AN INTERFACE STRIP PRECONDITIONER FOR DOMAIN DECOMPOSITION METHODS

by M. Storti, L. Dalcín, R. Paz

Centro Internacional de Métodos Numéricos

en Ingeniería - CIMEC

INTEC, (CONICET-UNL), Santa Fe, Argentina

<mstorti@intec.unl.edu.ar>

http://www.cimec.org.ar

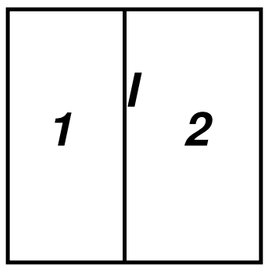
November 2, 2003

Parallel solution of large linear systems

- **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.**
- **Iteration on the global system of eqs. is highly uncoupled (low communication times) but has low convergence rates, specially for bad conditioned systems.**
- ***"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.**

Global iteration methods

$$\begin{bmatrix} A_{11} & 0 & A_{1I} \\ 0 & A_{22} & A_{2I} \\ A_{I1} & A_{I2} & A_{II} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_I \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_I \end{bmatrix}$$



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

Domain Decomposition Methods (DDM/SCMI)

Eliminate x_1 and x_2 to arrive to the condensed eq.

$$(A_{II} - A_{I1}A_{11}^{-1}A_{1I} - A_{I2}A_{22}^{-1}A_{2I})\,x_I = (b_I - A_{I1}A_{11}^{-1}\,b_1 - A_{I2}A_{22}^{-1}\,b_2)$$

$$\tilde{A}\,x_I = \tilde{b}$$

Evaluation of $y_I = \tilde{A}\,x_I$ implies

- **Solving the local equilibrium equations in each processor for x_j:**
  $A_{jj}\,x_j = -A_{jI}\,x_I$
- **Suming the interface and local contributions:**
  $y_I = A_{II}\,x_I + A_{I1}\,x_1 + A_{I2}\,x_2$
- **This method will be referred later as DDM/SCMI (*"Domain Decomposition Method/Schur complement matrix iteration"*).**

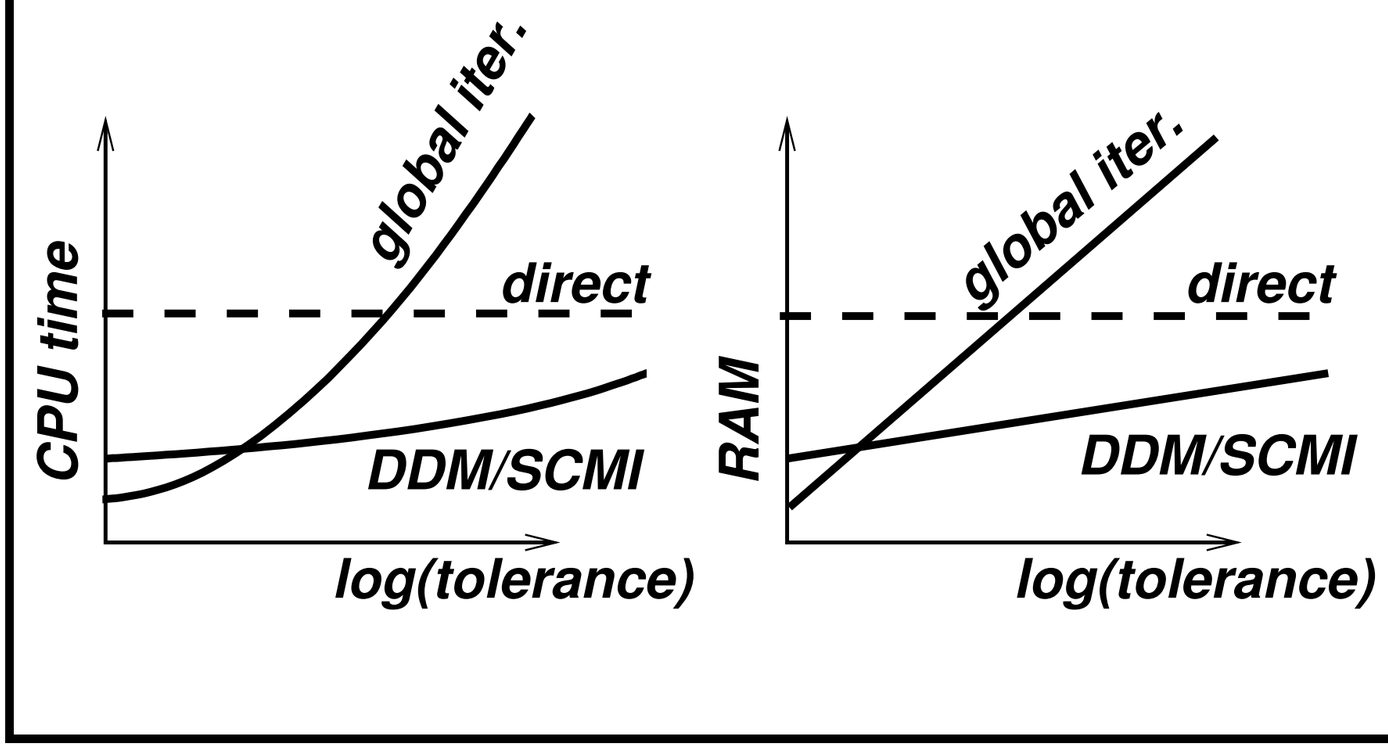
DDM/SCMI vs. Global iter.

