Iterative solution of panel method discretizations for potential flow problems. The modal multipolar preconditioning

J. D'Elía, M. Storti, and S. Idelsohn
Grupo de Tecnología Mecánica del INTEC
Güemes 3450, (3000) Santa Fe, Argentina
Phone: 54-42-55.91.75, Fax: 54-42-55.09.44

1. *Abstract*

The iterative solution of linear systems arising from panel method discretization of three-dimensional (3D) exterior potential problems coming mainly from aero-hydrodynamic engineering problems, is discussed. We propose an original preconditioning based on an approximate eigenspace decomposition, that corrects bad conditioning arising from pair of surfaces that are very close from each other, which is a very common situation in slender wings and other aerodynamic profiles. This preconditioning has been tested with the standard Bi-Conjugate Gradient (BCG) and Conjugate Gradient Squared (CGS) iterative methods.

Keywords

- panel method
- preconditioning
- iterative solver
- aerodynamics
- potential flow
- panel clustering

Running title:

Iterative Solution for Panel Method Discretizations

Send proofs to:

Jorge D'Elía
Grupo de Tecnología Mecánica del INTEC
Güemes 3450, (3000) Santa Fe, Argentina
Phone: 54-42-55.91.75, Fax: 54-42-55.09.44

Subject Classification:

- 35J05 Laplace equation, reduced wave equation (Helmholtz), Poisson equation
- 65N38 Boundary element methods
- 76G25 General aerodynamics and subsonic flows

3. Introduction

The Panel Method is a well established numerical technique for the solution of potential flow problems, especially in exterior aero and hydrodynamics due to its ability to cope with complex geometries and the lack of artificial infinite boundaries [1-4]. Our work is based on Morino's formulation [5-8] with plane low order panels for the potential field. A characteristic of such formulation is that it gives a full non-symmetric matrix with relatively low condition numbers. To be more specific, the condition number does not degrade under refinement as it is common in the *"in volume"*discretization methods, like FDM, FEM, FVM or the relatively young "mesh-less" methods. This advantage is somewhat compensated by the fact that the system matrix is full and a definitive assessment of the efficiency of the method, compared to the "in volume" ones is rather involved. However, a clear advantage of the method, mainly from the practical point of view, is the lack of domain interior meshing, specially in 3D.

Application of iterative solvers to panel/BEM (from "Boundary Element Method") problems is described in many papers [9-14]. A rough list of parameter-free iterative solvers for non symmetric systems of linear equations is given by Natchigal *et al* [17], where three basic types are considered as follows. First, those methods based on the normal equations: CGN = CGNR (Hestnes and Stiefel '52 [22], CGNE (Craig'55, LSQR (Paige and Saunders '82 [23]. Second, those ones based on orthogonalization: GCG (Concus and Golub '76, Widlund '78), ORTHOMIN (Vinsome '76), ORTHORES and ORTHODIR (Young and Jea '90), FOM (Saad '81), GCR (Elman '82, Eisenstat et al. [24], GMRES (Saad and Schultz [25]. Third, those ones based on biorthogonalization: BCG = BIOMIN (Lanczos [18], Fletcher [19], BIORES = BO (Lanczos '50, Jea and Young '83, BIODIR (Jea and Young '83), CGS = BIOMIN^2 (Sonneveld [20], BIORES^2 and BIODIR^2 (Gutknecht [26], BiCGSTAB (van der Vorst [27], QMR (Freund [28]. We also have the USYMLQ and USYMQR methods (Saunders et al [28]. These terminology approximately follows Nachtigal *et al.* and Gutknecht [26] and others references can be found in [29] and [30]. Iterative solvers are based on repetitive calculation of matrix-vector multiplication. As the matrices coming from panel discretizations are full, it is not possible to store the matrix coefficients in core memory, as it is usual in the "in volume" methods, where the matrix is sparse. Then, the interaction coefficients have to be recomputed at each matrix-vector operation and the overall cost is roughly the number of matrix-vector operations times the cost of evaluating one of them. Global efficiency is controlled thus by: a a choice of the iterative solver and preconditioning in order to improve the convergence rate and reduce the number of matrix-vector operations, and b) an efficient computation of the interaction coefficients.

Some preconditioners are purely algebraic, as those based on incomplete factorization (see [9] for instance), whereas others take into account the underlying physics. Most of the physically based preconditionings for the panel method are based on some kind of multi polar expansion of the field produced by a *"cluster"* of panels [13, 14,15]. The advantage of the algebraic preconditionings are that they can be applied to a broader range of problems, whereas physical based preconditionings yield better performances at the expense of being more specific.

Preconditioners in a general BEM context are extensively reviewed by Prasad et al [10], where some success has been reported with the conjugate gradient and GMRES when they are used in conjunction with preconditioning approaches. Also Hribersek et al [11] have considered Jacobi, incomplete factorization and row-sum type preconditioners for the BEM solution of viscous flow problems, showing improved convergence rates with the first two ones. Yan [21] obtained sparse preconditioners for dense system matrix in 2D BEM analysis through condensation by discrete Fourier transforms, whereas Vavasis [15] treats the panel/BEM case which is rather near to our case, ie. solving the Laplace equation in a exterior 3D domain, so we will give a brief account of his approach. Vavasis considers three basic preconditioners: the *mesh neighbor*, the *matrix entries* and the *hierarchical clustering* which have the following idea in common. From the system matrix $\mathbf{A}$ a small index list L is chosen, drawn from $\{1, 2, .., N_i\}$ such that the variables in L have the most influence on the variable i. Next, a small system of equations $\overline{\mathbf{A}}^T \overline{\mathbf{p}}_i = \overline{\mathbf{e}}_i$ is solved, where the bars denote that all the rows and the columns of $\mathbf{A}$ are deleted except for those in the index list L. Once this solution is known, it expands back to entries of the preconditioner and this procedure is done for all its rows. On one hand, in the *mesh neighbor* preconditioner we take into account that the matrix coefficient relating the control points i and j comes up roughly from a term like $1/|\mathbf{x}_i - \mathbf{x}_j|$. Therefore, the further apart two control points are, the less influence it would expect a change at one control point to have on the other. It is said that two control points are "neighbors" if they border a common panel side.

Since neighboring control points are the most interrelated, then they are put in the mesh neighbor preconditioner. Also we note that this strategy leads a sparse preconditioner where its sparsivity pattern will mirror the mesh connectivity. On the other hand, the mesh entries preconditioner constructs the index list L with the following criterion: if a_{ij}, a_{ji} satisfy $|a_{ij}a_{ji}| \geq t|a_{ii}a_{jj}|$, then the control point j is included in the index list L, where t denotes some user-specified tolerance, or "magic number". For instance, when $a_{ii} = 1/2$ and $|a_{ij}| < a_{ii}$, with $t = 1,0$ a diagonal and full populated preconditioner are obtained, respectively, whereas middle values of t does not offer any regular or predictable sparsivity pattern. In an algebraic context, Jemmings [21] proposed to adopt $0.01 \leq t \leq 0.10$. As it can be noted, this type of preconditioner does not depend on any panel/BEM formulation, hence it can be applied to an arbitrary system matrix. Finally the *hierarchical clustering* preconditioner classifies the control points according to how far away they lie from the control point i. The first step in its construction is to make a hierarchy of clusters, next to the center and radii r for each cluster C are obtained. Once the clusters, centers and radii are computed, the hierarchy preconditioner is constructed with the aid of the index list L obtained as the acceptable clusters of each control point. It is said that cluster C is acceptable to the control point i when the distance from the control point i to the center of C is at least tr, where $t > 1$ is a user-specified number. Vavasis reports good improvements with the first two types for rather thick 3D geometries on industrial applications. Further details about these three types of preconditioners adapted for a dense and unsymmetric matrix system can be found in the Vavasis's work and the reference listed there.

However, for thin wings we found that the performance of these kind of preconditioners is rather restricted. On one hand, the *mesh neighbor* preconditioner assumes that the importance of the influence matrix coefficient is only related with the mesh topology, ie. neighbor panels have a strong interaction and remote ones a weaker interaction. However, this assumption is not always the case for *dipolar* matrices, which are proportional also to a view factor, so that neighbor panels (in a topological sense) that are nearly coplanar have weak interaction, which is actually the case of well refined meshes on smooth surfaces. Moreover, remote panels (again in a topology sense) on opposite sides of thin wing geometries and facing each other, have a strong interaction. In other words, the mesh neighbor preconditioner strategy conduces to rather lower performances in such cases. On the other hand, the *matrix entries* preconditioner is a good alternative for thick geometries, with usually low rate in the filling of the preconditioner (ie, its non zero entries). But, for thin wing configurations (similar to the considered in the numerical examples) we had to use $t > 0.8$, with filling ratios greater than 80 %. Finally, the *hierarchical clustering* preconditioner shares similar characteristics with the *mesh neighbor* one.

For most iterative solvers, the convergence rate is closely related to the condition number of the linear system, which highly depends on the geometry. To be more specific, high condition numbers arise whenever two surfaces are close together, so that the distance between them is smaller than the average size of the panels. Note that this implies that, in contrast with the "in volume" methods, the condition number is reduced after refinement, since, eventually the average size of the panels will get smaller than the distance between the surfaces. Now, suppose for a moment that the profile is symmetric and with a small thickness ϵ, and we are interested in $\epsilon \to 0$. Any distribution of double layer density μ given by $\mu_{1,2}$ on each side of the wing can be decomposed as the sum of a symmetric one μ^+ and a skew-symmetric one μ^- where $\mu^\pm = 1/2(\mu_1 \pm \mu_2)$. Now in the limit of vanishing thickness, the field produced by a skew-symmetric density on both sides of the wing are added and the result is a flat surface at the plane of symmetry with a distribution of $2\mu^-$, whereas for the symmetric distribution the fields tend to cancel each other and the result is also a flat surface but with null double layer density distribution. This shows that the self interaction coefficient for the skew-symmetric distribution behaves like $O(1)$ whereas those for the symmetric one behaves like $O(\epsilon)$ for $\epsilon \to 0$. This explains why the condition number degrades as $\epsilon \to 0$ and suggests that a physically based preconditioning based on making the change of variables and scaling appropriately the symmetric part will correct this degradation. We call this the "modal preconditioning" and is, in our opinion, the main contribution of this paper.

One may argue that, if the preconditioning is efficient for profiles that are too thin, then it would be better to handle those case specially, i.e. modeling them as plain flat surfaces. Firstly, we will show in the examples that interesting gains are obtained for profiles of 8 % and even 25 % relative thickness. Secondly, even if the airfoil is not too thin, the thickness is smaller near the trailing edge and this is a cause of slower convergence. Thirdly and last, using the preconditioning allows us to solve arbitrarily thin airfoils without worrying about the degree of the approximation of replacing by a zero thickness airfoil.

With respect to an efficient evaluation of the interaction coefficients, it is a common practice to use approximated expressions based on far-field expansions, valid when both panels are separated by a distance that exceeds some threshold value scaled by the panel size. Typically, the cost of the far-field expression is faster than the exact evaluation in a ratio 1:8. However, this introduces a consistency error, and in order to eliminate it, we iterated an outer loop where a residual with the exact coefficients is computed and a correction is added. It is clear that if too much iterations of the outer loop (≥ 8) are performed, then it is cheaper to iterate directly on the exact coefficients, but it can be shown that this exterior problem is very well conditioned and, typically, 2 iterations are needed to reduce the error by a factor 10^{-7}. This issue is discussed in depth elsewhere [16].

