EFFICIENT PARALLEL ITERATIVE METHODS FOR ACOUSTIC SCATTERING WITH EXACT NONREFLECTING BOUNDARIES

PACS REFERENCE: 43.20.Fn

Ianculescu, Cristian; Thompson, Lonny
Clemson University, Department of Mechanical Engineering
Clemson, South Carolina 29634-0921 USA
tel: (864)656-7210; (864)656-5631
email: ciancul@ces.clemson.edu; lonny.thompson@ces.clemson.edu

ABSTRACT

Parallel iterative solutions of acoustic scattering in the frequency domain are developed using exact Dirichlet-to-Neumann (DtN) maps on an elliptical nonreflecting boundary. We exploit the special structure of the non-local DtN map as a low-rank update of the system matrix to efficiently compute the matrix-vector products found in Krylov subspace based iterative methods. For the complex non-hermitian matrices resulting from the Helmholtz equation, we use the BICG-STAB iterative method. Domain decomposition with interface minimization was performed to ensure optimal inter-processor communication. We show that when implemented as a low-rank update, the non-local character of DtN does not significantly decrease the scale up and parallel efficiency versus a purely approximate local boundary condition.

Introduction

Numerical solutions for acoustic radiation and scattering in unbounded domains require a large number of element unknowns to resolve high wavenumber/frequency problems. To solve these problems efficiently on today's computer architecture, it is required to distribute the work and memory on multi-processors and compute in parallel. A common method for solving unbounded problems governed by the Helmholtz equation is to apply an artificial truncation boundary surrounding the scatterer, where nonreflecting boundary conditions, infinite elements, or absorbing layers are defined. Some of the infinite elements, particularly the unconjugated elements, provide accurate solutions but may produce poorly conditioned matrices. Local approximate nonreflecting boundary conditions and absorbing layer elements, preserve the sparsity of the matrix equations within the bounded domains, but must be used with caution, as the errors produced by spurious reflection are generally unknown. This sometimes requires multiple solutions where the truncation boundary must be systematically moved further away from the scatterer to provide solutions within a given error tolerance.

A compelling alternative is the DtN map defined on a separable truncation boundary [1]. The DtN nonreflecting boundary is exact in that harmonics in the solution are exactly reproduced. By coupling all the nodes on the truncation boundary, therefore giving rise to a full lower rank submatrix, DtN was considered computationally prohibitive. However, a major breakthrough came, when, for a circle in two-dimensions or a sphere in three-dimensions, Malhotra et al. [2] and Oberai et al. [3] recognized that the special structure of the DtN map may be used to efficiently implement matrix-by-vector updates found in Krylov subspace based iterative solvers, thus making the DtN highly competitive with approximate local conditions, but with significantly improved accuracy. In Thompson et al. [4], the finite element formulation of the DtN map was extended to elliptic and spheroidal boundaries, thus allowing for efficient solutions of highly elongated structures such as a submarine or ship. In [5], it was shown that the interpretation of the DtN matrix as a low-rank update to the sparse interior matrices is preserved, allowing for efficient iterative solutions for the elliptic/spheroidal DtN. In this paper, we demonstrate an efficient algorithm to incorporate the matrix update resulting from the exact non-reflecting DtN map into a parallel iterative process.

We show that the effect of the DtN map on the parallel efficiency of the iterative solution to the sparse interior matrices is very small, and not significant.

We start by introducing the weak form of the exterior problem using the DtN map in elliptic coordinates, and follow by the finite element implementation and presentation of the parallel algorithm. A numerical example is given for scattering from an elliptic structure with parallel efficiency on two different parallel computer systems.

Exterior Problem and FE Implementation

We consider time-harmonic scattering and radiation of waves in an infinite region $\mathcal{D} \subset \mathcal{R}^2$, truncated by an artificial elliptical boundary Γ thus enclosing a finite computational domain Ω. The non-homogeneous Helmholtz equation is satisfied within Ω,

$$(\nabla^2 + k^2)\phi = -f(\mathbf{x}), \qquad \mathbf{x} \in \Omega \tag{1}$$

where ϕ is the scalar field, k is the wavenumber and f is the source, which is confined to Ω.

