Adaptive Coarse Space for Saddle Point Problem

Frédéric Nataf and Pierre-Henri Tournier

Laboratory J.L. Lions (LJLL), CNRS, Alpines Inria and Sorbonne University

---

Workshop Excalibur, November 2022, Paris

This work was given access to the HPC resources of OCCIGEN@CINES under the allocation 2021-067730 granted by GENCI.

Outline

Outline

Saddle Point Problem

Solve

$$\mathcal{A}\begin{pmatrix} \boldsymbol{u}_h \\ p_h \end{pmatrix} = \begin{pmatrix} \boldsymbol{F}_h \\ G_h \end{pmatrix} \text{ with } \mathcal{A} := \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix}.$$

Pervasive in scientific computing:

- (nearly) incompressible fluids or solids $\Rightarrow$ pressure formulation is usually mandatory.
- Coupled problems, A contains two physics and B the coupling conditions and $C = 0$.
- Multi Point Constraints (MPC) $\Rightarrow$ Lagrange multipliers.

For small enough problems, direct solvers are the method of choice (MUMPS, PARDISO, SUPERLU, $\ldots$)

Comparison with a Direct solver *MUMPS* on a steel-rubber 3D beam

Timings are in seconds. OOM means: Out Of available Memory

		MUMPS			DD saddle point solver			
n	#cores	setup	solve	total	setup	#It	gmres	total
134 000	16	7.1	0.1	7.2	27.1	18	19.7	46.8
1 058 000	32	85.7	0.8	86.5	166.2	20	137.2	303.4
1 058 000	65	71.0	0.6	71.6	91.0	21	77.1	168.1
1 058 000	131	63.2	0.5	63.7	59.7	24	49.7	109.4
3 505 000	55	477.8	3.7	481.5	404.1	24	430.1	834.2
3 505 000	110	392.3	2.3	394.6	242.5	23	212.8	455.3
3 505 000	221	387.0	2.1	389.1	134.8	23	109.4	244.2
3 505 000	442	453.9	2.2	456.1	88.2	24	68.6	156.8
8 235 000	262	OOM	/	/	278.5	25	264.3	542.8
8 235 000	525	1622.1	6.1	1628.2	172.1	24	136.0	308.1
8 235 000	1050	1994.3	7.4	2001.7	136.5	25	99.7	236.2

Maximum problem size with direct solver is around 10 million unknowns.

The GenEO domain decomposition solver introduced here will solve a problem with 1 billion unknowns.

Iterative Solvers

Difficulty: Matrix $\mathcal{A}$ is symmetric but not positive. If it is made positive, symmetry is lost $\Rightarrow$ issue for iterative solvers.

$$\mathcal{A} := \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} .$$

Algebraic multigrid and Domain Decomposition solvers:

As problems get large, penalization and augmented Lagrangian techniques may enhance convergence but at the expense of approximation errors and round-off error issues.

For saddle point problems with 3D nearly incompressible elasticity and arbitrary high heterogeneities, existing iterative solvers seem not be so usable.

Here, we propose an
Extension of the GenEO DDM to saddle point problems.

Outline

(Recall) An introduction to DDM I

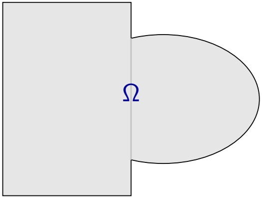
Consider the discretized Poisson problem: $Au = f \in \mathbb{R}^n$.
Given a decomposition of $[\![1; n]\!]$, $(\mathcal{N}_1, \mathcal{N}_2)$, define:

- the restriction operator R_i from $\mathbb{R}^{[\![1;n]\!]}$ into $\mathbb{R}^{\mathcal{N}_i}$,
- R_i^T as the extension by 0 from $\mathbb{R}^{\mathcal{N}_i}$ into $\mathbb{R}^{[\![1;n]\!]}$.

$u^m \longrightarrow u^{m+1}$ by solving concurrently:

$$u_1^{m+1} = u_1^m + A_1^{-1} R_1 (f - Au^m) \qquad u_2^{m+1} = u_2^m + A_2^{-1} R_2 (f - Au^m)$$

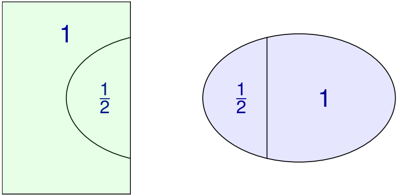
where $u_i^m = R_i u^m$ and $A_i := R_i A R_i^T$.