- **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 ☹.**
- **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 $LU(A_{jj})$ ☹.**

DDM/SCMI vs. global iter. *(contd...)*

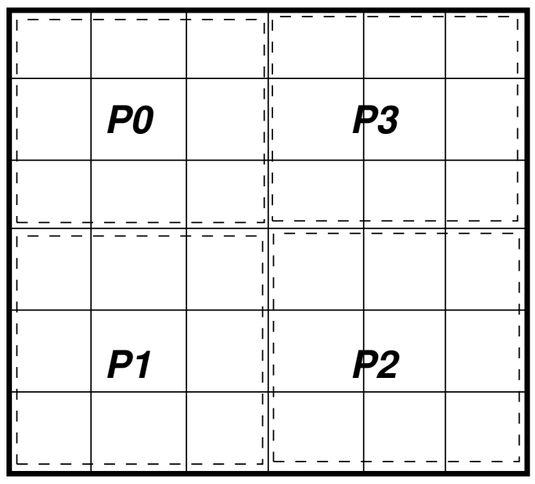
- **Better conditioning of the Schur comp. matrix prevents CG/GMRES convergence break-down ☺**
- **As GMRES CPU time is quadratic in iteration count (orthogonalization stage) and global iteration requires usually more iterations, Schur comp. matrix iteration is comparatively better for lower tolerances (☺/☹).**
- **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 ☹**

