Preconditioned iterative solvers for large-scale saddle point systems arising in constrained optimization problems

A. El Maliki
Hassan II University, ENCG-Morocco

Rencontre scientifique: Algorithmes Stochastiques et Applications
Université de Poitiers, Laboratoire de Mathématiques et Applications **,**
26 Novembre 2020

Outline

Introduction & motivation

Preconditioning of Karush-Kuhn-Tucker (KKT) systems

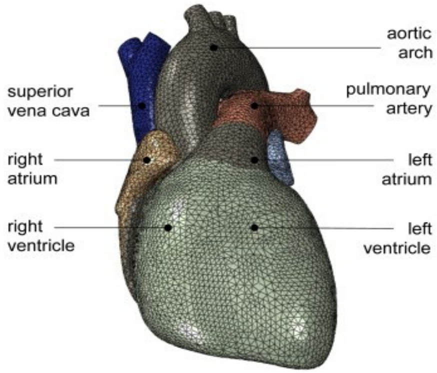
Primal-Dual Active-Set (PDAS) Method
(~ Semismooth Newton's method)

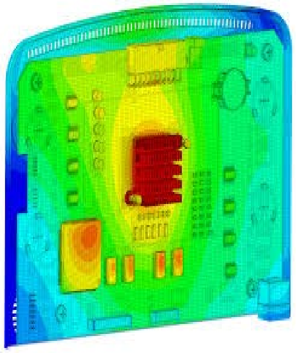
My Proposed Enhancements

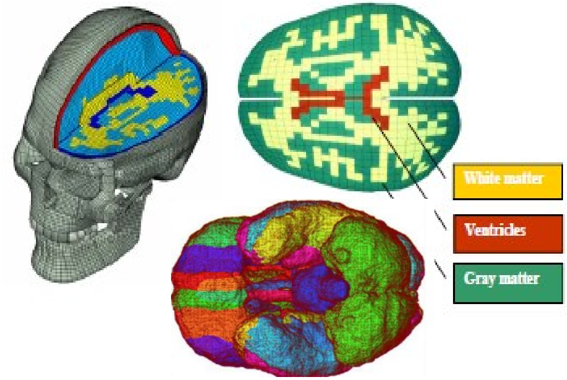
Examples: Finite Element approximation of nearly incompressible elasticity with contact constraints.

