Extension of the GenEO DDM to Saddle Point problem

Frédéric Nataf and Pierre-Henri Tournier

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

13th FreeFEM Days, Paris 2021

This work was given access to the HPC resources of OCCIGEN@CINES under the allocation 2020-067730 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.
- Multi Point Constraints (MPC) $\Rightarrow$ Lagrange multipliers.

Penalization may bypasse the problem in some situations but at the expense of approximation errors and round-off error issues

Difficulty: Matrix $\mathcal{A}$ is symmetric but not positive. If it is made positive, symmetry is lost $\Rightarrow$ issue for iterative solvers.
For small enough problems, direct solvers are the method of choice (MUMPS, PARDISO, SUPERLU, $\ldots$)

Large Scale Problems

Many millions or some billions of dof's on hundreds or thousands of cores.
For Symmetric Positive Definite (SPD) problems even with high heterogeneities, both

- Algebraic Multigrid solvers (AMG)
- Domain Decomposition Methods (DDM)

are quite efficient: robust and fast.

For saddle point problems with arbitrary high heterogeneities, even when A is SPD and C is symmetric positive semidefinite, these iterative solvers are not 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: $Au = f \in \mathbb{R}^n$.

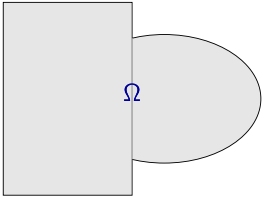
Given a decomposition of $[\![1;n]\!]$, $(\mathcal{N}_1, \mathcal{N}_2)$, define:

- 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 $A_i := R_i A R_i^T$.



(Recall) An introduction to DDM I

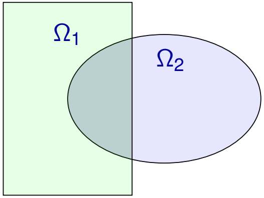
Consider the discretized Poisson problem: $Au = f \in \mathbb{R}^n$.
Given a decomposition of $[\![1;n]\!]$, $(\mathcal{N}_1, \mathcal{N}_2)$, define:

- 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 $A_i := R_i A R_i^T$.



(Recall) An introduction to DDM I

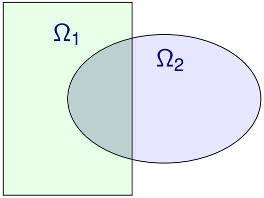
Consider the discretized Poisson problem: $Au = f \in \mathbb{R}^n$.
Given a decomposition of $[\![1;n]\!]$, $(\mathcal{N}_1, \mathcal{N}_2)$, define:

- 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 $A_i := R_i A R_i^T$.

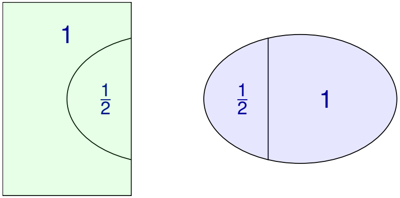


An introduction to DDM II

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, $u^{m+1} = \sum_{i=1}^{N} R_i^T D_i u_i^{m+1}$. $M_{RAS}^{-1} = \sum_{i=1}^{N} R_i^T D_i A_i^{-1} R_i$

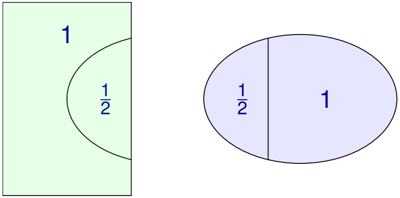
+ Krylov acceleration $\Rightarrow$ RAS algorithm (Cai & Sarkis, 1999)

An introduction to DDM II

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

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

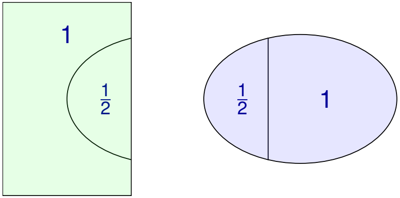
+ Krylov acceleration $\Rightarrow$ RAS algorithm (Cai & Sarkis, 1999)

An introduction to DDM II

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, $u^{m+1} = \sum_{i=1}^{N} R_i^T D_i u_i^{m+1}$. $\qquad M_{RAS}^{-1} = \sum_{i=1}^{N} R_i^T D_i A_i^{-1} R_i$

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

Adding a coarse space

One level methods are not scalable.
We add a coarse space correction (*aka* second level)
Let V_H be the coarse space and Z be a basis, $V_H = \mathrm{span}\, Z$, 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 kernel of the operator as a coarse space, this is the constant vectors, in local form this writes:

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

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

Strategy

Define an appropriate coarse space $V_{H2} = \mathrm{span}(Z_2)$ and use the framework previously introduced, writing $R_0 = Z_2^T$ the two level preconditioner is:

$$M_{ASM2}^{-1} := R_0^T (R_0 A R_0^T)^{-1} R_0 + \sum_{i=1}^{N} R_i^T A_i^{-1} R_i.$$

The coarse space must be

- Local (calculated on each subdomain) $\to$ parallel
- Adaptive (calculated automatically)
- Easy and cheap to compute
- Robust (must lead to an algorithm whose convergence is proven not to depend on the partition nor the jumps in coefficients)

Introduction to GenEO

Adaptive Coarse space for highly heterogeneous Darcy and (compressible) elasticity problems:
Geneo .EVP 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.

Introduction to GenEO

Adaptive Coarse space for highly heterogeneous Darcy and (compressible) elasticity problems:
Geneo .EVP 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.

Theory of GenEO

Two technical assumptions.