Schur comp. matrix iter. vs. global iter. *(contd...)*



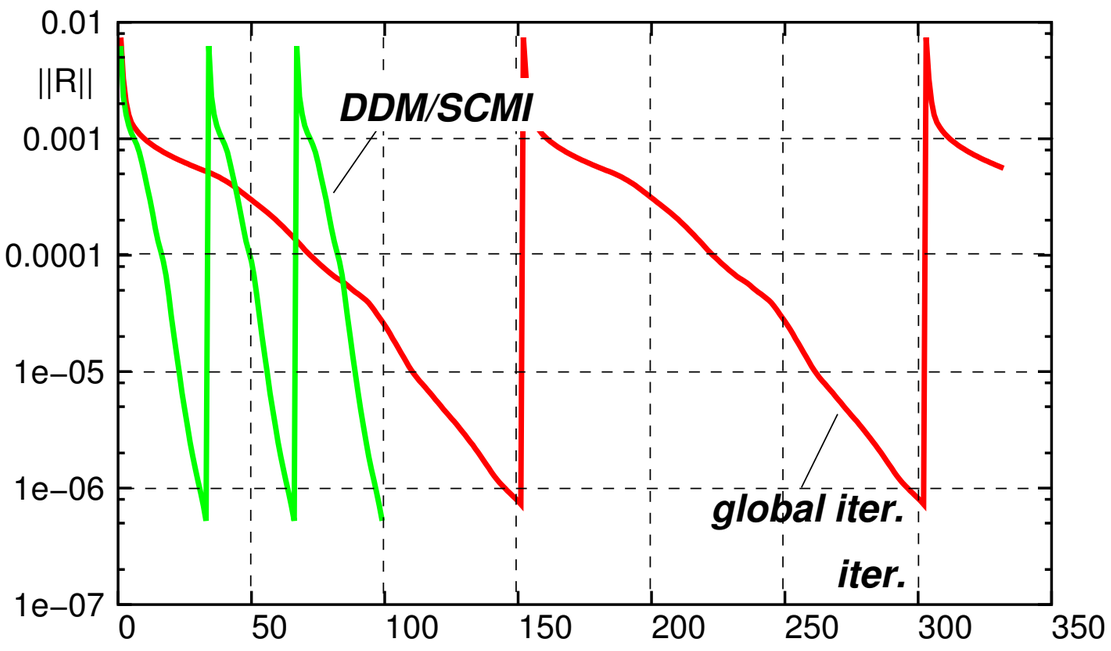
Subpartitioning

- **For large problems, the factorized part of the 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.**



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)]



Example. NS cubic cavity Re=1000 (contd...)

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

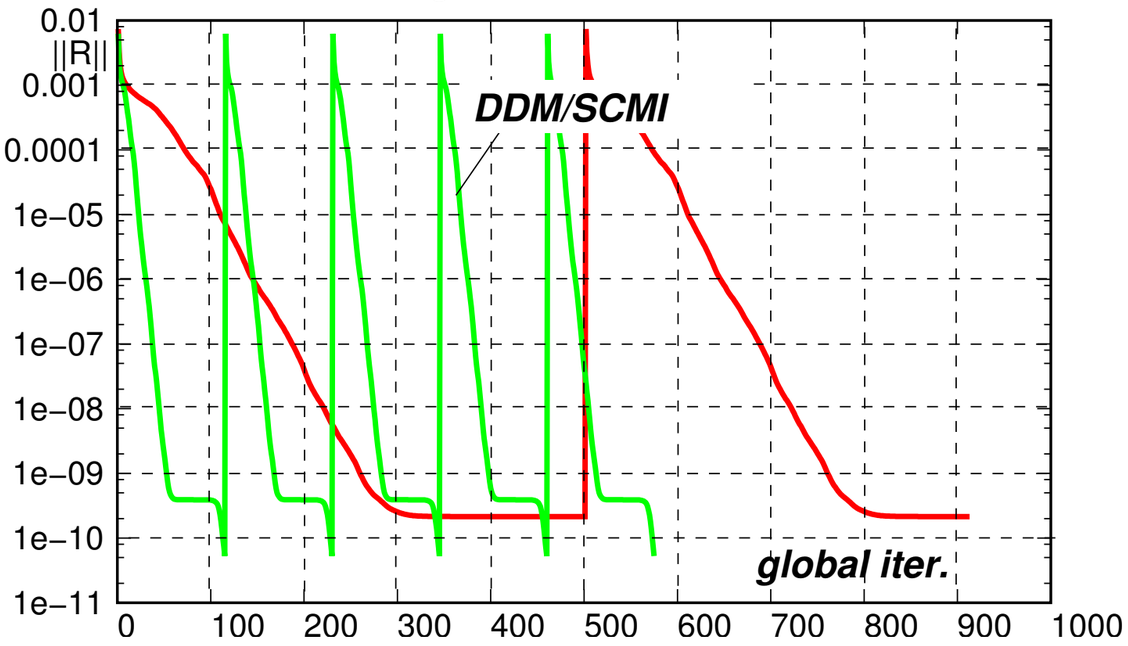
$$\text{CPU TIME(DDM/SCMI)} = 17.7 \text{ secs},$$

$$\text{CPU TIME(Global iteration)} = 63.8 \text{ secs}$$

- **Residuals are on the interface for Schur compl, 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** ☺

Example. NS cubic cavity Re=1000 (contd...)