4. Panel discretization overview

Let Γ be a closed surface, and $\Omega^{i(e)}$ the corresponding interior (exterior) domain. The governing equations for potential flow are the Laplace equation in Ω^e with slip condition on Γ which, in terms of the perturbation potential can be written as:

$$\begin{aligned} &\Delta\phi = 0, \quad \text{in } \Omega^e \\ &\frac{\partial \phi}{\partial n} = -\mathbf{u}_\infty \cdot \hat{\mathbf{n}}, \quad \text{at } \Gamma \end{aligned} \tag{1.a, b}$$

where $\mathbf{u}_\infty$ is the undisturbed velocity and ϕ the perturbation potential defined by:

$$\mathbf{u} = \mathbf{u}_\infty + \nabla\phi \tag{2}$$

where $\mathbf{u}$ is the total velocity. This problem can be rewritten as a Fredholm integral equation in terms of the single and double layer densities σ and μ as:

$$\mu(\mathbf{x}) - \frac{1}{2\pi}\int_\Gamma \mu(\mathbf{x}')\frac{\partial}{\partial n}\left(\frac{1}{|\mathbf{x}-\mathbf{x}'|}\right)\, d\Gamma' = \frac{1}{2\pi}\int_\Gamma \sigma(\mathbf{x}')\frac{1}{|\mathbf{x}-\mathbf{x}'|}\, d\Gamma' \tag{3}$$

for $\mathbf{x}$ belonging to the surface Γ, and $\hat{\mathbf{n}}$ is the normal pointing into Ω^e. For the slip or inlet/outlet boundary condition as in (1b) it turns out to be that σ is simply $\sigma = -\hat{\mathbf{n}} \cdot \mathbf{u}_\infty$ and then, the right hand side of (3) is known. Moreover, μ is equal to the perturbation potential ϕ at the surface. The panel method is based on approximating Γ by a polyhedral surface, composed of a certain number N_{pan} of non-overlapping flat panels $\{\Gamma_i\}_{i=1}^{N\mathrm{pan}}$. Assuming that μ and σ are constant over each panel and imposing (3) by collocation at the centroids of the panels, a linear system of the form:

$$\mathbf{A}\boldsymbol{\mu} = \mathbf{C}\boldsymbol{\sigma} \tag{4}$$

is obtained, where $\boldsymbol{\mu} = [\mu_1 \mu_2 \ldots]^T$ is the vector of panel potentials, $\boldsymbol{\sigma}$ the single layer densities per panel, computed as:

$$\sigma_i = -\mathbf{u}_\infty \cdot \hat{\mathbf{n}}_i \tag{5}$$

and $\hat{\mathbf{n}}_i$ is the normal to panel i. The interaction coefficients are computed from:

$$\begin{aligned} C_{ij} &= \int_{\Gamma_j} \frac{1}{|\mathbf{x} - \mathbf{x}_i|}\, d\Gamma \\ A_{ij} &= \int_{\Gamma_j} \frac{\partial}{\partial n} \left\{ \frac{1}{|\mathbf{x} - \mathbf{x}_i|} \right\}\, d\Gamma \end{aligned} \tag{6}$$

where $\mathbf{x}_i$ is the centroid of the i-panel.

5. The modal preconditioning

We will show that, for thin airfoils a symmetric/skew-symmetric decomposition reveals a conditioning which grows inversely proportional to the thickness. Then, a preconditioning based on this symmetric/skew-symmetric decomposition is proposed. We will show how the preconditioning corrects the bad scaling for a simple geometry consisting of two facing panels, then for two pairs of panels on a symmetric airfoil, and two pairs of panels on a non-symmetric airfoil. Finally the expression for the preconditioning in the general case is shown.

6. A pair of panels:

Consider first two identical flat square panels of side H, parallel to each other as in figure 1 separated by a distance $d = H\epsilon$. We should regard this as a very crude idealization of a slender airfoil, so that we take the normals as shown in the figure. The matrix of interaction coefficients is:

$$\mathbf{A} = \begin{bmatrix} 1/2 & -g(\epsilon) \\ -g(\epsilon) & 1/2 \end{bmatrix} \tag{7}$$

where we replaced $A_{11} = A_{22} = 1/2$, as is usual for the self-interaction coefficients, and by symmetry $A_{21} = A_{12} = -g(\epsilon)$, where $g(z)$ is the potential produced by a unit distribution of double layer potential at a distance z on an axis perpendicular to the panel and passing through its centroid, z is taken as positive if the point is on the side with positive charge. The qualitative behavior for $g(z)$ is like that in figure 2, it is antisymmetric with respect to $z = 0$, it has a jump of unit magnitude (i.e. equal to the double layer density) and continuous derivative at $z = 0$:

$$\begin{aligned} g\big|_{z=0^+} &= +1/2 \\ g\big|_{z=0^-} &= -1/2 \\ \left.\frac{dg}{dz}\right|_{z=0^+} &= \left.\frac{dg}{dz}\right|_{z=0^-} = -q \end{aligned} \tag{8}$$

Then for small ϵ we can put, to first order:

$$A = \begin{bmatrix} 1/2 & -1/2 + q\epsilon \\ -1/2 + q\epsilon & 1/2 \end{bmatrix} \tag{9}$$

As the problem is symmetric about the center plane, it decouples in a symmetric/antisymmetric basis. Let:

$$\mathbf{S} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{10}$$

be the change of basis matrix, then the transformed matrix is a diagonal one:

$$\begin{aligned} \bar{\mathbf{A}} &= \mathbf{S}^{-1}\,\mathbf{A}\mathbf{S} \\ &= \begin{bmatrix} q\epsilon & 0 \\ 0 & 1 - q\epsilon \end{bmatrix} \end{aligned} \tag{11.a,b}$$

and the bad conditioning is clear since for $\epsilon \to 0$ the first diagonal entry vanishes:

$$\mathrm{cond}(\mathbf{A}) = \mathrm{cond}(\bar{\mathbf{A}}) = \frac{1 - q\epsilon}{q\epsilon} \sim \frac{1}{q\epsilon} \quad \text{for large} \quad \epsilon \tag{12}$$

Let us take a closer look at what matrix $\mathbf{S}$ represents. If we take its columns as double-layer density distributions, then the first column corresponds to $\mu_{1,2} = 1$, and the second one to $\mu_1 = -\mu_2 = 1$ (see figure 3). We must recall that a double-layer density distribution can be thought of as two single layer distributions of equal strength but opposite sign in the limit when the distance between the single layer distributions tends to zero, keeping the product of the distance and the strength of the single layer distributions constant. By convention, the normal vector points from the negative charge side to the positive one for a positive double layer distribution. Note that this distribution corresponds to a symmetric arrangement of charge about the symmetry plane, whereas the second column, corresponds to a skew-symmetric one. The coefficients in matrix $\tilde{\mathbf{A}}$ are the interaction coefficient between these two distributions. It is clear from symmetry arguments that the symmetric arrangement can not have a non zero interaction with the skew-symmetric one, and this explains why the off-diagonal entries in $\tilde{\mathbf{A}}$ are null. Now, regarding the diagonal entries, for $\epsilon \rightarrow 0$ the double layer distribution of panel 1 tends to cancel that one of panel 2, so that the field vanishes to zero. The first diagonal entry in $\tilde{\mathbf{A}}$, see equation (11.b), represents the self-interaction coefficient of this arrangement, and then, vanishes for $\epsilon \rightarrow 0$. In contrast, in the skew-symmetric arrangement, the field of each panel tends to reinforce that of the other and, in the limit, a single panel with the original distribution which is twice as much is obtained. The second diagonal entry corresponds to the self-interaction coefficient for this charge distribution and then, it approaches a non-null value for $\epsilon \rightarrow 0$. As for this very simple ($N_{\mathrm{pan}} = 2$) case $\mathbf{S}^{-1}\,\mathbf{AS}$ is diagonal, then:

$$\mathbf{A} = \mathbf{S}\;\mathrm{diag}(\mathbf{S}^{-1}\,\mathbf{AS})\,\mathbf{S}^{-1} \tag{13}$$

where diag($\mathbf{X}$) stands for a diagonal matrix with the same diagonal entries of $\mathbf{X}$:

$$\mathrm{diag}(\mathbf{X}) = \begin{bmatrix} X_{11} & 0 & \dots & & \\ 0 & X_{22} & 0 & \dots & \\ \vdots & 0 & X_{33} & 0 & \dots \\ & \vdots & \vdots & \vdots & \end{bmatrix} \tag{14}$$

It is well known that the best preconditioning $\mathbf{Q}$ is that one that most resembles $\mathbf{A}$ for a given computational effort in inverting a linear system for the preconditioning matrix. In this simple case we can take $\mathbf{Q} = \mathbf{A}$, and from (13):

$$\mathbf{Q} = \mathbf{S}\;\mathrm{diag}(\mathbf{S}^{-1}\,\mathbf{AS})\,\mathbf{S}^{-1} \tag{15}$$

and this preconditioning will give a preconditioned matrix which is the identity, and then is optimal.

In the next section we will extend this preconditioning to the case with large number of panels and we will show that it has good preconditioning properties. Note that the computational effort in solving a system for $\mathbf{Q}$ as in (13) is negligible since the change of basis is performed by $N_{\mathrm{pan}}/2$ sums and differences, and the inversion of the diagonal part of the transformed matrix involves $O(N_{\mathrm{pan}})$ operations.

