EMrad code are necessary in order to increase the performance to a level useful for sub-bottom inversion. The main bottleneck in the existing single processor version of EMrad is the solution of the matrix equations. In the present report, parallel algorithms for the solution of dense systems of linear equations are discussed. These algorithms can be implemented in future parallelized versions of EMrad in order to significantly increase the performance of the code.

The report is organized in the following way. The governing integral equation and the corresponding discrete matrix equation are briefly discussed in Sec. 2. The structure of the single processor version of the EMrad code is reviewed in Sec. 3. In sections 4 and 5 direct and iterative solvers are discussed, respectively. The report is concluded by a summary in Sec. 6.

2 Governing equations

Many sub-bottom and bathymetry structures in marine electromagnetism can be modeled as a set of three-dimensional conductivity irregularities embedded in a horizontally stratified medium. The conductivity can thus be written as $\sigma(\mathbf{x}) = \sigma_0(z) + \Delta\sigma(\mathbf{x})$, where $\sigma_0(z)$ describes the horizontal stratification, and $\Delta\sigma(\mathbf{x})$ describes the conductivity irregularities. The positive z axis is pointing downward. One can show that the electric field is governed by the vector integral equation [6]

$$\mathbf{E}(\mathbf{x}) = \mathbf{E}_0(\mathbf{x}) + \iiint_D dV' \mathbf{G}_{\mathrm{e}}(\mathbf{x}, \mathbf{x}') \Delta\sigma(\mathbf{x}') \mathbf{E}(\mathbf{x}') \,, \tag{1}$$

where $\mathbf{E}$ is the total electric field and $\mathbf{E}_0$ is the incident electric field, i.e., the electric field in the absence of irregularities. $\mathbf{G}_{\mathrm{e}}$ is the 3×3 electric dyadic Green's function. D is the volume where $\Delta\sigma$ is different from zero. The dyadic Green's function satisfies a vector wave equation, where the local wave number of the medium depends on σ_0 [7]. In many applications, it is necessary to solve Eq. (1) for several frequencies and positions of the source. In frequency sounding experiments, the number of frequencies is typically of the order of 10. In order to solve the integral equation (1) numerically, the volume D is discretized into N rectangular volume elements. It is important to note that only the volume where $\Delta\sigma \neq 0$ need to be discretized. By using the method of collocation with pulse subsectional functions [8], an approximate solution of the continuous integral equation can be found as a solution of the matrix equation

$$\mathbf{A}\mathbf{e} = \mathbf{e}_0 \,, \tag{2}$$

where $\mathbf{e} = [E_{x,1}, \ldots, E_{x,N}, E_{y,1}, \ldots, E_{y,N}, E_{z,1}, \ldots, E_{z,N}]^T$ describes the total electric field in the grid points $\mathbf{x} = \mathbf{x}_i$, $i = 1, \ldots, N$. Analogously, the vector $\mathbf{e}_0$ describes the incident electric field. The system matrix is $\mathbf{A} = I - \widehat{\mathbf{G}}\widehat{\mathbf{S}}$, where

$$\widehat{\mathbf{G}} = \begin{pmatrix} \Gamma_{xx}^{11} \ldots \Gamma_{xx}^{1N} & \Gamma_{xy}^{11} \ldots \Gamma_{xy}^{1N} & \Gamma_{xz}^{11} \ldots \Gamma_{xz}^{1N} \\ & \vdots & \\ \Gamma_{xx}^{N1} \ldots \Gamma_{xx}^{NN} & \Gamma_{xy}^{N1} \ldots \Gamma_{xy}^{NN} & \Gamma_{xz}^{N1} \ldots \Gamma_{xz}^{NN} \\ \Gamma_{yx}^{11} \ldots \Gamma_{yx}^{1N} & \Gamma_{yy}^{11} \ldots \Gamma_{yy}^{1N} & \Gamma_{yz}^{11} \ldots \Gamma_{yz}^{1N} \\ & \vdots & \\ \Gamma_{yx}^{N1} \ldots \Gamma_{yx}^{NN} & \Gamma_{yy}^{N1} \ldots \Gamma_{yy}^{NN} & \Gamma_{yz}^{N1} \ldots \Gamma_{yz}^{NN} \\ \Gamma_{zx}^{11} \ldots \Gamma_{zx}^{1N} & \Gamma_{zy}^{11} \ldots \Gamma_{zy}^{1N} & \Gamma_{zz}^{11} \ldots \Gamma_{zz}^{1N} \\ & \vdots & \\ \Gamma_{zx}^{N1} \ldots \Gamma_{zx}^{NN} & \Gamma_{zy}^{N1} \ldots \Gamma_{zy}^{NN} & \Gamma_{zz}^{N1} \ldots \Gamma_{zz}^{NN} \end{pmatrix}_{3N \times 3N} , \tag{3}$$