We have effectively *divided*, but we have yet to *conquer*.

Duplicated unknowns coupled via a *partition of unity*:

$$I = \sum_{i=1}^{N} R_i^T D_i R_i.$$



Then, $u^{m+1} = \sum_{i=1}^{N} R_i^T D_i u_i^{m+1}$.

$$M_{RAS}^{-1} = \sum_{i=1}^{N} R_i^T D_i A_i^{-1} R_i$$

+ Krylov acceleration $\Rightarrow$ RAS algorithm (Cai & Sarkis, 1999)

ASM: a symmetrized version of RAS

$$M_{RAS}^{-1} := \sum_{i=1}^{N} R_i^T D_i A_i^{-1} R_i .$$

A symmetrized version: Additive Schwarz Method (ASM),

$$M_{ASM}^{-1} := \sum_{i=1}^{N} R_i^T A_i^{-1} R_i \quad (1)$$

is used as a preconditioner for the conjugate gradient (CG) method.

Although RAS is more efficient, ASM is amenable to condition number estimates.

Chronological curiosity: First paper on Additive Schwarz dates back to 1989 whereas RAS paper was published in 1998

Adding a coarse space

One level methods are not scalable: $M_{ASM}^{-1} := \sum_{i=1}^{N} R_i^T A_i^{-1} R_i$.

We add a coarse space correction (*aka* second level). Let V_H be the coarse space and Z be a basis, $V_H = \operatorname{span} Z$, writing $R_0 = Z^T$ we define the two level preconditioner as:

$$M_{ASM,2}^{-1} := R_0^T (R_0 A R_0^T)^{-1} R_0 + \sum_{i=1}^{N} R_i^T A_i^{-1} R_i .$$

The Nicolaides approach (1987) is to use the near-kernel of the local operators to build the coarse space:

$$R_0^T Z := (R_i^T D_i R_i \mathbf{1})_{1 \le i \le N} ,$$

where D_i are chosen so that we have a partition of unity: $\sum_{i=1}^{N} R_i^T D_i R_i = Id$. Key notion: Stable splitting (J. Xu, 1989)

Theoretical convergence result

Theorem (Widlund, Dryija)

Let $M_{ASM,2}^{-1}$ be the two-level additive Schwarz method:

$$\kappa(M_{ASM,2}^{-1} A) \leq C \left(1 + \frac{H}{\delta}\right)$$

where δ is the size of the overlap between the subdomains and H the subdomain size.

This does indeed work very well

Number of subdomains	8	16	32	64
ASM	18	35	66	128
ASM + Nicolaides	20	27	28	27

Fails for highly heterogeneous problems
You need a larger and adaptive coarse space

Introduction to GenEO

Adaptive Coarse space for highly heterogeneous Darcy and (compressible) elasticity problems:
A **Gen**eralized **E**igenvalue problem in the **O**verlap per subdomain:

Find $V_{j,k} \in \mathbb{R}^{\mathcal{N}_j}$ and $\lambda_{j,k} \geq 0$:

$$D_j R_j A R_j^T D_j V_{j,k} = \lambda_{j,k} A_j^{Neu} V_{j,k}$$

In the two-level ASM, let τ be a user chosen parameter:
Choose eigenvectors $\lambda_{j,k} \geq \tau$ per subdomain:

$$Z := \left(R_j^T D_j V_{j,k}\right)_{\lambda_{j,k} \geq \tau}^{j=1,\ldots,N}$$

This automatically includes Nicolaides CS made of Zero Energy Modes.

Adaptive Coarse space for highly heterogeneous Darcy and (compressible) elasticity problems:
A **Gen**eralized **E**igenvalue problem in the **O**verlap per subdomain:

Find $V_{j,k} \in \mathbb{R}^{\mathcal{N}_j}$ and $\lambda_{j,k} \geq 0$:

$$D_j\, R_j A R_j^T\, D_j\, V_{j,k} \;=\; \lambda_{j,k}\, A_j^{Neu}\, V_{j,k}$$

In the two-level ASM, let τ be a user chosen parameter:
Choose eigenvectors $\lambda_{j,k} \geq \tau$ per subdomain:

$$Z \;:=\; \left(R_j^T\, D_j\, V_{j,k}\right)_{\lambda_{j,k} \geq \tau}^{j=1,\ldots,N}$$

This automatically includes Nicolaides CS made of Zero

Energy Modes.

