Paper 70

Proceedings of the Sixth International Conference
on Engineering Computational Technology,
M. Papadrakakis and B.H.V. Topping, (Editors),
Civil-Comp Press, Stirlingshire, Scotland

Incomplete Factorization for Preconditioning Shifted Linear Systems Arising in Wind Modelling

E. Flórez, M.D. García, E. Rodríguez-Jiménez
H. Sarmiento, A. Suárez and G. Montero
University Institute for Intelligent Systems and
Numerical Applications in Engineering
University of Las Palmas de Gran Canaria, Spain

Abstract

The efficiency of a mass-consistent model for wind field adjustment depends on the stability parameter α. Each simulation with the wind model leads to the resolution of a linear system of equations, the matrix of which depends on a function $\varepsilon(\alpha)$, i.e., $(M + \varepsilon N)\, x_\varepsilon = b_\varepsilon$, where M and N are constant, symmetric and positive definite. Since each value of $\varepsilon(\alpha)$ yields a different linear system, we have to solve a set of them. So we could either construct a different preconditioner for each of them and improve the convergence of PCG at the expense of a high computational cost. On the other hand, a single preconditioner constructed with the first value of ε may be used for all the systems. Here, an intermediate solution is proposed by using a preconditioner, which is constructed once at first and updated for each ε at a low computational cost based on incomplete Cholesky factorization of A_ε.

Keywords: incomplete factorization, shifted linear systems, preconditioning, conjugate gradient, wind modelling, mass consistent models, genetic algorithms.

1 Introduction

The application of discretization techniques to problems defined by partial differential equations which model physical phenomena, sometimes leads to linear systems of equations of which matrix is given as a function of a parameter. This is specially true in the numerical simulation of wind fields with mass consistent models [1, 2].

In general, these problems are defined over regions with complex terrain, therefore a suitable discretization of the studied zone will be necessary. Here, we have used a technique for constructing tetrahedral meshes which are adapted to the terrain orography and have a higher density of nodes near the terrain surface [3]. Moreover some regions may need an additional refinement due, for example, to more accurate approximation in those zones. On the other hand, the combination of the model with a Gaussian plume approach make necessary the refinement along the trajectory of the plume. So, in general, we are going to work with meshes including elements of very different size. This fact may affect the conditioning of the linear system of equations that arises in this problem from this type of discretization, i.e.,

$$A_\varepsilon x_\varepsilon = b_\varepsilon \tag{1}$$

Thus a suitable preconditioning technique should be applied for an efficient conjugate gradient iteration. We are interested on the preconditioning of (1) for the case

$$A_\varepsilon = M + \varepsilon N \tag{2}$$

where matrices M and N stay constant along the process for a given discretization level. Such linear systems appear in each run of the model. There exist two different applications where a sequence of linear systems like (1) must be solved.

On one hand, we have applied genetic algorithms for the estimation of ε for each given set of station measures [2, 4]. Genetic algorithms (GAs) are optimisation tools based on the natural evolution mechanism. They produce successive trials that have an increasing probability to obtain a global optimum. The most important aspects of GAs are the construction of an initial population, the evaluation of each individual in the fitness function, the selection of the parents of the next generation, the crossover of those parents to create the children, and the mutation to increase diversity. In our experiments, initial population has been randomly generated and we use *iteration limit exceeded* as stopping criterion. The selection phase allocates an intermediate population on the basis of the evaluation of the fitness function. This evaluation, which is in general related to the difference between the observed and computed wind at the stations, leads us to solve a complete wind simulation for each individual of the population. In other words, we have to solve a linear system of equations like (1) for each value of ε.

On the other hand, for a given sequence of the parameter ε related to each set of measures along a simulation episode, a set of linear systems must be solved for each ε.

Two extreme strategies for preconditioning such linear systems may be applied. So we could either construct a different preconditioner for each one of them and improve the convergence of PCG at the expense of a high computational cost related to the construction of each preconditioner, or use a single preconditioner constructed with a given value of ε for all the systems. In this latter case the convergence will be getting worse as the value of the parameter moves away from the initial value.