For a lower tolerance (rtol=10^{-8}), Global iteration breaks down, while DDM/SCMI continues to converge.



Example. NS cubic cavity Re=1000 (contd...)

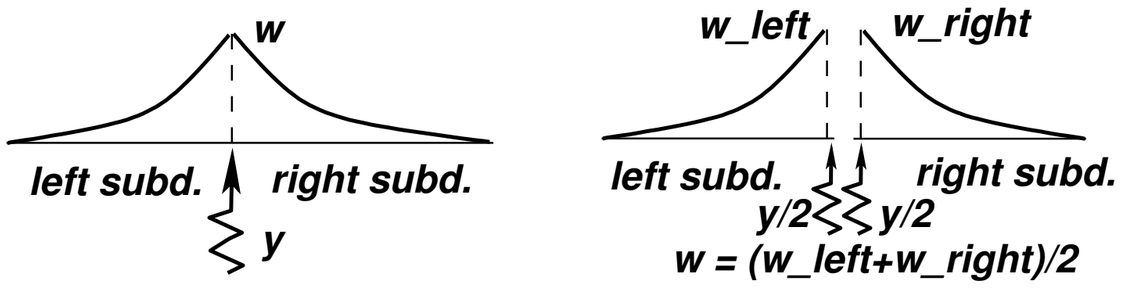
- **All results run in Beowulf cluster Geronimo at CIMEC (see http://www.cimec.org.ar/geronimo). 8 x P4 1.7 Ghz (512MB RIMM (Rambus)).**
- **NS code is PETSc-FEM (GPL FEM multi-physics code developed at CIMEC, see/download http://www.cimec.org.ar/petscfem).**
- **With DDM/SCMI we can run 2.2 Mtetras with NS monolithic (cubic cavity test) 252secs/iter (linear solution only, to rtol=10^{-4}).**
- **Global iteration crashes (out of memory!) at iteration 122 having converged a factor 5×10^{-3}.**
- **DDM/SCMI is specially suited to stabilized methods, since they are in general worse conditioned.**

Fractional step solver

- **Predictor and projection steps have $\kappa = O(1)$ are better solved by global iteration (even Richardson).**
- **Poisson step can be solved with DDM/SCMI.** *[Some of the following observations apply in general for symmetric, positive definite operators.]*
- **Better conditioning (than monolithic) makes global iteration more appropriate ☹. As Conjugate Gradient can be used in place of GMRES, CPU time vs. iteration is no more quadratic ☹**
- **Factorized internal subdomain matrices can be stored and no refactorized, so we can use larger internal subdomains (but requires more RAM) ☺**
- **We can solve 4.2 Mtetra mesh in 58sec (Poisson solution time only...) (rtol=10^{-4}, 130 iters.)**

