An Interface Strip Preconditioner For Domain Decomposition Methods: Application To Coupled Hydrology and CFD Problems.

Rodrigo R. PAZ - Mario A. STORTI - Norberto M. NIGRO

Centro Internacional de Métodos Computacionales en Ingeniería CIMEC-INTEC-CONICET-UNL
Santa Fe, Argentina
web page:http://www.cimec.org.ar/
e-mail:<rodrigop@intec.unl.edu.ar>

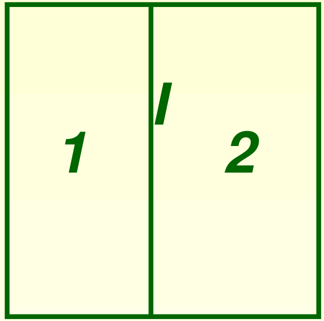
ENIEF 2004, November 8-11, 2004 - S.C. de Bariloche. Argentina

1. Introduction.

- **Parallel Solution for Coupled Surface/Subsurface Large Scale Hydrological Systems and several CFD problems in "Beowulf" clusters.**
- **Direct solvers are highly coupled and don't parallelize well (high communication times). Also they require too much memory, and they asymptotically demand more CPU time than iterative methods even in sequential mode. But they have the advantage that the computational cost do not depend on condition number ($\kappa(A) = ||\mathbf{A}||\,||\mathbf{A}^{-1}||$).**
- **Iteration on the global system of eqs. is highly uncoupled (low communication times) but has low convergence rates, specially for bad conditioned systems ($\kappa(A) \propto 1/h^2$).**
- **"Substructuring" or "Domain Decomposition" methods are somewhat a mixture of both: the problem is solved on each subdomain with a direct method and we iterate on the interface values in order to enforce the equilibrium equations there.**

2. Global iteration methods

$$\begin{bmatrix} \mathbf{A}_{11} & 0 & \mathbf{A}_{1I} \\ 0 & \mathbf{A}_{22} & \mathbf{A}_{2I} \\ \mathbf{A}_{I1} & \mathbf{A}_{I2} & \mathbf{A}_{II} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_I \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_I \end{bmatrix}$$



- **Computing matrix vector operations involve to compute diagonal terms $\mathbf{A}_{11}x_1$ and $\mathbf{A}_{22}x_2$ in each processor and,**
- **Communicate part of the non-diagonal contributions.**

Home Page
Title Page
Contents
Page 3 of 31
Go Back
Full Screen
Close
Quit

3. Domain Decomposition Methods (SD/SCMI)

Eliminate $\mathbf{x}_1$ and $\mathbf{x}_2$ to arrive to the condensed eq.

$$
\begin{aligned}
(\mathbf{A}_{II} - \mathbf{A}_{I1}\mathbf{A}_{11}^{-1}\mathbf{A}_{1I} - \mathbf{A}_{I2}\mathbf{A}_{22}^{-1}\mathbf{A}_{2I})\,\mathbf{x}_I \\
= (\mathbf{b}_I - \mathbf{A}_{I1}\mathbf{A}_{11}^{-1}\,\mathbf{b}_1 - \mathbf{A}_{I2}\mathbf{A}_{22}^{-1}\,\mathbf{b}_2) \\
\mathbf{S}\,\mathbf{x}_I = \tilde{\mathbf{b}}
\end{aligned}
$$

Evaluation of $\mathbf{y}_I = \mathbf{S}\,\mathbf{x}_I$ implies

- Solving the local equilibrium equations in each processor for $\mathbf{x}_j$: $\mathbf{A}_{jj}\,\mathbf{x}_j = -\mathbf{A}_{jI}\,\mathbf{x}_I$
- Suming the interface and local contributions: $\mathbf{y}_I = \mathbf{A}_{II}\,\mathbf{x}_I + \mathbf{A}_{I1}\,\mathbf{x}_1 + \mathbf{A}_{I2}\,\mathbf{x}_2$
- This method will be referred later as SDD/SCMI ("Sub-Domain Direct/Schur complement matrix iterative").

4. SDD/SCMI vs. Global iter.

- **Iteration on the Schur complement matrix (condensed system) is equivalent to iterate on the subspace where the local nodes (internal to each subdomain) are in equilibrium.**
- **The rate of convergence (per iteration) is improved ☺ due to:**

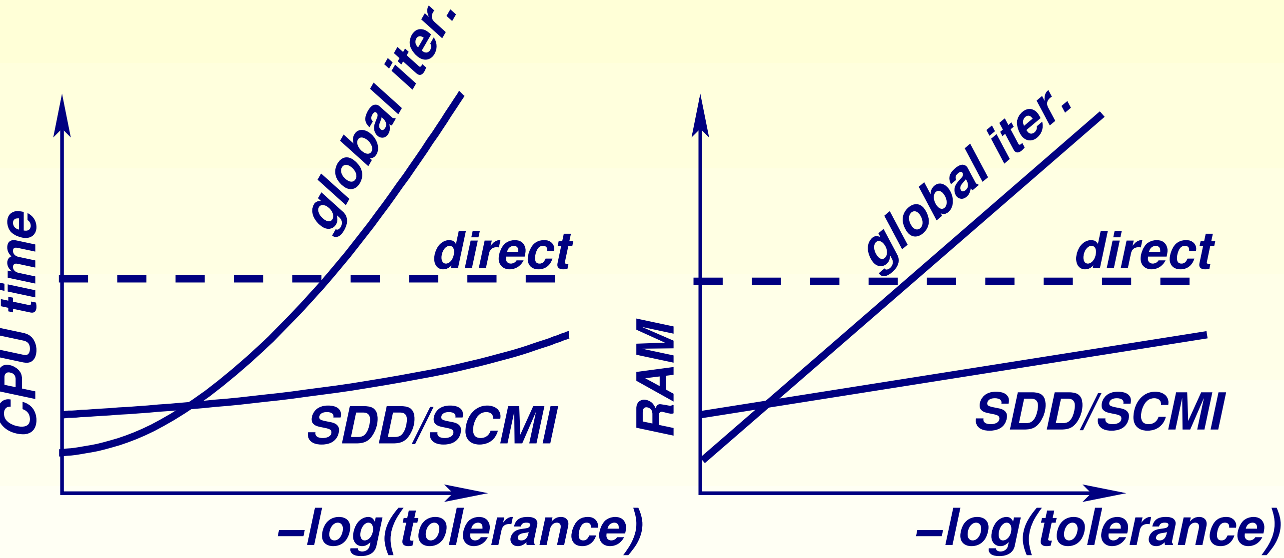
  a) **the condition number of the Schur complement matrix is lower,**

  b) **the dimension of the iteration space is lower (non-stationary methods like CG/GMRES tend to accelerate as iteration proceeds). However this is somewhat compensated by factorization time and backsubst. time in sub-domain problems ☹.**
- **As the iteration count is lower and the iteration space is significantly smaller, RAM requirement for the Krylov space is significantly lower ☺, but this is somewhat compensated by the RAM needed by the factorized internal matrices $LU(\mathbf{A}_{jj})$ ☹.**