7. Symmetric airfoil:

Consider now the case of a symmetric airfoil discretized with a large number of panels. It is easy to see that it suffices to consider the interaction coefficients between two pair of panels as shown in figure 4. As the airfoil is symmetric, panels 2 and 4 are obtained from panels 1 and 3 by reflection about the center plane of the airfoil. For simplicity, we will assume also that all the panels are parallel to the symmetry plane, and that panels 1 and 3 are identical (and then 2 and 4), but these assumptions are not essential. The structure of the matrix is then:

$$\mathbf{A} = \begin{bmatrix} 1/2 & -1/2 + q\epsilon & 0 & -p\epsilon \\ -1/2 + q\epsilon & 1/2 & -p\epsilon & 0 \\ 0 & -p\epsilon & 1/2 & -1/2 + q\epsilon \\ -p\epsilon & 0 & -1/2 + q\epsilon & 1/2 \end{bmatrix} \tag{16}$$

Of course, the 2 × 2 diagonal matrix blocks, are the the same as in the 2-panel example. Regarding the off-diagonal terms, the in-plane coefficients like A_{13}, A_{31}, A_{24} and A_{42} are null, which can be easily shown from (6), whereas the others, like A_{14} are $O(\epsilon)$ and negative, say $-p\epsilon$. Now, we write the change of basis matrix as:

$$\mathbf{S} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{bmatrix} \tag{17}$$

Note that, again, the first two columns correspond to symmetric (with respect to the horizontal plane of symmetry) distribution of potentials, whereas the last two correspond to skew-symmetric ones. The transformed matrix is:

$$\tilde{\mathbf{A}} = \begin{bmatrix} q\epsilon & -p\epsilon & 0 & 0 \\ -p\epsilon & q\epsilon & 0 & 0 \\ 0 & 0 & 1 - q\epsilon & p\epsilon \\ 0 & 0 & p\epsilon & 1 - q\epsilon \end{bmatrix} \tag{18}$$

and it is verified that the off diagonal 2 × 2 blocks are null. In addition, the first block diagonal entry corresponding to interaction coefficients between the symmetric distributions has terms $O(\epsilon)$, whereas the second block diagonal entry corresponding to the skew-symmetric mode is $O(1)$. The condition number is again

$O(1/\epsilon)$ and can be corrected with the preconditioning defined by (15). Effectively:

$$\begin{aligned}
\operatorname{cond}(\mathbf{A}\mathbf{Q}^{-1}) &= \operatorname{cond}(\mathbf{A}\,\mathbf{S}\ \operatorname{diag}(\mathbf{S}^{-1}\,\mathbf{A}\mathbf{S})^{-1}\,\mathbf{S}^{-1}) \\
&= \operatorname{cond}(\bar{\mathbf{A}}\ \operatorname{diag}(\bar{\mathbf{A}})^{-1}) \\
&= \begin{bmatrix} 1 & -p/q & 0 & 0 \\ -p/q & 1 & 0 & 0 \\ 0 & 0 & 1 & p\epsilon/(1-q\epsilon) \\ 0 & 0 & p\epsilon/(1-q\epsilon) & 1 \end{bmatrix} \\
&= O(1) \quad \text{for} \quad \epsilon \to 0
\end{aligned} \tag{19}$$

8 Non-symmetric airfoil:

In this case the change of basis does not decouple the problem as cleanly as in the symmetric case, but we can still show how the proposed preconditioner gives an $O(1)$ condition number, which is verified afterwards with numerical examples. As before, we will consider only two pair of panels. For convenience we make the following block decomposition of $\bar{\mathbf{A}}$:

$$\bar{\mathbf{A}} = \begin{bmatrix} \bar{\mathbf{A}}_{++} & \bar{\mathbf{A}}_{+-} \\ \bar{\mathbf{A}}_{-+} & \bar{\mathbf{A}}_{--} \end{bmatrix} \tag{20}$$

where each of the sub-matrices are 2×2, and we will assess the order with respect to ϵ of each of the elements in each of the blocks. This is done in the Appendix and the conclusions are:

$$\bar{\mathbf{A}}_{++} = \epsilon \bar{\mathbf{A}}^{*}_{++} + O(\epsilon^2) \tag{21}$$

$$\bar{\mathbf{A}}_{-+} = \epsilon^2 \bar{\mathbf{A}}^{*}_{-+} + O(\epsilon^3) \tag{22}$$

$$\bar{\mathbf{A}}_{+-} = \bar{\mathbf{A}}^{*}_{+-} + O(\epsilon) \tag{23}$$

$$\bar{\mathbf{A}}_{--} = \bar{\mathbf{A}}^{*}_{--} + O(\epsilon) \tag{24}$$

where $\bar{\mathbf{A}}^{*}_{..,..} = O(1)$.

9. Efficiency of the preconditioning:

From (21), (22), (23) and (24) we can write:

$$\bar{\mathbf{A}} \sim \begin{bmatrix} \epsilon \bar{\mathbf{A}}^{*}_{++} & \bar{\mathbf{A}}^{*}_{+-} \\ \epsilon^2 \bar{\mathbf{A}}^{*}_{-+} & \bar{\mathbf{A}}^{*}_{--} \end{bmatrix} \tag{25}$$

It is clear from this expression that $\mathrm{cond}(\mathbf{A}) \sim 1/\epsilon$ (or worse). In contrast, if it is preconditioned with (15):

$$\begin{aligned}
\mathbf{AQ}^{-1} &= \mathbf{A}\left[\mathbf{S}\,\mathrm{diag}(\mathbf{S}^{-1}\mathbf{AS})\mathbf{S}^{-1}\right]^{-1} \\
&= \mathbf{AS}\,\mathrm{diag}(\mathbf{S}^{-1}\mathbf{AS})^{-1}\mathbf{S}^{-1} \\
&= \mathbf{S}\bar{\mathbf{A}}\,\mathrm{diag}(\bar{\mathbf{A}})^{-1}\mathbf{S}^{-1} \\
&\sim \mathbf{S}\begin{bmatrix} \epsilon\bar{\mathbf{A}}^*_{++} & \bar{\mathbf{A}}^*_{+-} \\ \epsilon^2\bar{\mathbf{A}}^*_{-+} & \bar{\mathbf{A}}^*_{--} \end{bmatrix} \begin{bmatrix} \epsilon\,\mathrm{diag}(\bar{\mathbf{A}}^*_{++}) & 0 \\ 0 & \mathrm{diag}(\bar{\mathbf{A}}^*_{--}) \end{bmatrix}^{-1} \mathbf{S}^{-1} \qquad (26.\text{a-f}) \\
&\sim \mathbf{S}\begin{bmatrix} \bar{\mathbf{A}}^*_{++}\,\mathrm{diag}(\bar{\mathbf{A}}^*_{++})^{-1} & \bar{\mathbf{A}}^*_{+-}\,\mathrm{diag}(\bar{\mathbf{A}}^*_{--})^{-1} \\ \epsilon\bar{\mathbf{A}}^*_{-+}\,\mathrm{diag}(\bar{\mathbf{A}}^*_{++})^{-1} & \bar{\mathbf{A}}^*_{--}\,\mathrm{diag}(\bar{\mathbf{A}}^*_{--})^{-1} \end{bmatrix}\mathbf{S}^{-1} \\
&\sim \mathbf{S}\begin{bmatrix} \bar{\mathbf{A}}^*_{++}\,\mathrm{diag}(\bar{\mathbf{A}}^*_{++})^{-1} & \bar{\mathbf{A}}^*_{+-}\,\mathrm{diag}(\bar{\mathbf{A}}^*_{--})^{-1} \\ 0 & \bar{\mathbf{A}}^*_{--}\,\mathrm{diag}(\bar{\mathbf{A}}^*_{--})^{-1} \end{bmatrix}\mathbf{S}^{-1}
\end{aligned}$$

and then:

$$\mathrm{cond}(\mathbf{AQ}^{-1}) = O(1) \tag{27}$$

provided that the argument matrix in (26.d) is non-singular. Again, the bad conditioning is caused by the symmetric modes. Due to the tendency to cancel the field of the facing panel when $\epsilon \to 0$,the field produced by these modes is $O(\epsilon)$ and so are the interaction coefficients (the first matrix column in (25)). The diagonal preconditioning (in the transformed basis) successfully corrects this behavior.

10. Explicit expression for real ($N_{\mathrm{pan}} > 2$) meshes:

We give now the explicit expression for the change of basis matrix for the case where $N_{\mathrm{pan}} > 4$. We assume that the panels are numbered so that panel $2n-1$ and $2n$ are on opposite sides of the airfoil and they collapse to each other for $\epsilon \to 0$. The change of basis matrix is formed by putting first all the $N_{\mathrm{pan}}/2$ symmetric modes and after the skew-symmetric ones:

$$\mathbf{S} = \frac{1}{\sqrt{2}}\left[\begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes \mathbf{I} \quad \begin{bmatrix} 1 \\ -1 \end{bmatrix} \otimes \mathbf{I}\right] \tag{28}$$

where $\mathbf{I}$ stands for the identity matrix of $(N_{\mathrm{pan}}/2) \times (N_{\mathrm{pan}}/2)$ and the Kronecker product $\otimes$ of two matrices is defined by:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} B_{11}\mathbf{A} & B_{12}\mathbf{A} & B_{13}\mathbf{A} & \dots \\ B_{21}\mathbf{A} & B_{22}\mathbf{A} & B_{23}\mathbf{A} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{29}$$

With respect to the limits of applicability, the mesh on both surfaces have to be congruent, i.e. for $\epsilon \to 0$ the nodes and elements have to coincide with each other and the relative shift has to normal to the surface. Small deviation or distortions are acceptable provided that they are small with respect to the average element size (see figure 6).

11. Multipolar preconditioning

This preconditioning somewhat follows the lines of the modal one previously presented, but it is oriented towards to a single surface, instead. It also has some resemblance to multigrid methods. We will see in the next section how to combine it with the modal preconditioning. Consider, for instance, two pair of panels as shown in figure 7 and let us start with the same change of basis as before eq. (17). We assume that both panels are coplanar and have the same area. This is true or nearly true for highly structured meshes. We will discuss later how to extend the multipolar preconditioning for unstructured ones. We make the same block split as in (20) and we will analyze each block at a time. Consider an off-diagonal term in $\bar{\mathbf{A}}_{--}$ like:

$$\bar{A}_{34} = {}^1\!/\!_2(A_{31} - A_{32} - A_{41} + A_{42}) \tag{30}$$

As before, $A_{31} - A_{32}$ is the potential produced by a double layer density distribution of $\mu = +1$ on panel 1 and $\mu = -1$ on panel 2. Suppose now that panels 3 and 4 are far from panels 1 and 2, i.e.:

$$|\mathbf{x}| = |\mathbf{x}_{34} - \mathbf{x}_{12}| \gg h \tag{31}$$

where $\mathbf{x}_{34}$ is the centroid of the *"panel cluster"* composed of panels 3 and 4 and so on. Then, we can approximate $A_{31} - A_{32}$ by a quadrupole expansion:

$$A_{31} - A_{32} \approx -3ah\,\frac{xz}{r^5} = \phi_{12}(x, y, z) \tag{32}$$

where a is the area of the panels, the system xyz is chosen as in figure 8, and $r^2 = x^2 + y^2 + z^2$. Now, let $\hat{\mathbf{s}}$ be a unit vector going from the centroid of panel 3 to that of panel 4. Then:

$$\begin{aligned}\bar{A}_{34} &= -\,{}^1\!/\!_2[\phi_{12}(\mathbf{x}_{34} + {}^1\!/\!_2 h\hat{\mathbf{s}}) + \phi_{12}(\mathbf{x}_{34} - {}^1\!/\!_2 h\hat{\mathbf{s}})] \\ &\approx {}^1\!/\!_2 h(\hat{\mathbf{s}} \cdot \nabla\phi_{12}) \\ &= O(h^2/r^4)\end{aligned} \tag{33}$$

Note that this is $O[(h/r)^2]$ smaller than the typical interaction coefficient between two panels, which has typically a decay rate of a dipole $O(1/r^2)$. In contrast, the diagonal elements are $O(1)$. In a similar way, it can be shown that the off-diagonal elements in $\bar{\mathbf{A}}_{-+}$ and $\bar{\mathbf{A}}_{+-}$ are $O(h/r^3)$. The diagonal elements vanish by symmetry. Finally the elements in $\bar{\mathbf{A}}_{++}$ are basically the same as if we added each pair of panels in a single larger panel, with two points of collocation, i.e. it has the same structure of the original matrix $\mathbf{A}$ but it has a dimension which is smaller by half. In brief:

$$\bar{\mathbf{A}} = \begin{bmatrix} O(\mathbf{I}) + O(1/r^2) & O(h/r^3) \\ O(h/r^3) & O(\mathbf{I}) + O(h^2/r^4) \end{bmatrix} \tag{34}$$

where $O(\mathbf{I}) + O(1/r^2)$ means that the diagonal elements are $O(1)$ whereas the other off-diagonal terms are $O(1/r^2)$. The preconditioning we propose is obtained retaining only the $\bar{\mathbf{A}}_{++}$ block and the diagonal part of $\bar{\mathbf{A}}_{--}$:

$$\mathbf{Q} = \mathbf{S} \begin{bmatrix} \bar{\mathbf{A}}_{++} & 0 \\ 0 & \mathrm{diag}(\bar{\mathbf{A}}_{--}) \end{bmatrix} \mathbf{S}^{-1} \tag{35}$$