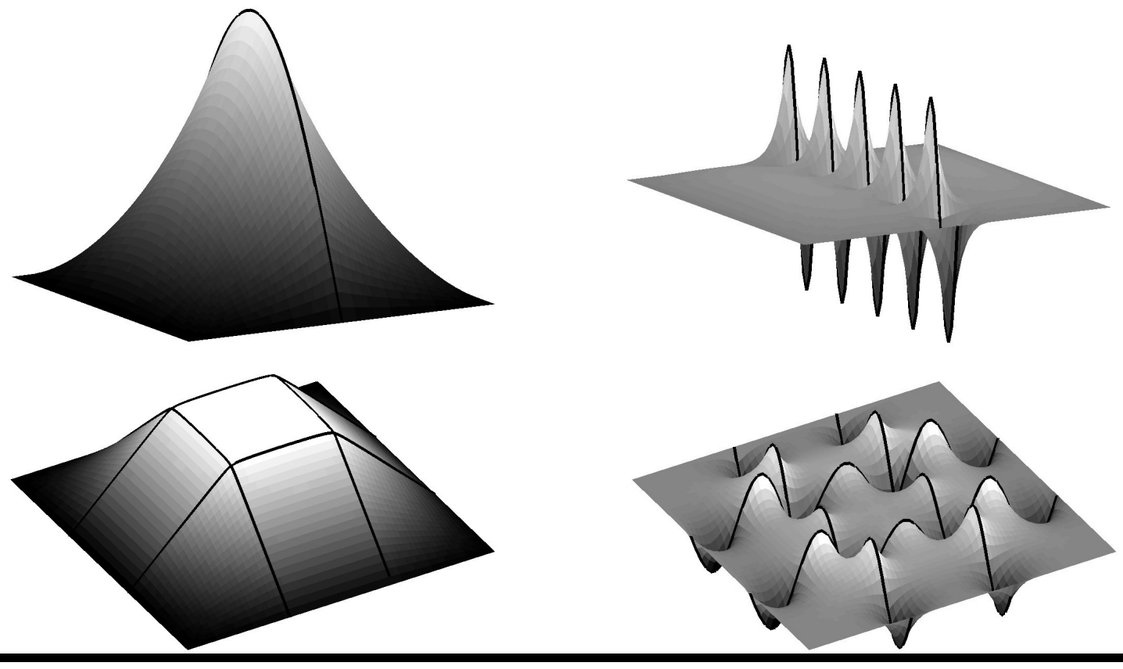
Schur CM preconditioning - NN

- **In order to further improve SCMI several preconditionings have been developed over years. When solving $\tilde{A}x = \tilde{b}$ a preconditioner should solve $\tilde{A}w = y$ 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).**

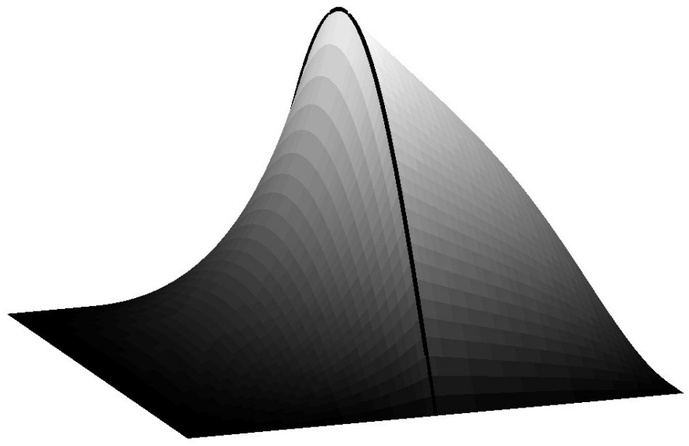


Flux splitting

Neumann-Neumann prec. works well whenever *"equal splitting"* is right: right subdomain equal to left subdomain, symmetric operator... Eigenfunctions of the Stekhlov operator (Schur comp. matrix) are symmetric.



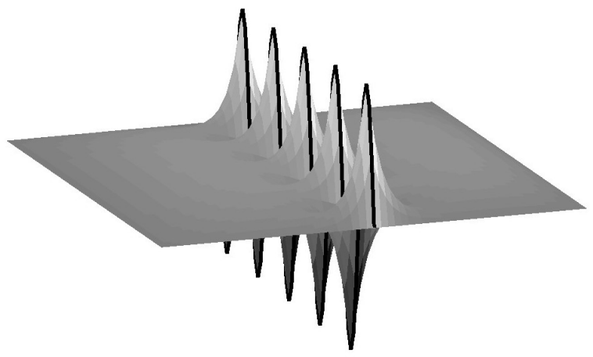
Flux splitting (advective case)