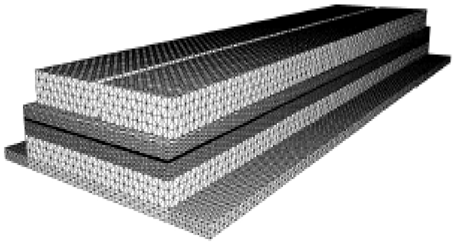
Summary

Introduction & motivation: Some applications

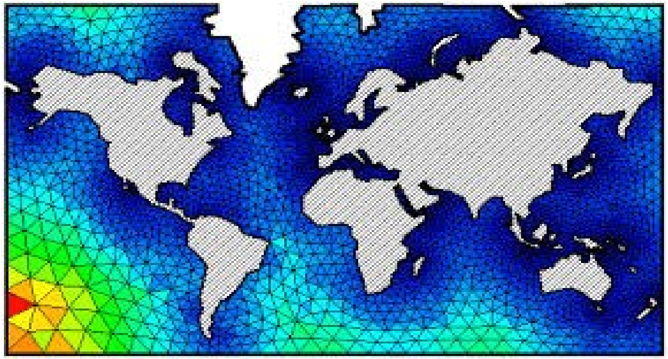


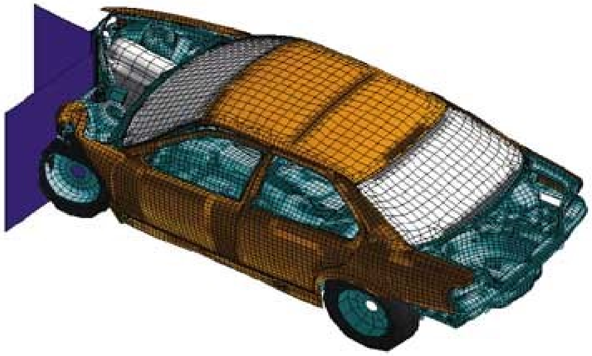


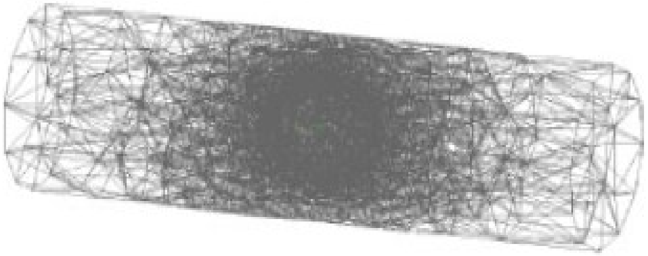


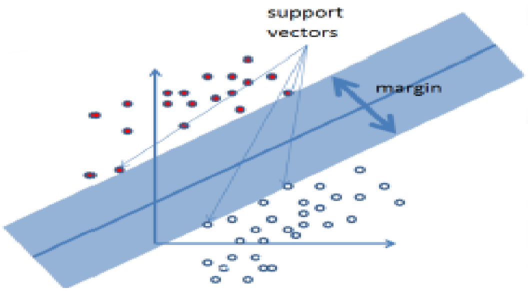


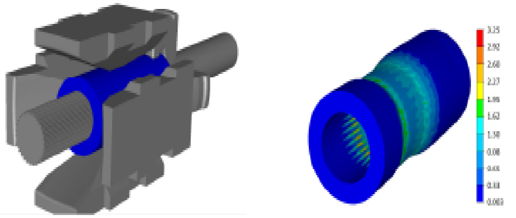
Electrolysis process











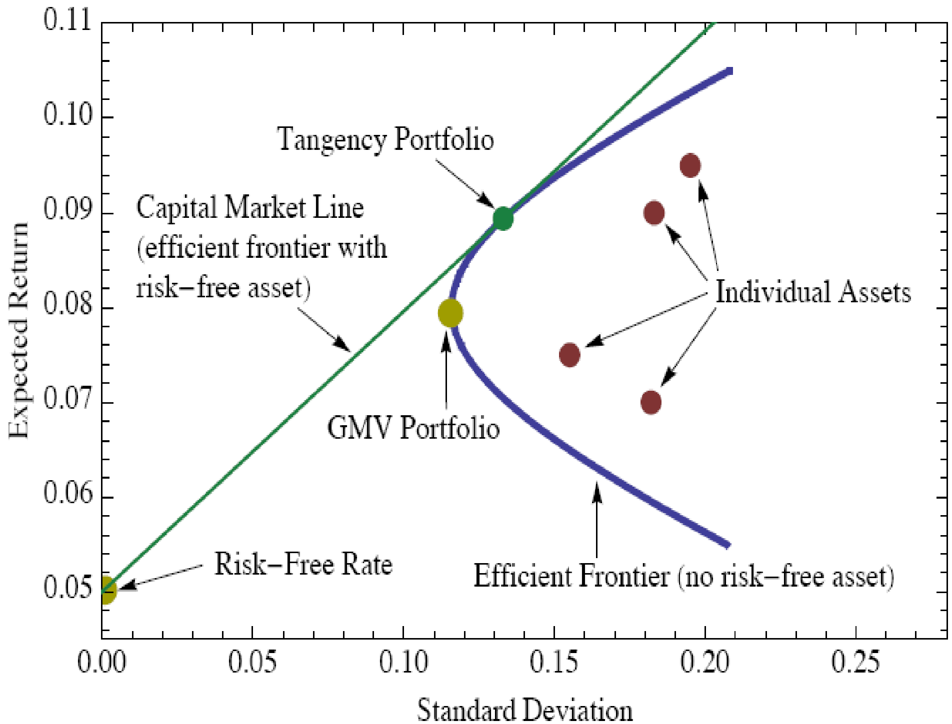
(a) Etat initial

(b) Déformation équivalente en fin de simulation

FIGURE 2.9 – Simulation de martelage rotatif

Some pictures come from WWW

Introduction & motivation: Some applications

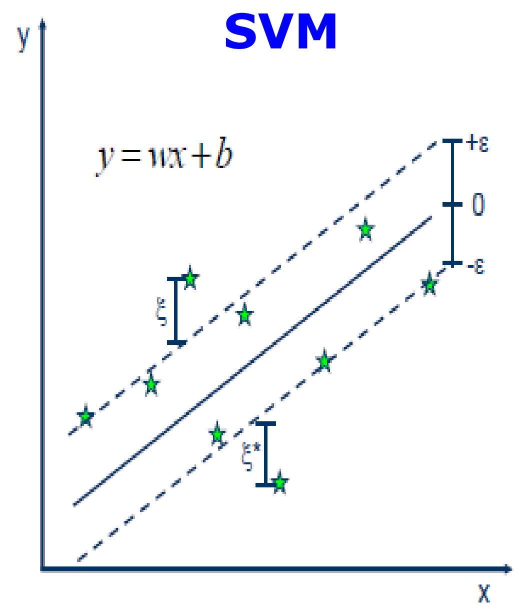


$$J_t(W_t, Z_t) = \max_{\{x_s\}_{s=t}^{T-1}} \mathbb{E}[u(W_T)|W_t, Z_t]$$

$$\text{s.t.} \quad W_{s+1} = W_s \cdot (x_s' R_{s+1}^{\mathrm{e}} + R^{\mathrm{f}}) \ \forall s \geq t$$