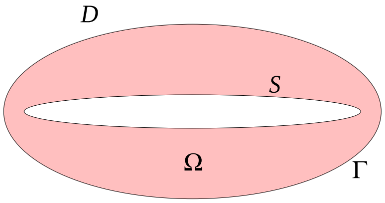

Figure 1: Scattering on unbounded domains. Exterior domain $\mathcal{D}$ is reduced to computational domain Ω enclosed by the scatterer $\mathcal{S}$ and the truncation boundary Γ

For simplicity, but not at the loss of generality, the elliptical scatterer is assumed to have a 'soft' boundary, i.e. $\phi = 0, \mathbf{x} \in \mathcal{S}$. The non-reflecting boundary condition can be written in abstract form as

$$\frac{\partial \phi}{\partial n} = M(\phi), \qquad \mathbf{x} \in \Gamma \tag{2}$$

For the elliptic case, the DtN operator $M(\phi)$ can be written as [6]

$$\begin{aligned} M_{B_1}(\phi) &= -\frac{\beta_1}{h_\theta} + \sum_{n=0}^{\infty} \frac{1}{\pi}(Zc_n + \beta_1) \cdot \frac{1}{h_\theta} ce_n \cdot T_c^n(\phi) + \\ & \quad \sum_{n=1}^{\infty} \frac{1}{\pi}(Zs_n + \beta_1) \cdot \frac{1}{h_\theta} se_n \cdot T_s^n(\phi) \qquad (3) \\ Zc_n &= \frac{Mc'_n(\mu_0, q)}{Mc_n(\mu_0, q)} \qquad (4) \\ Zs_n &= \frac{Ms'_n(\mu_0, q)}{Ms_n(\mu_0, q)} \qquad (5) \end{aligned}$$

where $Mc_n(\mu_0, q)$, $Ms_n(\mu_0, q)$, $Mc'_n(\mu_0, q)$, $Ms'_n(\mu_0, q)$ are the modified (even and odd) Mathieu functions of the third kind, and their derivatives, respectively, $ce_n(\theta, q)$, $se_n(\theta, q)$ are the regular even and odd Mathieu functions of the third kind, $T_c^n = \frac{1}{h_\theta}(\phi, ce_n)_\Gamma$, $T_s^n = \frac{1}{h_\theta}(\phi, se_n)_\Gamma$, $q = \frac{kf^2}{2}$ is the normalized wavenumber, μ_0 and f are the radial coordinate and the focal distance of the truncation boundary, respectively.

The weak form of (1)-(2) may be stated as: Find $\phi(\mathbf{x})$ in $\mathcal{V}$, such that for all admissible weighting functions $\bar{\phi}$ in $\mathcal{V}$, the following variational equation is satisfied,

$$K_\Omega(\bar{\phi}, \phi) + K_\Gamma(\bar{\phi}, \phi) = F(\bar{\phi}) \tag{6}$$

with inner products $(\cdot, \cdot) : \mathcal{V} \times \mathcal{V} \mapsto C$ defined by the sesquilinear forms,

$$\begin{aligned} K_\Omega(\bar{\phi}, \phi) &:= \int_\Omega (\nabla\bar{\phi} \cdot \nabla\phi - k^2 \bar{\phi}\phi) d\Omega \qquad (7) \\ K_\Gamma(\bar{\phi}, \phi) &:= -\int_\Gamma \bar{\phi} M(\phi) d\Gamma \qquad (8) \end{aligned}$$

and conjugate linear form $(\cdot) : \mathcal{V} \mapsto C$ defined by,

$$F(\bar{\phi}) := \int_\Omega \bar{\phi} f d\Omega \tag{9}$$

where $\mathcal{V}$ is a subspace of H^1 for which $\phi(\mathbf{x}) = 0, \mathbf{x} \in \mathcal{S}$.
The boundary operator in (8) is composed of both a local and DtN part:

$$K_\Gamma(\bar{\phi}, \phi) \quad = \quad \underbrace{B_\Gamma(\bar{\phi}, \phi)}_{\text{local}} + \underbrace{Z_\Gamma(\bar{\phi}, \phi)}_{\text{DtN}} \tag{10}$$