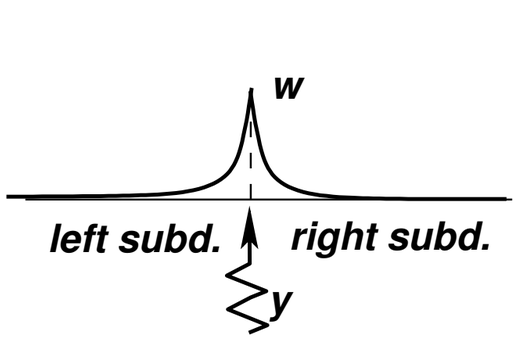


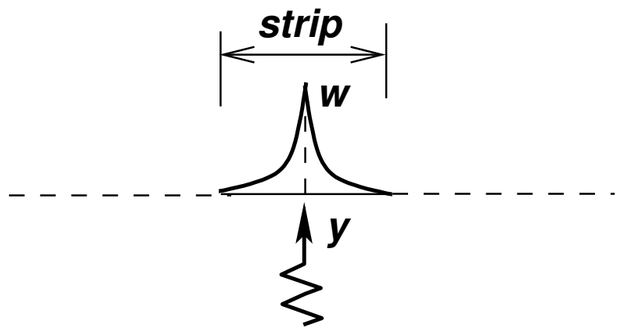
In the presence of advection splitting is biased towards the down-wind sub-domain. Eigenfunctions are no more symmetric.

Interface strip preconditioner

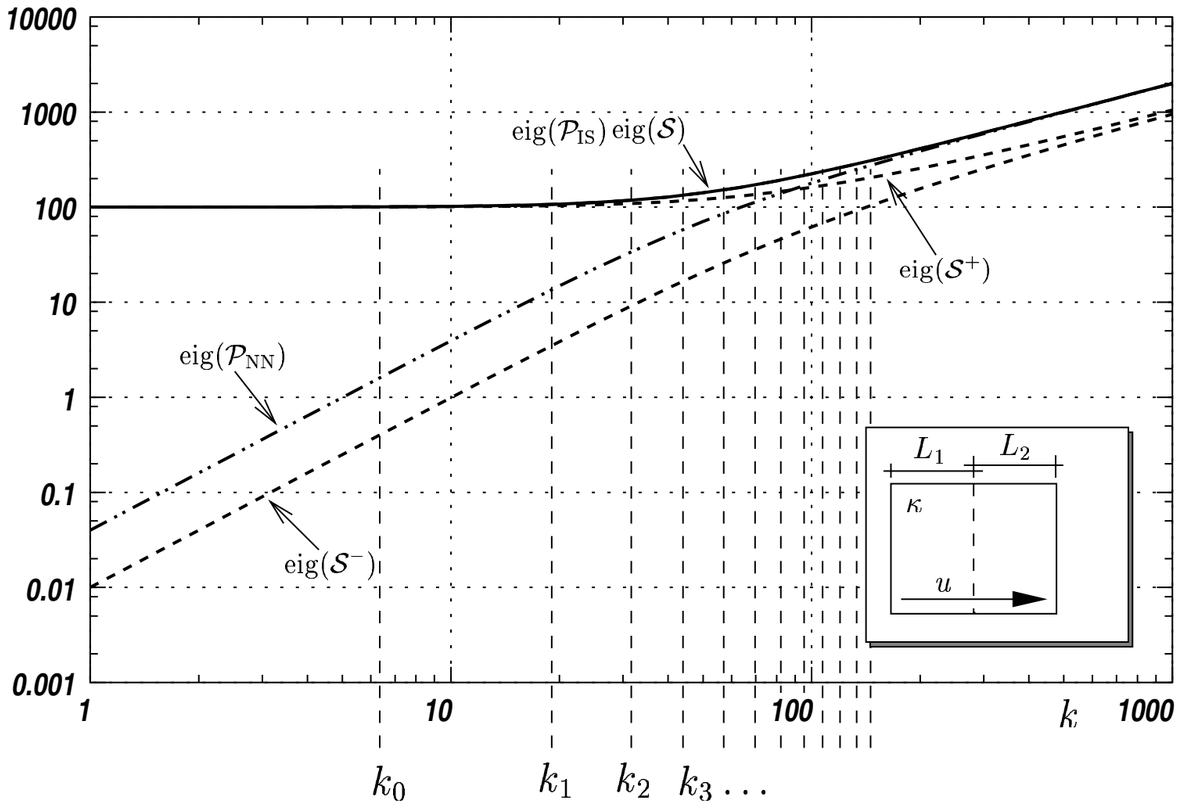
- **For high-frequency modes (high eigenvalues) the eigenfunctions concentrate near the interface.**
- **ISP: Solve the problem in a narrow strip around the interface**