5. SDD/SCMI vs. global iter. *(cont...)*

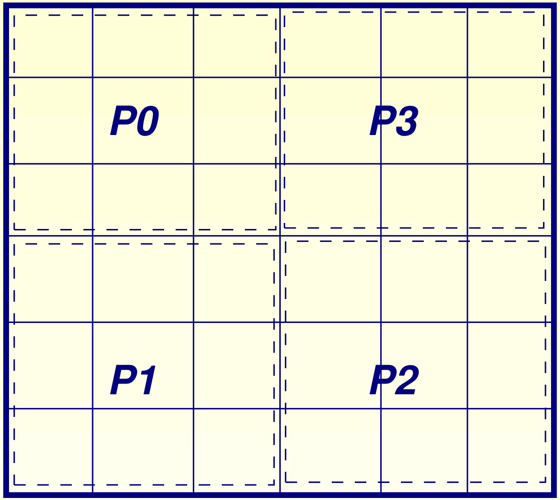
- **Better conditioning of the Schur comp. matrix prevents CG/GMRES convergence break-down due to deterioration of orthogonality ☺**
- **As GMRES CPU time is quadratic in iteration count (orthogonalization stage) and global iteration requires usually more iterations, Schur comp. matrix iteration is comparatively better for lower tolerances (☺/☹).**
- **Global iteration is usually easier to load balance since it is easier to predict computation time accordingly to the number of d.o.f.'s in the subdomain ☹.**

6. Schur comp. matrix iter. vs. global iter. *(cont...)*



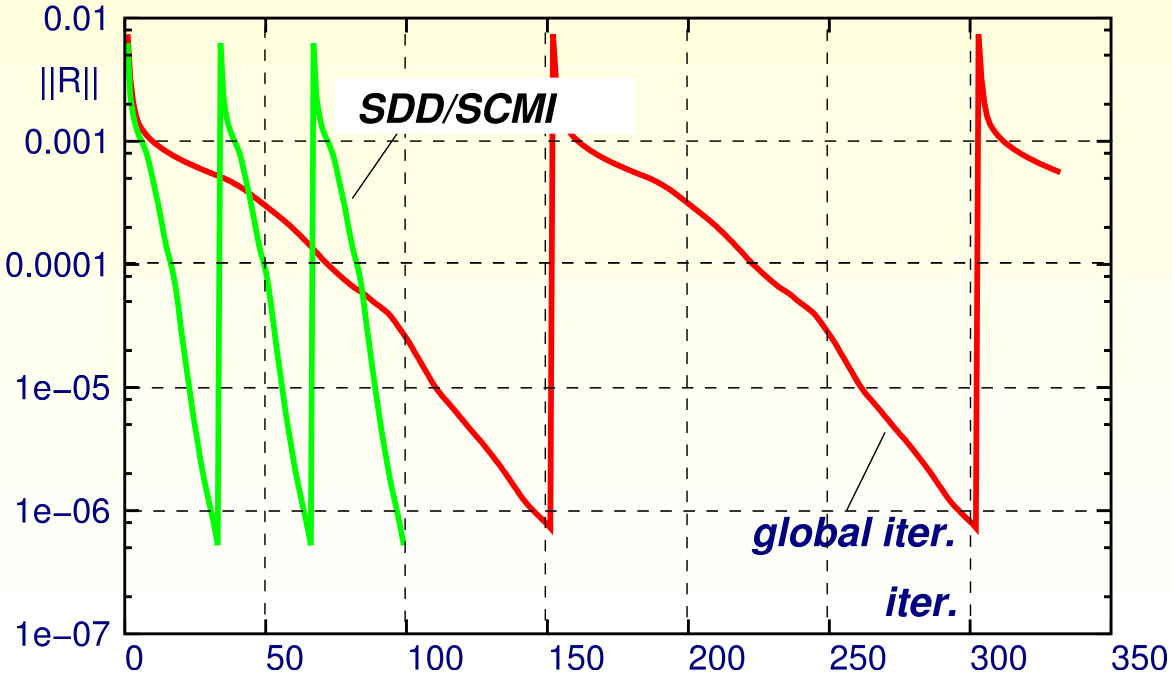
7. Subpartitioning

- For large problems, the factorized part of the $\mathbf{A}_{jj}$ matrix may exceed the RAM in the processor. So we can further subpartition the domain in the processor in smaller sub-subdomains.
- In fact, best efficiency is achieved with relatively small subdomains of 2,000-4,000 d.o.f.'s per subdomain.



8. Example. Navier Stokes cubic cavity Re=1000

625,000 tetras mesh, rtol=10^{-4}, NS monolithic, [Tezduyar et.al. TET (SUPG+PSPG) algorithm, CMAME, vol. 95, pp. 221-242, (1992)]



9. Example. NS cubic cavity Re=1000(*cont.*)

- **Of course, each iteration of SDD/SCMI takes more time, but finally, in average we have**

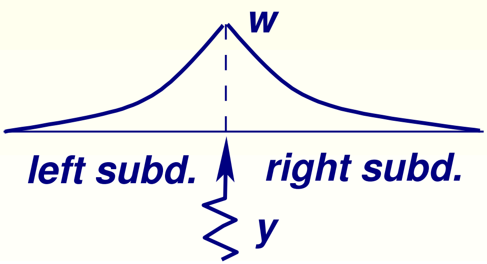
CPU TIME(SDD/SCMI) = 17.7 secs,

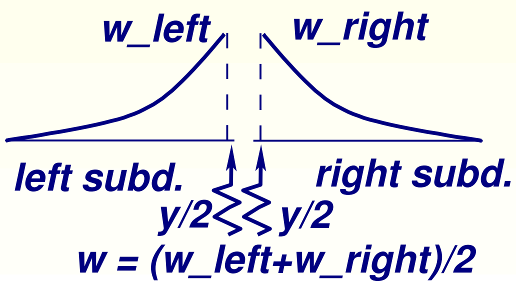
CPU TIME(Global iteration) = 63.8 secs

- **Residuals are on the interface for SCMI, global for Global iter. But vector iteration for the SCM is equivalent to a global vector with null residual on the internal nodes. (So that they are equivalent).**
- **SCMI requires much less communication ☺**

10. Schur CM preconditioning - NN

- In order to further improve SCMI several preconditionings have been developed over years. When solving $Sx = \tilde{b}$ a preconditioner should solve $Sw = y$ for w in terms of y approximately.
- For the Laplace eq. this problem is equivalent to apply a "concentrated heat flux" (like a Dirac's δ) y at the interface and solving for the corresponding temperature field. Its trace on the interface is w.
- Neumann-Neumann preconditioning amounts to split the heat flux one-half to each side of the interface ($\frac{1}{2}y$ for each subdomain).