The computational effort in inverting this conditioning corresponds to inverting the $\bar{\mathbf{A}}_{++}$ block. As this has half the dimension of the full matrix, the core memory requirement is one fourth smaller and the CPU time one eight smaller than those for the full matrix, which is a significant saving. However, panel applications are limited strongly by the matrix size, and then, we are interested in preconditionings requiring smaller amounts of memory at the expense of higher CPU times. This is achieved using larger clusters. Consider a cluster of four panels (two such clusters are shown in figure 9). The change of basis we consider is now:

$$\mathbf{S} = {}^1\!/_2 \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 1 & -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & -1 & 1 & -1 \\ 0 & 1 & 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & 1 & 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix} \tag{36}$$

Columns 1, 3, 4 and 5 correspond to double layer distributions on cluster 1 whereas the others correspond to cluster 2. Column 1 represents a constant distribution of double layer density so its far field expansion is a dipole, whereas columns 3, 4 and 5 have a null total sum of double layer density and their far field expansion are therefore at least of the order of a quadrupole, see figure 10. The same occurs, respectively, for columns 2, 6, 7 and 8 for cluster 2. If we call the first two columns as the "+ *modes*" and the rest as the "− *modes*", and a block split in the transformed matrix is induced. The matrices $\bar{\mathbf{A}}_{++}$, $\bar{\mathbf{A}}_{-+}$, $\bar{\mathbf{A}}_{+-}$ and $\bar{\mathbf{A}}_{--}$ have, respectively, sizes of 2×2, 2×6, 6×2 and 6×6. The behavior with respect to h/r are as before or higher: for instance $\bar{A}_{58}$ corresponds to the interaction of an octupole on cluster 2 double differentiated at the center of cluster 1. This is then $O(h^4/r^6)$. The size of the $\bar{\mathbf{A}}_{++}$ block in the preconditioning given by (35) is now 2×2, one fourth the size of $\mathbf{A}$.

In this way, by recursion, larger and larger clusters can be used and the dimension of the matrix to be inverted is $O(N_{\rm pan}/m)$ where $m = 2^n$ is the number of panels in a cluster. Consider a cluster of 2^n panels, then the *"in-cluster"* change of basis matrix $\mathbf{S}_{\rm cluster}$ is defined recursively as:

$$\begin{aligned} \mathbf{S}_{\rm cluster}(2^n) &= \mathbf{S}_{\rm cluster}(2) \otimes \mathbf{S}_{\rm cluster}(2^{n-1}) \\ \mathbf{S}_{\rm cluster}(2) &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \end{aligned} \tag{37}$$

We split $\mathbf{S}_{\rm cluster}$ in its first column representing a constant distribution of double layer density and all the rest representing higher order distributions:

$$\mathbf{S}_{\rm cluster} = [\, \mathbf{S}^{+}_{\rm cluster} \quad \mathbf{S}^{-}_{\rm cluster} \,] \tag{38}$$

where:

$$\mathbf{S}^{+}_{\rm cluster} = 2^{-(n-1)} \, [1 \quad 1 \quad \ldots \quad 1]^T \tag{39}$$

and $\mathbf{S}^{-}_{\rm cluster}$ is a matrix of $2^n \times (2^n - 1)$ with the rest of the columns. The change of basis matrix is, then:

$$\mathbf{S}_{\rm mp} = [\, \mathbf{S}^{+}_{\rm cluster} \otimes \mathbf{I} \quad \mathbf{S}^{-}_{\rm cluster} \otimes \mathbf{I} \,] \tag{40}$$

where $\mathbf{I}$ is an identity matrix of size $N_{\rm pan}/2^n$. Finally we would like to say a few words on the implementation of the change of basis for large clusters. As long as not so large clusters are used, the cost of a change of basis is negligible. If very large clusters are used, and a *"naive"* implementation (i.e. as a literal matrix-vector product) is used, the cost could affect performance. However, this kind of change of basis can be seen as the *"discrete Haar transform"*, which is well known in the theory of discrete signals. There are efficient implementations of this kind of transformation, in the same spirit of the well known *Fast Fourier Transform* algorithm.

12. The modal and multipolar preconditionings combined

To combine both preconditionings for a thin airfoil, we first transform the matrix to the modal symmetric/skew-symmetric basis, as explained in section §7 and §8. The change of basis matrix is:

$$\mathbf{S}_{\mathrm{mod}} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & & \dots & 1 & 0 & & \dots \\ 1 & 0 & \ddots & & -1 & 0 & \ddots & \\ 0 & 1 & \ddots & \ddots & 0 & 1 & \ddots & \ddots \\ 0 & 1 & 0 & & 0 & -1 & 0 & \\ \vdots & 0 & 1 & \ddots & \vdots & 0 & 1 & \ddots \\ & 0 & 1 & \ddots & & 0 & -1 & \ddots \\ & \vdots & 0 & \ddots & & \vdots & 0 & \ddots \\ & & 0 & \ddots & & & 0 & \ddots \\ & & \vdots & & & & \vdots & \\ & & & & & & & \end{bmatrix} \tag{41}$$

And the transformed matrix is:

$$\bar{\mathbf{A}}_{\mathrm{mod}} = \mathbf{S}_{\mathrm{mod}}{}^{-1}\mathbf{A}\mathbf{S}_{\mathrm{mod}} = \begin{bmatrix} \bar{\mathbf{A}}_{\mathrm{mod}}^{++} & \bar{\mathbf{A}}_{\mathrm{mod}}^{+-} \\ \bar{\mathbf{A}}_{\mathrm{mod}}^{-+} & \bar{\mathbf{A}}_{\mathrm{mod}}^{--} \end{bmatrix} \tag{42}$$

For thin airfoils ($\epsilon \to 0$) $\bar{\mathbf{A}}_{\mathrm{mod}}^{--}$ tends to a matrix with interaction coefficients for panels with double layer density distributions, as if the finite thin airfoil were replaced by a zero thickness airfoil at the center plane. We can then apply the multi-polar preconditioning with, say, 4-panels clusters. The change of basis matrix is the identity for the symmetric modes and the multipolar matrix $\mathbf{S}_{\mathrm{mp}}$ for $N_{\mathrm{pan}}/2$ panels given by (40):

$$\mathbf{S} = \mathbf{S}_{\mathrm{mod}} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{\mathrm{mp}} \end{bmatrix} \tag{43}$$

The transformed matrix has a structure of the following form:

$$\bar{\mathbf{A}}_{\mathrm{md/mp}} = \mathbf{S}^{-1}\mathbf{A}\mathbf{S} = \begin{bmatrix} \bar{\mathbf{A}}_{\mathrm{mod}}^{++} & \bar{\mathbf{A}}_{\mathrm{mod}}^{+-}\mathbf{S}_{\mathrm{mp}} \\ \mathbf{S}_{\mathrm{mp}}^{-1}\bar{\mathbf{A}}_{\mathrm{mod}}^{-+} & \mathbf{S}_{\mathrm{mp}}^{-1}\bar{\mathbf{A}}_{\mathrm{mod}}^{--}\mathbf{S}_{\mathrm{mp}} \end{bmatrix} \tag{44}$$

The $\mathbf{S}_{\mathrm{mp}}$ change of basis matrix induces a block decomposition (20) on $\bar{\mathbf{A}}^{--}_{\mathrm{mod}}$ and the same estimation of the order of coefficients as in equation (34), section §11, is valid.

$$\mathbf{S}^{-1}_{\mathrm{mp}}\bar{\mathbf{A}}^{--}_{\mathrm{mod}}\mathbf{S}_{\mathrm{mp}} = \begin{bmatrix} \bar{\bar{\bar{\mathbf{A}}}}^{++}_{\mathrm{md/mp}} & \bar{\bar{\bar{\mathbf{A}}}}^{+-}_{\mathrm{md/mp}} \\ \bar{\bar{\bar{\mathbf{A}}}}^{-+}_{\mathrm{md/mp}} & \bar{\bar{\bar{\mathbf{A}}}}^{--}_{\mathrm{md/mp}} \end{bmatrix} \tag{45}$$

The combined modal and multipolar preconditioning is obtained in this basis by neglecting all the non-diagonal elements but those in $\bar{\bar{\bar{\mathbf{A}}}}^{++}_{\mathrm{mod}}$ so that the combined preconditioning is:

$$\mathbf{Q}_{\mathrm{md/mp}} = \begin{bmatrix} \mathrm{diag}(\bar{\mathbf{A}}^{++}_{\mathrm{mod}}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \bar{\bar{\bar{\mathbf{A}}}}^{++}_{\mathrm{md/mp}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathrm{diag}(\bar{\bar{\bar{\mathbf{A}}}}^{--}_{\mathrm{md/mp}}) \end{bmatrix} \tag{46}$$

The cost of this preconditioning is related to solving a system for $\bar{\bar{\bar{\mathbf{A}}}}^{++}_{\mathrm{mod}}$ which is a system $N_{\mathrm{pan}}/2m$, i.e. $2m$ times smaller than that of the original one. In the limit of a very large cluster size ($2m = N_{\mathrm{pan}}$) the modal/multipolar preconditioning approaches the pure modal one, since the preconditioning is diagonal (in a somewhat confuse notation we refer to $2m$ as the size of the cluster so that a "modal/multipolar preconditioning with 64 panels/cluster" stands for a modal preconditioning combined with a multipolar preconditioning with 32 panels per cluster). For simplicity, we have considered so far that the geometry is composed of a single thin airfoil. In the general case, some other elements not necessarily thin as, for instance, a fuselage are present in the geometry. In that case the change of basis matrix is block diagonal, with the identity matrix for those elements not in the thin airfoil part, and the usual modal change of basis matrix for those which are on the thin airfoil. After this, the change of basis matrix for the multipolar preconditioning is constructed by clustering those elements in the thin airfoil, as well as those in the other (non-thin) parts. The extension to several thin airfoils is also simple.