Strongly advective case (Pe=50)



Condition number, 50x50 mesh, Pe=50

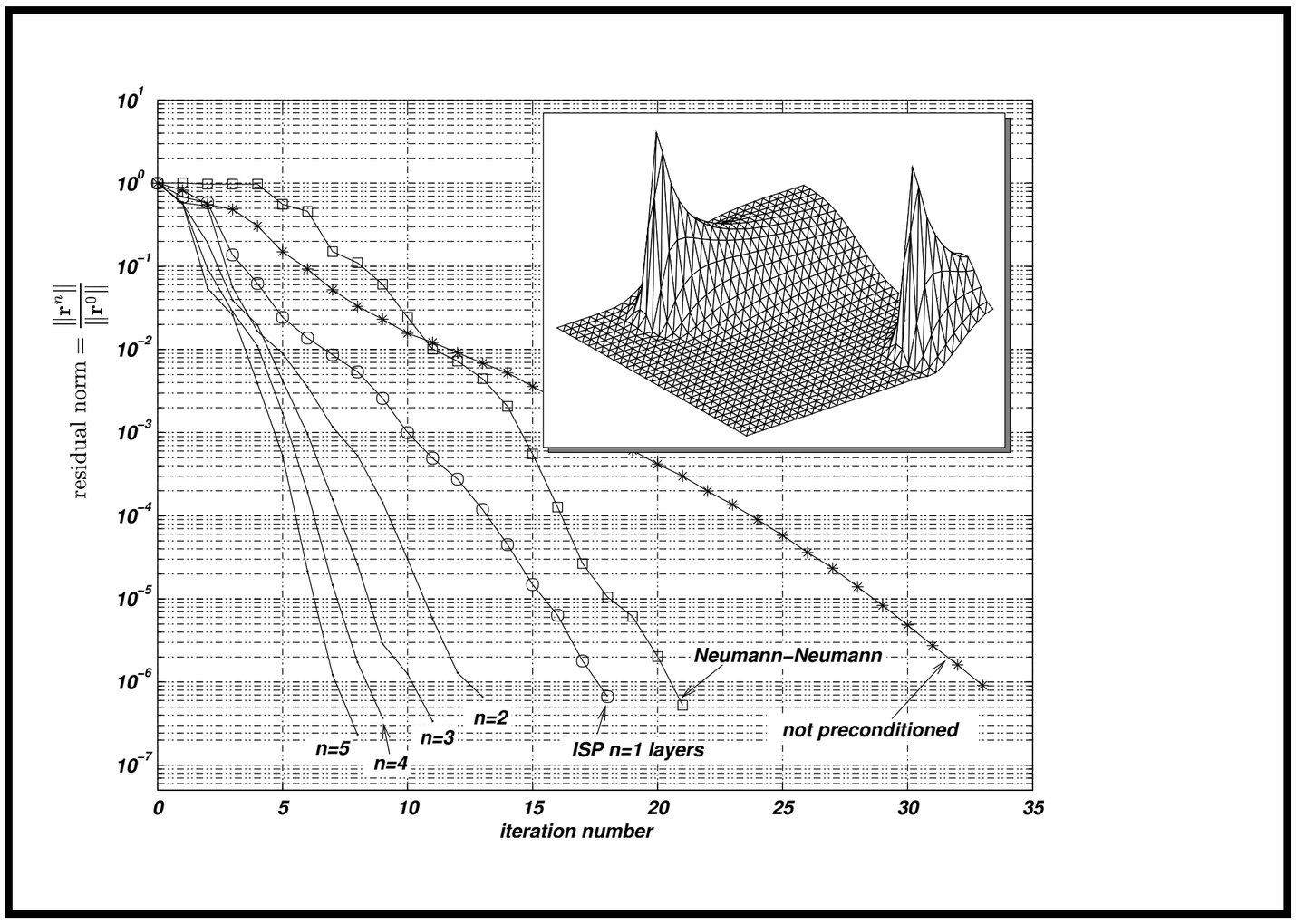
u	$\mathrm{cond}(\mathcal{S})$	$\mathrm{cond}(\mathcal{P}_{\mathrm{NN}}^{-1}\mathcal{S})$	$\mathrm{cond}(\mathcal{P}_{\mathrm{IS}}^{-1}\mathcal{S})$
0	**41.00**	**1.00**	**4.92**
1	**40.86**	**1.02**	**4.88**
10	**23.81**	**3.44**	**2.92**
50	**5.62**	**64.20**	**1.08**

Table 1: Condition number for the Stekhlov operator and several preconditioners for a mesh of 50×50 elements.

