A hierarchical preconditioner for Stokes and connected problems

M. Fortin, A. Elmaliki

October 23, 2007

The Problem

- Our goal is to solve large 3D incompressible elasticity problems. In practice, this will mean some form of the Mooney-Rivlin model. We will also want to introduce contact.
- We use tetrahedra for ease of generation and for mesh adaptation methods
- The choice of elements for 3D incompressible materials is limited.
  1. MINI
  2. SMALL
  3. Taylor -Hood $P_2 - P_1$.

What should we choose? Let us try to compare the cost of each of them.
To do so, let us consider a mesh of n^2 cubes subdivided into 6 tetrahedra. For n large we have approximately

- n^2 vertices,
- $6n2$ tetrahedra,
- $7n^2$ edges
- $12n^2$ faces.

Let us compare our elements

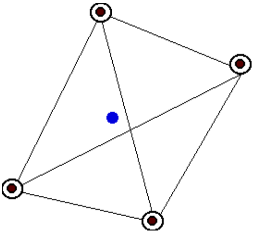


Figure: MINI

We have 3 n^2 displacement d.o.f. (+ 18 n^2 internal nodes)+ n^2 pressure d.o.f. For contact problems, MINI produces bad contact pressure.

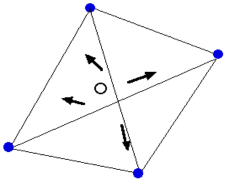


Figure: SMALL

We have 3 n^2 + 12 n^2 displacement d.o.f + 6 n^2 pressure d.o.f.for a total of 21 n^2

This is much for a first order element

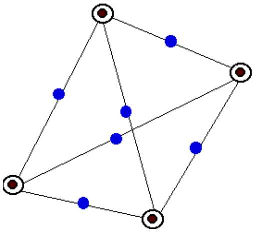


Figure: Taylor-Hood

We have 3 n^2 +3x7n^2 displacement + n^2 pressure d.o.f. for a total of 25 n^2.
But we have a second order element.

The Goal

We choose the Taylor-Hood element

- We want a fast and simple solver for the $P_2 - P_1$ Taylor-Hood element.
- The key will be the hierarchical basis for P_2
- The rest will be a pile of preconditiners and Krylov subspaces methods.

Hierarchical basis for P_2

The idea is simple.

- Instead of using the standard Lagrange basis, we use the basis of P_1, associated to vertices.
- We add the shape functions associated with edges. Their coefficient now represents a correction to the value of the linear approximation.
- This is a standard idea which has been employed in hierarchical error estimation.
- We can write

$$P_2 = P_1 \oplus C_2$$

For a standard elliptic problem,

$$a(u,v) - < \; f, v \; > \quad \forall v \in P_2,$$

the associated matrix can be written as

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \tag{1}$$

- Matrix A_{11} is the matrix for the P_1 element. It is about 7 times smaller than A
- Matrix A_{22} has an O(1) condition number.

If we have a good (approximate) solver for A_{11}, we can think of solving the global problem

$$Au = F$$

by a sequence of simpler ones

- Solve (approximately) $A_{11}u_1^{k+1} = F_1 - A_{12}u_2^k$
- Solve (approximately) $A_{22}u_2^{k+1} = F_2 - A_{21}u_1^{k+1}$ This can be seen as a block SOR method. solution in u_2 can be done through a few iterations of SOR or CG.
- We shall rather write this as a preconditioner. We also want to preserve symmetry.

Preconditioner form

To recover symmetry, we do a block SSOR sweep

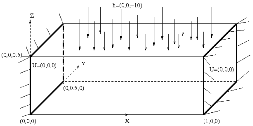
- Solve (approximately) $A_{11}\delta_0 u_1 = F_1 - A_{12}u_2^k - A_{11}u_1^k = r_1^k$
- Solve (approximately) $A_{22}\delta u_2 = F_2 - A_{21}u_1^k - A_{22}u_2^k - A_{21}\delta_0 u_1 = r_2^k - A_{21}\delta_0 u_1$
- Solve (approximately) $A_{11}\delta u_1 = r_1^k - A_{11}\delta_0 u_1 - A_{12}\delta u_2$

This can be fed to a standard CG method

Solving the P_1 part

- This depends on the size of the problem and your cleverness. We can think of,
  1. A direct solver if the problem is not too large.
  2. A multigrid solver if you are clever.
  3. A few iterations of an IC-CG,
  4. Or whatever...
- Our own choice was IC-CG, directly available in PetSc.
- The direct solver was used as a comparison.

A test problem



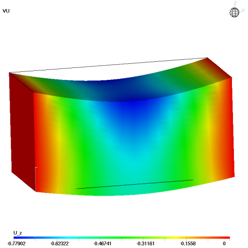


Figure: A simple elasticity problem

Some sketchy results

- When using the method as a solver: 27 iterations (42 seconds).
- As a preconditioner: 14 iterations (22 seconds).
- The same behaviour was obtained on other problems.