13. Numerical examples

Firstly, we will show several small scales (approx. 200 panels). The interest of these examples is that we can compute the full eigenvalue distribution and condition number using standard LAPACK routines (embedded in the high level scientific programming language Octave). Secondly, convergence curves for examples with large number of panels ($> 12,000$) will be presented. We consider first a thin biconvex spherical lens as shown in figure 11 of relative thickness t. The expression for the surface of the lens is:

$$[|z| + R/\tan(\theta)]^2 + x^2 + y^2 = (R/\sin(\theta)^2 \tag{47}$$

where the angle θ is given by $t = 2\tan(\theta/2)$. In figure 12 we see the behavior of the condition number as a function of the relative thickness t for the modal/multipolar preconditioned ($\mathbf{Q}^{-1}\mathbf{A}$) and non-preconditioned ($\mathbf{A}$) matrices. Four values of thicknesses ranging from 25 % to 0.2 % have been considered. It is verified that the condition number for the non-preconditioned matrix is $O(1/\epsilon)$ for $\epsilon \to 0$, whereas it is almost independent of ϵ for the preconditioned matrix. Moreover, the condition number is nearly the same for all cluster sizes. Remember that the computational effort involved in the preconditioning is roughly inversely proportional to the cluster size. However, we will see that in practice the rate of convergence does depend on the cluster size. The mesh was composed of 192 triangular panels. In figure 13 we see the eigenvalue spectra for the lens for two thicknesses. The mesh was the same as in the previous paragraph, the modal/multipolar preconditioning has been used. Note that in all cases the eigenvalues are clearly separated in two branches for the non-preconditioned case, and each branch has $N_{\mathrm{pan}}/2$ eigenvalues. This is typical for problems where a symmetry "*separates*" branches of eigenvalues. As the thickness gets smaller the lower branch, i.e. that one with the lowest (in absolute value) eigenvalues, gets smaller too, and for very small thicknesses (as $t = 0.2\%$ in the figure) the lower branch is clustered near the origin, whereas the upper branch is clustered near $|\lambda| = 1$. In contrast, the spectra for the preconditioned matrices remains almost unaltered. In figure 14 we see the convergence history for the 5 % thick lens at 0, 5 and 90 degrees angles of attack, and cluster sizes n_c ranging from 2 to 64, where the Bi-Conjugate Gradient (BCG) [18 -19] algorithm was used. The curve labeled as "J" stands for the non-preconditioned problem. For incidence at 0 degrees the flow is symmetric, and then the skew-symmetric part of the equations is not excited, then the rate of convergence is poorly improved by the preconditioning. In the other cases, the improvement in the rate of convergence monotonically decreases with cluster size. For a typical case of incidence at 5 degrees we see that even with a large cluster of 64 panels the improvement in the rate of convergence is significant. Similar plots for thickness of 0.2 % are shown in figure 15. The improvements are more significant for thinner lenses, as expected.

Figures 16-18 show similar results but for a symmetric delta wing, with a wing section consisting of a symmetric Joukowski profile. Figure 19 represents curves of convergence for a delta wing with a typical section being a Joukowski profile, where 12,288 panels were used and the airfoil thickness was 8 %, see figure 20. In this case both Conjugate Gradient Squared (CGS) [20] and Bi-Conjugate Gradient (BCG) methods were used without and with a preconditioning of 32 panels per cluster. We can see that the improvement in the rate of convergence by the preconditioning is significant with both methods and the performance with CGS is better than with BCG. In all cases we show convergence history versus number of matrix-vector products (which we will call here on *"work units"*) and CPU time. A work unit involves computation of the interaction coefficients by columns and standard dot products or DAXPY (the LINPACK vector sum routine) operations. The expression for the interaction coefficients involve transcendental functions and is, by far, the most time consuming part of the work unit. To estimate the speedup obtained with the preconditioning we estimated the rates of convergence as the reciprocal of the mean slope of the curves with and without preconditioning (as the curves are often noisy, the choice of such a mean slope is somewhat arbitrary, the dashed lines in the figures represent the mean slope we adopted). In the case of the residual versus work units curves these rates are expressed in work units/order that is, the number of work units needed to reduce the residual by a factor of ten. In the case of the residual versus time, the corresponding unit is CPU hrs/order. Work units rates have the advantage that are independent of processor speed but do not take into account the overhead needed by the preconditioning. It includes changes to and from the modal basis and a back-substitution for the preconditioning matrix. For the 12,288 mesh with a 32 panels/cluster preconditioning the overhead represents a 6.9 % of the matrix-vector product operation but depends strongly on details of the implementation. For instance, on a vector processor the overhead tend to be smaller, since the preconditioning operations (mainly the back-substitution) is much more prone to vectorization than the computation of coefficients. On the other hand, it is common to evaluate the interaction coefficients for distant panels using far-field expansions. In this case the overhead tends to be higher. The speedup is defined as the quotient between the rates with and without preconditioning, and is a little higher for the work units rates than those based on CPU time:

$$\begin{aligned}\frac{\text{Speedup (work units)}}{\text{Speedup (CPU time)}} &= \frac{\text{CPU time for a work unit with prec.}}{\text{CPU time for a work without prec.}} \\ &= 1 + \frac{\text{preconditioning CPU time overhead}}{\text{CPU time for a work unit without prec.}} \\ &> 1\end{aligned} \tag{48}$$

Note that solution of the full system matrix would require 1.2 Gbytes of RAM, whereas the preconditioning matrix only requires 1.2 Mbytes. Indeed, this problem was ran on a DEC ALPHA/200 workstation of 233 Mhz where the core memory requirement was only of 5.2 Mbytes.

14. Conclusions

The modal/multipolar preconditioning successfully corrects the bad conditioning arising from thin airfoils. It has been shown that the condition number for the resulting linear system is $O(1/\epsilon)$ whereas for the preconditioned system it is almost constant. Numerical results for a biconvex lens and a symmetric airfoil are shown, and significants in condition number are obtained even for airfoils as thick as 25 %.

15. Acknowledgment

This work has received financial support from *Consejo Nacional de Investigaciones Científicas y Técnicas* (CONICET, Argentina) through grant BID 802/OC-AR PID Nr. 26, and from *Universidad Nacional del Litoral* (Argentina). We made extensive use of software distributed by the *Free Software Foundation / GNU-Project*: Linux ELF-OS, Octave, Fortran f2c compiler, Tgif and others.

16. Appendix :

17. $\bar{\mathbf{A}}_{-+}$ *block:* Consider for instance coefficient $\bar{A}_{11}$ which from (11.a) and (17) is:

$$\bar{A}_{11} = ½(A_{11} + A_{12} + A_{21} + A_{22}) \tag{49}$$

Recall that A_{ij} is the potential produced by a constant distribution of double layer density on the i-panel on the centroid of the j-panel. If we call ϕ the potential produced by a distribution of double layer density $\mu = 1$ on panel 1 and $\mu = 1$ on panel 2, then:

$$A_{i1} + A_{i2} = \phi(\mathbf{x}_i) \tag{50}$$

But, as we discussed previously, for $\epsilon \to 0$, the double layer densities tend to cancel each other and we have:

$$\phi \to \epsilon\bar{\phi} \tag{51}$$

where $\bar{\phi}$ is the potential produced by a constant *"quadruple layer"* density. Now:

$$\begin{aligned} \bar{A}_{11} &= {}^{1}\!/_{2}[\phi(\mathbf{x}_1) + \phi(\mathbf{x}_2)] \\ &\to {}^{1}\!/_{2}\epsilon\,[\bar{\phi}(\mathbf{x}_1) + \bar{\phi}(\mathbf{x}_2)] \\ &\to {}^{1}\!/_{2}\epsilon\,\bar{\phi}(\mathbf{x}_{12}) \end{aligned} \tag{52}$$

where $\mathbf{x}_{12}$ is the point lying in the middle of the segment joining $\mathbf{x}_1$ and $\mathbf{x}_2$. This is so for all the elements in the block and then (21) is obtained.

18. $\bar{\mathbf{A}}_{-+}$ ***block:***

Consider first a non-diagonal coefficient:

$$\begin{aligned} \bar{A}_{41} &= {}^{1}\!/_{2}(A_{31} + A_{32} - A_{41} - A_{42}) \\ &= {}^{1}\!/_{2}[\phi(\mathbf{x}_3) - \phi(\mathbf{x}_4)] \\ &\to {}^{1}\!/_{2}\epsilon\,[\bar{\phi}(\mathbf{x}_3) - \bar{\phi}(\mathbf{x}_4)] \\ &\to {}^{1}\!/_{2}\epsilon^2\,\frac{\partial\bar{\phi}}{\partial z}(\mathbf{x}_{34}) \end{aligned} \tag{53}$$

where z is a coordinate normal to the panels 3 and 4 and we have used the fact that $\bar{\phi}$ is well behaved at $\mathbf{x}_{3,4}$. This reasoning is valid for all the non-diagonal elements in $\bar{\mathbf{A}}_{-+}$, as $\bar{\mathbf{A}}_{32}$. For the diagonal elements:

$$\begin{aligned} \bar{\mathbf{A}}_{31} &= {}^{1}\!/_{2}(A_{11} + A_{12} - A_{21} - A_{22}) \\ &= {}^{1}\!/_{2}[\phi(\mathbf{x}_1) - \phi(\mathbf{x}_2)] \\ &= {}^{1}\!/_{2}\epsilon\,[\bar{\phi}(\mathbf{x}_1) - \bar{\phi}(\mathbf{x}_2)] + O(\epsilon^2) \end{aligned} \tag{54}$$

but now $\bar{\phi}$ has some degree of discontinuity across the panel and it can not be expanded in power series. However, as mentioned above, the *"quadruple layer density"* is a symmetric distribution of density about the center-plane and then $\bar{\phi}(\mathbf{x}_{12} + z\hat{\mathbf{n}})$ is even about $z = 0$ so that:

$$\begin{aligned} \bar{\mathbf{A}}_{31} &= {}^{1}\!/_{2}\epsilon\,[\bar{\phi}(\mathbf{x}_1) - \bar{\phi}(\mathbf{x}_2)] + O(\epsilon^2) \\ &= {}^{1}\!/_{2}\epsilon\,[\bar{\phi}(\mathbf{x}_{12} + {}^{1}\!/_{2}\epsilon\hat{\mathbf{n}}) - \bar{\phi}(\mathbf{x}_{12} - {}^{1}\!/_{2}\epsilon\hat{\mathbf{n}})] + O(\epsilon^2) \\ &= O(\epsilon^2) \end{aligned} \tag{55}$$

As all the elements (diagonal and non-diagonal) are $O(\epsilon^2)$, and then (22) is obtained.

19. $\bar{\mathbf{A}}_{+-}$ ***block:***

Consider the off-diagonal coefficients:

$$\begin{aligned} A_{23} &= 1/2(A_{31} - A_{32} + A_{41} - A_{42}) \\ &= \psi(\mathbf{x}_3) + \psi(\mathbf{x}_4) \end{aligned} \tag{56}$$

where ψ is one half the potential produced by a distribution of double layer density $\mu = 1$ on panel 1 and $\mu = -1$ on panel 2. In this case, the double layer densities tend to reinforce each other and:

$$\psi \to \bar{\psi} \quad \text{for} \quad \epsilon \to 0 \tag{57}$$

In addition, ψ is well behaved at $\mathbf{x}_{3,4}$ so that:

$$\begin{aligned} A_{23} &\to \bar{\psi}(\mathbf{x}_3) + \bar{\psi}(\mathbf{x}_4) \\ &\to 2\bar{\psi}(\mathbf{x}_{34}) \\ &= O(1) \end{aligned} \tag{58}$$

in contrast, the diagonal terms are of the form:

$$\begin{aligned} A_{13} &\to \bar{\psi}(\mathbf{x}_1) + \bar{\psi}(\mathbf{x}_2) \\ &= O(\epsilon) \end{aligned} \tag{59}$$

since $\bar{\psi}$ is odd across the panel. Then, (23) is obtained.

