

Iterative Solution methods

Basic Iterative Solution methods

The ideas to use iterative methods for solving linear systems of equations go back to Gauss (1823), Liouville (1837) and Jacobi (1845).

Introduction:

Before considering iterative solution methods for **linear** systems of equations, we recall how do we solve **nonlinear** problems

Let

$$f(x) = 0$$

have to be solved and $f(x)$ is a nonlinear function in x .

The usual way to approach the problem is:

$$F(x) \equiv x - f(x).$$

If x^* is the solution of $f(x) = 0$, then x^* is a stationary point for

$$x = F(x). \tag{1}$$

Then we proceed with finding the stationary point for (1) and this is done **iteratively**, namely,


$$x^{(k+1)} = F(x^{(k)}), k = 0, 1, \dots, x^{(0)} \text{ given.}$$

Convergence of the fixed point iteration:

For any initial guess $x^{(0)}$, there exists a unique fixed point x^* for $F(x)$,
 $x^* = \lim_{k \rightarrow \infty} x^{(k)}$ **if and only if** F is a contracting mapping, i.e.

$$\|F(x) - F(y)\| \leq q\|x - y\|$$

for some $q \in (0, 1)$.



Fixed point for linear problems:

Let now $f(\mathbf{x}) \equiv A\mathbf{x} - \mathbf{b}$ be linear. We use the same framework:

$$\begin{aligned} F(\mathbf{x}) &= \mathbf{x} - (A\mathbf{x} - \mathbf{b}) \\ \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} - (A\mathbf{x}^{(k)} - \mathbf{b}) = \mathbf{x}^{(k)} + \mathbf{r}^{(k)} \end{aligned}$$

where $\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}$ is called the **residual** at iteration k .

In this way we obtain the simplest possible iterative scheme to solve

$$A\mathbf{x} = \mathbf{b},$$

namely,

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} - (A\mathbf{x}^{(k)} - \mathbf{b}), \quad k = 0, 1, \dots \\ \mathbf{x}^{(0)} &\text{ given.} \end{aligned}$$

Simple iteration

For many reasons the latter form of the simple iteration is replaced by

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \tau \mathbf{r}^{(k)}, \quad (2)$$

where τ is some properly chosen method parameter.

Relation (2) defines the so-called *stationary basic iterative method of first kind*.

Stationary iterative methods ...

If we permit τ to change from one iteration to the next, we get

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \tau_k \mathbf{r}^{(k)}, \quad (3)$$

which latter defines the so-called
non-stationary basic iterative method of first kind.

So far τ and τ_k are some scalars. Nothing prevents us to replace the method parameter by some matrix, however, if this would improve the convergence of the iterative method.

(cont)

Nothing prevents us to replace the method parameter by some matrix, however, if this would improve the convergence of the iterative method. Thus, we can consider

$$\begin{aligned}\mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + C^{-1}(\mathbf{b} - A\mathbf{x}^{(k)}) \\ \text{or} \\ \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + C^{-1}\mathbf{r}^{(k)},\end{aligned}\tag{4}$$

It is easy to see that we obtain (4) by replacing $A\mathbf{x} = \mathbf{b}$ with

$$C^{-1}A\mathbf{x} = C^{-1}\mathbf{b}$$

and use the simple iteration framework. In this case the iterative scheme takes the form

$$\begin{aligned}C\mathbf{d}^{(k)} &= \mathbf{r}^{(k)}, \\ \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \mathbf{d}^{(k)}\end{aligned}\tag{5}$$

The scheme (5) has in general a higher computational complexity than (3), since a solution of a system with the matrix C is required at each iteration.

Concerns:

C1 Does the iteration process converge to the solution, i.e. does $\mathbf{x}^{(k)} \rightarrow \mathbf{x}^*$?

C2 If 'yes', how fast does it converge?

The number of iterations it needed for the iterative method to converge with respect to some convergence criterion, is a function of the properties of A . For instance, $it = it(n)$, where n is the size of A . If it turns out that $it = O(n^2)$, we haven't gained anything compared to the direct solution methods.