$$\Gamma_{\alpha\beta}^{jk} = \iiint_{V_k} dV' G_{\alpha\beta}(\mathbf{x}_j, \mathbf{x}'_k) \,, \tag{4}$$

$$\widehat{\mathbf{S}} = \mathrm{diag}\left([\Delta\sigma_1, \ldots, \Delta\sigma_N, \Delta\sigma_1, \ldots, \Delta\sigma_N, \Delta\sigma_1, \ldots, \Delta\sigma_N]\right) \,, \tag{5}$$

V_k is the volume occupied by the kth volume element $j, k = 1, \ldots, N$, and $\alpha, \beta = x, y, z$.

Due to the singularity of the dyadic Green's function, a principal value integral has to be performed for $\Gamma_{\alpha\beta}^{jj}$, $j = 1, \ldots, N$ [6]. The electric field can be calculated in any desired field point by inserting the solution of Eq. (2) into Eq. (1). The system matrix is dense, i.e., all matrix elements are non-zero, and the size of the system matrix is $3N \times 3N$. N is typically larger than 10,000.

3 The structure of the EMrad code

The volume integral equation method described in Sec. 2 has been implemented in the forward wave propagation model EMrad [5]. The program is implemented for single processor computers and the code is mainly written in FORTRAN 77. For FOI users, the code can be downloaded from a local SUBVERSION server. The dyadic Green's functions for the horizontally stratified medium are computed by the NLAYER code [3]. The structure of the EMrad code is summarized in Alg. (1).

Algorithm 1 EMrad pseudo-code

```
for all Frequencies do
    for i = 1 : 3N do                          ▷ Assemble the system matrix A
        for j = 1 : 3N do
            Calculate the dyadic Green's function using NLAYER
            Calculate and store the matrix element A_ij
        end for
    end for
    for all Source positions do
        Calculate and assemble the right-hand side e_0
        Solve the system of linear equations
        for all Field points do
            Calculate E at the field point
        end for
    end for
end for
```

Equation (2) can be solved either by iterative or direct methods. In early versions of the EMrad code the matrix equation was solved iteratively using the generalized minimal residual (GMRES) algorithm. When using the GMRES algorithm, it is necessary to apply preconditioning of the system matrix to achieve convergence. Following Ref. [9], the extended Born approximation was used in the preconditioning procedure. However, the iterative solution showed very poor convergence for some important applications of the EMrad code. In fact, the convergence was so slow that a direct solver gave a better performance [10]. Direct methods such as Gaussian elimination suffer from an $\mathcal{O}(N^3)$ scaling of the arithmetic work [11]. An iterative method with a promising preconditioner is discussed in Sec. 5.

The volume integral method requires a substantial amount of memory when applied to realistic problems. The reason is that the system matrix is dense. Both for an iterative and a direct solver, $9\,N^2$ complex valued matrix elements must be stored. Consider, for example, a block shaped volume which is discretized with 30 points in each direction, i.e., $N = 30^3 = 27,000$. If the matrix elements are represented in double precision, one needs to allocate $16(3N)^2 = 16 \cdot 9 \cdot (30^3)^2$ bytes $\sim$ 100 Gbytes of memory for the storage of the matrix. This is by far more memory than is available in modern workstations. Attempts to circumvent this problem has been made by discarding "small" matrix elements [12]. On the other hand, cluster computers can be employed to solve systems of this size, since a cluster may contain hundreds of processors each equipped with a few Gbytes of memory.