Saddle-Point Problems

- We now consider a problem of the standard mixed form

$$\begin{pmatrix} A & B^t \\ B & 0 \end{pmatrix} = \begin{pmatrix} F \\ G \end{pmatrix} \tag{2}$$

- This can be the Stokes problem but we shall also consider the case of contact problems.
- The standard preconditioner is based on the factorisation of the matrix

$$\begin{pmatrix} A & B^t \\ B & 0 \end{pmatrix} = \begin{pmatrix} A & 0 \\ B & -S \end{pmatrix} \begin{pmatrix} I & A^{-1}B^t \\ 0 & I \end{pmatrix} \tag{3}$$

- $S = BA^{-1}B^t$

To get a preconditioner

- Approximate A by $\hat{A}$
- Approximate S. For Stokes, the (lumped ?) mass matrix M is appropriate as S is an operator of order 0.
- If $\hat{A} = A$ this is Uzawa's Method.
- Changing A into $A_r = A + rB^tB$ improves the condition number of the dual but makes the primal problem bad.
- This is equivalent to the method of Arrow-Hurwicz. This attacks the saddle-point by alternating minimisation in u and maximisation in p

The method of Arrow-Hurwicz

1. Initialisation: Let p_k and u_k be given,
2. Compute

$$u_{k+1} = u_k + \rho_u \hat{A}^{-1}(F - Au_k - B^t p_k) = u_k - \rho_u \hat{A}^{-1} r_u \quad (4)$$

$$p_{k+1} = p_k + \rho_p M^{-1}(Bu_{k+1} - G) \quad (5)$$

This is not symmetric. To recover symmetry, we solve again in u. Experience shows that there is an optimal ratio ρ_p / ρ_u.

Arrow-Hurwicz-Preconditioner Form

- Let $r_u = Au + B^t p - F$ and $r_p = Bu - G$
- Solve,

$$
\begin{aligned}
\delta_0 u &= \hat{A}^{-1}(f - Au_k - B^t p_k) = -\rho_u \hat{A}^{-1} r_u \\
\delta p &= \alpha M^{-1}(r_p + B\delta_0 u) \\
\delta u &= -\hat{A}^{-1}(r_u - A\delta_0 u - B^t \delta p)
\end{aligned}
\tag{6}
$$

- We use αM, with $\alpha =?$

A numerical example

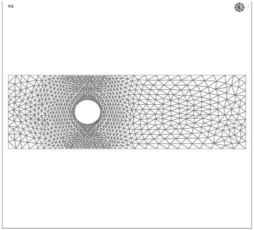


Figure: Cylinder

- Mesh 1(1959 elements) A-H+GCR:21 iteration(0.99 seconds), Uzawa: 18
- Mesh 2(7836 elements) A-H+GCR 20 iterations (4.1seconds) Uzawa:18 iterations (7.9 seconds)
- The number of iterations behaves well and also the time

A 3D example

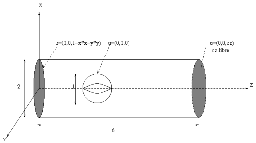


Figure: 3D obstacle

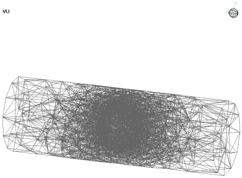


Figure: Mesh 1

Results

We introduced a penalty term in integral form. Obviously, r must remain small.

- On mesh 1 (9067 elements) Uzawa 32 iterations (273.18s) with $r = 0$, 21 iterations (243.22s) with $r = 5$
- A-H +GCR) 35 iterations (25.95) with $r = 0$, 25 iterations (18.41s) with $r = 5$
- On mesh 2, (72536 elements) 38 iterations (276.61s) with $r = 0$, 28 iterations (204.64s) with $r = 5$

Contact Problems

A simple case: an elastic body Ω in contact with a rigid surface S.

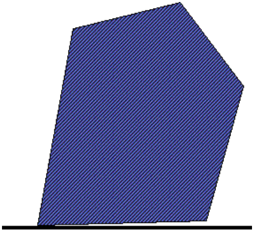


Figure: An elastic body on a rigid surface

- The precise formulation of the elastic part is irrelevant and we shall consider a generic case. In practice, we will have to deal with a non linear material.
- The real problem is 3D.

Sliding Contact

- We first consider the sliding case. We denote by $d(\cdot)$ the distance of Ω to S.
- $d(x)$ is computed by projecting the point x of Ω on S.

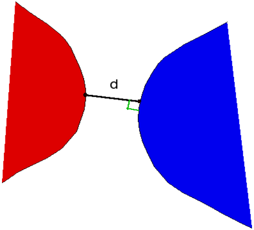


Figure: Distance

- $d(\cdot)$ is a non linear function.
- The constraint is $d(x) \geq 0$. The bodies must not interpenetrate.
- Thus if v is the displacement of Ω and $J(v)$ an elastic energy functional, we want to solve,

$$\inf_{d(v)\geq 0} J(v) \tag{7}$$

Contact Pressure

- We introduce a Lagrange multiplier for the constraint,

$$\inf_{v} \sup_{\lambda_n \geq 0} J(v) + < \lambda_n, d(v) > . \tag{8}$$

- The optimality conditions are then

$$< A(u)u - F, v > + < \lambda_n, v \cdot n >= 0 \ \forall v \tag{9}$$

$$< d(u), \mu > \quad \geq 0 \quad \forall \mu \geq 0. \tag{10}$$

- We have standard Kuhn-Tucker conditions :