The best one can hope for is to get $it \leq Const$, where $Const$ is independent of n . Since the the computational complexity of one iteration is in many cases proportional to n (for sparse matrices, for instance) then the complexity of the whole solution process will be

$$O(n).$$

C3 Is the method robust with respect to the method parameters (τ, τ_k) ?

Concerns (cont.):

C4 Is the method robust with respect to various problem parameters?

$$A = A(\rho, \nu, E, \dots)$$

C5 When we are using the scheme $C^{-1}A\mathbf{x} = C^{-1}\mathbf{b}$, it must be easy to solve systems with C .

C6 Is the method parallelizable?

Parallelization aspects become more and more important since n is XXL.

Concerns (cont.):

Suppose the method converges to the exact solution \mathbf{x}^* .

Then more questions arise:

C7 When do we stop the iterations?

→ We want $\|\mathbf{x}^* - \mathbf{x}^{(k)}\| \leq \varepsilon$ but \mathbf{x}^* is not known.

→ What about checking on $\mathbf{r}^{(k)}$?

→ Is it enough to have $\|\mathbf{r}^{(k)}\| \leq \tilde{\varepsilon}$?

Will the latter guarantee that $\|\mathbf{x}^* - \mathbf{x}^{(k)}\| \leq \varepsilon$?

Denote $\mathbf{e}^{(k)} = \mathbf{x}^* - \mathbf{x}^{(k)}$ (the error at iteration k). Then

$$\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)} = A(\mathbf{x}^* - \mathbf{x}^{(k)}) = A\mathbf{e}^{(k)}.$$

In other words $\mathbf{e}^{(k)} = A^{-1}\mathbf{r}^{(k)}$.

Scenario: Suppose $\|A^{-1}\| = 10^8$ and $\tilde{\varepsilon} = 10^{-4}$. Then

$$\|\mathbf{e}^{(k)}\| \leq \|A^{-1}\| \|\mathbf{r}^{(k)}\| \leq 10^4, \quad \text{which is not very exiting.}$$

Example: Discrete Laplace Δ_h^5 :

$$\|A^{-1}\| \approx \lambda_{min} = \frac{1}{2}(\pi h)^2 \approx 10^4 \text{ for } h = 10^{-2}.$$

Concerns (cont.):

- C8 How do we measure (estimate) the convergence rate?
 - C9 How do we find good method parameters (τ, τ_k, C) , which will speed up the convergence?
-

Stopping tests:

In practice, most used stopping tests are:

(S1) $\|\mathbf{r}^{(k)}\| \leq \varepsilon$, residual based, absolute

Stopping tests:

In practice, most used stopping tests are:

(S1) $\|\mathbf{r}^{(k)}\| \leq \varepsilon$, residual based, absolute

(S2) $\|\mathbf{r}^{(k)}\| \leq \varepsilon \|\mathbf{r}^{(0)}\|$, residual based, relative

Stopping tests:

In practice, most used stopping tests are:

(S1) $\|\mathbf{r}^{(k)}\| \leq \varepsilon$, residual based, absolute

(S2) $\|\mathbf{r}^{(k)}\| \leq \varepsilon \|\mathbf{r}^{(0)}\|$, residual based, relative

(S3) $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\| \leq \varepsilon$

Stopping tests:

In practice, most used stopping tests are:

(S1) $\|\mathbf{r}^{(k)}\| \leq \varepsilon$, residual based, absolute

(S2) $\|\mathbf{r}^{(k)}\| \leq \varepsilon \|\mathbf{r}^{(0)}\|$, residual based, relative

(S3) $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\| \leq \varepsilon$

(S4) $\|\mathbf{x}^* - \mathbf{x}^{(k)}\| \leq \varepsilon_0 \|\mathbf{x}^* - \mathbf{x}^{(0)}\|$.

If the latter is wanted, then we must check on (S3) and choose ε such

that $\varepsilon \leq \frac{\|B\|}{1-\|B\|} \varepsilon_0 \|\mathbf{x}^* - \mathbf{x}^{(0)}\|$.

Either estimate of $\|A^{-1}\|$ or of $\|B = C^{-1}R\|$ is required.

Concerns (cont.):

C9 How do we find good method parameters (τ, τ_k, C) , which will speed up the convergence?