As pointed out in Ref. [5], the volume integral equation formulation is ideal for parallelization as the method is inherently divided into several independent steps. A number of subtasks in the

code can be identified, which could be parallelized in future developments of the code. They are:

1. Assembling the system matrix (calculate $\Gamma_{\alpha\beta}^{jk}$ using NLAYER)
2. Solving the system of linear equations
3. Evaluation of the EM fields for several source locations
4. Evaluation of the EM fields at several sensor points
5. Several frequencies in parallel

4 Direct solver

In this section, we explore the possibility to use a parallelized direct solver for solving the system of linear equations given in Eq. (2). As mentioned above, direct solvers are slow due to the $\mathcal{O}(N^3)$ scaling of the execution time. By using parallel computers, this scaling can to some extent be counterbalanced by using a large number of processors. Algorithms for direct solution can be found in the literature, e.g., see Ref. [13]. Here, we have chosen to use an existing implementation found in the linear algebra routine library ScaLAPACK [14], which is a parallel computer version of the well-known LAPACK library.

The ScaLAPACK routine *PZGESV* computes the solution of a complex system of linear equations in double precision. The routine uses LU decomposition with partial pivoting and row interchanges to factor $\mathbf{A}$ as $\mathbf{A} = \mathbf{PLU}$, where $\mathbf{P}$ is a permutation matrix, $\mathbf{L}$ is unit lower triangular, and $\mathbf{U}$ is upper triangular. No additional memory needs to be allocated in order to store the matrices $\mathbf{L}$ and $\mathbf{U}$. Instead, the original system matrix $\mathbf{A}$ is overwritten to store $\mathbf{L}$ and $\mathbf{U}$. The elements of the system matrix must be distributed to submatrices stored in the individual nodes of the computer cluster. ScaLAPACK uses the block-cyclic partition of the system matrix to minimize load imbalance and idling problems [13, 14].

4.1 Direct solution of a one-dimensional test problem

In order to investigate the performance of the routine *PZGESV*, we have implemented a numerical solution of a one-dimensional test problem. The considered model equation is

$$\psi(x) + \int_{x_1}^{x_2} G(x,x')\eta(x')\psi(x')dx' = \psi_0(x;x_\mathrm{s})\,, \tag{6}$$

where $G(x,x') = -ik_0\exp(-ik_0|x-x'|)/2$ and k_0 is a constant. The right hand side $\psi_0(x;x_\mathrm{s}) = 2\,G(x,x_\mathrm{s})/k_0$ represents the field generated by a point source located at $x = x_\mathrm{s}$. The wave field generated by the source is scattered off the irregularity $\eta(x)$, where $\eta(x) = \eta_0\exp[-(x-x_c)^2/(2L^2)]$ for $x \in [x_1, x_2]$ and $\eta = 0$ otherwise. The parameters η_0, x_c, and L describe the position and size of the irregularity. Equation (6) is of the same type as Eq. (1), i.e., a Fredholm integral equation of the second kind. A numerical method for the solution of Eq. (6) can be formulated as a matrix equation. The field $\mathbf{X}$ inside the irregularity is determined by the matrix equation