$$\lambda_n \geq 0, \ d(u) \geq 0, < d(u), \lambda_n >= 0. \tag{11}$$

Linearised Problem

- This is a non linear inequality problem which we solve by a Newton's method (SQP).
- We temporarily fix the normal but this should also be linearised.
- Defining $g_0^n = d(u_0)$ for the initial configuration u_0, we get,

$$\begin{cases} < A'(u_0)\delta u, v > + < \lambda_n, v \cdot n > = < F - Au_0, v > \quad \forall v, \\ < g_0^n - \delta u \cdot n, \phi > \geq 0 \quad \forall \phi \geq 0. \end{cases}$$

We notice that this is an inequation problem for which we must use a suitable algorithm. Once it is solved, we update the configuration, compute a new gap and iterate to convergence

Conjugate Projected Gradient

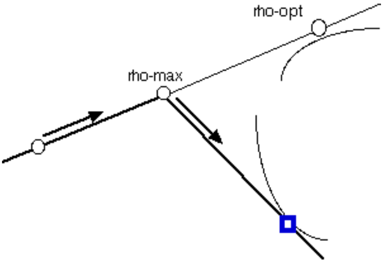


Figure: Conjugate projected gradient

Dual form

- In our applications, the CPG algorithm will be applied to a dual problem of the form,

$$\inf_{\lambda} (K\lambda, \lambda) - (BA^{-1}F, \lambda) - (G, \lambda), \tag{12}$$

where

$$K = BA^{-1}B^t \tag{13}$$

- The gradient in λ, at some point λ_k, is conveniently computed by solving

$$Au_k = F - B^t\lambda_k \tag{14}$$

and then computing, $g_k - Bu_k - G$.

CPG for Sliding Contact

- The gradient is $g = B_n u - G_n$ where the matrix B_n is defined as previously
- In the simplest form, the projected gradient is computed component wise

$$\begin{cases} Pg_i = g_i & \text{if} \quad \lambda_i > 0, \\ Pg_i = g_i & \text{if} \quad \lambda_i = 0 \quad \text{and} \quad g_i \geq 0, \\ Pg_i = 0 & \text{if} \quad \lambda_i = 0 \quad \text{and} \quad g_i < 0. \end{cases}$$

Remarks

- $K = BA^{-1}B^t$ is now a first order operator and convergence will depend on h but also on the shape of elements. With conjugation, we have $\sqrt{h}$ which is not so bad...
- Preconditioning with penalty is not very convenient for the contact points change while iterating, which would imply changing the matrix.
- It is not obvious to get a good approximation of $K = BA^{-1}B^t$. T
- The minimum is $S = M$, that is the mass matrix. Experience shows that this cures the dependency on the shape of elements. But obviously we still have a dependency on mesh size. Moreover, the projection becomes more difficult.

Projection of the gradient

- We suppose that M is some approximation to K. To project the gradient, we now have to solve

$$\inf_{d_i \geq 0,\, i \in AS} \frac{1}{2} < Md, d > - < g, d > \qquad (15)$$

- AS is the active set. We have $i \in AS$, If $\lambda_i = 0$
- This is local only if M is diagonal.
- For a non diagonal M, we may use a CPG. This is a small problem.

Arrow-Hurwicz with projection-Preconditioner Form

- Let $r_u = Au + B^t p - F$ and $r_p = Bu - G$
- Solve,

$$
\begin{aligned}
\delta_0 u &= \hat{A}^{-1}(f - Au_k - B^t p_k) = -\rho_u \hat{A}^{-1} r_u \\
\delta p &= \alpha P_M (r_p + B\delta_0 u) \\
\delta u &= -\hat{A}^{-1}(r_u - A\delta_0 u - B^t \delta p)
\end{aligned}
\tag{16}
$$

- We would need a ProjectedMinres. The solve and project strategy should be easy to implant. Anybody has experience with this?

The real problem : choosing M

- Any idea better than the mass matrix?
- Would an IC factorisation of the P_1 part be better?
- Even with the mass matrix, there should be a gain with respect to the present method.

Nested iterations

- This also contains a lot of nested iterations.
- Should we have the iteration on internal pressure at the same level as that for the contact pressure?

Frictional contact

The displacement method presented Monday relies on unconstrained or unilateral saddle-point problems, for instance

$$\begin{cases} < A'\phi, v > + \frac{1}{\epsilon} < C\phi_T, v_T >_\Gamma + < \gamma, v \cdot n >_\Gamma = < d^k, v \cdot n >_\Gamma \\ \phi \cdot n \leq 0, \\ \gamma \geq 0, \\ \gamma(\phi \cdot n) = 0. \end{cases} \tag{17}$$

Conclusion and Perspective

- This is a promising avenue for large scale industrial problems.
- There is room for new ideas for preconditioning.
- There is also another iteration on the threshold...