$$\sum_{i=1}^{n} x_i = 1, \ x \geq 0.$$

SVM



- Minimize:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$$

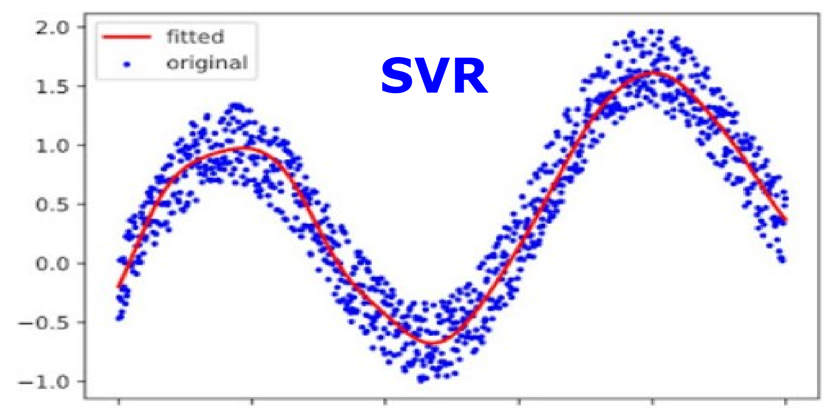
- Constraints:

$$y_i - wx_i - b \leq \varepsilon + \xi_i$$

$$wx_i + b - y_i \leq \varepsilon + \xi_i^*$$

$$\xi_i, \xi_i^* \geq 0$$

SVR



Some pictures come from WWW

Introduction & motivation: Some applications

Many application areas in science, engineering, and finance give rise to 1X1, 2X2 or 3X3 sparse systems **with equality and/or inequality constraints**, including

- **chemical engineering**
- fuid flow
- **oceanography**
- multiphysics
- **acoustics**
- economic modelling
- **Large-Scale *Portfolio Optimization***
- magnetohydrodynamic
- **structural engineering …**
- **PDE-constrained optimization**
- Least Square approximations and estimation
- **Sequential quadratic programming (SQP) methods for NLP**
- Signal and image processing, computer
- **Machine Learning (SVM, SVR, …)**

- **Nonlinear** optimisations(NLO) and nonlinear PDE models capture the complex nature of many **real-world problems**

Newton's Method

Problem: F(u) = 0, F : $IR^n \rightarrow IR^n$ continuously differentiable.

Newton's Method

Given an initial u.

Iterate:

Solve $F'(u^k)\delta = -F(u^k)$

Update $u^{k+1} = u^k + \delta$

Include the constraints: In many circumstances some unknowns are **linked** by **additional equations or inequalities:**

- Bu=b
  - linear constraints occur when one or more regions behave as rigid bodies, B and b are obtained by prescribing constant distances between all nodes belonging to that region
  - prescribed normal displacement
- Cu<=g
  - linear constraints represent non penetration conditions
- **Mixture of the both equations**

Inexact Newton

Inexact Newton methods (Dembo-Eisenstat-Steihaug, 1982) provide a framework for analysis and implementation.

Inexact Newton Method
Given an initial u.
Iterate:
Find some $\eta \in [0; 1)$ and s that satisfy
$$\|F(u) + F'(u)\, s\| \leq \eta \|F(u)\|$$
Update u: u + s.

Newton-Krylov methods:
Choose $\eta \in [0; 1)$
Apply the Krylov solver to $F'(u)\, s = -F(u)$ until
$$\|F(u) + F'(u)\, s\| \leq \eta \|F(u)\|$$
A small value of η may make computing a step that satisfies very expensive.

The issue of when to stop the linear iterations becomes the issue of choosing the « forcing term » η .

We apply relative and absolute criteria tests

$$\frac{\|F'(u^k) + F(u^k)\delta\|}{\|F(u^0)\|} \leq 10^{-6} \quad OR$$

$$\|F'(u^k) + F(u^k)\delta\| \leq 10^{-8}$$

Problem: Resolution of large-scale linear system

- **Direct Solvers**
  - Factorization (LU, MultFront, MUMPS, PARDISO, ...).
  - **Exact solution, robust.**
  - Direct method can be prohibitively resource intensive as far as memory and CPU are concerned. (systems with high connectivity (3D problems)).

- **Krylov Method:**
  - Use a Krylov subspace method to approximately solve F'(u) **δ**=-F(u) (Ax=b)
  - Construction of the sequence (xk) which converge to the solution.
  - Need only the application of A on the vector (free matrix).

Krylov Subspace Method
Given x0, determine . . .

$$x_k = x_0 + z_k,$$
$$z_k \in K_k \equiv \text{span}\{r_0, Ar_0, \ldots, A^{k-1}r_0\}$$

Examples:
CG/CR, GMRES,TFQMR, BiCGSTAB, FGMRES, GMRESR, **GCR**, MINRES, . . .

- **Preconditioners are a key to successful of these iterative methods**

Preconditioning techniques

Preconditioning techniques: $Ax = b$. $\Longleftrightarrow$ $M^{-1}Ax = M^{-1}b$

- *The matrix $M^{-1}A$ need not to be formed explicitly.*
  However, $Mw = v$ need to be solved whenever needed.