We consider [C9].

Intuitively, C has to do something with A .

Note that if $C = A$, then $C^{-1} = A^{-1}$ and we will get convergence in one step!

However, the computational effort to construct A^{-1} is higher than to use a direct solution method.

We try the following choice. Consider the following so-called **splitting** of A ,

$$A = C - R,$$

where C is nonsingular and R can be seen as an error matrix.

The matrix $B = C^{-1}R$ is referred to as the **iteration matrix**.

Equivalent formulation using the splitting:

Using the splitting $A = C - R$ we obtain the following equivalent form of the iterative procedure:

$$\begin{aligned} A = C - R &\longrightarrow R = C - A \\ \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + C^{-1}(\mathbf{b} - A\mathbf{x}^{(k)}) \\ &= \mathbf{x}^{(k)} + C^{-1}\mathbf{b} - C^{-1}(C - R)\mathbf{x}^{(k)} \\ &= C^{-1}\mathbf{b} + C^{-1}R\mathbf{x}^{(k)} \end{aligned}$$

$$C\mathbf{x}^{(k+1)} = R\mathbf{x}^{(k)} + \mathbf{b} \tag{6}$$

The matrix C is called a **preconditioner** to A . Its general purpose is to improve the properties of A in order to achieve a better (faster) convergence of the method.

Choices of the matrix C

Choice 'J'

Let $A = D - L - U$, where D is diagonal, U is strictly upper triangular and L is strictly lower triangular.

Let $C \equiv D$, $R = L + U$. The iterative scheme is known as **Jacobi** iteration:

$$D\mathbf{x}^{(k+1)} = (L + U)\mathbf{x}^{(k)} + \mathbf{b}$$

$$\text{Entry-wise } x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{i \neq j} a_{ij} x_j \right) ..$$

For the method to converge: $B = D^{-1}(L + U)$

$$\rho(B) \leq \|D^{-1}(L + U)\|_{\infty} = \max_{1 \leq i \leq n} \sum_{\substack{j=1 \\ j \neq i}}^n \left| \frac{a_{ij}}{a_{ii}} \right|$$

We want $\rho(B) < 1$. One class of matrices, for which Jacobi method converges is when A is strictly diagonally dominant.

Choices of the matrix C

- Choice GS-B Choose $C \equiv D - U$, $R = L$

Backward Gauss-Seidel

$$(D - U)\mathbf{x}^{(k+1)} = L\mathbf{x}^{(k)} + \mathbf{b}$$

Choices of the matrix C

- Choice GS-B Choose $C \equiv D - U$, $R = L$

Backward Gauss-Seidel $(D - U)\mathbf{x}^{(k+1)} = L\mathbf{x}^{(k)} + \mathbf{b}$

- Choice GS-F Choose $C \equiv D - L$, $R = U$

Forward Gauss-Seidel $(D - L)\mathbf{x}^{(k+1)} = U\mathbf{x}^{(k)} + \mathbf{b}$

Choices of the matrix C

- Choice GS-B Choose $C \equiv D - U, R = L$

Backward Gauss-Seidel $(D - U)\mathbf{x}^{(k+1)} = L\mathbf{x}^{(k)} + \mathbf{b}$

- Choice GS-F Choose $C \equiv D - L, R = U$

Forward Gauss-Seidel $(D - L)\mathbf{x}^{(k+1)} = U\mathbf{x}^{(k)} + \mathbf{b}$

- G-S is convergent for s.p.d. matrices.

Choices of the matrix C

- Choice GS-B Choose $C \equiv D - U, R = L$

Backward Gauss-Seidel $(D - U)\mathbf{x}^{(k+1)} = L\mathbf{x}^{(k)} + \mathbf{b}$

- Choice GS-F Choose $C \equiv D - L, R = U$

Forward Gauss-Seidel $(D - L)\mathbf{x}^{(k+1)} = U\mathbf{x}^{(k)} + \mathbf{b}$

- G-S is convergent for s.p.d. matrices.