Two technical assumptions.

Theorem (Spillane, Dolean, Hauret, N., Pechstein, Scheichl (Num. Math. 2013))

If for all j: $0 < \lambda_{j,m_{j+1}} < \infty$:

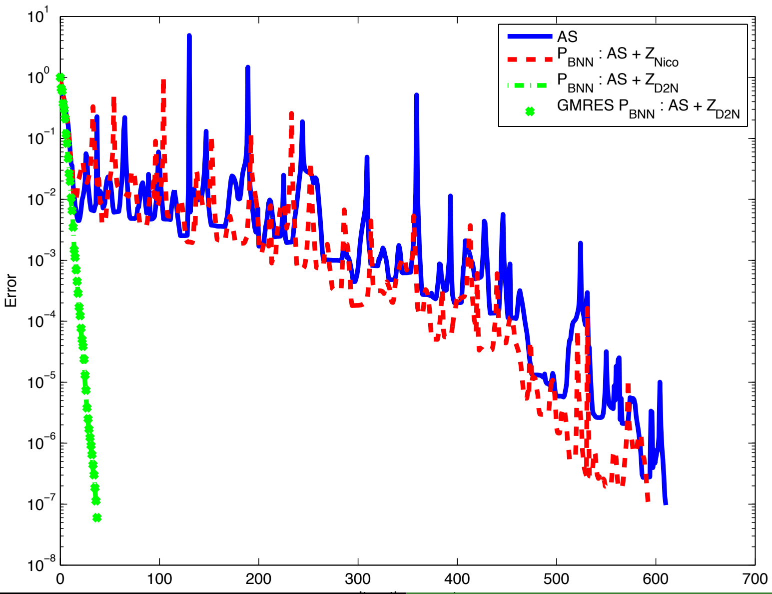
$$\kappa(M_{ASM,2}^{-1}A) \;\leq\; (1+k_0)\Big[2 + k_0\,(2k_0+1)\Big(1+\tau\Big)\Big]$$

Possible criterion for picking τ: (used in our Numerics)

$$\tau := \min_{j=1,\ldots,N} \frac{H_j}{\delta_j}$$

$H_j \ldots$ subdomain diameter, $\delta_j \ldots$ overlap

Convergence on a Highly Heterogeneous diffusion problem



Outline

Extension of GenEO to Saddle Point problem

Preconditioning $\mathcal{A}$ (e.g. Stokes, Nearly incompressible elasticity):

$$\mathcal{A} := \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} .$$

is equivalent to preconditioning A and $S := C + BA^{-1}B^T$.
Starting with a Schwarz preconditioner $A^{-1} \approx M_{ASM2}^{-1}$ as above, we have

$$S \approx C + BM_{ASM2}^{-1}B^T \approx S_0 + \underbrace{\sum_{i=1}^{N} \tilde{R}_i^T (\tilde{C}_i + \tilde{B}_i \, (R_i A R_i^T)^{-1} \, \tilde{B}_i^T) \tilde{R}_i}_{S_1} ,$$

where $S_0 := B\, Z_{GenEO} \, (Z_{GenEO}^T \, A \, Z_{GenEO})^{-1} \, Z_{GenEO}^T \, B^T$.
The operator S_1 is dense and has to be preconditioned.

Extension of GenEO to Saddle Point problem

But as a sum of local Schur complements, S_1 can be preconditioned by a Neumann-Neumann preconditioner

$$M_{S_1,\text{one level}}^{-1} := \sum_{i=1}^{N} \tilde{R}_i^T \tilde{D}_i \, (\tilde{C}_i + \tilde{B}_i \, (R_i A R_i^T)^{-1} \, \tilde{B}_i^T)^\dagger \, \tilde{D}_i \tilde{R}_i \, .$$

made scalable and robust with a GenEO type correction (see N., Num. Math., 2020):

$$M_{S_1}^{-1} := Z_{S_1} \, (Z_{S_1}^T S_1 Z_{S_1})^{-1} \, Z_{S_1}^T + \left(\sum_{i=1}^{N} \tilde{R}_i^T \tilde{D}_i \, (I_d - \xi_i)(\tilde{C}_i + \tilde{B}_i \, (R_i A R_i^T)^{-1} \, \tilde{B}_i^T)^\dagger \, (I_d - \xi_i^T) \tilde{D}_i \tilde{R}_i \right) \, .$$

where Z_{S_1} is populated with weighted local eigenvectors corresponding to the largest eigenvalues of the following GEVP:

$$\tilde{D}_i \tilde{R}_i S_1 \tilde{R}_i^T \tilde{D}_i \, \tilde{\mathbf{P}}_{ik} = \mu_{ik} (\tilde{C}_i + \tilde{B}_i \, (R_i A R_i^T)^{-1} \, \tilde{B}_i) \, \tilde{\mathbf{P}}_{ik} \, , \qquad (2)$$

and ξ_i denotes an orthogonal projection on the local contribution of the subdomain to the coarse space.