$$B_\Gamma(\bar{\phi}, \phi) \quad = \quad \frac{\beta_1}{h_\theta}(\bar{\phi}, \phi)_\Gamma \tag{11}$$

$$\begin{aligned} -Z_\Gamma(\bar{\phi}, \phi) \quad = \quad & \sum_{n=0}^{\infty} \frac{1}{\pi}(Zc_n + \beta_1) \cdot {T_c}^n(\bar{\phi}) {T_c}^n(\phi) + \\ & \sum_{n=1}^{\infty} \frac{1}{\pi}(Zs_n + \beta_1) \cdot {T_s}^n(\bar{\phi}) {T_s}^n(\phi) \end{aligned} \tag{12}$$

where $h_\theta = f\sqrt{\sinh^2 \mu + \sin^2 \theta}$ is the metric for the elliptic coordinate system, and $\beta_1 = \frac{1}{2}\tanh \mu_0 - ikf_0 \sinh \mu_0$.

We discretize the domain Ω into a finite number of elements, $\Omega \approx \Omega^h = \bigcup_{e=1}^{N_e} \Omega_e^h$ and apply a standard Galerkin approximation for both the trial and test functions, $\phi(\mathbf{x}) \approx \phi^h(\mathbf{x}) = \mathbf{N}(\mathbf{x})\mathbf{d}$, $\bar{\phi}(\mathbf{x}) \approx \bar{\phi}^h(\mathbf{x}) = \mathbf{N}(\mathbf{x})\bar{\mathbf{d}}$, where $\mathbf{N} \in R^{N_{dof}}$ is a row vector of standard C^0 Lagrangian interpolation functions and $\mathbf{d} \in C^{N_{dof}}$ the column vector of the nodal values. This leads to the following linear system:

$$\mathbf{Kd} = \mathbf{f} \tag{13}$$

with $\mathbf{K} \in C^{N_{dof} \times N_{dof}}$ an indefinite complex-symmetric matrix defined as follows:

$$\mathbf{K} \quad = \quad \underbrace{(\mathbf{K}_\Omega + \mathbf{B}_\Gamma)}_{\text{sparse}} \quad + \quad \underbrace{\mathbf{M}^T \cdot \mathbf{Z}_\Gamma \cdot \mathbf{M}}_{\text{DtN low rank update}} \tag{14}$$

where $\mathbf{K}_\Omega = K_\Omega(\mathbf{N}^T, \mathbf{N})$ is the term associated with the discretization of the Helmholtz equation in Ω, $\mathbf{B}_\Gamma = B_\Gamma(\mathbf{N}^T, \mathbf{N})$ is the matrix resulted from applying the local radiation boundary operator, $\mathbf{M}$ is the permutation matrix, and $\mathbf{Z}_\Gamma$ is the low rank update associated with the DtN operator:

$$\mathbf{Z}_\Gamma = \mathbf{C}^T \cdot \mathbf{\Delta} \cdot \mathbf{C} \tag{15}$$

where $\mathbf{C} = [\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{N_m-1}, \mathbf{s}_1, \ldots, \mathbf{s}_{N_m-1}]$,
$\mathbf{\Delta} = -\frac{1}{\pi}\text{diag}(Zc_0, Zc_1, \ldots, Zc_{N_m-1}, Zs_1, \ldots, Zs_{N_m-1})$ and $\mathbf{c}_n = {T_c}^n(\mathbf{N}^T)$, $\mathbf{s}_n = {T_s}^n(\mathbf{N}^T)$.