- The preconditioned system should converge faster
- Linear systems with coefficient matrix M are easy to solve.
- Note that if **M = A** any iterative method converges in one iteration.

➢ **Standard preconditioners:** ILU, Jacobi, SOR ,……

➢ ***Advanced preconditioners:***

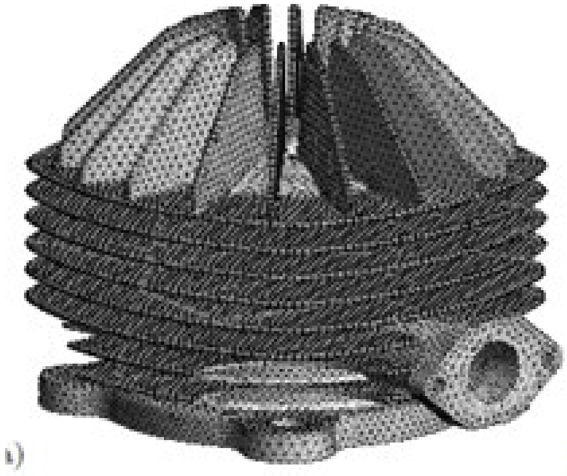
- *Multigrid, Multilevel*
- *Domain decomposition*

➢ ***Preconditioning of Karush-Kuhn-Tucker (KKT) systems or saddle point systems***

➢The robust and efficient preconditioner, **depends completely on the problem**

Discretized nonlinear PDEs.

- **The linearized system must be solved on the domain discretization**



- **FEM: this technique** is more widely used in many applications for its several favourable characteristics such as capability of dealing with **complex boundary conditions**

- **AIM: Computing the solution with high-precision and low costs in the CPU Execution time and memory storage**

Systems with equality constraints

- *Convection-diffusion*
- *Linear elasticity*

$$Au = b$$

- *Navier-Stokes problem*
- *Nearly incompressible elasticity*
- *Elasticity with equality Contact constraints*

$$A\begin{pmatrix} u \\ p \end{pmatrix} = \begin{bmatrix} F & B^t \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}.$$

- *Incompressible elasticity with equality contact constraints*

$$A\begin{bmatrix} \mathrm{u} \\ p \\ \eta \end{bmatrix} = \begin{bmatrix} K & B^T & C^T \\ B & -\varepsilon M_p & 0 \\ C & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{u} \\ p \\ \eta \end{bmatrix} = \begin{bmatrix} f \\ 0 \\ g \end{bmatrix}$$

Nearly incompressible elasticity problem with contact *equality constraints*

- *Using the finite element discretization gives the* **3x3 sparse system below:**

$$\mathcal{F}\begin{bmatrix} \mathbf{u} \\ p \\ \eta \end{bmatrix} = \begin{bmatrix} K & B^T & C^T \\ B & -\varepsilon M_p & 0 \\ C & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \\ \eta \end{bmatrix} = \begin{bmatrix} f \\ 0 \\ g \end{bmatrix}$$

We can approximate this matrix by a block factorization:

$$\mathcal{F} = \begin{bmatrix} K & B^T & C^T \\ B & -\varepsilon M_p & 0 \\ C & 0 & 0 \end{bmatrix} = \begin{bmatrix} K & 0 & 0 \\ B & -S_p & 0 \\ C & -CK^{-1}B^T & -S_\eta \end{bmatrix} \begin{bmatrix} I & K^{-1}B^T & K^{-1}C^T \\ 0 & I & -S_p^{-1}BK^{-1}C^T \\ 0 & 0 & I \end{bmatrix} = \mathcal{L}\mathcal{U}$$

MIGCR(m): GCR + Approximate Block Factorization

We use the block matrix

$$\mathcal{L} = \begin{bmatrix} \tilde{K} & 0 & 0 \\ B & -\tilde{S}_p & 0 \\ C & 0 & -\tilde{S}_\eta \end{bmatrix}$$

as preconditioner for Generalised Conjugate Residual (GCR) algorithm and then we obtain the following algorithm :

MIGCR(m): GCR(m) + $\mathcal{L}$

Submatrices approximation

There are several ways to solve the submatrices($\tilde{K}$, $\tilde{S}_p$, $\tilde{S}_\eta$) :

- Exact solves. This is in general (too) expensive.
- Inexact solves using an incomplete decomposition ILUP.
- Inexact solves using an algebraic multigrid: AMG, ML HEPRE.
- Inexact solves using an iterative method to solve the subproblems.

Since in this case the **preconditioner is variable**, the outer iteration **should be flexible**, for example GCR.