Condition number, 100x100 mesh, Pe=50

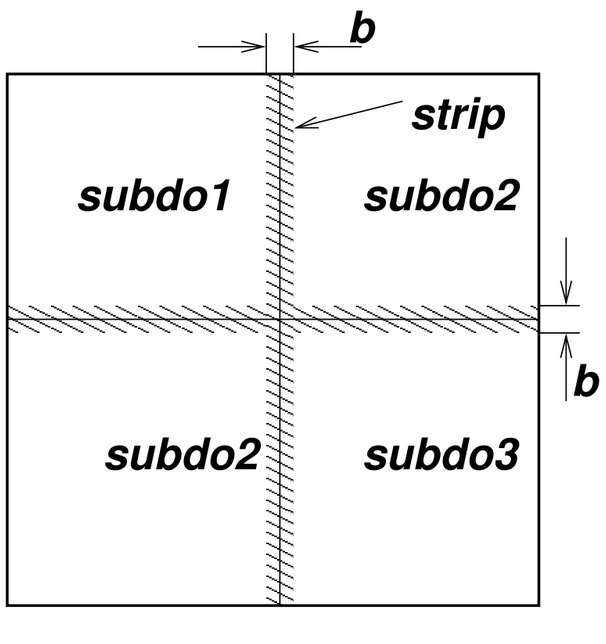
u	$\mathrm{cond}(\mathcal{S})$	$\mathrm{cond}(\mathcal{P}_{\mathrm{NN}}^{-1}\mathcal{S})$	$\mathrm{cond}(\mathcal{P}_{\mathrm{IS}}^{-1}\mathcal{S})$
0	**88.50**	**1.00**	**4.92**
1	**81.80**	**1.02**	**4.88**
10	**47.63**	**3.44**	**2.92**
50	**11.23**	**64.20**	**1.08**

Table 2: Condition number for the Stekhlov operator and several preconditioners for a mesh of 100×100 elements.


residual norm = $\frac{\|\mathbf{r}^n\|}{\|\mathbf{r}^0\|}$
iteration number
Neumann–Neumann
not preconditioned
ISP n=1 layers
n=2
n=3
n=4
n=5

ISP features

- **Better than NN for strong advective case ☺ (NN can improve splitting, but here splitting should depend on wave-length)**
- **No floating subdomains.**
- **Cost depends on strip width. Good conditioning can be obtained with thin strips (≪NN) ☺**
- **Solution of strip problem should be done iteratively. Two possibilities: communicates (weakly coupled) or doesn't communicate (diagonal dominant).**



ISP Implementation

- **ISP matrix is *"global"* (has elements in all processors): can't be inverted with a direct solver.**
- **Iterative solution can't use a CG/GMRES solver (can't nest CG/GMRES inside CG/GMRES)**
- **Use either preconditioned Richardson iteration or disconnect (i.e. modify) ISP matrix in some way in order to get a more disconnected matrix and use a direct method.**
- **Only preconditioned Richardson iteration is currently implemented in PETSc-FEM.**

Cubic cavity

- **Use N^3 regular hexahedral mesh or $5N^3$ tetra mesh.**
- **Very dense connectivity (high LU band-width, strong 3D).**
- **Covers many real flow features (separation, boundary layers...)**
- **For N=30 (135Ktet), using 128 subdomains, 50% savings in CPU time, 25% savings in RAM.**

N=30 cubic cavity stat.

nlay	iters
0	340
1	148
2	38

Conclusions

- **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.**
- **DDM/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).**
- **Interface strip preconditioner improves convergence, specially for advection dominated problems or floating subdomains.**