Benzi [5] and Meurant [6] propose an intermediate solution by using two type of preconditioners, respectively, which are constructed once at first and updated for each ε at a low computational cost. These strategies also lead to an intermediate rate of convergence between the above extreme options. Benzi develops the study factorised approximate inverses using the SAINV algorithm [7] for the special case of $A_\varepsilon = M+\varepsilon I$, with I being the unit matrix. However Meurant does it for $A_\varepsilon = M+\varepsilon D$

with D being a diagonal matrix, constructing the preconditioner from an incomplete Cholesky factorization of M. In this work, we generalize Meurant's algorithm to the case of $A_\varepsilon = M + \varepsilon N$, with M and N SPD matrices.

In section 2, the mass consistent model is presented. It generates a velocity field for an incompressible fluid which adjusts to an initial one obtained from experimental measures and physical considerations. The construction of the initial field may be found in [2] and it is not directly involved in the updating of matrix A_ε. We remark the study of the stability parameter α of the wind model which is directly related with ε.

The construction of the preconditioner is set in terms of an incomplete Cholesky factorization for the matrix $A_\varepsilon = M + \varepsilon N$, with the corresponding simplifications that allows its undating at a reasonable cost is described in section 3.

Section 4 is devoted to illustrate the performance of this preconditioner in same numerical experiments. Finally, our conclusions are presented in section 5.

2 Wind model

This model [1] is based on the continuity equation for an incompressible flow where the air density is constant in the domain Ω and *no-flow-through* conditions on Γ_b (terrain and top) are considered

$$\vec{\nabla} \cdot \vec{u} = 0 \qquad \text{in } \Omega \tag{3}$$
$$\vec{n} \cdot \vec{u} = 0 \qquad \text{on } \Gamma_b \tag{4}$$

The problem is formulated as a least-square approach in Ω, with $\vec{u}(\widetilde{u}, \widetilde{v}, \widetilde{w})$ to be adjusted

$$E(\vec{u}) = \int_\Omega \left[\alpha_1^2 \left((\widetilde{u} - u_0)^2 + (\widetilde{v} - v_0)^2 \right) + \alpha_2^2 (\widetilde{w} - w_0)^2 \right] \, d\Omega \tag{5}$$

where the interpolated wind $\vec{v}_0 = (u_0, v_0, w_0)$ is obtained from experimental measurements and physical considerations, and α_1, α_2 are the Gauss precision moduli.

In practice, we use the so called stability parameter of the wind model,

$$\alpha = \frac{\alpha_1}{\alpha_2} \tag{6}$$

since the minimum of the functional given by (5) is the same if we divide it by α_2^2. So, if $\alpha >> 1$ flow adjustment in the vertical direction predominates. However if $\alpha << 1$ flow adjustment occurs primarily in the horizontal plane. Thus, the selection of α allows the air to go over a terrain barrier or around it, respectively. Moreover, the behaviour of mass consistent models in many numerical experiments has shown that they are very sensitive to the value chosen for α. In [4, 2] a brief discussion about the selection of α by several authors is presented.

Solving (5) constrained by (3) and (4) is equivalent to find a saddle point $(\vec{v}, \phi)$ of the Lagrangian

$$E(\vec{v}) = \min_{\vec{u} \in K} \left[E(\vec{u}) + \int_{\Omega} \phi \vec{\nabla} \cdot \vec{u} \, d\Omega \right] \tag{7}$$

being $\vec{v} = (u, v, w)$, ϕ the Lagrange multiplier and K the set of admissible functions. The Lagrange multipliers technique is used to minimise the problem (7), whose minimum comes to form the Euler-Lagrange equations

$$u = u_0 + \tfrac{1}{2\alpha_1^2} \frac{\partial \phi}{\partial x}, \quad v = v_0 + \tfrac{1}{2\alpha_1^2} \frac{\partial \phi}{\partial y}, \quad w = w_0 + \tfrac{1}{2\alpha_2^2} \frac{\partial \phi}{\partial z} \tag{8}$$

Since α_1 and α_2 are constant in Ω, the variational approach results in an elliptic problem substituting (8) in (3)