20. $\bar{\mathbf{A}}_{--}$ ***block:***

For the off diagonal terms:

$$\begin{aligned} \bar{\mathbf{A}}_{34} &= 1/2(A_{31} - A_{32} - A_{41} + A_{42}) \\ &= \psi(\mathbf{x}_3) - \psi(\mathbf{x}_4) \\ &\to \bar{\psi}(\mathbf{x}_3) - \bar{\psi}(\mathbf{x}_4) \\ &\to \epsilon \frac{\partial \bar{\psi}}{\partial z}(\mathbf{x}_{34}) \\ &= O(\epsilon) \end{aligned} \tag{60}$$

since $\bar{\psi}$ is well behaved on $\mathbf{x}_{3,4}$. On the other hand, for the diagonal terms:

$$\begin{aligned} \bar{\mathbf{A}}_{33} &= 1/2(A_{11} - A_{12} - A_{21} + A_{22}) \\ &= \psi(\mathbf{x}_1) - \psi(\mathbf{x}_2) \\ &\to \bar{\psi}(\mathbf{x}_1) - \bar{\psi}(\mathbf{x}_2) \\ &\to 2\bar{\psi}(\mathbf{x}_1) \\ &= O(1) \end{aligned} \tag{61}$$

since $\bar{\psi}$ is odd across the panel. Then, (24) is obtained.

21. References

[1] Katz J. and Plotkin A., *Low-Speed Aerodynamics, from Wing Theory to Panel Methods*, McGraw-Hill, Inc, 1991.

[2] D'Elía J., Storti M. and Idelsohn S., "A CVBEM formulation for multiple profiles and cascades", *Applied Mechanics Review* , Vol. 48, pp. S203-S210 (1995).

[3] D'Elía J., Storti M. and Idelsohn S., "A 3D Panel code for wave resistance calculations, part I: general formulation and discretization", *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, Vol. 13, No. 4, pp. 515-530 (1997).

[4] Storti M., D'Elía J. and Idelsohn S., "Algebraic Discrete Non-Local (DNL) absorbing boundary condition for the ship wave resistance problem", to appear in *Journal of Computational Physics* (1998).

[5] Morino L., Kuo C.C., "Subsonic potential aerodynamics for complex configurations: a general theory", *AIAA Journal*, Vol. 12, pp. 191-197 (1974).

[6] Morino L., (edt.) *Computational Methods in Potential Aerodynamics*, Springer-Verlag, 1985.

[7] Gennaretti M., Morino L., "A boundary element method for the potential, compressible aerodynamics of bodies in arbitrary motion", *AIAA Journal*, Vol. 96, pp. 15-19 (1992).

[8] Maître T., *Modelisation de l'ecouléménnt autour d'une hélice marine par la méthode du potentiel*, Ph. D. Tesis Institut National Polytechnique de Grenoble, 1988.

[9] Evans D.J. (edt.) *Preconditioning Methods, Theory and Applications*, Gordon and Breach Science Publishers, 1983.

[10] Prasad K.G., Kane J.H., Keyes D.E., "Iterative Solution techniques in boundary element analysis", *International Journal for Numerical Methods in Engineering*, Vol. 31, pp. 1511-1536 (1991).

[11] Hriberšek M., Škerget P., Mang H. "Preconditioned conjugate. grad. methods for boundary domain integral method", *Engineering Analysis with Boundary Elements* Vol. 12, pp. 111-118 (1993).

[12] Manzur W.J., Araújo F.C., Malaghini E.B., "Solution of BEM systems of equations via iterative techniques", *International Journal for Numerical Methods in Engineering*, Vol. 33, pp. 1823-1841 (1992).

[13] Hackbusch W. and Nowak Z.P., "On the fast matrix multiplication in the boundary element method by panel clustering", *Numerische Mathematik*, Vol. 54, pp. 463-491 (1989).

[14] Nabors K., Korsmayer F.T., Leighton F.T. and White J., "Preconditioned, adaptative, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory", *SIAM J. Sci. Comput.*, Vol. 15, No. 3, pp. 713-735 (1994).

[15] Vavasis, S.A., "Preconditioning for boundary integral equations", *SIAM J. Matrix Anal. Appl.*, Vol. 13, No 3, pp. 905-925 (1992).

[16] D'Elía J., *Numerical Methods for the Ship Wave Resistance Problem*, Ph. D. Thesis, 1997.

[17] Nachtigal N.M., Reddy S.C., Trefethen L.N., "How fast are non-symmetric matrix iterations ?", *SIAM J. Matrix Anal. Appl.*, Vol. 13, No. 3, pp. 778-795 (1992).

[18] Lanczos C., "Solution of systems of linear equations by minimized iteration", *J. Res. Nat. Bur. Standards*, Vol. 49, pp. 33-53 (1952).

[19] Fletcher R., "Conjugate gradient methods for indefinite systems", *Lecture Notes in Math.*, Vol. 506, pp. 73-89 (1976).

[20] Sonneveld P., "CGS, a fast Lanczos-type solver for nonsymmetric linear systems", *SIAM J. Sci. Statist. Comput.*, Vol. 10, pp. 36-52 (1989).

[21] Yan Y., "Sparse Preconditioned Iterative Methods for Dense Linear Systems", *SIAM J. Sci. Comput.*, Vol. 15, No. 5, pp. 1190-1200 (1994).

[22] Hestenes M.R. and Stiefel E., "Methods of conjugate gradients for solving linear systems", *J. Res. Nat. Bur. Standards*, Vol. 49, pp. 409-436 (1952).

[23] Paige C.C. and Saunders A., "LSQR: An algorithm for sparse linear equations and sparse least squares", *ACM Trans. Math. Software*, Vol. 8, pp. 43-71 (1982).

[24] Eisentat S.C., Helman H.C. and Schultz M.H. "Variational iterative methods for nonsymmetric systems of linear equations", *SIAM J. Numer. Anal.*, Vol. 20, pp. 345-357 (1983).

[25] Saad Y., Schultz, "Conjugate gradient-like algorithms for solving nonsymmetric linear systems", *Math. Comput.*, Vol. 44, pp. 417-424 (1985).

[26] Gutknecht M.H., " The unsymmetric Lanczos algorithms and their relations to Pade approximation, continued fractions and the qd algorithm", Preliminary

Proceedings, *Copper Mountain Conference on Iterative Methods*, 1990.

[27] Van der Vorst H.A., "Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems", *SIAM J. Sci. Statist. Comp.*, Vol. 13, No. 3, pp. 631-644 (1992).

[28] Freund R.W., "Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices", *SIAM J. Sci. Statist. Comp.*, Vol. 13, pp. 425-448 (1992).

[29] Saunders M.A., Simon H.D. and Yip E.L., "Two conjugate gradient-type methods for unsymmetric linear equations", *SIAM J. Numer. Anal.*, Vol. 25, pp. 927-940 (1988).

[30] Saad Y., "Krylov subspace methods on supercomputers", *SIAM J. Sci. Statist. Comp.*, Vol. 10, No. 6, pp. 1299-1232 (1989).

[31] Greebaum A.,"Estimating the attainable accuracy of recursively computed residual methods", *SIAM J. Matrix Anal. Appl.*, Vol. 18, No 3, pp. 535-551 (1997).

22. Notation and symbols

22.1. Notation

$\mathbf{A} \otimes \mathbf{B}$ = Kronecker product of two matrices defined by (29)

22.2. Greek letters

Γ = airfoil surface

Ω = exterior domain to the airfoil

ϵ = vanishing parameter scaling the airfoil thickness.

ϕ = velocity potential

μ = double layer density on the surface.

σ = single layer density

22.3. Latin letters

$\mathbf{A}$ = system matrix of interaction coefficients for double layer density distribution

$\bar{\mathbf{A}}$ = matrix $\mathbf{A}$ in the modal or multipolar basis

$\mathbf{C}$ = system matrix of interaction coefficients for single layer density distribution

H = length side of the square panels

$\mathbf{Q}$ = preconditioning matrix

$\mathbf{S}$ = change of basis matrix

$\mathrm{cond}(\mathbf{X})$ = condition of matrix $\mathbf{X}$

d = distance between panels

$\mathrm{diag}(\ )$ = 1) for matrices $\mathrm{diag}(\mathbf{X})$ is the diagonal part of $\mathbf{X}$, see equation (14), whereas 2) $\mathrm{diag}(a, b, c, \ldots)$ stands for a matrix whose diagonal entries are $a, b, c, \ldots$

g = potential produced by a single panel on a normal axis passing through its centroid

m = number of panels in a cluster

$\hat{\mathbf{n}}$ = normal unit vector

N_{pan} = number of panels

t = airfoil thickness

$\mathbf{u}_\infty$ = undisturbed velocity

Figure Captions

Figure 1: Crude realization of a thin airfoil with two panels.
Figure 2: Potential across the panel produced by constant double layer density
Figure 3: Symmetric and skew-symmetric distributions of charge
Figure 4: Two pair of panels on a symmetric airfoil.
Figure 5: Two pair of panels on a non-symmetric airfoil.
Figure 6: Effect of small distortions in the efficiency of the preconditioned.
Figure 7: Two pair of panels on a surface.
Figure 8: Local system at a panel cluster.
Figure 9: Two clusters of 4 panels each.
Figure 10: Distribution of double layer densities on the 4 panel cluster.
Figure 11: Geometrical description of the lens.
Figure 12: Condition number versus thickness for the lens with several cluster-sizes n_c.
Figure 13: Eigenvalue distribution for the lens with thicknesses 5 % and 0.2 %.
Figure 14: Convergence histories of residual for the 5 % thick lens at several angles of attack and cluster size n_c, where "J" stands for the non-preconditioned problem.
Figure 15: Same as figure 14 for the 0.2 % thick lens.
Figure 16: Condition number versus thickness t for several cluster sizes n_c.
Figure 17: Convergence histories of residual for the 5% thick symmetric delta wing at several angles of attack and cluster size n_c.
Figure 18: Same as figure 17 for the 0.2 % thick delta wing.
Figure 19: Convergence curves for symmetric delta wing 8 % thick, 12,288 panels, cluster size $n_c = 32$.
Figure 20: 12,288 panels mesh for the 8 % thick delta wing.