Define N_S^{-1} a spectrally equivalent preconditioner to S:

$$N_S := S_0 + M_{S_1} \, .$$

The application of the preconditioner N_S^{-1} consists in solving:

$$N_S \, \mathbf{P} = \mathbf{G} \, ,$$

by a Krylov solver with $M_{S_1}^{-1}$ as a preconditioner.

Saddle point algorithm in three solves:

INPUT: $\begin{pmatrix} \mathbf{F}_U \\ \mathbf{F}_P \end{pmatrix} \in \mathbb{R}^{n+m}$ OUTPUT: $\begin{pmatrix} \mathbf{U} \\ \mathbf{P} \end{pmatrix}$ the solution.

1. Solve $A\mathbf{G}_U = \mathbf{F}_U$ by a PCG with M_A^{-1} as a preconditioner
2. Compute $\mathbf{G}_P := \mathbf{F}_P - B\,\mathbf{G}_U$
3. Solve $S\,\mathbf{P} := (C + BA^{-1}B^T)\mathbf{P} = -\mathbf{G}_P$ by a PCG with N_S^{-1} as a preconditioner (nested loops).
4. Compute $\mathbf{G}_U := \mathbf{F}_U - B^T\mathbf{P}$
5. Solve $A\mathbf{U} = \mathbf{G}_U$ by a PCG with M_A^{-1} as a preconditioner

Outline

Nearly incompressible elasticity

The mechanical properties of a solid are characterized by its elastic energy:

$$\int_\Omega 2\,\mu\,\underline{\underline{\varepsilon}}(\boldsymbol{u}) : \underline{\underline{\varepsilon}}(\boldsymbol{u}) + \lambda\,|\mathrm{div}\,(\boldsymbol{u})|^2$$

where the Lamé coefficients λ and μ are defined in terms of the Young modulus E and Poisson ratio ν:

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \;\text{ and }\; \mu = \frac{E}{2(1+\nu)}\,,$$

As ν is close to $1/2^-$, $\lambda \to \infty$ so that $\mathrm{div}(u) \to 0$, but the pressure p:

$$p := \lambda\,\mathrm{div}\,(\boldsymbol{u}) \to p_{incompressibility}$$

and has thus to be introduced for stability, e.g. $\nu_{rubber} = 0.4999$.

Saddle point system

The resulting discretized variational formulation reads:

$$\begin{cases} \int_\Omega 2\,\mu\,\underline{\underline{\varepsilon}}(\boldsymbol{u}_h) : \underline{\underline{\varepsilon}}(\boldsymbol{v}_h) dx & - \int_\Omega p_h \,\mathrm{div}\,(\boldsymbol{v}_h) dx = \int_\Omega \boldsymbol{f} \boldsymbol{v}_h dx \\ \\ - \int_\Omega \mathrm{div}\,(\boldsymbol{u}_h) q_h dx & - \int_\Omega \frac{1}{\lambda} p_h q_h = 0. \end{cases} \tag{3}$$

where we take the lowest order Taylor-Hood finite element $C0P2 - C0P1$ so that the pressure p_h is continuous. In matrix form we have:

$$\begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_h \\ p_h \end{pmatrix} = \begin{pmatrix} \boldsymbol{F}_h \\ 0 \end{pmatrix} .$$

with an arbitrary domain decomposition .

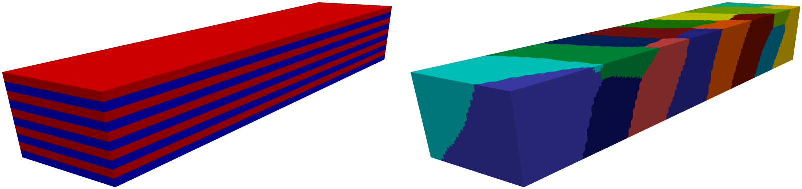


Figure: Heterogeneous beam of rubber and steel. Coefficient distribution (left) and mesh partitioning by the automatic graph partitioner *Metis* (right).