$$\mathbf{A}\,\mathbf{X} = \mathbf{B} \tag{7}$$

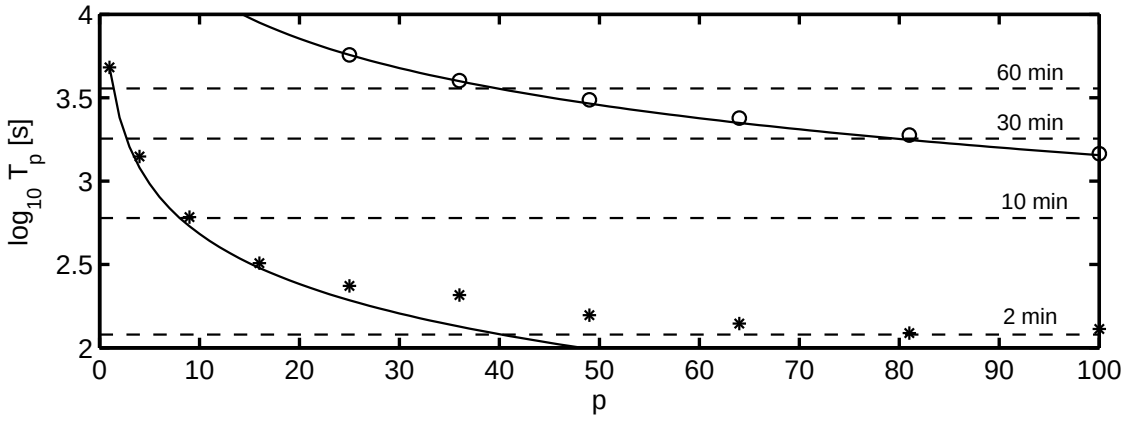

Figure 1: Execution time of the parallel algorithm as a function of the number of processors. The result for N =10,000 and N =30,000 are drawn with $*$ signs and $\circ$ signs, respectively.

where the matrix elements are $A_{ij} = \delta_{ij} + G(x_i, x_j)\eta(x_j)\Delta x_j$ and $B_{ij} = 2\,G(x_i, x_{\mathrm{s},j})/k_0$, where $i, j = 1, \ldots, N$. The $N \times N$ matrix $\mathbf{A}$ is dense and nonsymmetric. When Eq. (7) is solved, the solution can be used to calculate ψ at any field point.

The numerical method outlined above has been implemented in a FORTRAN 90 code named `fie2k1d`. The code is designed for a distributed memory architecture using the Message Passing Interface (MPI) standard. Below, we will highlight some important features of the code. The submatrices of the distributed system matrix are allocated dynamically and the size of the problem can be determined at run time. This allows the user to execute the code with an arbitrary matrix size on an arbitrary number of processors. The elements of the system matrix are computed by the node which store that particular element. This is important in cases where the computation of the matrix elements is time consuming, e.g., when it is necessary to call the NLAYER code to calculate the dyadic Green's function. The direct solver allows us to treat several positions of the source in parallel. Only the right hand side of Eq. (7) is changed when the position of the source is altered. Consequently, the computationally demanding LU decomposition of the system matrix can be reused for all positions. The total number of floating-point operations to solve the problem for multiple source configurations is still not higher than $\mathcal{O}(N^3)$. Finally, the evaluation of the field at an arbitrary number of points is parallelized by multiplication of distributed matrices. Thus, the implemented program shows possible solutions to the first four items listed in Sec. 3. The software solutions must be modified somewhat in order to apply to the three dimensional case, but the same principles can be used.

4.2 Numerical investigation of the parallel efficiency

A common figure of merit in parallel computing is the speedup. Speedup refers to how much faster a parallel algorithm is compared with its sequential counterpart [13]. It is defined by the equation $S_p = T_{\mathrm{seq}}/T_p$, where T_{seq} is the execution time of the sequential algorithm and T_p is the execution time of the parallel algorithm with p processors. In an ideal parallel system, the speedup is equal to p. In practice, $S_p < p$ as the processors cannot devote their entire time to the computations of the algorithm. Part of the time the processors are communicating with other nodes or are idling. Another figure of merit is the parallel efficiency, which is a measure of the fraction of time for which a processor is usefully employed. The parallel efficiency