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \alpha^2 \frac{\partial^2 \phi}{\partial z^2} = -2\alpha_1^2 \left(\frac{\partial u_0}{\partial x} + \frac{\partial v_0}{\partial y} + \frac{\partial w_0}{\partial z} \right) \ \text{ in } \ \Omega \tag{9}$$

We consider Dirichlet condition for *flow-through* boundaries and Neumann condition for terrain and top

$$\phi = 0 \ \text{ on } \ \Gamma_a \tag{10}$$

$$\vec{n} \cdot T \, \vec{\nabla} \mu = -\vec{n} \cdot \vec{v}_0 \quad \text{on } \ \Gamma_b \tag{11}$$

with $T = diag\left[\frac{1}{2\alpha_1^2}, \frac{1}{2\alpha_1^2}, \frac{1}{2\alpha_2^2}\right]$. The problem given by (9)-(11), is solved using tetrahedral finite elements (see [4, 2]) which leads to a set of 4×4 elemental matrices and 4×1 elemental vectors related to element Ω_e, with $\hat{\psi}_i$ being the form function of the i-th node, $i = 1, 2, 3, 4$, defined in the reference element $\hat{\Omega}_e$ and $|\mathbf{J}|$ the Jacobian of the transformation from Ω_e to $\hat{\Omega}_e$,

$$\begin{aligned} \{\mathbf{A}^e\}_{ij} &= \int_{\hat{\Omega}_e} \Bigg\{ (\frac{\partial \hat{\psi}_i}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial \hat{\psi}_i}{\partial \eta}\frac{\partial \eta}{\partial x} + \frac{\partial \hat{\psi}_i}{\partial \varphi}\frac{\partial \varphi}{\partial x})(\frac{\partial \hat{\psi}_j}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial \hat{\psi}_j}{\partial \eta}\frac{\partial \eta}{\partial x} + \frac{\partial \hat{\psi}_j}{\partial \varphi}\frac{\partial \varphi}{\partial x}) \\ &+ (\frac{\partial \hat{\psi}_i}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial \hat{\psi}_i}{\partial \eta}\frac{\partial \eta}{\partial y} + \frac{\partial \hat{\psi}_i}{\partial \varphi}\frac{\partial \varphi}{\partial y})(\frac{\partial \hat{\psi}_j}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial \hat{\psi}_j}{\partial \eta}\frac{\partial \eta}{\partial y} + \frac{\partial \hat{\psi}_j}{\partial \varphi}\frac{\partial \varphi}{\partial y}) \\ &+ \alpha^2 (\frac{\partial \hat{\psi}_i}{\partial \xi}\frac{\partial \xi}{\partial z} + \frac{\partial \hat{\psi}_i}{\partial \eta}\frac{\partial \eta}{\partial z} + \frac{\partial \hat{\psi}_i}{\partial \varphi}\frac{\partial \varphi}{\partial z})(\frac{\partial \hat{\psi}_j}{\partial \xi}\frac{\partial \xi}{\partial z} + \frac{\partial \hat{\psi}_j}{\partial \eta}\frac{\partial \eta}{\partial z} + \frac{\partial \hat{\psi}_j}{\partial \varphi}\frac{\partial \varphi}{\partial z}) \Bigg\} \\ &\cdot \ |\mathbf{J}| \ \ d\xi \, d\eta \, d\varphi \end{aligned} \tag{12}$$

$$\begin{aligned} \{\mathbf{b}^e\}_i &= \int_{\hat{\Omega}_e} -2\alpha_1^2 \Bigg\{ u_0 (\frac{\partial \hat{\psi}_i}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial \hat{\psi}_i}{\partial \eta}\frac{\partial \eta}{\partial x} + \frac{\partial \hat{\psi}_i}{\partial \varphi}\frac{\partial \varphi}{\partial x}) \\ &+ v_0 (\frac{\partial \hat{\psi}_i}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial \hat{\psi}_i}{\partial \eta}\frac{\partial \eta}{\partial y} + \frac{\partial \hat{\psi}_i}{\partial \varphi}\frac{\partial \varphi}{\partial y}) \\ &+ w_0 (\frac{\partial \hat{\psi}_i}{\partial \xi}\frac{\partial \xi}{\partial z} + \frac{\partial \hat{\psi}_i}{\partial \eta}\frac{\partial \eta}{\partial z} + \frac{\partial \hat{\psi}_i}{\partial \varphi}\frac{\partial \varphi}{\partial z}) \Bigg\} \cdot |\mathbf{J}| \ \ d\xi \, d\eta \, d\varphi \end{aligned} \tag{13}$$