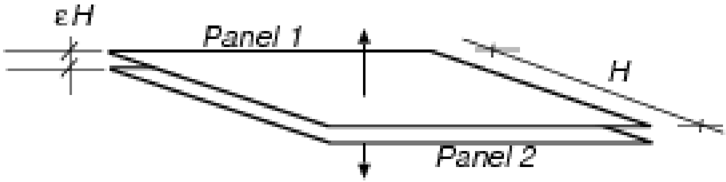


Figure 1. Iterative solution of panel method ... by J. D'Elía , M. Storti and S. Idelsohn. IJNMF

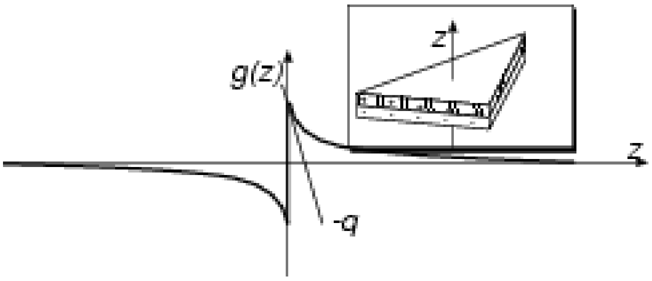


Figure 2. Iterative solution of panel method ... by J. D'Elía , M. Storti and S. Idelsohn. IJNMF

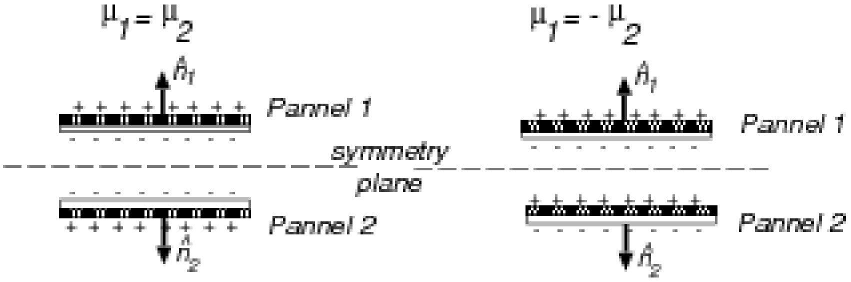


Figure 3. Iterative solution of panel method ... by J. D'Elía , M. Storti and S. Idelsohn. IJNMF

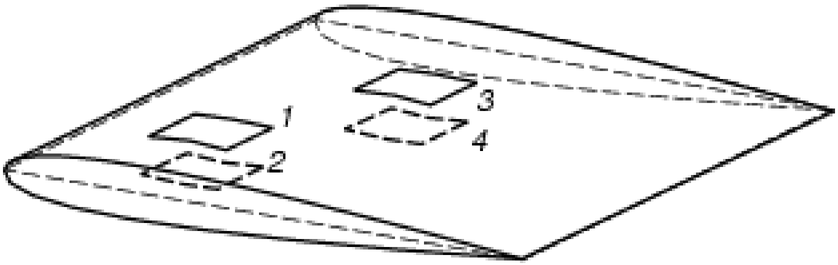
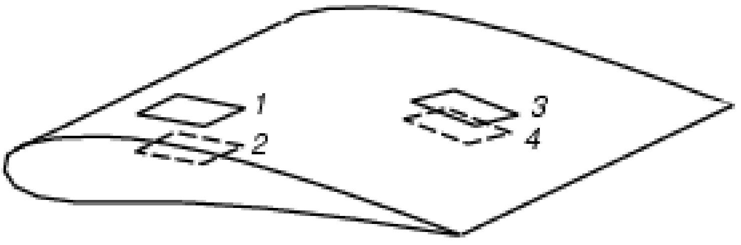
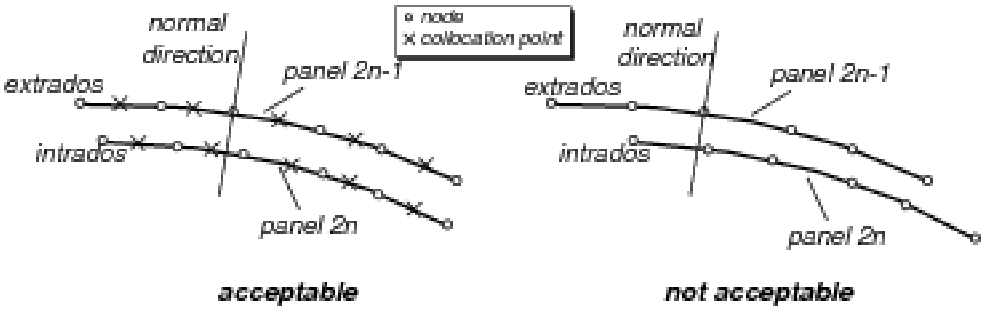
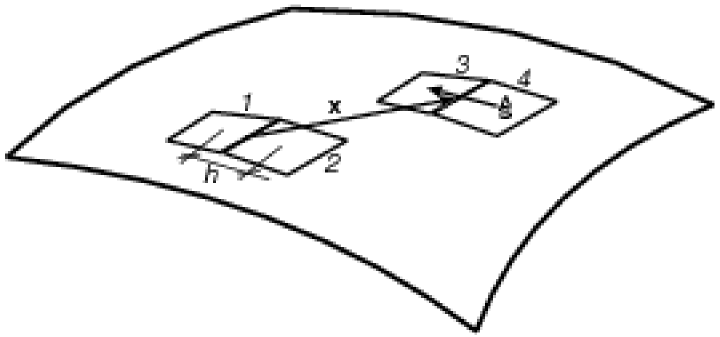


Figure 4. Iterative solution of panel method ... by J. D'Elía , M. Storti and S. Idelsohn. IJNMF







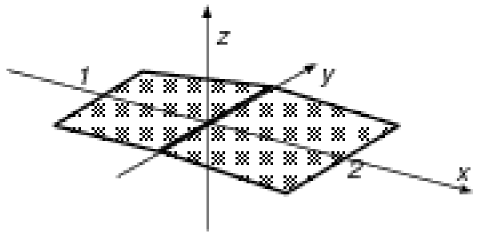
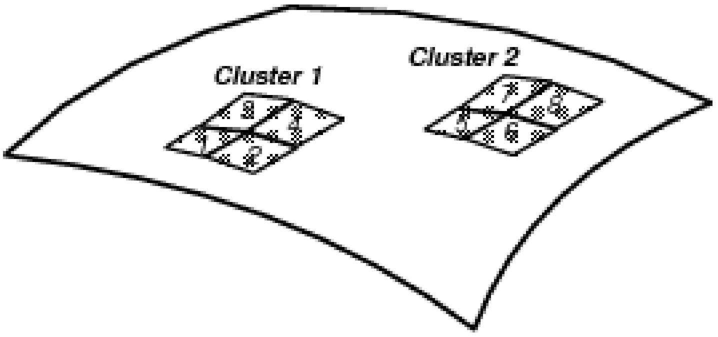


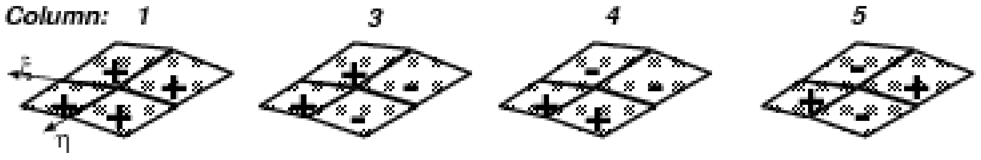
Figure 5. Iterative solution of panel method ... by J. D'Elía , M. Storti and S. Idelsohn. IJNMF

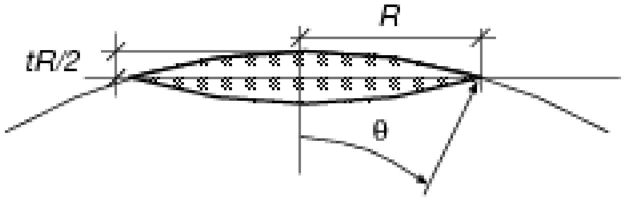
Figure 6. Iterative solution of panel method ... by J. D'Elía , M. Storti and S. Idelsohn. IJNMF

Figure 7. Iterative solution of panel method ... by J. D'Elía , M. Storti and S. Idelsohn. IJNMF

Figure 8. Iterative solution of panel method ... by J. D'Elía , M. Storti and S. Idelsohn. IJNMF







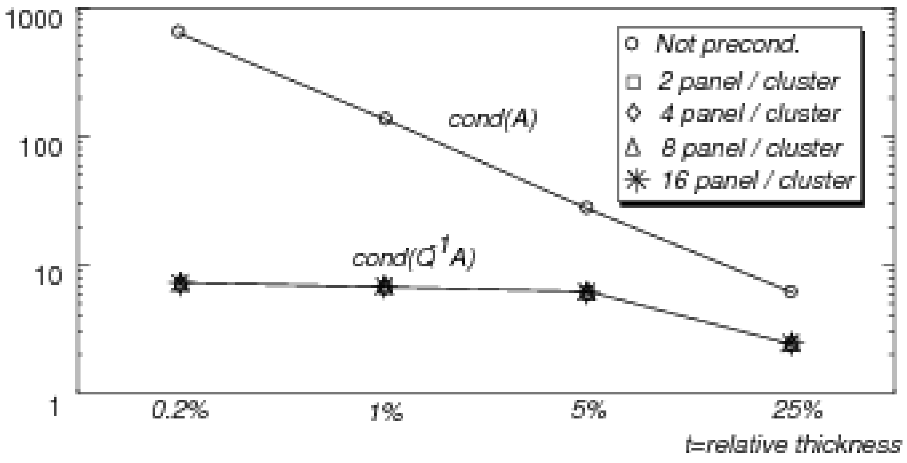
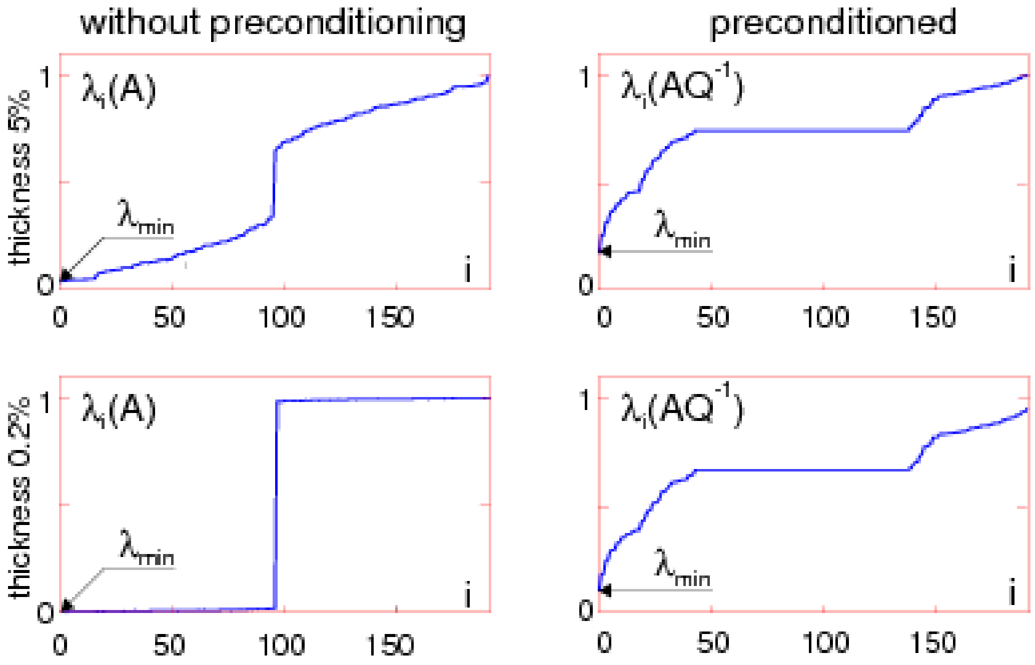