Parallel algorithm

In the context of iterative methods, the matrix-by-vector products, dot-products and vector updates are the most computationally intensive operations, consequently it is important to distribute them to processes handling submatrices of similar size. Ideally, a perfectly balanced mesh accompanied by a minimal interface will maximize the parallel efficiency. Consider the discretized domain Ω^h divided into N_s non-overlapping subdomains Ω_i^h such that $\bigcup_{i=1}^{N_s} \Omega_i^h = \Omega^h$. In the following, subscript index i represents that respective quantity (matrix/vector) restricted on subdomain Ω_i^h, and indices Ω and $\partial\Omega$ indicate further restrictions to the interior and interface of that subdomain, respectively. For example, $\mathbf{y}_i = [\mathbf{y}_{i\Omega}, \mathbf{y}_{i\partial\Omega}]$, where $\mathbf{y}_i$ is a vector projected onto Ω_i^h, $\mathbf{y}_{i\Omega}$ is $\mathbf{y}_i$ restricted to the interior of Ω_i^h, and $\mathbf{y}_{i\partial\Omega}$ is $\mathbf{y}_i$ restricted to the boundary of Ω_i^h. Similarly,

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{A}_{i\Omega,\Omega} & \mathbf{A}_{i\Omega,\partial\Omega} \\ \mathbf{A}_{i\partial\Omega,\Omega} & \mathbf{A}_{i\partial\Omega,\partial\Omega} \end{bmatrix} \tag{16}$$

is a matrix projected on subdomain Ω_i^h with corresponding contributions from the subdomain interior and interface, respectively.

Matrix-vector product

Following the decomposition in (14) and the notation convention (16), the matrix vector product can be written in general form as

$$\begin{aligned}
\mathbf{y} &= \left[\mathop{\mathbf{A}}_{i=1}^{N_s}(\mathbf{K}_i+\mathbf{B}_i)\right]\mathbf{x}+\left[\mathop{\mathbf{A}}_{i=1}^{N_s}\mathbf{M}_i^T\cdot\mathbf{Z}_i\cdot\mathbf{M}_i\right]\mathbf{x} \\
&= \mathop{\mathbf{A}}_{i=1}^{N_s}\left[(\mathbf{K}_i+\mathbf{B}_i)\mathbf{x}_i\right]+\mathop{\mathbf{A}}_{i=1}^{N_s}\left[(\mathbf{M}_i^T\cdot\mathbf{Z}_i\cdot\mathbf{M}_i)\mathbf{x}_i\right] \\
&= \mathop{\mathbf{A}}_{i=1}^{N_s}\mathbf{y}_i+\mathop{\mathbf{A}}_{i=1}^{N_s}\mathbf{z}_i \qquad (17)
\end{aligned}$$

where $\mathop{\mathbf{A}}_{i=1}^{N_s}$ is an assembly operation over subdomains. The first term in the right hand side of (17) is sparse and we use efficient compact storage schemes (CSR) in our implementation. Each subdomain contains the local matrix partitioned into interior and interface blocks, as in (16). We compute first the interface matrix-vector product and then initiate its communication to the adjacent processes, via non-blocking `MPI_ISEND` calls. We proceed then to compute the interior part, for which no communication is necessary, and in the end, update the interface data with the results received from neighboring subdomains. The latter is a blocking operation since at the end of it, all nodal values are to be used for other operations (e.g. dot-products, vector updates, calculation of DtN term, etc).

Algorithm for computing $(\mathbf{K}_i+\mathbf{B}_i)\mathbf{x}_i$

For each process i:

1. Get $\mathbf{x}_{i\Omega}$, $\mathbf{x}_{i\partial\Omega}$
2. Compute $\mathbf{y}_{i\partial\Omega}=[\mathbf{K}_{i\partial\Omega,\Omega}+\mathbf{B}_{i\partial\Omega,\Omega},\mathbf{K}_{i\partial\Omega,\partial\Omega}+\mathbf{B}_{i\partial\Omega,\partial\Omega}]\begin{bmatrix}\mathbf{x}_{i\Omega}\\ \mathbf{x}_{i\partial\Omega}\end{bmatrix}$
3. `MPI_ISEND`($\mathbf{y}_{i\partial\Omega}$) to process j, adjacent to i (non-blocking)
4. Compute $\mathbf{y}_{i\Omega}=[\mathbf{K}_{i\Omega,\Omega}+\mathbf{B}_{i\Omega,\Omega},\mathbf{K}_{i\Omega,\partial\Omega}+\mathbf{B}_{i\Omega,\partial\Omega}]\begin{bmatrix}\mathbf{x}_{i\Omega}\\ \mathbf{x}_{i\partial\Omega}\end{bmatrix}$
5. `MPI_RECV`($\mathbf{y}_{j\partial\Omega}$) from process j, adjacent to i (blocking)
6. $\mathbf{y}_{i\partial\Omega}=\mathbf{y}_{i\partial\Omega}+\mathbf{y}_{j\partial\Omega}$, j adjacent to i