Rubber is nearly incompressible $\nu_{rubber} = 0.4999$ and soft $E_{rubber} = 0.01\,GPa$ whereas steel is compressible $\nu_{steel} = 0.35$ and hard $E_{steel} = 200.GPa$.

#cores	n	$dim(V_0)$	$dim(\tilde{W}_0)$	setup(s)	#It	gmres(s)	total(s)	#It N_S^{-1}
262	15 987 380	5 383	3 319	710.7	24	631.6	1342.3	11
525	27 545 495	9 959	2 669	526.6	21	519.5	1046.1	12
1 050	64 982 431	17 837	4 587	675.2	22	665.9	1341.1	11
2 100	126 569 042	32 361	7 995	689.2	25	733.8	1423.0	10
4 200	218 337 384	59 704	13 912	593.0	27	705.4	1298.4	10
8 400	515 921 881	141 421	25 949	735.8	32	1152.5	1888.3	10
16 800	1 006 250 208	260 348	41 341	819.2	29	1717.9	2537.1	12

Table: Weak scaling experiment.

#cores	n	$dim(V_0)$	$dim(\tilde{W}_0)$	setup(s)	#It	gmres(s)	total(s)	#It N_S^{-1}
525	27 545 495	9 959	2 669	526.6	21	519.5	1046.1	12
1 050	27 545 495	15 078	4 082	265.7	21	224.7	490.4	11
2 100	27 545 495	23 172	6 453	168.8	23	131.1	299.9	10
4 200	27 545 495	37 768	11 152	103.8	23	91.3	195.1	9

Table: Strong scaling experiment.

Adaptivity of the Coarse Space to the problem

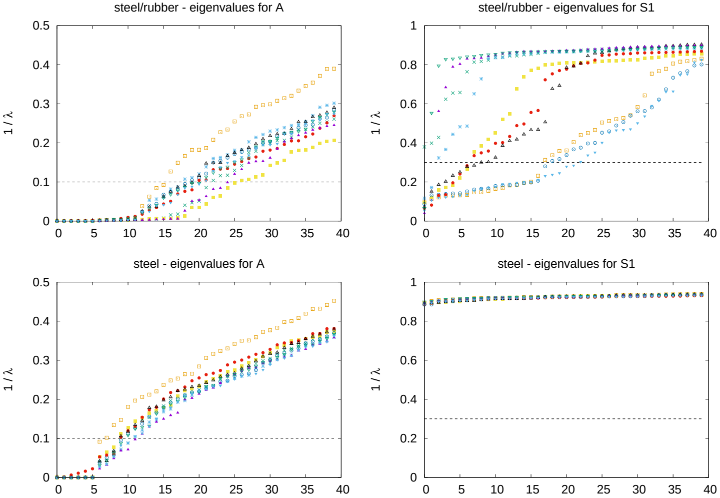


Figure: Top: Steel/Rubber beam. Bottom: Steel only beam. Inverse of the eigenvalues of the local GenEO eigenvalue problems for both coarse spaces, V_0 for A (left) and $\tilde{W}_0$ for S_1 (right)

Comparison with AMG *GAMG (PETSc)*

Comparisons on the velocity (only) formulation since we were unable to run GAMG on the saddle point formulation.

525 cores	GAMG		DD solver				
ν	#It	total(s)	$dim(V_0)$	setup(s)	#It	gmres(s)	total(s)
0.48	56	25.5	41 766	60.4	18	5.0	65.4
0.485	60	26.1	41 984	60.9	20	5.3	66.2
0.49	116	33.3	42 000	60.4	23	5.9	66.3
0.495	$>$2000	/	42 000	60.4	32	7.6	68.1
0.499	$>$2000	/	42 000	60.6	95	20.3	81.0

Table: GAMG (PETSc) versus standard GenEO for a homogeneous beam discretized with 7.9 million unknowns.

As ν gets close to 0.5, GAMG fails to compute a solution.