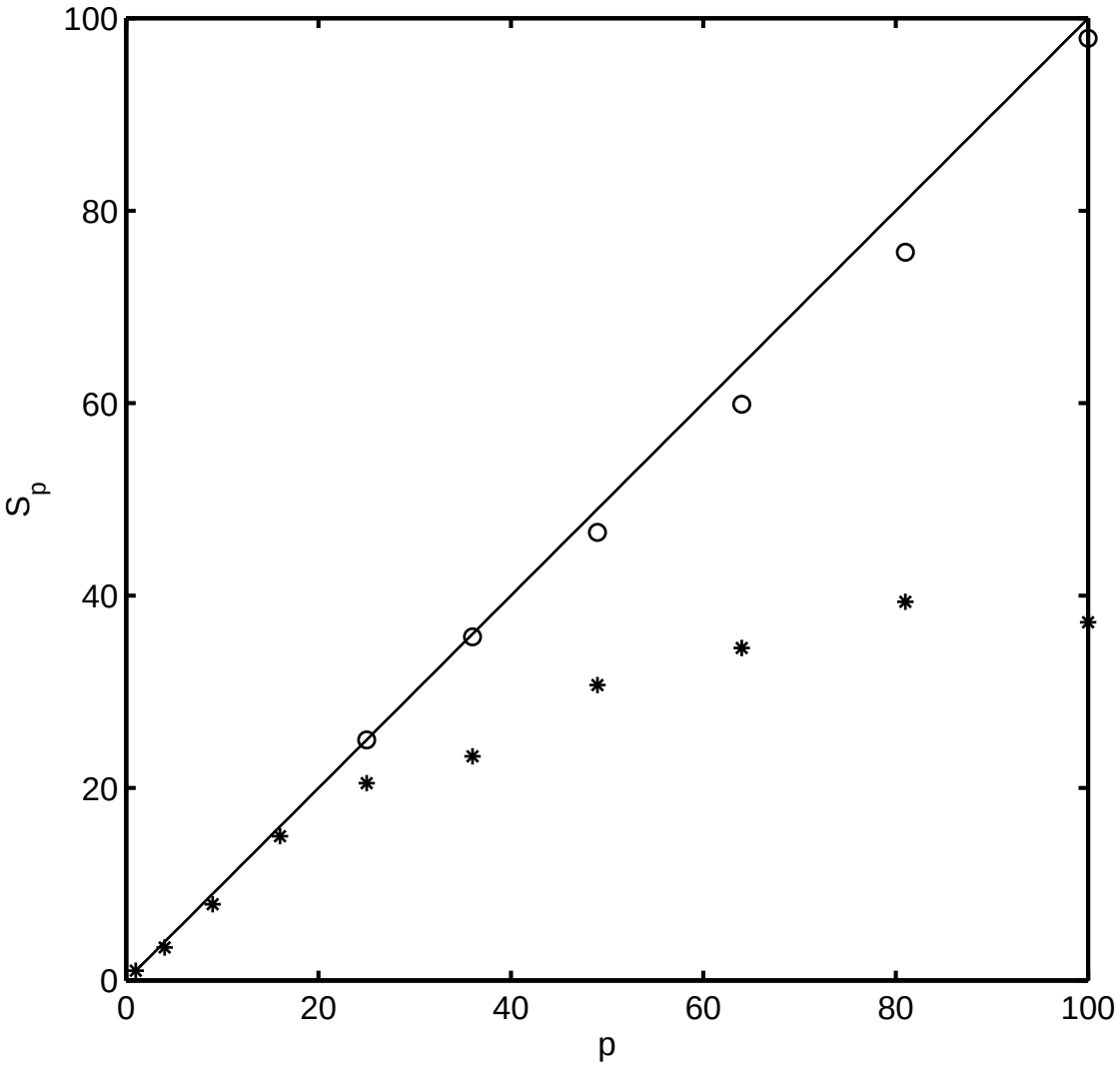

Figure 2: Speedup as a function of the number of processors. The result for N =10,000 and N =30,000 are drawn with $*$ signs and $\circ$ signs, respectively.

is defined as $E = S_p/p$. In a nonideal parallel system, the efficiency is smaller than one. In this section the execution time, speedup, and parallel efficiency of the code `fie2k1d` is investigated by using the cluster computer *Tunnan*. The test problem has been solved with 10,000 and 30,000 unknown field components, respectively, and 1000 source locations (number of right-hand sides). The number of processors is varied from 1 to 100.