Stiffness matrix K approximation

- **Taylor-Hood $P_2 - P_1$ element is used**
  - Quadratic finite element (P_2) for **u**
  - Linear finite element (P_1) for **p**
  - Linear finite element (P_1) for λ

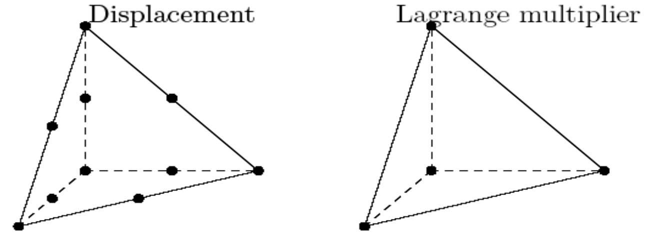


Figure 1. Taylor-Hood $P_2 - P_1$ element ($O(h^2)$)

- ***To approximate* the matrix K, we exploit its structure**
- ***Using quadratic hierarchical basis, the matrix K takes the form***

$$K\delta = \begin{pmatrix} K_{ll} & K_{lq} \\ K_{ql} & K_{qq} \end{pmatrix} \begin{pmatrix} \delta_l \\ \delta_q \end{pmatrix} = \begin{pmatrix} r_l \\ r_q \end{pmatrix} \quad \begin{cases} K_{ll} & \text{associated with vertices, linear part} \\ K_{qq} & \text{associated with midside nodes, quadratic part.} \end{cases} \quad cond(K_{qq}) = O(1)$$

- *For P_2 tetrahedral finite element* $size(K) \approx 7 \times size(K_{ll})$

Algorithm 7. Symmetric Hierarchical Preconditioner (SHP)

- Make a few iterations of SSOR on the system (pre-smoothing) $K\delta = \begin{pmatrix} K_{ll} & K_{lq} \\ K_{ql} & K_{qq} \end{pmatrix} \begin{pmatrix} \delta_l \\ \delta_q \end{pmatrix} = \begin{pmatrix} r_l \\ r_q \end{pmatrix}$
- Calculate the residual $d_l = r_l - K_{ll}\delta_l - K_{lq}\delta_q$.
- Solve $K_{ll}\delta_l^* = d_l$.
- Update: $\delta^f = (\delta_l + \delta_l^*, \ \delta_q)^T$.
- Update residual $r = r - K\delta^f$.
- Make a few iterations of SSOR on the global system $K\delta^b = r$ (post-smoothing).
- Update $\delta = \delta^b + \delta^f$.

- ***The preconditioner HP developed is efficient and robust.***

Schur complement approximation

- *Schur complement is too expensive to compute and has to be approximated:*

$$\tilde{S}_p = \varepsilon M_p + BK^{-1}B^T \approx \theta M_p \qquad \tilde{S}_\mu = CK^{-1}C^T \approx M_\mu$$

For equality contraints :

- M_p *the mass matrix on pressure space or its diagonal diag(M_p)*

For inequality contraints :

- M_μ *the mass matrix on pressure contact space or its diagonal diag(M_μ)*

Systems with inqualities constraints

- *Elasticity with inequality Contact constraints*

$$\begin{cases} Ku + C^T\lambda = f \\ Cu \leq g,\ \lambda \geq 0,\ (Cu-g).\lambda = 0 \end{cases}$$

- *Incompressible elasticity with Inequality contact constraints*

$$\begin{cases} Ku + B^T p + C^T \eta = f \\ Bu - \varepsilon M_p = 0 \\ Cu - g \leq 0, \quad \eta \geq 0 \quad \eta \cdot (Cu - g) = 0 \end{cases}$$

In this study, we are interested in 3x3 sparse systems with equality and/or inequality constraints

There are two classes of algorithms

Primal-Dual Active-Set method (PDAS)

- Nonlinear optimization algorithms: (**Inequality constraints**)
  - ✓ **Active-Set Methods (ASM)**
  - ✓ ***Penalty method,***
  - ✓ ***Augmented Lagrangian methods***
  - ✓ ***Interior Point Method (IPM)***
  - ✓ ***Projected gradient type.*** ***(May(1986), Dilintas et al. (1988), Renouf & Alart (2005),…)***

- Strengths of active-set methods:
  - can be run concurrently with the Newton iteration
  - warm-start easily
  - accurate solutions despite degeneracy and ill-conditioning

- Our goals:

  Active-set methods for **large-scale** sequential quadratic optimization problems

To do this, we combine some advanced preconditioning techniques, Krylov Subspace method and primal-dual active set strategy.