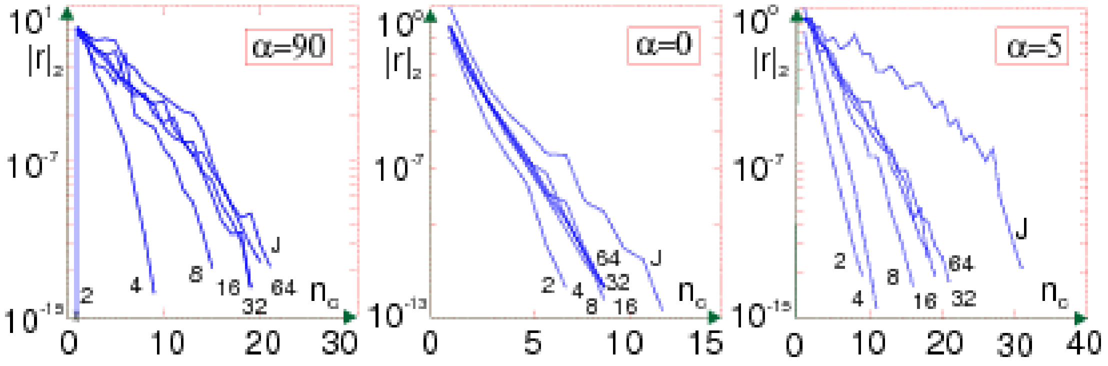
Figure 9. Iterative solution of panel method ... by J. D'Elía , M. Storti and S. Idelsohn. IJNMF

Figure 10. Iterative solution of panel method ... by J. D'Elía , M. Storti and S. Idelsohn. IJNMF

Figure 11. Iterative solution of panel method ... by J. D'Elía , M. Storti and S. Idelsohn. IJNMF

Figure 12. Iterative solution of panel method ... by J. D'Elía , M. Storti and S. Idelsohn. IJNMF





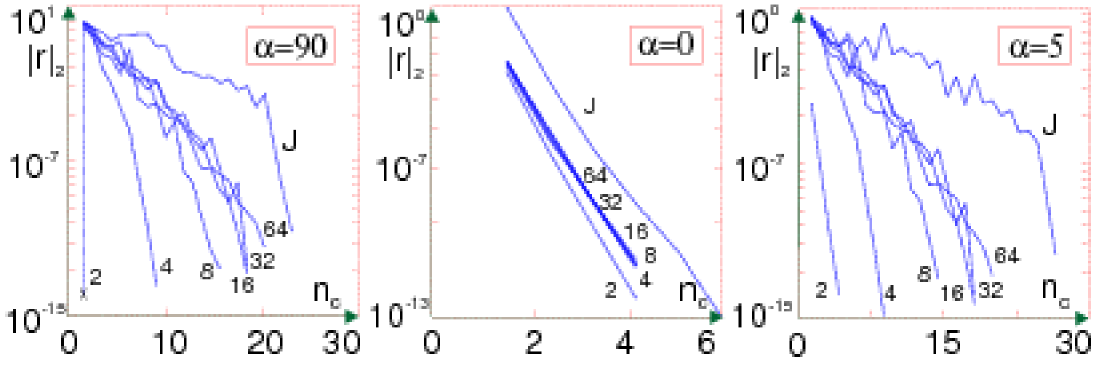
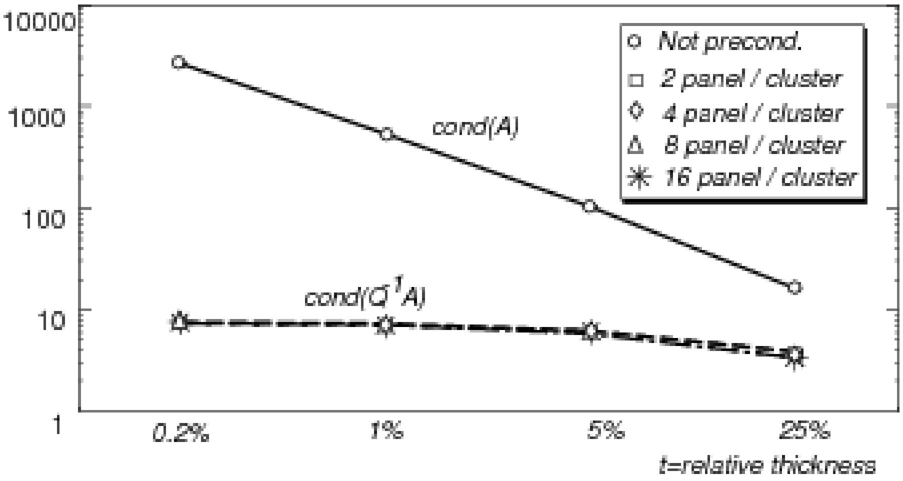
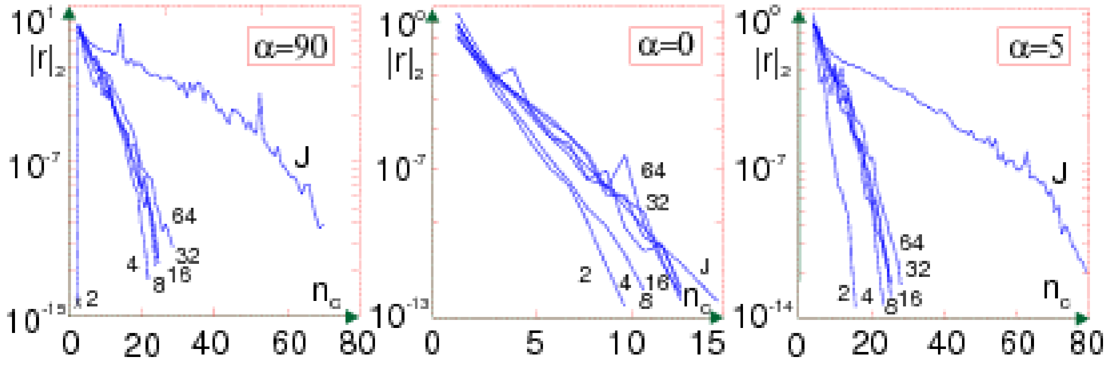


Figure 13. Iterative solution of panel method ... by J. D'Elía , M. Storti and S. Idelsohn. IJNMF

Figure 14. Iterative solution of panel method ... by J. D'Elía , M. Storti and S. Idelsohn. IJNMF

Figure 15. Iterative solution of panel method ... by J. D'Elía , M. Storti and S. Idelsohn. IJNMF





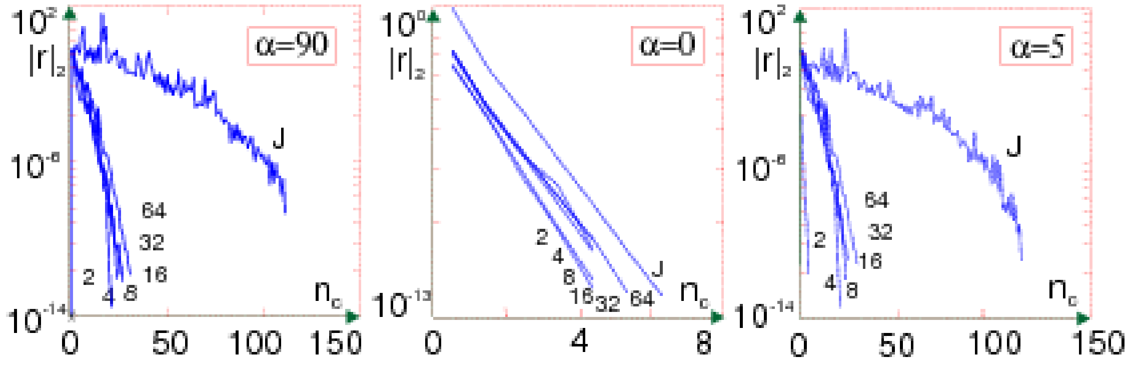


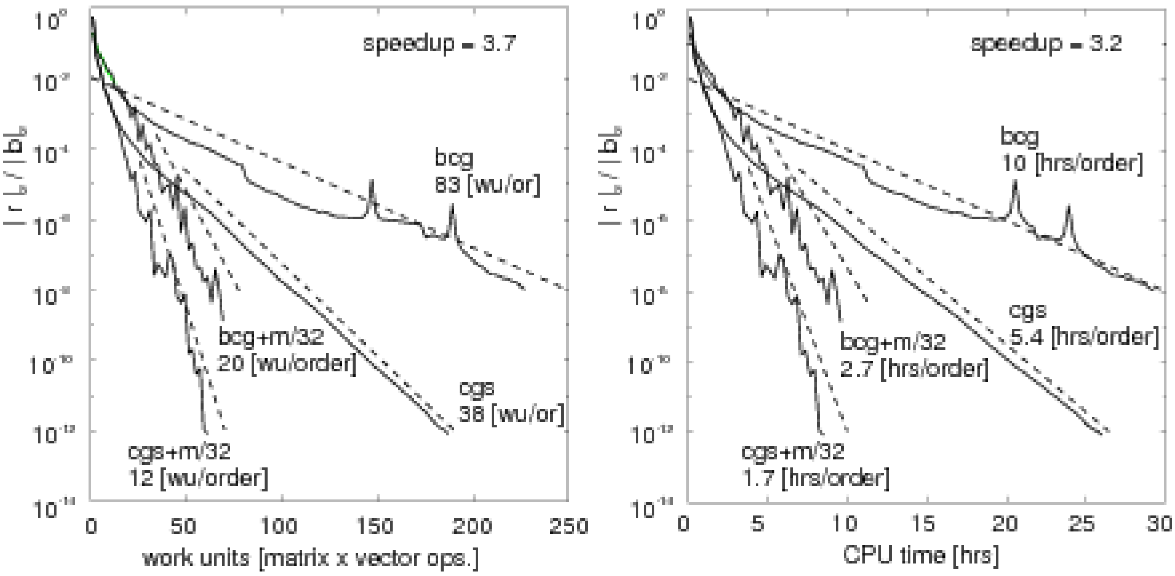
Figure 16. Iterative solution of panel method ... by J. D'Elía , M. Storti and S. Idelsohn. IJNMF

Figure 17. Iterative solution of panel method ... by J. D'Elía , M. Storti and S. Idelsohn. IJNMF

Figure 18. Iterative solution of panel method ... by J. D'Elía , M. Storti and S. Idelsohn. IJNMF

Figure 19. Iterative solution of panel method ... by J. D'Elía , M. Storti and S. Idelsohn. IJNMF

Figure 20. Iterative solution of panel method ... by J. D'Elía , M. Storti and S. Idelsohn. IJNMF



$| r |_{\circ} / | b |_{\circ}$
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
speedup = 3.7
bcg
83 [wu/or]
bcg+m/32
20 [wu/order]
cgs
38 [wu/or]
cgs+m/32
12 [wu/order]
0
50
100
150
200
250
work units [matrix x vector ops.]
speedup = 3.2
bcg
10 [hrs/order]
cgs
5.4 [hrs/order]
bcg+m/32
2.7 [hrs/order]
cgs+m/32
1.7 [hrs/order]
0
5
10
15
20
25
30
CPU time [hrs]