Denote by $\Omega_\Gamma^h=\left\{\Omega_i^h \mid \Omega_i^h\bigcap\Gamma\neq\emptyset\right\}$ the group of subdomains to have an edge on the DtN boundary. Because of the non-local character of DtN, an 'all-to-all' type of communication is required between subdomains in Ω_Γ^h. By projecting the angular functions onto the finite element space the DtN term becomes [2]

$$\mathbf{Z}_\Gamma=\mathbf{M}_\Gamma\cdot\mathbf{C}^T\cdot\mathbf{\Delta}\cdot\mathbf{C}\cdot\mathbf{M}_\Gamma \qquad (18)$$

where $\mathbf{C}$ is now a matrix $(2N_m-1)N_\Gamma$ of angular modes computed at nodes on the truncation boundary and $\mathbf{M}_\Gamma$ is a boundary mass matrix, which simplifies to a diagonal if integrated at nodal points. We emphasize that the DtN matrices $\mathbf{Z}_i$, defined on the subdomains adjacent to the truncation boundary Γ, are never explicitly assembled, but their decomposition products from (15), $\mathbf{C}_i$ and $\mathbf{\Delta}$, are stored locally on each subdomain in Ω_Γ^h.

The algorithm to compute the DtN term is the following:

Algorithm for computing $\mathbf{Z}_i\mathbf{x}_i$

For each process i handling a subdomain in Ω_Γ^h:

1. Get $\mathbf{x}_{i\Omega}$, $\mathbf{x}_{i\partial\Omega}$
2. $\mathbf{t}_1^i=\mathbf{M}_i\begin{bmatrix}\mathbf{x}_{i\Omega}\\ \mathbf{x}_{i\partial\Omega}\end{bmatrix}$ - extract nodes on the truncation boundary
3. $\mathbf{t}_2^i=\mathbf{M}_{i\Gamma}\mathbf{t}_1^i$; $\mathbf{t}_2^i=\mathbf{t}_2^i+\mathbf{t}_2^j$ - multiply by $\mathbf{M}_\Gamma$ on each subdomain followed by interface update with adjacent subdomains in Ω_Γ^h
4. $\mathbf{t}_3^i=\mathbf{C}^i\mathbf{t}_2^i$ - multiply by submatrix of angular modes

5. $\mathbf{t}_3^i = \sum_{i=1}^{N_s}{}' \mathbf{t}_3^i$ - the prime under the sum indicates that values of the interface nodes are divided by the number of subdomains sharing them
6. $\mathbf{t}_3^i = \mathbf{\Delta t}_3^i$ - multiply by the matrix of radial modes
7. $\mathbf{t}_4^i = \mathbf{C}^{T^i} \mathbf{t}_3^i$ - multiply by the transpose of the submatrix of angular modes
8. $\mathbf{z}_i = \mathbf{M}_{i\Gamma} \mathbf{t}_4^i$; $\mathbf{z}_i = \mathbf{z}_i + \mathbf{z}_j$ - multiply by $\mathbf{M}_\Gamma$ on each subdomain followed by interface update with adjacent subdomains in Ω_Γ^h

In the algorithm above, $\mathbf{M}_i$ is the permutation matrix on that subdomain. Technically, the mapping described in step 2 is achieved with a pointer indicating the positions of boundary nodes. $\mathbf{M}_{i\Gamma}$ and $\mathbf{C}^i$ are, respectively, the boundary mass and the matrix of angular modes, assembled on subdomain i.