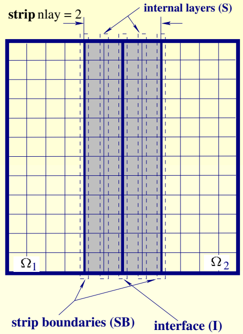


11. Flux splitting

- **Neumann-Neumann prec. works well whenever “equal splitting” is right: right subdomain equal to left subdomain, symmetric operator.**
- **In the presence of advection, splitting is biased towards the downwind sub-domain. Then, eigenfunctions are no more symmetric.**

12. Interface strip preconditioner

Consider the problem in figure. The preconditioning consists in, given a vector f_I defined on the nodes on I, compute an approximate solution v_I given by



$$\begin{bmatrix} A_{II} & A_{IS} & A_{I,SB} \\ A_{SI} & A_{SS} & A_{S,SB} \\ A_{SB,I} & A_{SB,S} & A_{SB,SB} \end{bmatrix} \begin{bmatrix} v_I \\ v_S \\ v_{SB} \end{bmatrix} = \begin{bmatrix} f_I \\ 0 \\ 0 \end{bmatrix}, \tag{1}$$

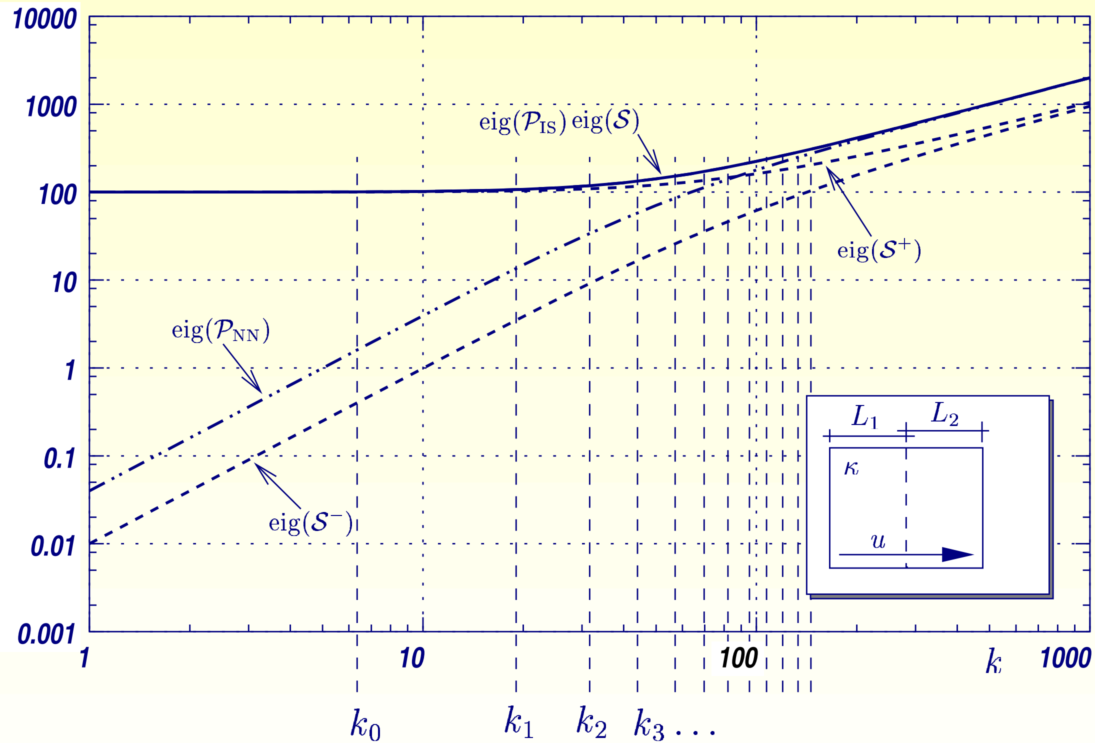
with **"Dirichlet boundary conditions"** on I / $v_{SB} = 0$, then, it reduces to

$$\begin{bmatrix} A_{II} & A_{IS} \\ A_{SI} & A_{SS} \end{bmatrix} \begin{bmatrix} v_I \\ v_S \end{bmatrix} = \begin{bmatrix} f_I \\ 0 \end{bmatrix}, \tag{2}$$

Once this equation is solved, v_I is the value of the proposed preconditioner applied to f_I, i.e.

