LOCAL PRECONDITIONING FOR PARALLEL ITERATIVE SOLVERS

P.CÓRDOBA[†], G. HOUZEAUX[†]

[†]BARCELONA SUPERCOMPUTING CENTER (BSC - CNS), BARCELONA (SPAIN)

paula.cordoba@bsc.es

ABSTRACT

The **discretization of partial differential equations** *coming from different complex physical problems often involves solving* **large sparse linear systems of equations** *with a great number of unknowns. These systems can be solved either with direct or with iterative methods. Iterative solvers are often the ones preferred, as they are cheaper in terms of computer storage and CPU-time, but at the same time they are less robust than direct methods and often converge slowly to the desired solution. To cope with this problem,* **equivalent preconditioned systems** *can be solved instead of the original one. Finding a good preconditioner for solving sparse linear systems of equations is not an easy task and several aspects have to be taken into account. The* **values of the sparse matrix highly depend on the physics of the problem**, *depending on the problem we have, different patterns or dependencies in the same matrix can be observed.* **Adapting the preconditioner to the physics of the problem** *and detecting different behaviours, looking at the values of a sparse matrix can improve convergence in a simple way.*

INTRODUCTION

Numerical modeling can help to understand better and predict the behaviour of certain problems such as fluid dynamics, heat transfer or solid mechanics in physics and engineering. In many cases these problems require the solution of complex PDE's which have to be **discretized** and **solved numerically** to obtain **good approximations of real life solutions**.

The resulting matrix obtained highly depends on the physical phenomena studied in each case this is the reason why before considering any numerical method to solve PDE's it is important to understand the physics that is behind.

Physical Phenomena

Taking the **Advection Diffusion (A-D) Equation with** k = **constant** three behaviours can be observed:

$$\underbrace{\overbrace{\rho c_p \frac{\partial u}{\partial t}}^{Parabolic} + \mathbf{v}\cdot\rho c_p \nabla u}_{Hyperbolic} \underbrace{\overbrace{-k\nabla^2 u = \underbrace{f}_{Hyperbolic}}^{Parabolic}}_{Elliptic}$$

Limiting behaviours of the A-D equation can easily be seen considering its non-dimensional form:

Non-Dimensional Variables

$$t^* = \frac{t}{t_0} \quad \mathbf{x}^* = \frac{\mathbf{x}}{\mathbf{L_0}} \quad \mathbf{v}^* = \frac{\mathbf{v}}{\mathbf{v_0}} \quad u^* = \frac{u}{u_0}$$

Non-Dimensional Equation & Peclet's Number

$$\frac{\partial u^*}{\partial t^*} + \mathbf{v}^*\cdot\nabla u^* - \frac{1}{\mathrm{Pe}}\nabla^2 u^* = f'$$

$$\text{with } \mathrm{Pe} = \frac{\mathbf{v_0 L_0}}{\alpha} = \frac{\mathbf{v_0 L_0}\rho c_p}{k} \text{ and } f' = \frac{f}{u_0}$$

Limiting behaviours in the A-D:

- **Hyperbolic behaviour:**

$$\mathrm{Pe} \to \infty \Rightarrow \frac{1}{\mathrm{Pe}} \to 0 \text{ then} \frac{\partial u^*}{\partial t^*} + \mathbf{v}^*\cdot\nabla u^* = f'$$

- **Parabolic behaviour:**

$$\mathrm{Pe} \to 0 \Rightarrow \frac{1}{\mathrm{Pe}} \to \infty \text{ then} \frac{\partial u^*}{\partial t^*} - \frac{1}{Pe}\nabla^2 u^* = f'$$

Numerical Treatment

- Discretization of conitnuum equations of physics (Finite Element, Finite Diference, Finite Volume) leads us to **sparse linear systems of equations (SLSE)** $\mathrm{A}\mathbf{x} = \mathbf{b}$ ⇒ Few matrix entries of A differ from zero.

- Solution to SLE's can be done using:

$$\Rightarrow \begin{cases} \textbf{Direct Methods} \Rightarrow \begin{cases} \text{Computationally Expensive} \\ \text{Robust} \end{cases} \\ \textbf{Iterative Methods} \Rightarrow \begin{cases} \text{Computationally Cheaper} \\ \text{Less robust than direct methods} \end{cases} \end{cases}$$

PRECONDITIONING

To improve the convergence of **iterative methods**, equivalent preconditioned systems can be solved instead of the original one, this means multiplying the system by a matrix called preconditioner, which has part of the information contained in the original matrix.

$$\mathrm{M}^{-1}\mathrm{A}\mathbf{x} = \mathrm{M}^{-1}\mathbf{b} \Rightarrow \textbf{Left Preconditioning}$$
$$\mathrm{A}\mathrm{M}^{-1}\mathrm{M}\mathbf{x} = \mathbf{b} \quad \Rightarrow \textbf{Right Preconditioning}$$

TYPES OF PRECONDITIONERS

- **'Brute Force Methods'** ⇒ Multigrid, Domain Decomposition Methods etc.
  - Expensive
- **'Simple Methods'** ⇒ Jacobi, Gauss-Seidel, Linelet etc.
  - Cheap
  - Can be adapted to the physics of the problem ⇒ **LOCAL PRECONDITIONING**

Local Preconditioners

Nodal reordering according with the physics of the problem in each case.