Step 5 contains the 'all-to-all' additive communication, as we need to sum the contributions of all the subdomains in Ω_Γ^h and then communicate the result back to each subdomain in Ω_Γ^h. This is accomplished using `MPI_ALLREDUCE`($\mathbf{t}_3^i$, ... , `MPI_SUM`,`MPI_DTN`) the last variable being the communicator created to synchronize the processes handling subdomains in Ω_Γ^h.

The size of $\mathbf{t}_3^i$ is $2N_m - 1$ which typically is not a large number, and as we will see in the results section, for one of the architectures tested, the latency time when calling `MPI_ALLREDUCE` is dominant vs the actual communication of these short-length vectors.

Numerical experiments

Consider scattering of a plane wave $\phi(\mathbf{x}) = e^{-ik\mathbf{x}}$ by an elliptic cylinder with focal distance $f = 1$ and radial coordinate $\mu = 0.1$, such that the major and minor axes are, respectively, $a = f \cosh \mu$ and $b = f \sinh \mu$. Also, as mentioned previously, we assume a 'soft' scattering object, such that $\phi = \phi^s + \phi^i = 0$ on S, where we defined ϕ as the total field, ϕ^i as the incident field and ϕ^s as the scattered field.

The computational domain is enclosed by a truncation boundary placed at $\mu = 0.5$ and it is discretized using an orthogonal mesh of 80 × 1400 quadrilateral elements (112,000 nodes). For a wavenumber $k = 2\pi$ we use N_m=0 (purely local B_1 condition), 5, 10, 15, and 20 angular modes in the DtN condition. The algorithm was implemented with MPI on 2 machines, using appropriate FORTRAN 90 compilers, and the BICG-STAB iterative method. A 16 cpu Sun UltrasparcII with 450MHz cpu clock and 4GB shared memory, and a 32 cpu Linux cluster with Intel 900 MHz processors and 16 GB of distributed memory.

Figure (2) shows the scaleup, while Fig. (3) shows the parallel efficiency for the 2 machines, respectively. For the Sun machine, super-linear scaleup is shown up to 5 processors, while the Intel machine maintains good scaleup beyond the 7 processors tested. We expect better scaleup for larger test cases, since the ratio of computation/communication increases.

Conclusions

We show that incorporating an exact non-local DtN map as a low-rank update in an efficient parallel algorithm for solving the Helmholtz's equation provides accurate results with little extra cost involved in communication when choosing the appropriate number of modes on the truncation boundary. When implemented as a low-rank update, the non-local character of DtN does not significantly decrease the scale up and parallel efficiency versus a purely approximate local boundary condition.

Support for this work was provided by the National Science Foundation under Grant CMS-9702082 in conjunction with a Presidential Early Career Award for Scientists and Engineers (PECASE).

References

[1] J.B. Keller and D. Givoli, 'Exact non-reflecting boundary conditions', *J. Comp. Phys.*, **82**, 172-192, 1989.

[2] M. Malhotra and P.M. Pinsky, 'A matrix-free interpretation of the nonlocal Dirichlet-to-Neumann radiation boundary condition', *Internat. J. Numer. Methods Engrg.*, **39**, 3705-3713, 1996.

[3] A.A. Oberai, M. Malhotra and P.M. Pinsky, 'On the implementation of the Dirichlet-to-Neumann radiation condition for iterative solution of the Helmholtz equation', *Applied Numerical Mathematics*, **27**, 443-464, 1998.