Based on work by:

- Hintermüller, Ito, Kunisch (2002)
- A. El Maliki, M. Fortin, J. Deteix & A. Fortin (2013)

PMI-GCR algorithm for nearly incompressible material problems with inequality constraints contact

- *Using the finite element discretization gives the saddle-point problems with Kuhn-Tucker conditions:*

$$\begin{cases} Ku + B^T p + C^T \eta = f \\ Bu - \varepsilon M_p = 0 \\ Cu - g \leq 0, \quad \eta \geq 0 \quad \eta \cdot (Cu - g) = 0 \end{cases}$$

represents the non-penetration condition on the contact zones

The KKT condition:
Can be written as

$$Cu - g \leq 0, \quad \eta \geq 0 \quad \eta \cdot (Cu - g) = 0$$

$$P(\eta, Cu - g) = \eta - max(0, \eta + c(Cu - g)) = 0.$$

We define the active set $\mathcal{A}_c = \{j \mid \lambda_j + c(Cu - g)_j > 0\}$

Let $\tilde{C}$ be the restriction of C to A_c

$$\begin{cases} Au + B^t p + \tilde{C}^t \lambda = f \\ B^t u - \varepsilon M_p p = 0 \\ \tilde{C} u - \tilde{g} = 0 \end{cases}$$

Primal-Dual Active-Set (PDAS) Method

The authors Hintermüller, Ito, Kunisch show the **PDAS** algorithms is equivalent to a semi-smooth Newton's method

Algorithm 3 The general active set strategy

- Initialize u_0, λ_0 . Set $k = 0$.
- Set $\mathcal{A}_c = \{j \mid \lambda_j + c(Cu - g)_j > 0\}$ and $\mathcal{I}_c = \{j \mid \lambda_j + c(Cu - g)_j \leq 0)\}$
- Let $\tilde{C}$ be the restriction of C to $\mathcal{A}_k$. We then solve

$$\begin{cases} Au + B^t p + \tilde{C}^t \lambda = f \\ B^t u - \varepsilon M_p p = 0 \\ \tilde{C} u - \tilde{g} = 0 \end{cases} \tag{21}$$

- We project λ on the admissible set $\Lambda_+ = \{\lambda \geq 0\}$ and recompute the active and inactive sets $\mathcal{A}_c$ and $\mathcal{I}_c$
- We iterate till those sets are stabilized.

Projected Mixed Iteration Generalized Conjugate Residual Algorithm (PMI-GCR)

GCR(m)

+

$$\mathcal{L} = \begin{bmatrix} \tilde{K} & 0 & 0 \\ B & -\tilde{S}_p & 0 \\ C & 0 & -\tilde{S}_\eta \end{bmatrix}$$

+ Projection procedure (Primal-Dual Active Set strategy)

+

Subject to $\lambda \geq 0$

⇓

PMI-GCR(m)

Remark: in the case of equality contraints, we use the above algorithm without the prejection procedure: we obtain the algorithm MI-GCR(m).

To avoid the oversolving of inner iteration in Newton algorithm, we use

Algorithm 3 Summary of Inexact-Newton and P-Mix-It-GCR

Let $(\delta_u, \delta_p, \delta_\lambda)$ the correction on the u, p and λ. The Inexact-Newton method combined with P-Mix-It-GCR is summarized as follow:

PMI-GCR(m)

- While $\|(\delta_u, \delta_p, \delta_\lambda)\|_2 \geq \varepsilon_{newton}$ solve the system

$$\begin{cases} K(u_k)\delta_u + B^T\delta_p + C^T\delta_\lambda = r_f \\ B\delta_u - \varepsilon M_p \delta_p = r_p \\ C\delta_u - r_g \leq 0, \quad \lambda_k \geq 0, \qquad (C\delta_u - r_g, \lambda_k) = 0 \end{cases}$$

by P-Mix-It-GCR using the two critirions

$$\|z_k^{(0)}\|_2 \geq tol_r * \|z_0^{(0)}\|_2 \quad or \quad \|z_k^{(0)}\|_2 \geq tol_a$$

$u_{k+1} = u_k + \delta_u, p_{k+1} = p_k + \delta_p$ and $\lambda_{k+1} = \lambda_k + \delta_\lambda$

Preconditioning strategies for mixed system

- *The success of the combination* ***PMI-GCR*** *involves*
  - ✓ *an efficient and robust preconditioner for K*
  - ✓ *a good approximation of the Shur complement S*
  - ✓ *Primal-Dual Active Set method*

- *The* ***PMI-GCR*** *algorithm is equivalent to apply* ***MI-GCR****(m)* *on the system with equality constraints with a modified residual*

- *Contrarily to standard projected gradient type methods,* ***PMI-GCR(m)*** *consists in solving the saddle-point system "**all-at-once**".*

- *The methods do not require the exact solution (direct method)*

Some numerical results

- *Taylor-Hood* $P_2 - P_1$ *element is used. This element satisfies the* *inf-sup* *condition and is second order* $O(h^2)$
- *The elastic body is represented by the Brick* $[0,3] \times [0,1] \times [0,1]$ *with material proprieties: E=200MPa (Young's Modulus), Poisson's ratio* $\nu=0.4999$
- *For* ***nearly incompressible material*** *we take the bulk modulus* $\kappa = 10^6$
- *The candidate contact surface* $\Gamma_C = (0,3) \times (0,1) \times \{0\}$
- *The Dirichlet condition is imposed on the border* $\Gamma_D = (0,3) \times (0,1) \times \{1\}$
- Problem sizes: 6,460 to **2,555,142** unknowns.
- *Newton algorithm tolerance*= 10^{-6} *and for inner solver* 10^{-6} *and* 10^{-12}
- *Inexact-Newton algorithm tolerance*= 10^{-6} *and for inner solver* 10^{-4} *and* 10^{-6}