$$v_I = \mathcal{P}_{\mathrm{IS}}^{-1} f_I \tag{3}$$

13. Strongly advective case (Pe=50)

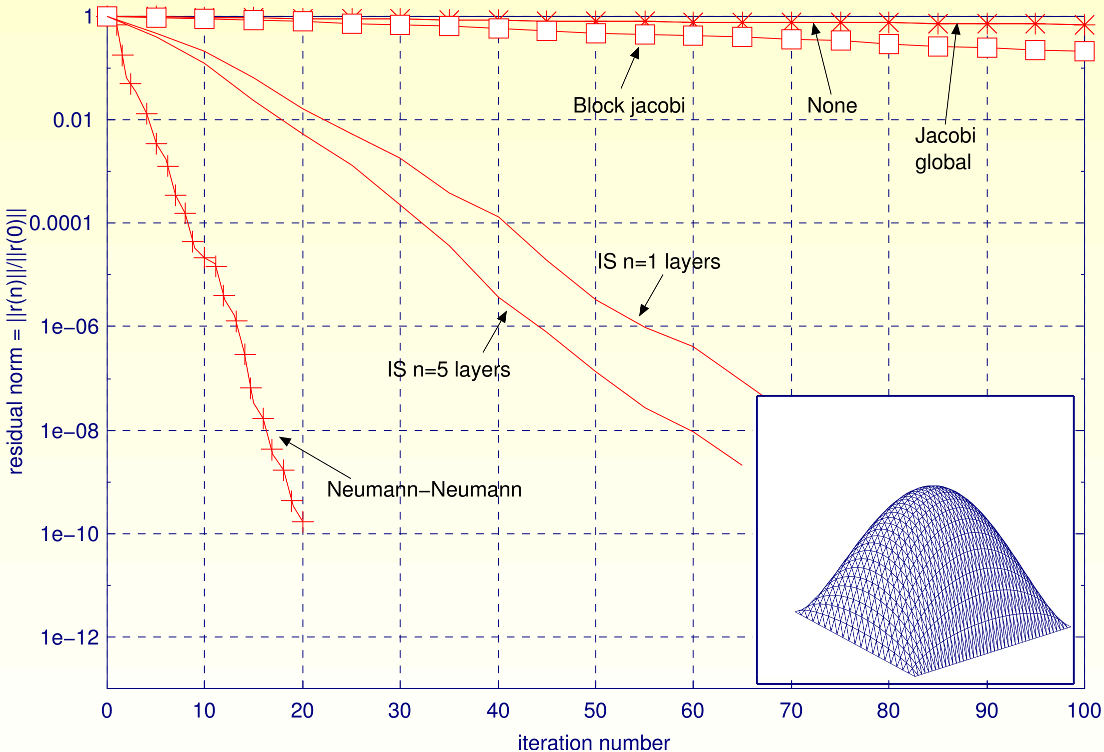


14. Condition number, 100x100 mesh, Pe=50

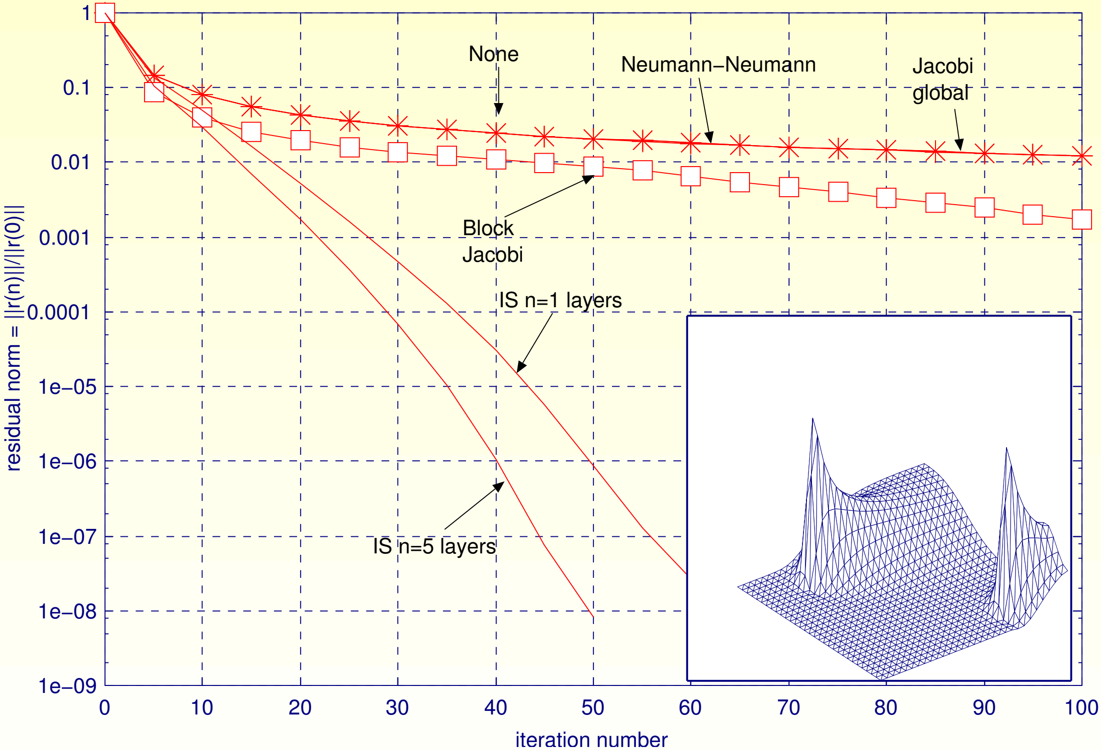
u	$\text{cond}(\mathcal{S})$	$\text{cond}(\mathcal{P}_{\text{NN}}^{-1}\mathcal{S})$	$\text{cond}(\mathcal{P}_{\text{IS}}^{-1}\mathcal{S})$
0	88.50	1.00	4.92
1	81.80	1.02	4.88
10	47.63	3.44	2.92
50	11.23	64.20	1.08

Table 1: Condition number for the Stekhlov operator and several preconditioners for a mesh of 100×100 elements.

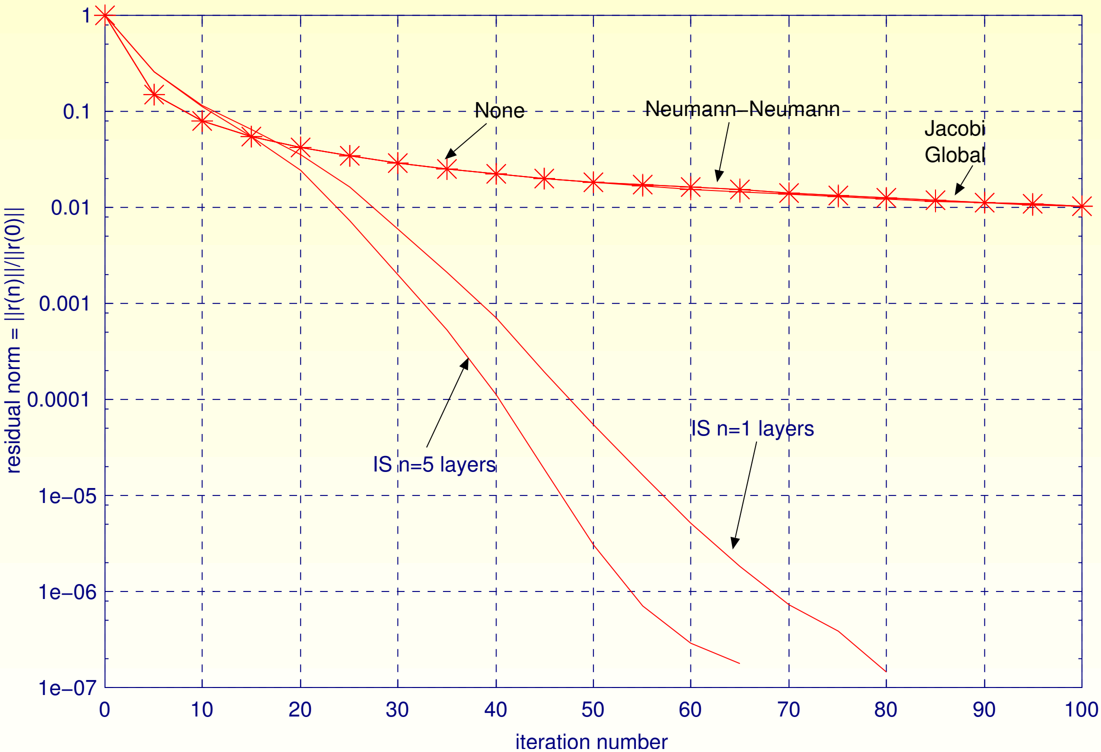
14.1. Solution of Poisson problem (*mesh* 500×500), 4 subdomains (one per processor).



14.2. Solution of advective-diffusive problem (*mesh* 500×500), 4 subdomains (one per processor).



14.3. Solution of advective-diffusive problem (*mesh* 1000×1000), 7 subdomains (one per processor).



15. Shallow Water equations

Conservative form of Mass and Momentum eqs. $\mathbf{U} = (h, u, v)^T$**:**

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathcal{F}_i(\mathbf{U})}{\partial x_i} = \mathbf{G}_i(\mathbf{U}), \quad i = 1, 2, \quad \textbf{on } \Omega_{st} \times (0, t], \tag{4}$$

Flux Functions:

$$\mathcal{F}_1(\mathbf{U}) = (hu, hu^2 + g\tfrac{h^2}{2}, huv)^T, \tag{5}$$

$$\mathcal{F}_2(\mathbf{U}) = (hv, huv, hv^2 + g\tfrac{h^2}{2})^T, \tag{6}$$

Source Term:

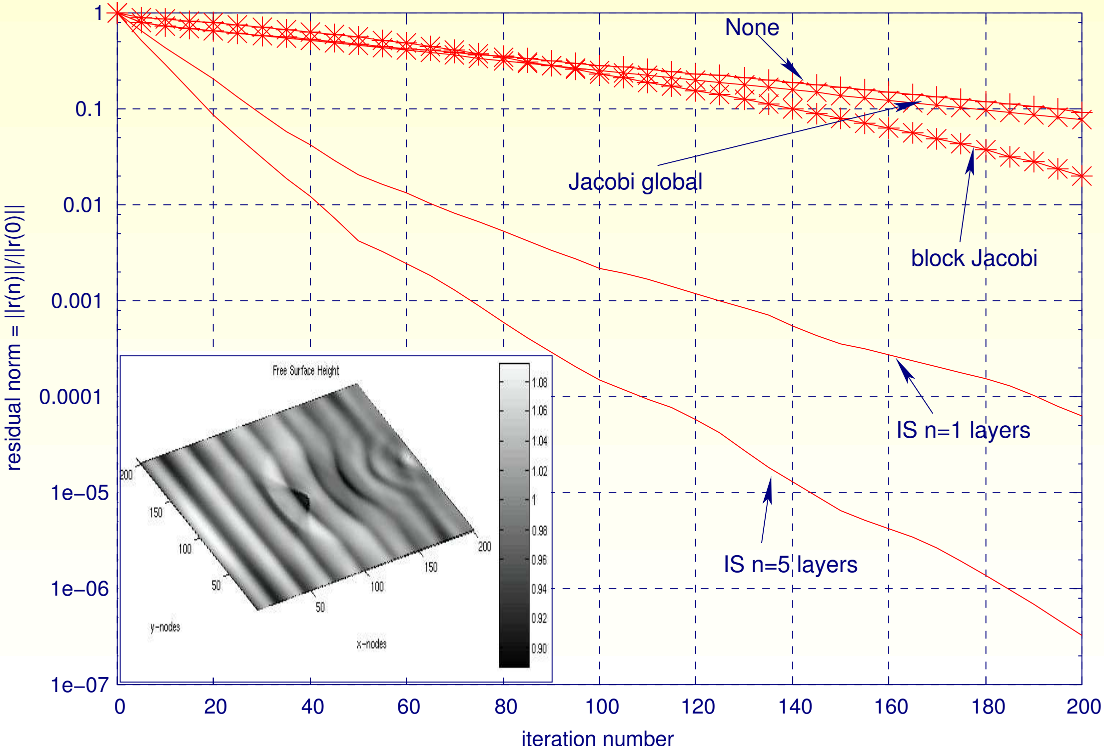
$$\mathbf{G}(\mathbf{U}) = (G_s,\ gh(S_{0x} - S_{fx}),\ gh(S_{0y} - S_{fy}))^T \tag{7}$$

Bed-Channel Friction Models:

$$S_{fx} = \frac{1}{C_h^2 h} u|\bar{u}|, \quad S_{fy} = \frac{1}{C_h^2 h} v|\bar{u}| \qquad \textbf{Chézy model.} \tag{8}$$

$$S_{fx} = \frac{n^2}{h^{4/3}} u|\bar{u}|, \quad S_{fy} = \frac{n^2}{h^{4/3}} v|\bar{u}|, \qquad \textbf{Manning model.} \tag{9}$$

15.1. 2D Shallow Water subcritical flow over an impermeable unit square channel with a parabolic bump at the bottom. Mesh: 10^5 linear triangles partitioned with METIS into 5 sub-domains (one per processor).



16. Coupled Surface-Subsurface Water Flow/

- **Subsurface Flow in a confined (freatic) aquifer integrated in the vertical direction:**

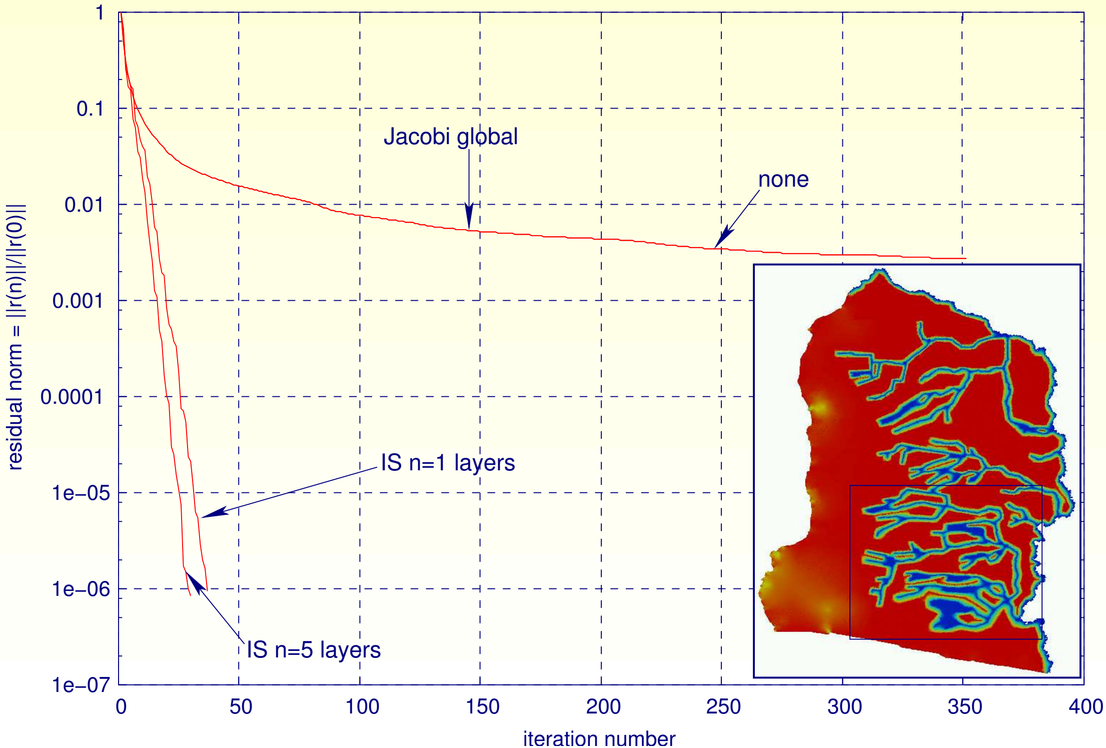
$$\frac{\partial}{\partial t}(S(\phi-\eta)\phi) = \nabla\cdot(K(\phi-\eta)\nabla\phi) + \sum G_a, \quad \textbf{on } \Omega_{aq}\times(0,t]. \tag{10}$$