Note that if we set $\varepsilon = \alpha^2$, the elemental matrix may be written as

$$\{\mathbf{A}^e\}_{ij} = \{\mathbf{M}^e\}_{ij} + \varepsilon \{\mathbf{N}^e\}_{ij} \tag{14}$$

The assembling of such matrices yields a symmetric matrix given by equation (2). Here we propose to solve the corresponding linear (1) system of equations by using PCG algorithm. Once we have obtained ϕ, the resulting wind field is computed using equation (8).

3 Updating of the incomplete Cholesky factorization

We will generalize the incomplete factorization proposed by Meurant [6] for the case of matrices $A_\varepsilon = M + \varepsilon D$, with D being diagonal, to matrices $A_\varepsilon = M + \varepsilon N$, with M and N being two $n \times n$ symmetric positive definite matrices. We can write A_ε as follows,

$$A_\varepsilon = (m_i j) + \varepsilon (n_i j) = \begin{pmatrix} m_{11} + \varepsilon n_{11} & (f_{1M} + \varepsilon f_{1N})^T \\ f_{1M} + \varepsilon f_{1N} & M_2 + \varepsilon N_2 \end{pmatrix}$$

where f_{1M}, f_{1N} represent $(n-1) \times 1$ column matrices and M_2, N_2, $(n-1) \times (n-1)$ matrices.

A factorization of the first row and column of A_ε is carried out,

$$A_\varepsilon =$$

$$\begin{pmatrix} m_{11} + \varepsilon n_{11} & \mathbf{0} \\ l_{1M} + \varepsilon l_{1N} & \mathbf{I} \end{pmatrix} \begin{pmatrix} (m_{11} + \varepsilon n_{11})^{-1} & \mathbf{0} \\ \mathbf{0} & C_2 \end{pmatrix} \begin{pmatrix} m_{11} + \varepsilon n_{11} & (l_{1M} + \varepsilon l_{1N})^T \\ \mathbf{0} & \mathbf{I} \end{pmatrix} =$$

$$L_1 Z_1 L_1^T$$

with $l_{1M} = f_{1M}$ and $l_{1N} = f_{1N}$.

Then, identifying term with term, we obtain for matrix C_2,

$$C_2 = M_2 + \varepsilon N_2 - \frac{1}{m_{11} + \varepsilon n_{11}} (l_{1M} + \varepsilon l_{1N}) (l_{1M} + \varepsilon l_{1N})^T \tag{15}$$

If, in order to build the preconditioner, we consider only the diagonal entries of N as first approximation, equation (15) is simplified since $l_{1N} = 0$,

$$C_2 = \varepsilon D_2 + M_2 - \frac{1}{m_{11} + \varepsilon n_{11}} l_{1M} l_{1M}^T,$$

An order 0 algorithm is derived from,

$$C_2 = \varepsilon D_2 + M_2 - \frac{1}{m_{11}} l_{1M} l_{1M}^T$$

and the entries of C_2 are computed by adding εD_2 to what we would have obtained for the incomplete decomposition of M.
Another approximation consists of considering all the entries in N_2 and neglecting the products εl_{1N} in (15). So, the successive computations of matrices C_i do not involve ε and those may be obtained easily from the M decompositions,

$$C_2 = \varepsilon N_2 + M_2 - \frac{1}{m_{11}} l_{1M} l_{1M}^T$$

and thus, in matrix form,

$$C_2 = \varepsilon N_2 + \begin{pmatrix} m_{22}^{(2)} & f_{2M}^T \\ f_{2M} & M_3 \end{pmatrix} = \begin{pmatrix} m_{22}^{(2)} + \varepsilon n_{22} & (f_{2M} + \varepsilon l_{2N}) \\ f_{2M} + \varepsilon l_{2N} & M_3 + \varepsilon N_3 \end{pmatrix}$$