- **Anisotropy Linelet**
  - Solving **Poisson's equation** ⇒ $\nabla^2 u = f$ in an anisotropic mesh (ie. Boundary layer problem)
  - Dominant terms perpendicular direction to the wall ⇒ A linelet is a bunch of nodes in that direction
  - The matrix tends to be tri-diagonal

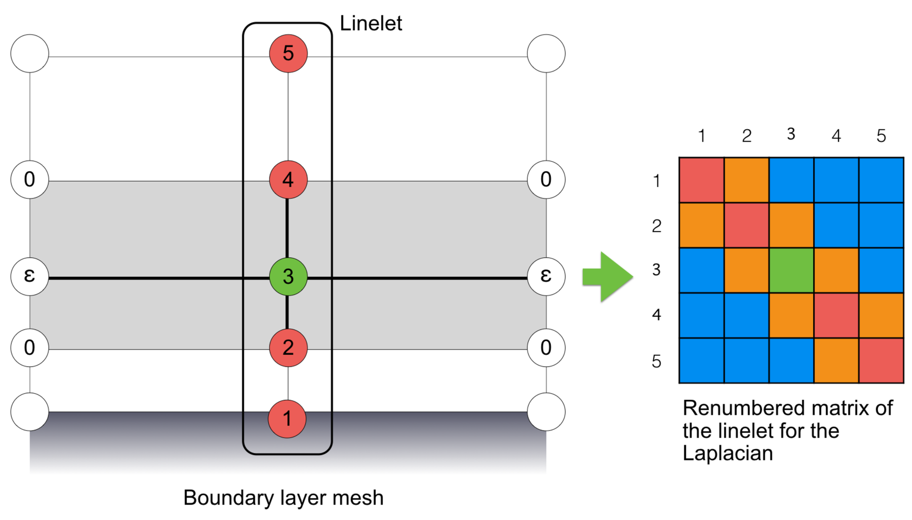


Figure 1: *Nodal Renumbering in Anisotropy Linelet*

- **Streamline Linelet** ⇒ Numbering the mesh nodes in the flow direction, useful in convection dominated flows ie. velocity is such that $\mathbf{v} = (v_x, 0)$

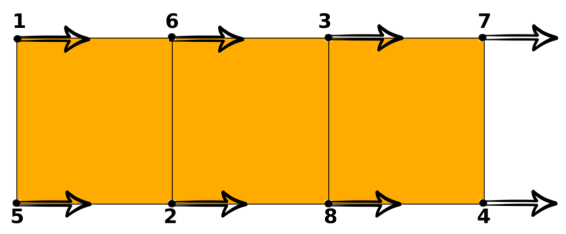


Figure 2: *Random mesh numbering.*

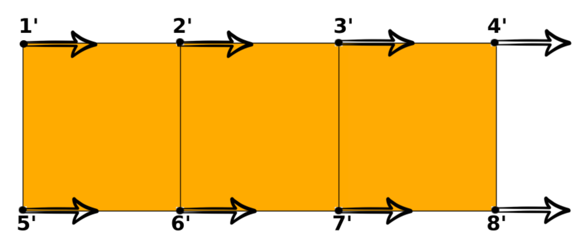


Figure 3: *Mesh numbering along flow direction*

Resultant matrices before and after nodal reordering:

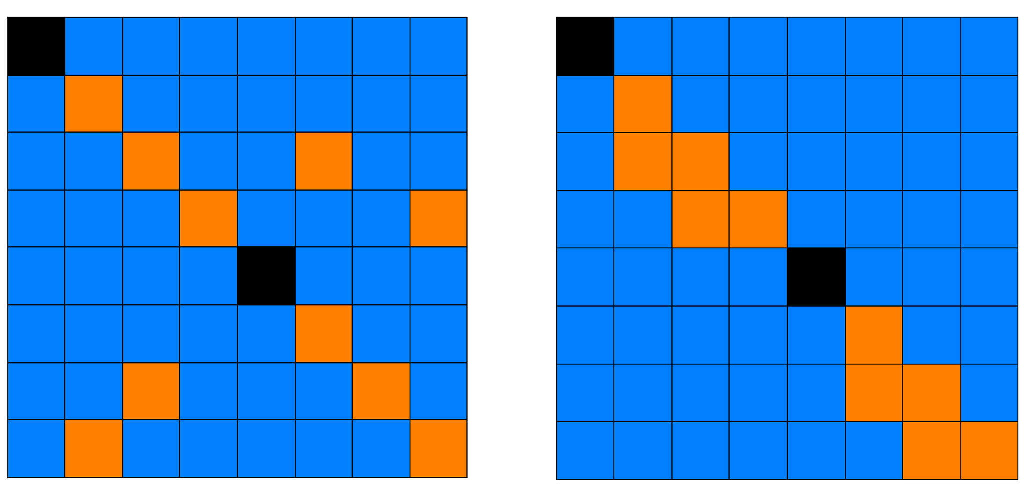


Figure 4: *Left: Resultant matrix from random numbering. Right: Resultant matrix numbered along flow direction.*

The resultant matrix obtained after renumbering, suggests that using a **Gauss Seidel preconditioner** could be a **good option**, as in this particular case it will **converge in just one iteration.**

$$x_i^{(k)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j<i} a_{ij}x_j^{(k)} - \sum_{j>i} a_{ij}x_j^{(k-1)}\right)$$

PARALLELIZATION STRATEGY

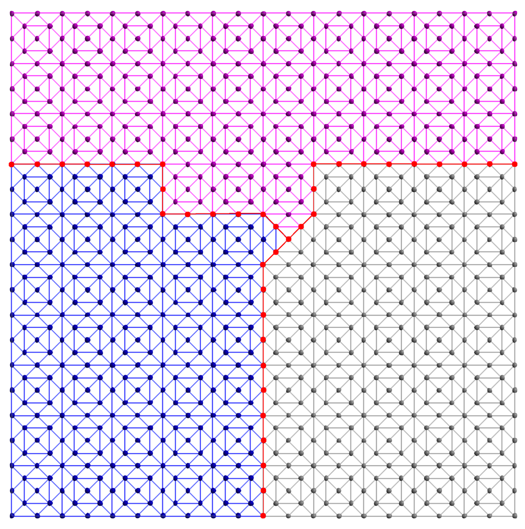


Figure 5: Mesh Partitioned into 3 subdomains

- Pure MPI, no multithreaded or hybrid approaches such as OpenMP or MPI+OpenMP
- Applied to the INTERIOR nodes of each subdomain
- In the INTERFACE NODES ⇒ JACOBI preconditioner

RESULTS

Anisotropy Linelet Preconditioner

Pressure equation for turbulent heat transfer in a channel and nasal cavity:

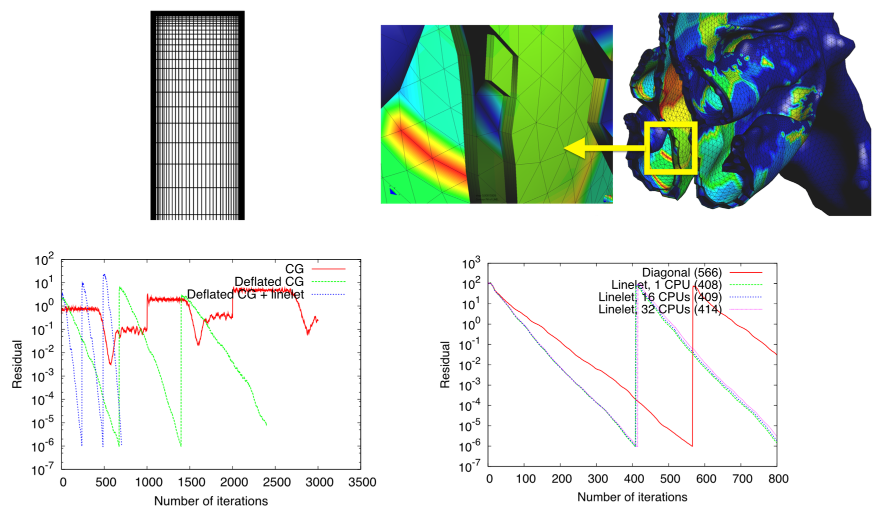
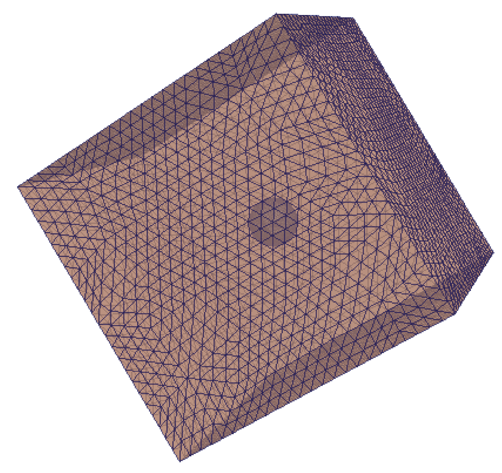