- **Surface Flow: (1D[variable cross section)/2D Shallow Water equations.)**

- **Coupling Term.**

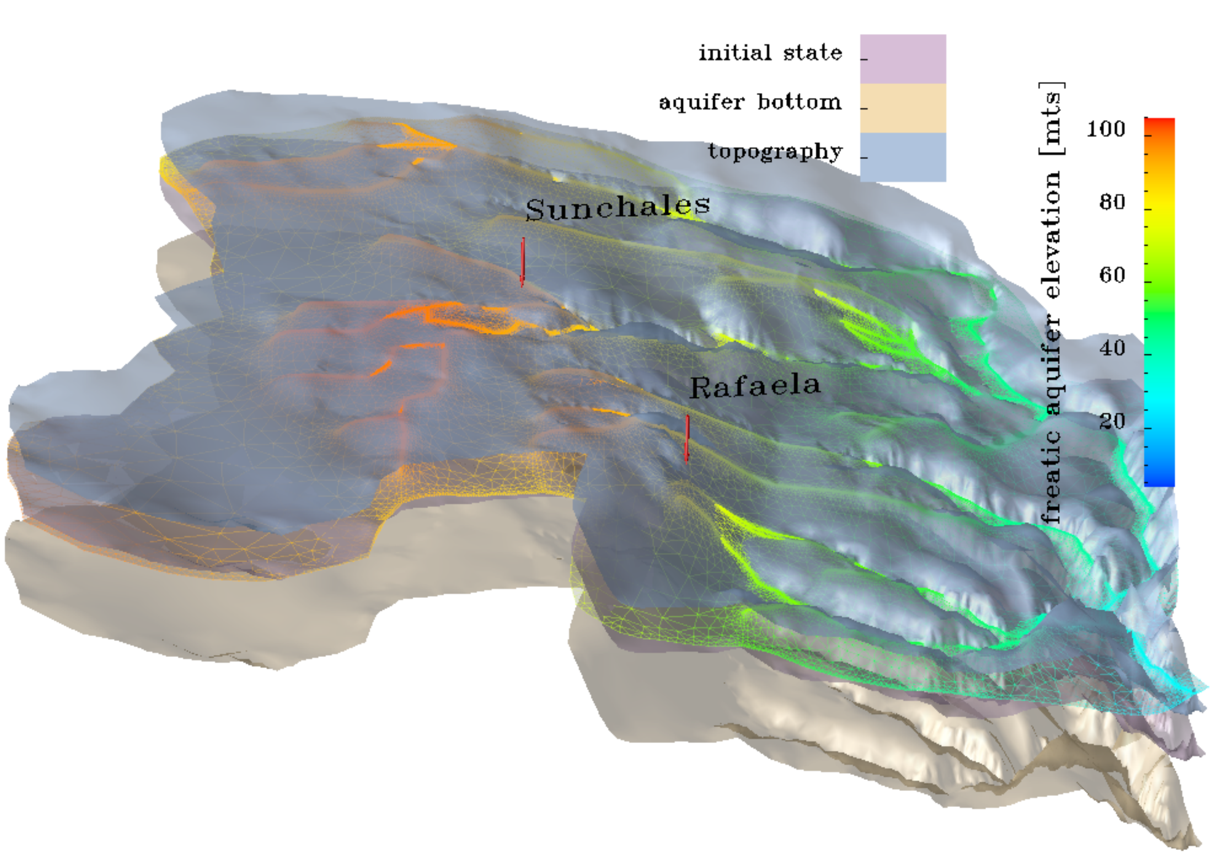
$$\begin{aligned} \textbf{Stream gain/loss:} \quad G_s &= P/R_f(\phi-h_b-h), && (11)\\ \textbf{Aquifer gain/loss:} \quad G_a &= -G_s\,\delta_{\Gamma_s}. && (12) \end{aligned}$$

16.1. 1D Saint-Venant/Groundwater interaction over several basins in Sta Fe. 32900$\mathbf{Km}^2$, 1.65**M** triangles, 9 sub-domains (one per processor). Annual average periodical raindrop.