- make it more fancy: $A = D - L - U$. Then

$$\begin{aligned}\omega A &= \omega D - \omega L - \omega L + D - D \leftarrow \text{overrelaxation} \\ &= (D - \omega L) - (\omega U + (1 - \omega)D)\end{aligned}$$

Choose $C \equiv D - \omega L, R = \omega U + (1 - \omega)D$:

SOR $(D - \omega L)\mathbf{x}^{(k+1)} = [\omega U + (1 - \omega)D]\mathbf{x}^{(k)} + \omega \mathbf{b}$

SOR - back to 1940

One can see SOR as a generalization of G-S ($\omega = 1$). Rewrite

$$(D - \omega L)\mathbf{x}^{(k+1)} = [\omega U + (1 - \omega)D]\mathbf{x}^{(k)} + \omega \mathbf{b}$$

as $(\frac{1}{\omega}D - L)\mathbf{x}^{(k+1)} = [(\frac{1}{\omega} - 1)D + U]\mathbf{x}^{(k)} + \mathbf{b}$

For the iteration matrix $B_\omega = (\frac{1}{\omega}D - L)^{-1} [(\frac{1}{\omega} - 1)D + U]$

One can show that $\rho(B_\omega) < 1$ for $0 < \omega < 2$. Furthermore, there is an optimal value of ω , for which $\rho(B_\omega)$ is minimized:

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho(\hat{B})^2}}, \quad \hat{B} = I - D_A^{-1}A.$$

Splittings of A

Let $A, C, R \in \mathbb{R}^{n \times n}$ and consider $A = C - R$. A splitting of A is called

- **regular** if C is monotone and $R \geq 0$ (elementwise)
- **weak regular** if C is monotone and $C^{-1}R \geq 0$
- **nonnegative** if C^{-1} exists and $C^{-1}R \geq 0$
- **convergent** if $\rho(C^{-1}R) < 1$.

Recall: A matrix is called *monotone* if $A\mathbf{x} > 0$ implies $\mathbf{x} > 0$.

Theorem: A - monotone $\Leftrightarrow A^{-1} \geq 0$.

- SSOR- Symmetric Successive Overrelaxation
- AOR - Accelerated Overrelaxation

$$(D + \sigma L)\mathbf{x}^{(k+1)} = [(1 - \omega)D - (\omega - \sigma)L - \omega U]\mathbf{x}^{(k)} + \omega \mathbf{b}$$

- GAOR Generalized AOR - for saddle point systems
- ...

Let A be symmetric matrix.

$$\mathbf{x}_0 \text{ given, } \quad \mathbf{x}_1 = \mathbf{x}_0 + \frac{1}{2}\beta_0\mathbf{r}_0$$

For $k = 0, 1, \dots$ until convergence

$$\mathbf{x}_{k+1} = \alpha_k\mathbf{x}_k + (1 - \alpha_k)\mathbf{x}_{k-1} + \beta_k\mathbf{r}_k.$$

$$\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k.$$

$$\alpha_k = \frac{a+b}{2}\beta_k, \quad \frac{1}{\beta_k} = \frac{a+b}{2} - \left(\frac{b-a}{4}\right)^2 \beta_{k-1}, \quad \beta_0 = \frac{4}{a+b}.$$

Note that $\alpha_k > 1, k \geq 1$.

Modifications for nonsymmetric matrices exist.

The preconditioned modified Hermitian Skew-Symmetric (PMHSS) method

Consider the complex system

$$Cz = h,$$

where $C = A + iB$, $z = \mathbf{x} + iy$ and $h = \mathbf{f} + ig$. Thus,
 $(A + iB)(\mathbf{x} + iy) = \mathbf{f} + ig$, where A, B are real matrices, $\mathbf{x}, \mathbf{y}, \mathbf{f}, \mathbf{g}$ are real
vectors and $i = \sqrt{-1}$ is the imaginary unit. Then

$$\begin{aligned} A\mathbf{x} - B\mathbf{y} &= \mathbf{f} \\ B\mathbf{x} + A\mathbf{y} &= \mathbf{g} \end{aligned}$$

This system can be rewritten in a matrix form

$$\begin{bmatrix} A & -B \\ B & A \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}.$$

Let A, B be symmetric and B - positive definite. We want to solve

$$\begin{bmatrix} A & -B \\ B & A \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}.$$