Only the entries of f_{2M} corresponding to non null entries of M are computed in order to avoid the fill-in, obtaining l_{2M}. So the decomposition of C_2 results,

$$C_2 \approx$$
$$\begin{pmatrix} m_{22}^{(2)} + \varepsilon n_{22} & \mathbf{0} \\ l_{2M} + \varepsilon l_{2N} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \left(m_{22}^{(2)} + \varepsilon n_{22}\right)^{-1} & \mathbf{0} \\ \mathbf{0} & C_3 \end{pmatrix} \begin{pmatrix} m_{22}^{(2)} + \varepsilon n_{22} & (l_{2M} + \varepsilon l_{2N})^T \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$$

where, identifying,

$$C_3 = M_3 + \varepsilon N_3 - \frac{1}{m_{22}^{(2)} + \varepsilon n_{22}} (l_{2M} + \varepsilon l_{2N}) (l_{2M} + \varepsilon l_{2N})^T$$

Similarly, with the same simplifications, we have,

$$C_3 = M_3 + \varepsilon N_3 - \frac{1}{m_{22}^{(2)}} l_{2M} l_{2M}^T = \begin{pmatrix} m_{33} + \varepsilon n_{33} & (f_{3M} + \varepsilon l_{3N})^T \\ f_{3M} + \varepsilon l_{3N} & M_4 + \varepsilon N_4 \end{pmatrix}$$

that is constructed following the same procedure of C_2.

In this way, obtaining all the matrices C_i, the incomplete decomposition of A_ε results,

$$A_\varepsilon \approx L_1 Z_1 L_1^T = L_1 L_2 Z_2 L_2^T L_1^T = (L_1 L_2 \cdots L_n) Z_n (L_1 L_2 \cdots L_n)^T \qquad (16)$$

being Z the diagonal matrix,

$$\begin{pmatrix} (m_{11} + \varepsilon n_{11})^{-1} & & & . & \\ & \left(m_{22}^{(2)} + \varepsilon n_{22}\right)^{-1} & & . & \\ & & \left(m_{33}^{(3)} + \varepsilon n_{33}\right)^{-1} & . & \\ . & . & . & . & . \\ & & & . & \left(m_{nn}^{(n)} + \varepsilon n_{nn}\right)^{-1} \end{pmatrix}$$

The diagonal entries of the lower triangular matrix $L_1 L_2 \cdots L_n$ are $m_{ii}^{(i)} + \varepsilon n_{ii}$. The respective columns below the diagonal entries are defined by $(n - i) \times 1$ matrices $l_{jM} + \varepsilon l_{jN}$.

4 Numerical experiments

In this section we present the results obtained using CG with the proposed preconditioners for solving the linear systems of equations arising from an elliptic equation related to a 3-D mass consistent model for wind field adjustment [1, 2]. All the experiments were carried out in a XEON Precision 530 with Fortran Double Precision. In the resolution, we always started from the null vector and stopped if $\|r_k\|_2 \leq 10^{-10}\|r_0\|_2$ or if the number of iterations was greater than 10000.

Two cases are presented, for each one we generated two sets of values of parameter α. For the set of extreme values 0 y 20, the initial population is randomly generated and should be as diverse as possible in order to obtain satisfactory results with GAs [8]. For the set with extremes 0 and 100, the initial population is generated following a normal distribution. The first case is related to a wind simulation in a region of Gran Canaria Island and we have used one mesh to produce linear systems of 98999 equations. The second one, also related to a wind simulation, but now for the whole Gran Canaria Island, we have used one mesh to produce linear systems of 100643 equations. Matrices M, N corresponding to both problems, are SPD.

The results have been represented in several graphics for a wide range of values of α and timings for reaching convergence in each case. ICHOL$_D$ and ICHOL$_N$ represent the ICHOL preconditioners obtained with the approaches developed in section 3. These preconditioners are compared with full-ICHOL of matrix A_ε, that is, computing a new ICHOL decomposition for each ε, and with the use of unique preconditioner, ICHOL(A_{ε_0}), along the whole process.