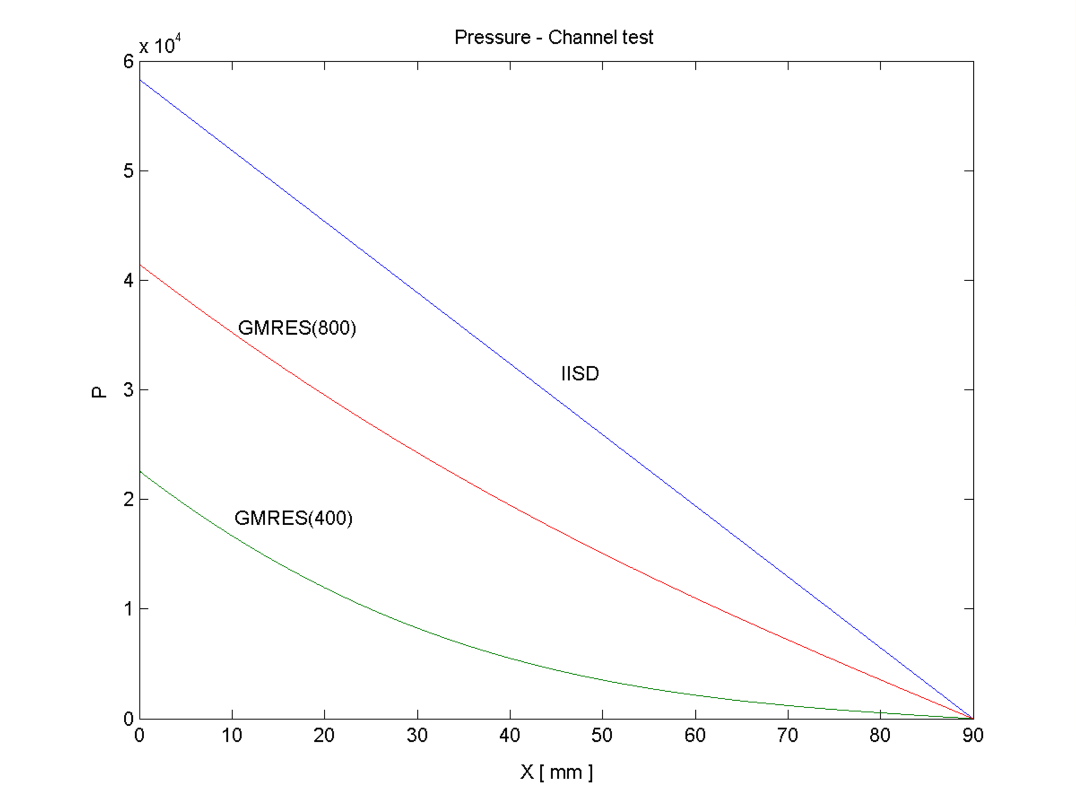


Home Page
Title Page
Contents
Page 22 of 31
Go Back
Full Screen
Close
Quit

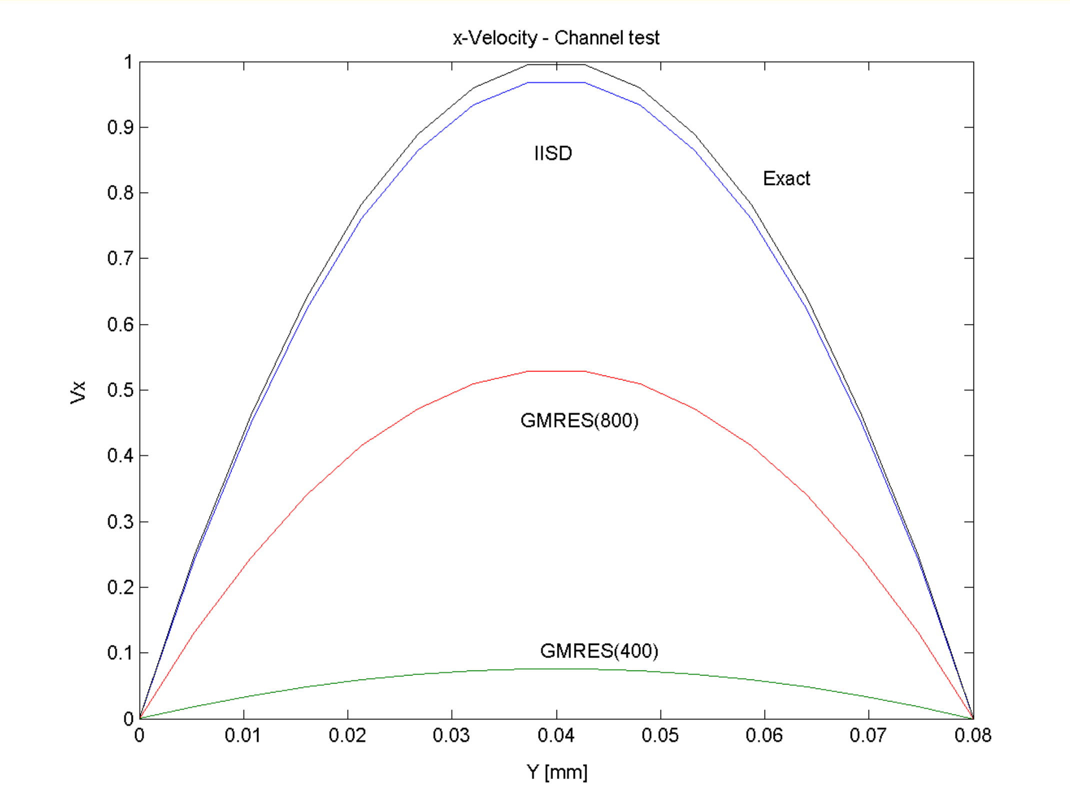
16.2. Periodic Steady Solution in a Cululu basin.