Figure 6: *Convergence curves for the Anisotropy Linelet preconditioner*

Streamline Linelet Preconditioner

Heat Convection 3D:



- Stationary with an initial velocity in the x, y and z directions
- 3D + 559000 elements

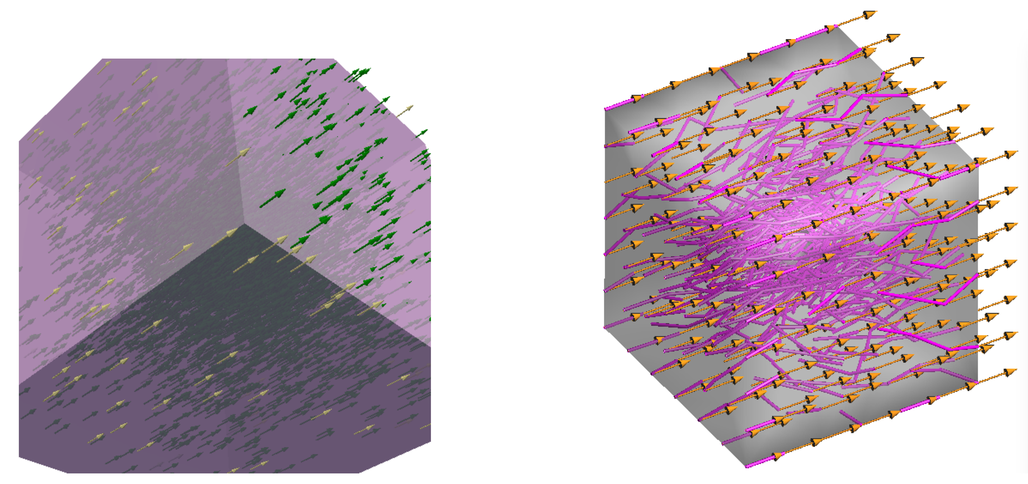


Figure 7: *Streamlines after numbering*

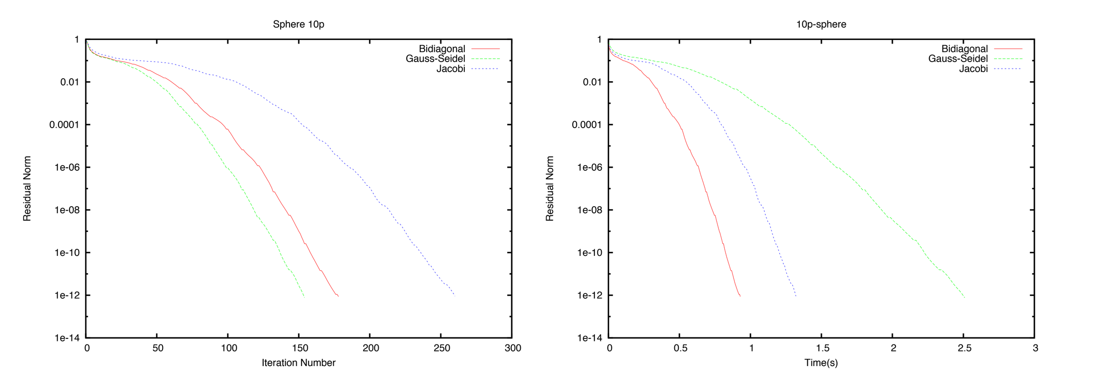


Figure 8: *Convergence and time curves for Streamline Linelet preconditioner*

REFERENCES

[1] F. Magoulès, F.X. Roux, G. Houzeaux, *Parallel Scientific Computing. December 2015, Wiley-ISTE* (December 2015, Wiley-ISTE).

[2] J. Saad. *Iterative Methods for Sparse Linear Systems*. Siam, 2003

[3] P.Córdoba, G. Houzeaux, J.C. Cajas. *Streamwise Numbering for Gauss-Seidel and Bidiagonal Preconditioners in Convection Dominated Flows*.PMAA16: The 9th International Workshop on Parallel Matrix Algorithms and Applications.

[4] O. Soto, R. Löhner and F. Camelli *A linelet preconditioner for incompressible flow solvers*International Journal of Numerical Methods for Heat & Fluid Flow. Vol 13. pp. 133-147, March, 2003