The problem with $N = 10{,}000$ unknowns can be solved on a single node in the cluster, and the execution time was found to be $T_1 = T_{\text{seq}} \approx 4337$ s. Figure 1 shows T_p as a function of $p = n^2$, for $n = 1, 2, \ldots, 10$. The execution time decreases rapidly when p is increased from 1 to 25. In fact, T_p is close to the ideal execution time T_{seq}/p, which is indicated by the solid curve in Fig. 1. Figure 2 shows S_p as a function of p. The solid line shows the ideal speedup $S_p = p$. For $p > 16$, S_p starts to deviate from the ideal speedup. Consequently, the parallel efficiency decreases with increasing number of processors. For $p = 100$, the efficiency has decreased from the maximum value 93 % at p =16 to $\sim$35%. This behavior is illustrated in Fig. 3, where E is plotted as a function of p.

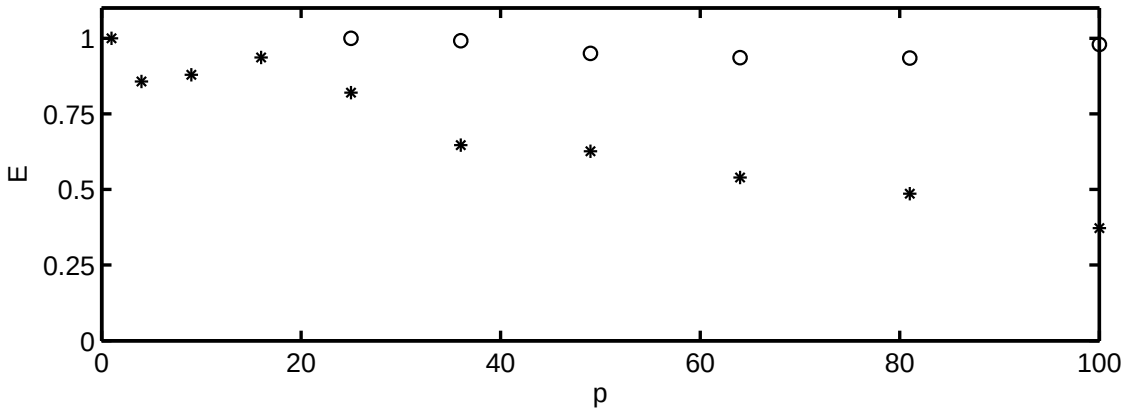

Figure 3: Parallel efficiency as a function of the number of processors. The result for N =10,000 and N =30,000 are drawn with $*$ signs and $\circ$ signs, respectively.

The larger problem with $N = 30{,}000$ unknowns cannot be solved on a single node as the memory per node is limited to 2 Gbytes. When 25 processors are used the execution time is roughly 1.5 h. In contrast to the smaller problem discussed above, T_p is close to the ideal execution time for all investigated values p, and the estimated speedup is almost linear. For p =100, the execution time is reduced to 24 minutes.

An approximate expression for T_p is given in Ref. [14], where it was suggested that

$$T_p = \frac{C_f N^3}{p} t_f + \frac{C_v N^2}{\sqrt{p}} t_v + \frac{C_m N}{N_b} t_m \,, \tag{8}$$

where t_f is the time to execute one floating-point operation by one processor, $C_f N^3$ is the total number of floating-point operations, t_v is the time per data item communication, $C_v N^2/\sqrt{p}$ is the total number of data items communicated, t_m is the time to communicate a message of zero length (also called the latency), N_b is the data distribution block size, and $C_m N/N_b$ is the total number of messages. The execution time of the sequential algorithm is $T_{\text{seq}} \approx C_f N^3 t_f$, giving

$$E = \left(1 + \frac{1}{N_b} \frac{C_m t_m}{C_f t_f} \frac{p}{N^2} + \frac{C_v t_v}{C_f t_f} \frac{\sqrt{p}}{N} \right)^{-1} . \tag{9}$$

As seen in Eqs. (8) and (9), the impact of communication and latency is more severe for small N, where the last two terms can be dominating. This is in qualitative agreement with the numerical experiments discussed above.

5 Iterative solvers

Since the coefficient matrix in Eq. (2) is dense, the arithmetic complexity for Gaussian elimination grows cubically in N (the number of cells). For large values of N, this arithmetic cost becomes prohibitively high. A possible remedy is to use a rapidly convergent iterative method instead, where the arithmetic complexity might be as low as $\mathcal{O}(N^2)$. Thus, there is a potential to reduce the arithmetic work by an order of magnitude in N.