17. Stokes Flow in a long lubricated channel.



18. Stokes Flow in a long lubricated channel. (*cont..*)



Home Page

Title Page

Contents

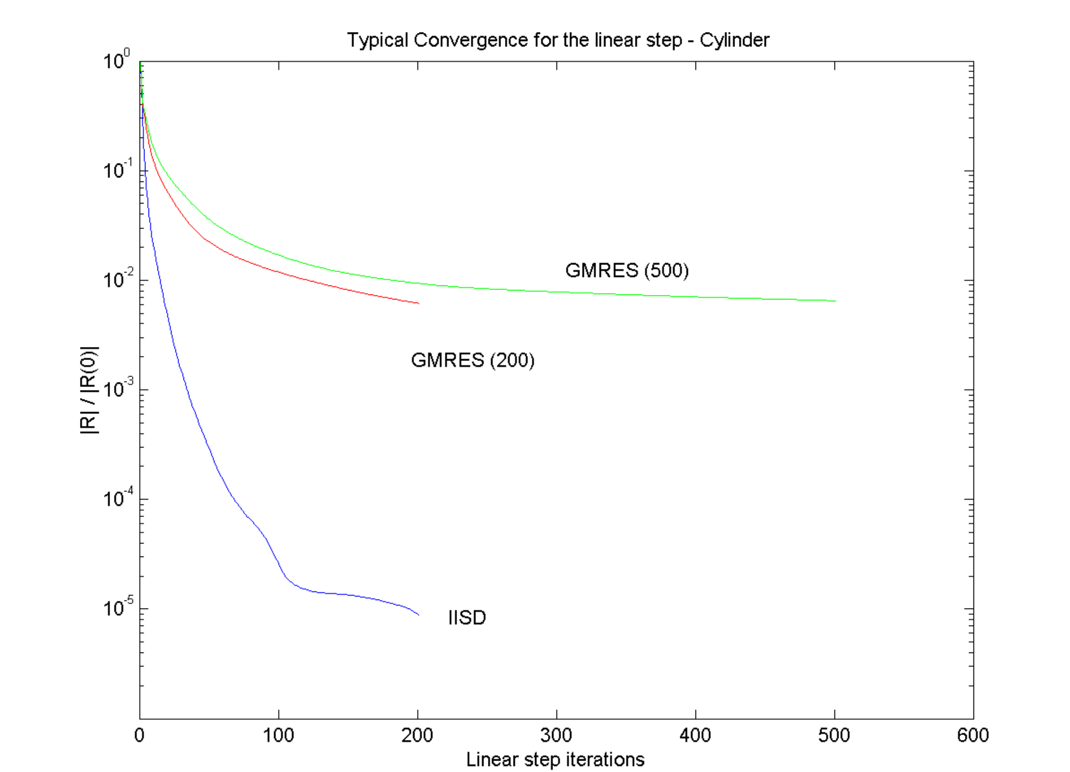
Go Back

Full Screen

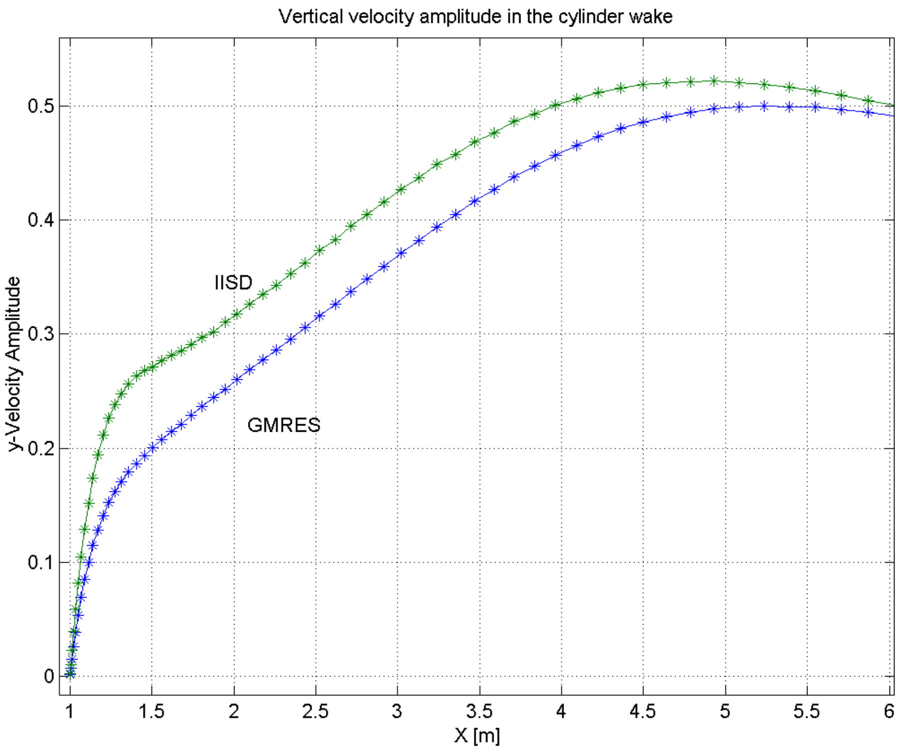
Close

Quit

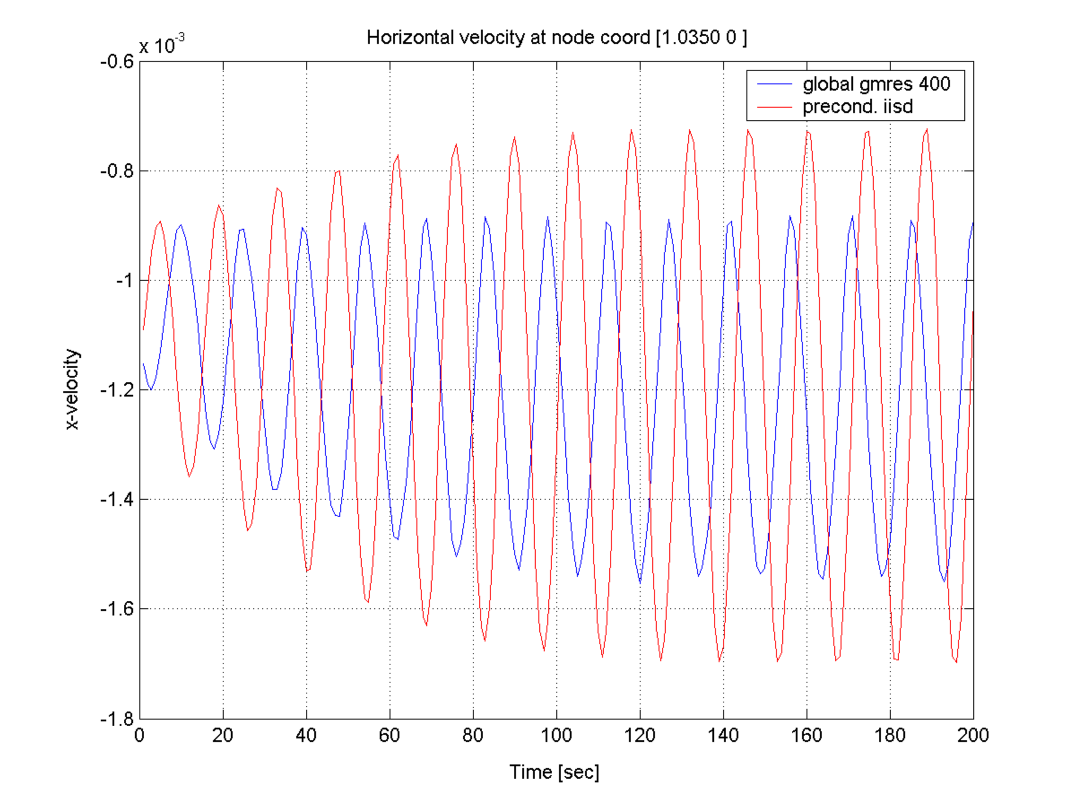
19. N-S Flow arround a cylynder. Iteration count.



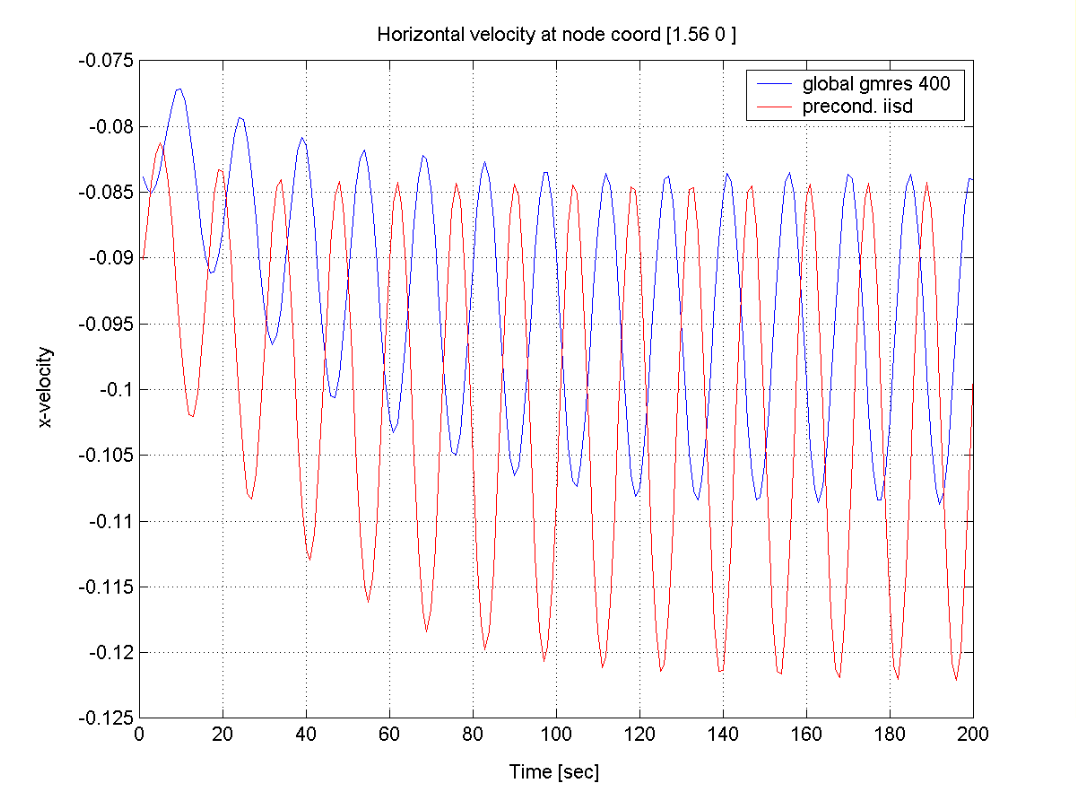
20. N-S Flow arround a cylynder. Vertical velocity in the wake.



21. N-S Flow arround a cylynder. Horizontal velocity near the cyl.



22. N-S Flow arround a cylynder. Vertical velocity near the cyl.



23. Conclusion

- Domain Decomposition + iteration on the Schur complement matrix is an efficient algorithm for solving large linear systems in parallel or sequentially.
- Specially suited for ill-conditioned problems.
- Interface strip preconditioner improves convergence, specially for advection dominated problems or floating subdomains.
- SDD/SCMI + ISP performes better the most inner step in the solution of a non-stationary/non-linear CFD problem.
- SDD/SCMI examples using PETSc-FEM (Finite Element parallel, general purpose, multi-physics code developed at CIMEC). Can be donwloaded at `http://www.cimec.org.ar/petscfem`

24. Future Work

- **Efficiency Tests of the IISD+IS Preconditioner on Disaggregate Schemes (Fractional Step Scheme).**
- **Solve the predictor step (advection-diffusion term) in FS schemes with IISD+IS preconditioner, then the pressure step (Laplacian Problem) with Neumann-Neumann Preconditioner.**
- **Construction of a preconditioner with mixed multigrid (no more than 3 levels) and IISD+ISP schemes.**
- **Test the IISD+ISP solver on turbulent problems (high Reynolds.)**