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

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

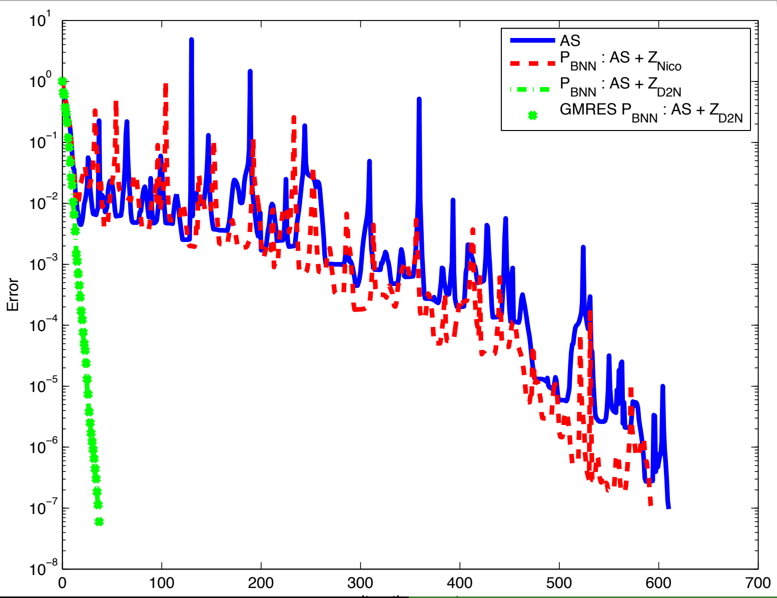
$$\kappa(M_{ASM,2}^{-1}A) \le (1+k_0)\Big[2 + k_0\,(2k_0+1)\Big(1+\tau\Big)\Big]$$

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

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

$H_j \dots$ subdomain diameter, $\delta_j \dots$ 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 $A^{-1} \approx M_{ASM2}^{-1}$ as above, we have

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

The operator M_{S_1} is dense and has to be preconditioned. But as a sum of local Schur complements, it can be preconditioned by a GenEO Neumann-Neumann type method:

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

Two Stage Algorithm

Define N_S a spectrally equivalent preconditioner to S:

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

The application of the preconditioner N_S 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 two stages:

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.
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 $1/2$, λ tends to infinity so that the solid is nearly incompressible, e.g. $\nu_{rubber} = 0.4999$.
The pressure p:

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

is stable in the incompressible limit and has thus to be introduced for stability.

Saddle point system

The resulting discretized variational formulation reads:

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

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:

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

Mechanical test case

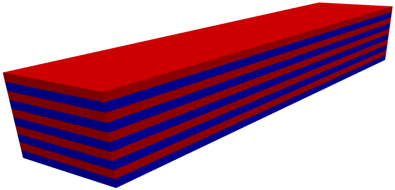
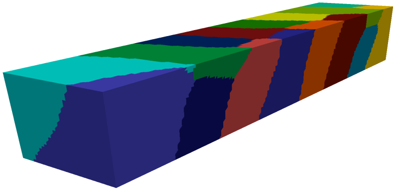



Figure: Heterogeneous beam composed of 10 alternating layers. Coefficient distribution (left) and mesh partitioning into 16 subdomains by the automatic graph partitioner *Metis* (right).

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 $E_{steel} = 200. GPa$.

Weak scalability

#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 for 3D heterogeneous elasticity: beam with 10 alternating layers of steel and rubber .

Our numerical results can be reproduced running the script `https://github.com/FreeFem/FreeFem-sources/blob/develop/examples/ffddm/elasticity_saddlepoint.edp` available in the FreeFem distribution starting from version 4.10.

Comparison with a Direct solver

		MUMPS			DD saddle point solver			
$n\,10^3$	#cores	setup(s)	solve(s)	total(s)	setup(s)	#It	gmres(s)	total(s)
134	16	7.1	0.1	7.2	27.1	18	19.7	46.8
1058	32	85.7	0.8	86.5	166.2	20	137.2	303.4
1058	65	71.0	0.6	71.6	91.0	21	77.1	168.1
1058	131	63.2	0.5	63.7	59.7	24	49.7	109.4
3505	55	477.8	3.7	481.5	404.1	24	430.1	834.2
3505	110	392.3	2.3	394.6	242.5	23	212.8	455.3
3505	221	387.0	2.1	389.1	134.8	23	109.4	244.2
3505	442	453.9	2.2	456.1	88.2	24	68.6	156.8
8235	262	OOM	/	/	278.5	25	264.3	542.8
8235	525	1622.1	6.1	1628.2	172.1	24	136.0	308.1
8235	1050	1994.3	7.4	2001.7	136.5	25	99.7	236.2

Table: Comparison with the parallel sparse direct solver *MUMPS* for 3D heterogeneous elasticity: beam with 10 alternating layers. Reported timings for four discretization levels while also varying the number of cores (OOM means the computation ran out of available memory).

Comparison with AMG

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

525 cores	GAMG		DD saddle point 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 the velocity formulation on the homogeneous beam discretized with 7.9 million unknowns.

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

Conclusion and Prospects

- Iterative solver for saddle point problem with highly heterogeneous coefficients that works for linear elasticity, Stokes (not shown here) systems
- Available to FreeFem users via `https://github.com/FreeFem/FreeFem-sources/blob/develop/examples/ffddm/elasticity_saddlepoint.edp`
- Prospects
  - Inclusion into HPDDM for PETSc users
  - Multilevel version
  - Black box version
  - Multiscale finite element for saddle point problem
  - 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.