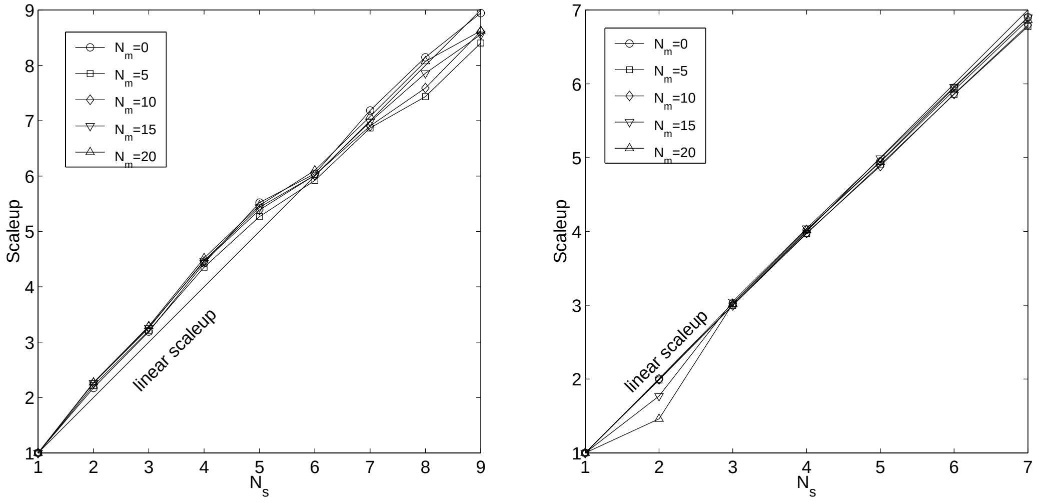

Figure 2: Scaleup for the two parallel systems. (Left): Sun HPC with UltrasparcII cpu's. (Right): Linux cluster with Intel PIII cpu's.

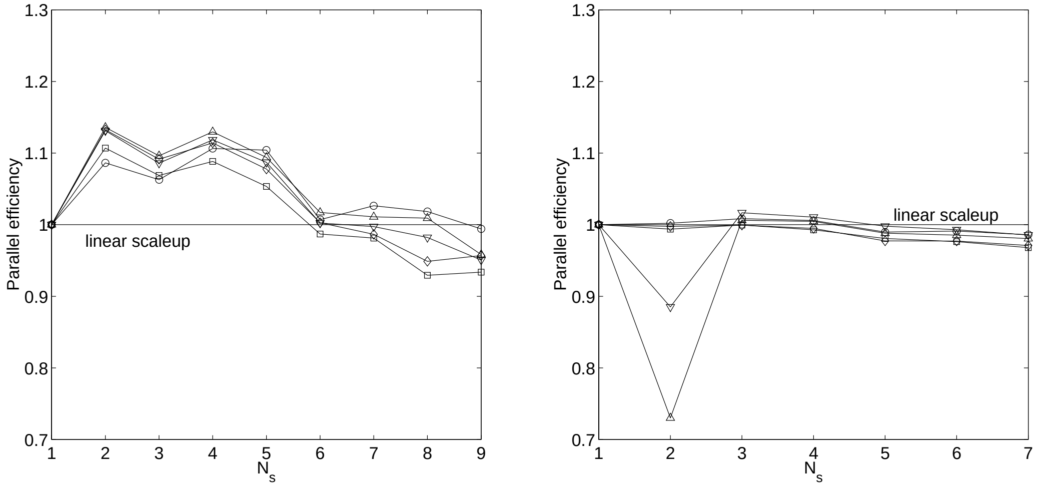

Figure 3: Parallel fficiency for the two parallel systems. (Left): Sun HPC with UltrasparcII cpu's. (Right): Linux cluster with Intel PIII cpu's.

[4] L.L. Thompson, R. Huan and C. Ianculescu, "Finite Element Formulation of Exact Dirichlet-to-Neumann Radiation Conditions on Elliptic and Spheroidal Boundaries", *Proceedings of the ASME Noise Control and Acoustics Division - 1999* ASME 1999 International Mechanical Engineering Congress and Exposition, NCA-Vol. 26, Symposium on Computational Acoustics, Nashville, TN, Nov. 14-19, 497-510, 1999.

[5] L.L. Thompson, C. Ianculescu and R. Huan, "Exact radiation conditions on spheroidal boundaries with sparse iterative methods for efficient computation of exterior acoustics", *Proceedings of the 7th International Congress on Sound and Vibration*, Session on Finite Elements in Acoustics, Garmisch-Partenkirchen, Germany, Jul. 4-7, 2101-2108, 2000

[6] M.J. Grote and J.B. Keller, 'On nonreflecting boundary conditions', *J. of Comput. Phys.*, **122**, 231-243, 1995.

[7] H.A. Van der Vorst, 'Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems', *SIAM J. Sci. Stat. Comput.*, **13**, 631-644, 1992.