Algorithm assessment on Stokes computations

#cores	n	$dim(V_0)$	$dim(\tilde{W}_0)$	setup(s)	#It	gmres(s)	total(s)	#It N_S^{-1}
4	717 837	93	4	420.0	11	107.2	527.2	6
8	717 837	151	8	197.9	11	56.2	254.1	6
16	717 838	267	16	115.2	12	35.6	150.8	7
32	717 842	420	36	71.5	13	21.9	93.4	8
64	717 838	616	65	44.0	14	15.2	59.2	9
8	2 867 499	327	8	792.7	11	236.5	1029.2	7
16	2 867 499	577	16	371.6	12	148.2	519.8	9
32	2 867 499	877	32	291.3	12	86.8	378.1	10
64	2 867 503	1 306	66	164.9	13	55.9	220.8	11
128	2 867 503	1 985	133	118.6	13	41.4	160.0	13
8	11 462 307	606	8	3365.4	11	1146.3	4511.7	8
16	11 462 307	1 133	16	1753.6	11	640.4	2394.0	11
32	11 462 307	1 827	32	1099.9	12	404.8	1504.7	13
64	11 462 307	2 760	64	628.0	12	213.9	841.9	13
128	11 462 307	4 124	134	438.5	13	162.1	600.6	15

Table: 2D Stokes – Air bubbles in Water.

Going further: Comparisons for 3D flows with multigrid solvers, Cahouet-Chabard method on this or other problems, . . .

Outline

Domain Specific Language for finite element method

Why use a DS(E)L (FreeFem++, Feel++, Dune, Fenics or Firedrake) instead of `C/C++/Fortran/..` ?

- performances close to low-level language implementation,
- hard to beat something as simple as:

```
varf a(u, v) = int3d(mesh)([dx(u), dy(u), dz(u)]' * [dx(v), dy(v), dz(v)])
               - int3d(mesh)(f * v) + on(boundary_mesh)(u = 0) ,
```

- access to the variational formulation is then natural and that's what we need.

A few facts

- 1987: First version by O. Pironneau written in Pascal on Macintosh
- Since 1992: the main developer is Frédéric Hecht

Some FreeFem features

- Integrates many state of the art libraries.
- Automatic Mesh refinement native in 2d and via the plugin "Mmg" (Frey at al.) in 3D
- Interpolate between different finite element spaces defined on different meshes, clouds of points to mesh
- Extensible via dynamic plugins
- NEW: coupled FEM/BEM thanks to X. Claeys with $\mathcal{H}$-matrix compression thanks to P. Marchand
- Interface to MPI, PETSc
- parallel version runs on Linux, Windows, Mac since 2017
- Docker on Qarnot and Rescale cloud computing platform
- Web browser (Javascript port thanks to A. Le Hyaric (LJLL))

Data Distribution for parallel computing

Domain Partition via Metis or Scotch interface

Overlap is done by FreeFem based on the mesh connectivity

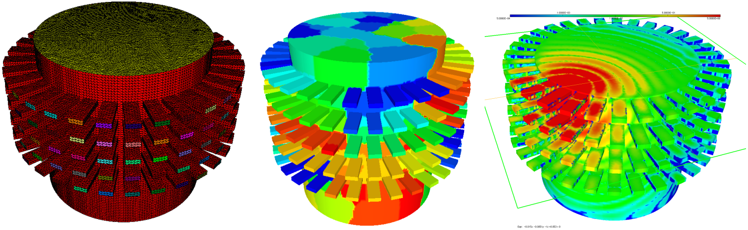


Figure: Electromagnetic chamber – Harmonic Maxwell DD solver with Robin or PML interface conditions

Conclusion and Prospects

- Iterative solver for saddle point problem with highly heterogeneous coefficients that works for linear elasticity, Stokes systems
- Reproducibility and availability to FreeFem users via https://github.com/FreeFem/FreeFem-sources/blob/develop/examples/ffddm/elasticity_saddlepoint.edp
- Preprint available on HAL:

F Nataf and P.-H. Tournier, "A GenEO Domain Decomposition method for Saddle Point problems", https://hal.archives-ouvertes.fr/view/index/docid/3450974 , HAL Archive.

- Prospects
  - More than 2-level
  - Inclusion into HPDDM for PETSc users
  - Multiscale finite element for saddle point problem

Acknowledgements

Special thanks to GENCI for their support:
Developments have been developed, benchmarked and used on national HPC resources at IDRIS and at CINES under Allocations 20XX-067730.

THANK YOU
FOR YOUR ATTENTION!

Acknowledgements

Special thanks to GENCI for their support:
Developments have been developed, benchmarked and used on national HPC resources at IDRIS and at CINES under Allocations 20XX-067730.

THANK YOU
FOR YOUR ATTENTION!