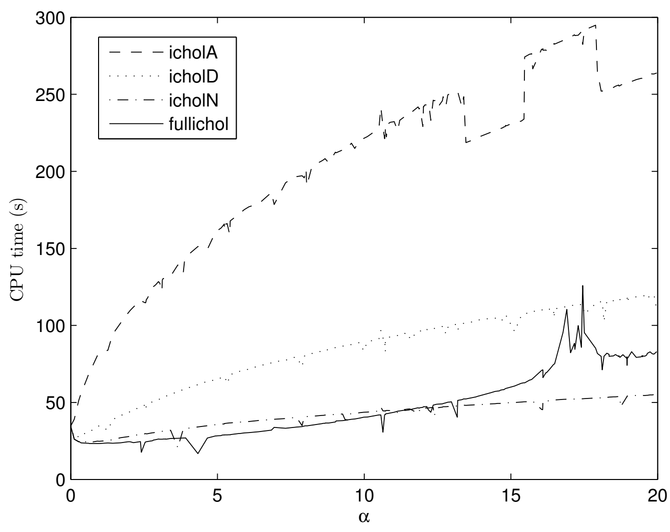

Figure 1: Windfield.98999: Convergence of Conjugate Gradient with several preconditioners ICHOL for different values of parameter α randomly calculated.

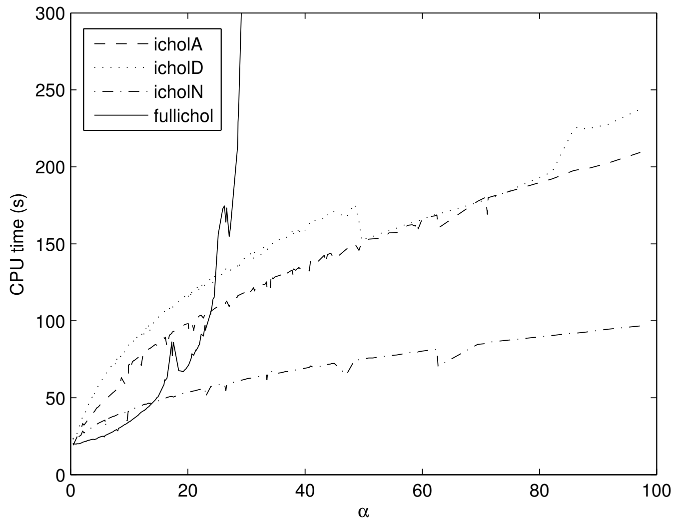

Figure 2: Windfield.98999: Convergence of Conjugate Gradient with several preconditioners ICHOL for different values of parameter α using normal distribution.

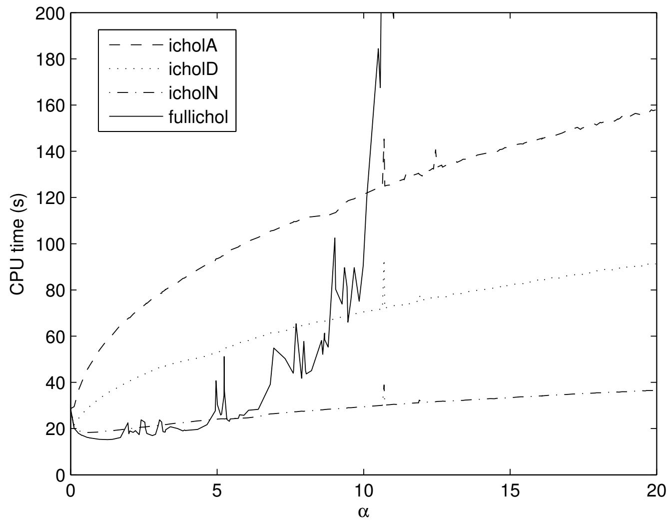

Figure 3: Windfield.100643: Convergence of Conjugate Gradient with several preconditioners ICHOL for different values of parameter α randomly calculated.