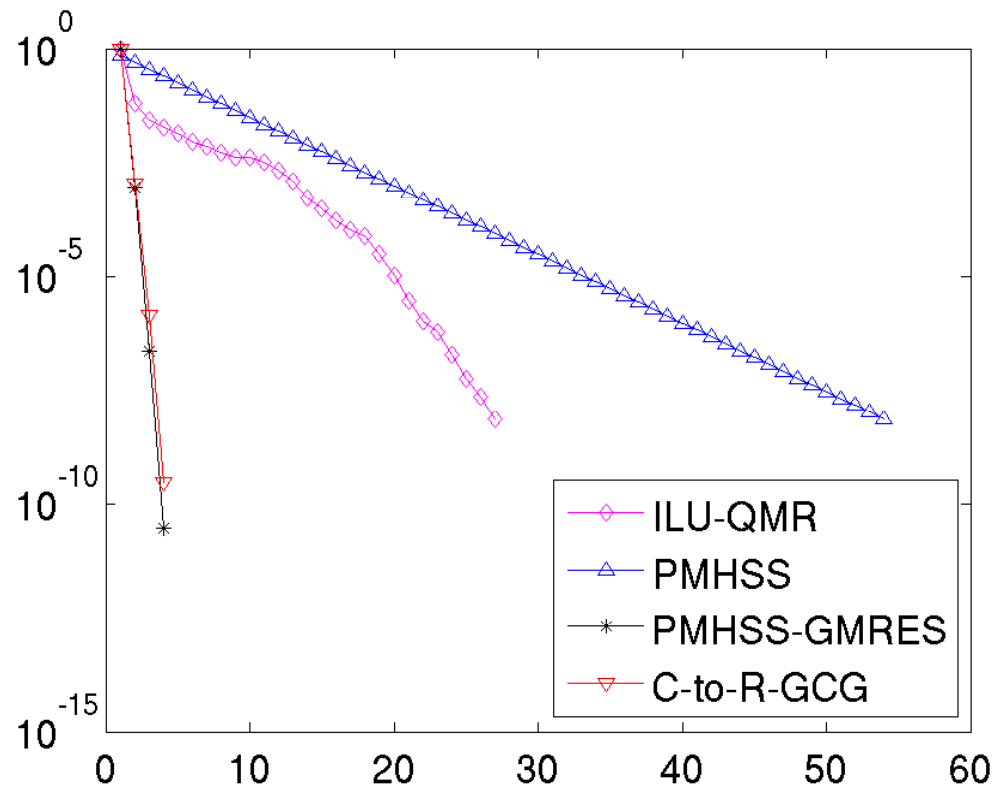
$$\begin{aligned} (\alpha V + A)\mathbf{x}^{k+1/2} &= (\alpha V + iB)\mathbf{x}^k - i\mathbf{b} \\ (\alpha V + B)\mathbf{x}^{k+1} &= (\alpha V - iA)\mathbf{x}^{k+1/2} + \mathbf{b} \end{aligned}$$

Let $\alpha = 1, V = B$. Then the algorithm becomes:

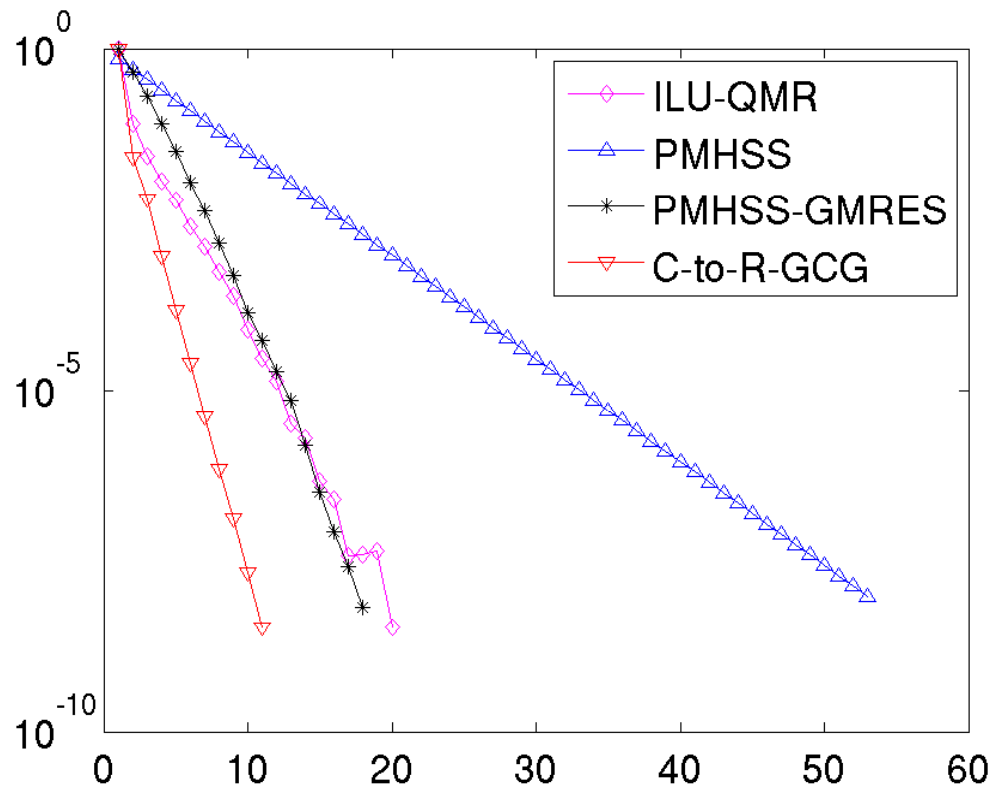
$$\begin{aligned} (A + B)\mathbf{z}^k &= \mathbf{q}^k \\ \mathbf{x}^k &= 0.5 * (1 - i)\mathbf{z}^k, \end{aligned}$$

where \mathbf{q}^k is the current residual in the iterative solution method.

(G. Golub, Z.-Z. Bai, M. Benzi and others.)



Convergence comparisons, 2D FEM, problem size 16641, $\omega = 0.01$



Convergence comparisons, 2D FEM, problem size 16641, $\omega = 1$

Problem size	Direct time	ILU-QMR		C -to- R -GCGMR		PMHSS-GMRES	
		iter	time err.	iter	time err.	iter	time err.
$\omega = 0.01$							
16641	0.2652	26	1.8462 2.9051e-5	3(6)	0.22638 6.7551e-5	3	0.1084 6.5565e-8
66049	1.5852	52	13.311 1.2776e-4	3(6)	0.88911 1.8912e-4	3	0.5652 1.7679e-7
263169	9.735	103	98.19 1.4075e-3	3(6)	4.0115 6.1502e-4	3	2.5433 8.5698e-7
$\omega = 100$							
16641	0.2918	19	1.6017 6.8632e-6	10(7)	0.6094 2.8814e-5	17	0.6866 3.6833e-6
66049	1.6299	36	10.675 8.0457e-5	10(7)	2.6054 1.3499e-4	17	3.1166 1.6439e-5
263169	9.9345	70	73.228 8.2749e-4	9(8)	11.803 3.5128e-3	17	15.154 6.7764e-5

2D, FEM, M -mass matrix, $C = L + i\omega M$; inner solver AGMG

Problem size	Direct time	ILU-QMR		C -to- R -GCGMR		PMHSS-GMRES	
		iter	time err.	iter	time err.	iter	time err.
$\omega = 0.01$							
4913	0.5755	69	0.2761 1.3967e-4	3(4)	0.2133 8.1647e-05	3	0.1226 7.5096e-07
35937	34.608	138	4.522 3.4811e-4	3(5)	1.2087 1.2793e-3	3	4.1468 4.7588e-06
274625	–	278	87.563	3(6)	12.982	–	–
$\omega = 100$							
4913	0.5374	42	0.1818 6.0743e-06	11(3)	0.2125 4.3167e-06	18	0.6515 7.5454e-06
35937	33.129	78	2.5282 8.9689e-05	10(5)	2.3168 2.5178e-4	19	20.839 1.5175e-05
274625	–	149	46.315	10(6)	30.578	–	–

3D, FEM, M -mass matrix, $C = L + i\omega M$; inner solver AGMG