5.1 Krylov subspace methods

The state of the art for iterative solution of linear systems is to employ some Krylov subspace method [15]. For simplicity and robustness, we advocate the restarted generalized minimal residual (GMRES(ℓ)) algorithm [16], where ℓ is the restarting length. Within the current project, a serial version of the restarted GMRES algorithm has been thoroughly implemented in FORTRAN 90.

5.2 Preconditioning

In order to achieve an acceptable rate of convergence and, more importantly, a short total execution time, it is crucial to construct an effective, parallelizable preconditioner. For the type of systems that we are considering, standard preconditioning techniques like incomplete LU factorizations are likely to perform poorly. A viable alternative is to use preconditioners based on sparse approximate inverses (SPAI) [17], the construction of which are inherently parallel. Furthermore, they are possible to tune by altering a tolerance parameter ε.

6 Summary

In the present report, we have discussed parallel computer algorithms for the solution of volume integral equation models in marine electromagnetics. In particular, parallel algorithms for the solution of systems of linear equations have been discussed. These algorithms could be implemented in future parallelized versions of EMrad in order to significantly increase the performance of the code.

A parallel computer implementation of a direct solver has been evaluated by solving a one-dimensional test problem. The solver shows high parallel efficiency for large problems. For the iterative solvers, we suggest that the continued work should be focused on parallelization of the GMRES code using MPI, implementation of sequential SPAI preconditioners, and parallelization of SPAI preconditioners.

References

[1] P. Krylstedt and J. Mattsson. Environmental assessment for underwater electric sensors. In *Proceedings of UDT Europe*, 2003.

[2] P. Krylstedt and M. Karlsson. Progress report – electromagnetic rapid environmental assessment. Technical Report FOI-R–1857–SE, Swedish Defence Research Agency, 2005.

[3] L. Abrahamsson and B. L. Andersson. NLAYER– An ELFE code for horizontally stratified media. Technical Report FOA-R–97-00586-409–SE, Swedish Defence Research Agency, 1997.

[4] D. B. Avdeev. Three-dimensional electromagnetic modeling and inversion from theory to application. *Surv. Geophys*, 26:767–799, 2005.

[5] J. Mattsson. An integral equation method for low frequency electromagnetic fields in three-dimensional marine environments. Technical Report FOI-R–1838–SE, Swedish Defence Research Agency, 2005.

[6] G. W. Hohmann. Three-dimensional induced polarization and electromagnetic modeling. *Geophysics*, 40:309–324, 1975.

[7] J. D. Jackson. *Classical electrodynamics*. John Wiley & sons, second edition, 1975.

[8] R. F. Harrington. *Field computation by moment methods*. MacMillan Co., New York, 1968.

[9] G. Hursán and M. Zhdanov. Contraction integral equation method in three-dimensional electromagnetic modelling. *Radio Sci.*, 37, 2002.

[10] J. Mattsson. private communication, 2007.

[11] G. H. Golub and J. M. Ortega. *Scientific Computing and Differential Equations*. Academic Press, Boston, 1992.

[12] O. N. Portniaguine, G. Hursán, and M. S. Zhdanov. Compression in 3D electromagnetic modeling. In *Proceedings of the Second International Symposium on Three-Dimensional Electromagnetics*, pages 209–212. Univ. of Utah, Salt Lake City, 1999.

[13] A. Grama, A. Gupta, G. Karypis, and V. Kumar. *Introduction to parallel computing*. Pearson Education Limited, Harlow, second edition, 2003.

[14] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley. *ScaLAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997.

[15] R. W. Freund, G. H. Golub, and N. M. Nachtigal. Iterative solution of linear systems. *Acta Numer.*, 1:57–100, 1992.

[16] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7:856–869, 1986.

[17] M. J. Grote and T. Huckle. Parallel preconditioning with sparse approximate inverses. *SIAM J. Sci. Comput.*, 18:838–853, 1997.