S_η \ Meshes	M1	M2	M3	M4
M_η	41/ 0.22s	40/ 1.44s	36/ 10s	34/ 92s
$diag(M_\eta)$	53/ 0.28s	49/ 1.76s	43/ 13s	35/ 95s
$S\widetilde{S}$	43/ 0.28s	38/ 1.44s	41/ 12s	49/ 135s

Table 3: Contact of a Brick on the rigid flat surface with equality constraints and (ν = 0.499999). $\mathbf{MI-GCR(30)}$: Iterations count / CPU time(seconde)

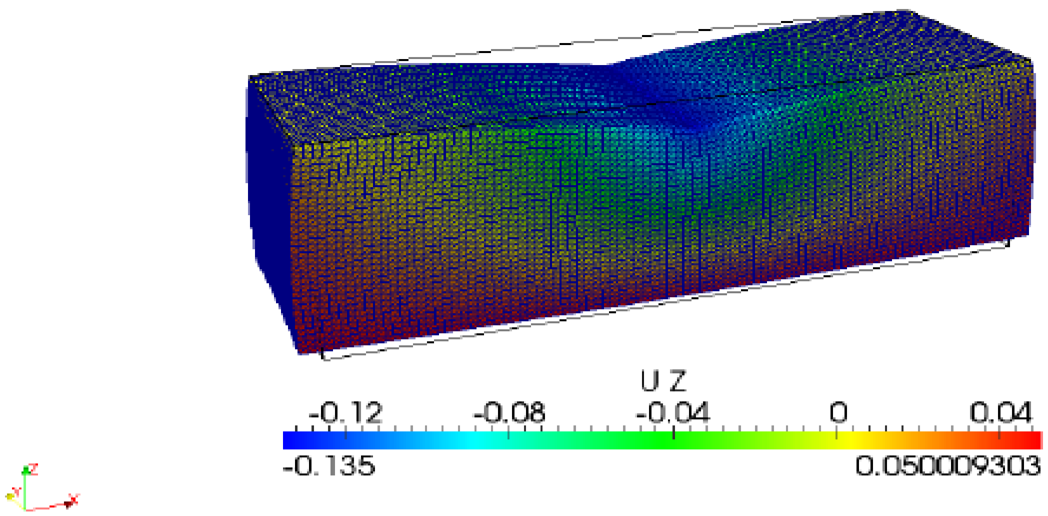
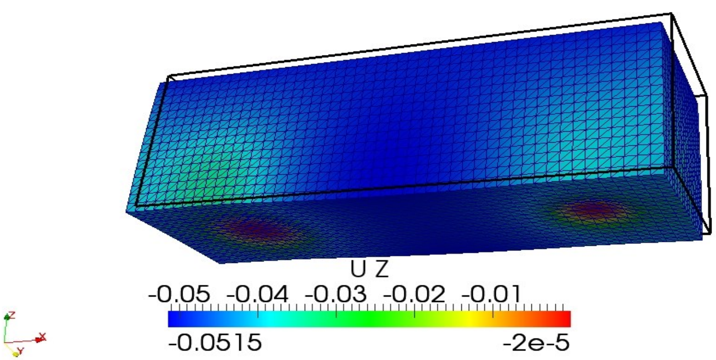


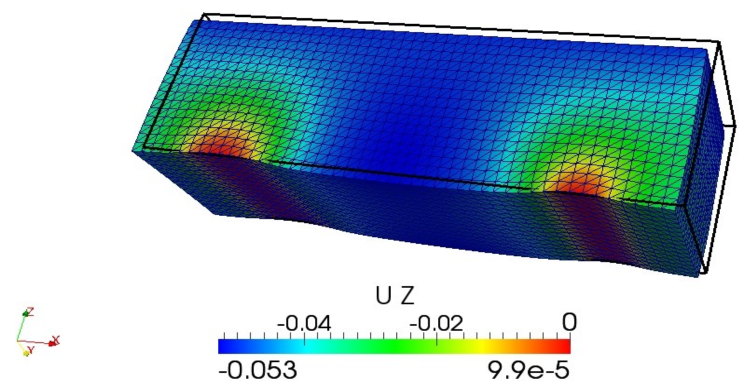
Figure 1: (a) Contact on the rigid flat surface: Case of equality contraints

Some numerical results:

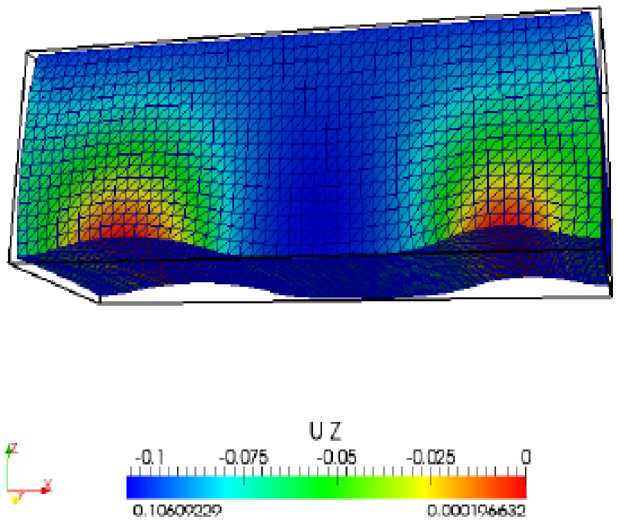
Table 4: Contact on the two rigid cylinders surface (CS) and two ridid spheres surface (SS): case of inequality constraints and ($\nu = 0.499999$). Nonlinear iterations / Average inner iteration count / CPU time(s) for PMI-GCR(30). t=10

Problem	$\tilde{S}$ \meshes	M_1	M_2	M_3	M_4
CS	$diag(M_\eta)$	6/ 38/ 0.99 s	6/ 47/ 9 s	6/ 63/112 s	6/ 87/ 1362 s
	M_η	6/ 37/ 0.99 s	6/ 40/ 8 s	6/ 48/83 s	6/ 68/ 1099 s
SS	$diag(M_\eta)$	6/ 35/ 0.92 s	6/ 41/ 8 s	6/ 62/109 s	6/ 91/ 1434 s
	M_η	6/ 36/ 0.95 s	6/ 39/ 8s	6/ 48/ 83 s	6/ 68/ 1070s

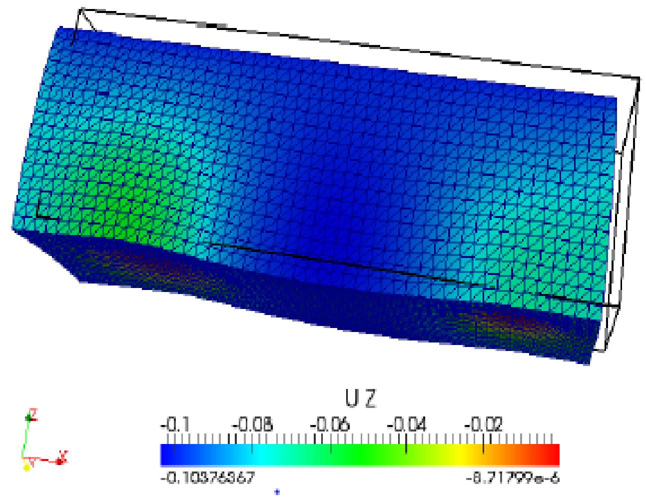




Some numerical results: solution
Case: Moony-Rivlin



(a)



(b)

Figure 3: Moony-Rivlin with bulk modulus $\kappa = 10^6$(a)Contact on the two rigid cylinders surface, (b) Contact on the two rigid spheres surface

Some numerical results
Case: Moony-Rivlin and Néo-Hookéen

Table 5: Contact of Brick body on a two spheres and two cylinders. Cases of a Mooney-Rivlin and néo-hookéen material ($\kappa = 10^6$) with Newton and Inexact-Newton combined with PMI-GCR(30). Nonlinear iterations / Average inner iteration count / CPU time(s).

	Meshes	M_3		M_4	
Problem	material parameters	$c_{01} = 1.0$ $c_{10} = 0.1$	$c_{01} = 1.0$ $c_{10} = 0.0$	$c_{01} = 1.0$ $c_{10} = 0.1$	$c_{01} = 1.0$ $c_{10} = 0.0$
CS	Newton	99/ 42/ 1284s	99/ 41/ 1165s	105/ 55/ 15390s	105/ 53/ 14719s
	Inexact-Newton	99/ 35/ 1008s	99/ 34/ 1010s	105/ 47/ 13198s	105/ 48/ 12712s
SS	Newton	98/ 40/ 1464s	99/ 41/ 1222s	105/ 52/ 14874s	105/ 69/ 19152s
	Inexact-Newton	98/ 33/ 1008s	99/ 34/ 1028s	105/ 44/ 12198s	105/ 58/ 16072s

Time Gain using Inexact-Newton: more than 20%

Summary

- *The proposed methods do not require a direct solver*
- *Numerical results on the examples show that the proposed methods*
  - *are scalable for equality constraints*
  - *show a little dependence on the mesh for inequality constraints*
  - Cost per iteration typically only slightly more than linear system solve

- *Inexact-Newton has been successfully applied.*

- *Large and realistic 3D simulations of elasticity with constraints contacts problems are now possible at low computing costs*

- *Still needed:*

  - extensions to multiphysics problems;

  - MPGCR in the Blackbox Model (For machine Learning,...)

Thank you

For your attention