Figures 1 to 4 show the behavior of the about mentioned preconditioners. For all cases, preconditioner ICHOL_N presented in this work, leads to the best convergence. On the other hand, for small values of parameter α, preconditioner full-ICHOL has a better behavior. However, for large values of α, it gets worse very fast, making 10000 iterations with no convergence. This result could be explained since we can not always

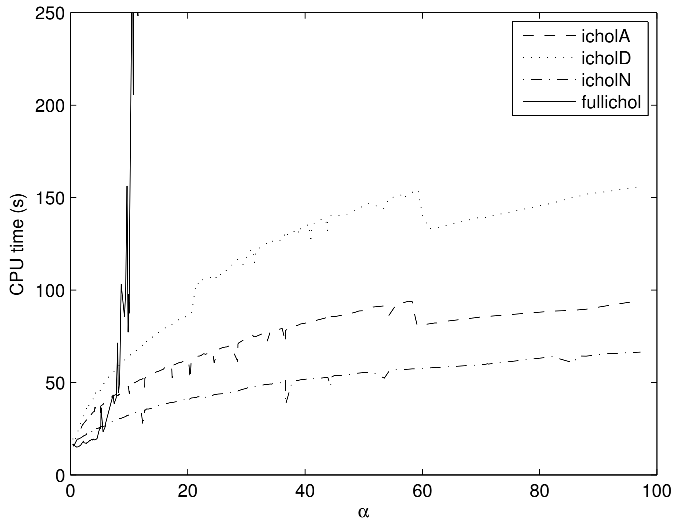

Figure 4: Windfield.100643: Convergence of Conjugate Gradient with several preconditioners ICHOL for different values of parameter α using normal distribution.

obtain an incomplete factorization ILU(0) from the definite positive matrix $(M + \varepsilon N)$ which preserves symmetry and positivity. In consequence, this factorization is not a good preconditioner for Conjugated Gradient method . Moreover, an inversion in the behavior of preconditioners ICHOL(A_{ε_0}) and ICHOL$_D$ can be observed in figures for the two different populations of α used in the numerical experiments. This inversion can be explained because the values of the parameter taking to build the preconditioners ICHOL(A_{ε_0}) are different in both cases.

5 Conclusion

The incomplete factorization for each value of the stability parameter of the shifted linear systems, arising from numerical simulation of wind fields in mass consistent models, is not a good preconditioner. However, the preconditioner ICHOL$_N$, presented in this work leads to better results and thus it is a good choice for systems obtained in the performance of genetic algorithms for those problems of simulation of wind fields.

References

[1] G. Montero, R. Montenegro, J.M. Escobar, *"A 3-D Diagnostic Model for Wind Field Adjustment"*, J. Wind Eng. Ind. Aer., 74-76, 249-261, 1998.

[2] G. Montero, E. Rodríguez, R. Montenegro, J.M. Escobar, J.M. González- Yuste, *"Genetic Algorithms for an Improved Parameter Estimation with Local Refinenent of Tetrahedral Meshes in a Wind Model"*, Adv. Engng. Soft., 36, 3-10, 2005.

[3] G. Montero, R. Montenegro, J.M. Escobar, E. Rodríguez, J.M. González- Yuste, *"Tetrahedral Mesh Generation for Environmental Problems over Complex Terrain"*, Lect. Notes in Comp. Sci., 2329, 335-344, 2002.

[4] E. Rodríguez, G. Montero, R. Montenegro, J.M. Escobar, J.M. González- Yuste, *"Parameter Estimation in a Three-dimensional Wind Field Model Using Genetic Algorithms"*, Lect. Notes in Comp. Sci., 2329, 950-959, 2002.

[5] M. Benzi, D. Bertaccini, *"Approximate Inverse Preconditioning for Shifted Linear Systems"*, BIT Num. Math., 43, 231-244, 2003.

[6] G. Meurant, *"On the incomplete Cholesky Decomposition of a class of perturbed matrices"*, SIAM J. Sci. Comput., 23, 2, 419-429, 2001.

[7] M. Benzi, J.K. Cullum, M. Tůma, *"Robust approximate inverse preconditioning for the conjugate gradient method"*, SIAM J. Sci. Comput. 22, 1318-1332, 2000.

[8] M. Vose, *"Modeling Simple Genetic Algorithm"*, Foundations of Genetic Algorithm 2, 63-74, Whitley (Ed.) Vail, CO: Morgan Kaufman.