Efficient solvers for constrained optimization in parameter identification problems

Naoufal NIFA

CentraleSupélec, Université Paris-Saclay
Laboratoire de mécanique des sols, structures et matériaux

Directeur de thèse : Denis AUBRY
Responsable industriel : Mathieu CORUS

Soutenance de thèse – 24 novembre 2017

Industrial framework

Context

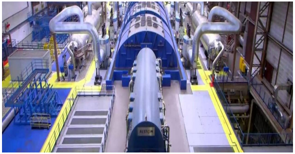
- EDF is an operator and not a manufacturer of energy structures
- Exposition to high levels of vibration due to ageing installations and maintenance tasks

Vibration issues

- Reduction of structural integrity
- Impact on the quality of service and performance

Industrial needs

- To understand the mechanical behavior of equipment
- Predictive model to asses vibration issues



Building a predictive model : the hybrid model

Context

- The hybrid model = the numerical model + the experimental model
- Using the experimental data to model-updating ➜ an inverse problem
- Least-squares methods is used in industrial context

Building a predictive model : the hybrid model

Context

- The hybrid model = the numerical model + the experimental model
- Using the experimental data to model-updating ➜ an inverse problem
- Least-squares methods is used in industrial context

The energy-based functional approach [1,2,3] (minimization of energy residual)

- Origin : an error indicator to check the quality of finite element solutions[4]
- We seek to extend the solutions identified experimentally on the numerical model

[1] **E. Balmès**. Review and evaluation of shape expansion methods. 2000

[2] **M. Reynier**. Sur le contrôle de modélisations par EF : recalage à partir d'essais dynamiques. Thèse de doctorat Paris VI. 1990

[3] **A. Kuczkowiak et al.**. Robust expansion of experimental mode shapes, Proceedings of the 32nd IMAC, 3 (2014), pp. 419–427.

[4] **P. Ladevèze**. Comparaison des modèles des milieux continus. Thèse de doctorat UPMC. 1975

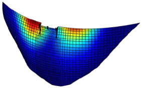
Building a predictive model : the hybrid model

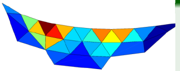
Context

- The hybrid model = the numerical model + the experimental model
- Using the experimental data to model-updating ➜ an inverse problem
- Least-squares methods is used in industrial context

The energy-based functional approach [1,2,3] (minimization of energy residual)

- Origin : an error indicator to check the quality of finite element solutions[4]
- We seek to extend the solutions identified experimentally on the numerical model





Good properties

- Locate erroneously modeled regions in space
- Robust even in presence of noisy data
- Good convexity properties of cost functions

[1] **E. Balmès**. Review and evaluation of shape expansion methods. 2000
[2] **M. Reynier**. Sur le contrôle de modélisations par EF : recalage à partir d'essais dynamiques. Thèse de doctorat Paris VI. 1990
[3] **A. Kuczkowiak et al.**. Robust expansion of experimental mode shapes, Proceedings of the 32nd IMAC, 3 (2014), pp. 419–427.
[4] **P. Ladevèze**. Comparaison des modèles des milieux continus. Thèse de doctorat UPMC. 1975

The energy-based functional approach numerical issues

Why it is not implemented for industrial purposes ?

- The resulting linear system is a difficult challenge for mechanical softwares
- High cost in CPU time and memory when using direct solvers
- The repeated use of the approach in mechanical applications

The energy-based functional approach numerical issues

Why it is not implemented for industrial purposes ?

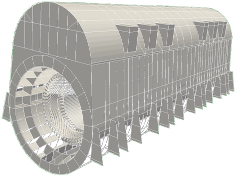
- The resulting linear system is a difficult challenge for mechanical softwares
- High cost in CPU time and memory when using direct solvers
- The repeated use of the approach in mechanical applications

Why not within model reduction framework ?[1]

- Modal Reduced basis downgrades the error localization properties
- Enriched basis : normalization problems, numerical conditioning problems

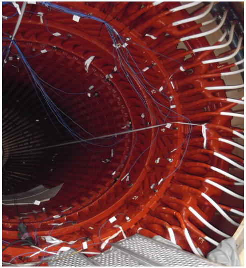
[1] **A. Bobillot and E. Balmès**. Solving minimum dynamic residual expansion and using results for error localization, In Proceeding of IMAC XIX, 4359 (2001), pp. 179–185.

The energy-based functional approach applied to a large scale problem



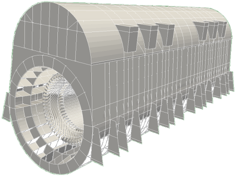
- Numerical model with more than 10^6 dofs
- 3D elements, shell elements, quadrangles, straight beams ...
- Experimental mesh using 684 sensors

➜ Robust expansion of shape modes[1]



[1] **A. Kuczkowiak, S. Cogan, M. Ouisse, E. Foltete, and M. Corus**. Robust expansion of experimental mode shapes under epistemic uncertainties, Proceedings of the 32nd IMAC, A Conference and Exposition on Structural Dynamics, 3 (2014), pp. 419–427.

The energy-based functional approach applied to a large scale problem



- Numerical model with more than 10^6 dofs
- 3D elements, shell elements, quadrangles, straight beams ...
- Experimental mesh using 684 sensors

➜ Robust expansion of shape modes[1]

Computational cost[1]

- *MD Nastran*® direct solver provides a huge computation cost for a single calculation
  ➜ 5h for one mode shape expansion, memory 36GB – cluster Intel 64 bit 2GHz
- Some options (ordering / shift trick) reduce computational time from 5h to 3h

[1] **A. Kuczkowiak, S. Cogan, M. Ouisse, E. Foltete, and M. Corus**. Robust expansion of experimental mode shapes under epistemic uncertainties, Proceedings of the 32nd IMAC, A Conference and Exposition on Structural Dynamics, 3 (2014), pp. 419–427.

Objectives

Academic objective

- Specific linear system structure
- Recent research work devoted to solving comparable linear systems[1,2,3,4,5]
- Develop proper solution methods for this specific structure

Industrial objectives

- Solve the linear systems on a **standard desktop computer**
- Develop powerful solution approach using **standard packages**
- Transfer efficient tools to engineering through the mechanical software Code_Aster ®

[1] **M. Benzi, G.H. Golub, J. Liesen**. Numerical solution of saddle point problems. Acta numerica. 2005 May ;14 :1-37.

[2] **N.I. Gould, M.E. Hribar, J. Nocedal** . On the solution of equality constrained quadratic programming problems arising in optimization. SIAM Journal on Scientific Computing. 2001 ;23(4) :1376-95.

[3] **M. Benzi, G.H. Golub**. A preconditioner for generalized saddle point problems. SIAM Journal on Matrix Analysis and Applications. 2004 ;26(1) :20-41.

[4] **Y. Little, Y. Saad**. Block preconditioners for saddle point problems. Numerical Algorithms. 2003 Aug 1 ;33(1-4) :367-79.

[5] **C. Keller, N.I. Gould, A.J. Wathen**. Constraint preconditioning for indefinite linear systems. SIAM Journal on Matrix Analysis and Applications. 2000 ;21(4) :1300-17.

Overview

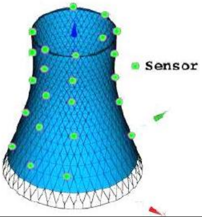
Numerical and experimental data

Ingredients

- Unknown model parameters θ
- FE model of the structure $(M(\theta), K(\theta))$ such that
    - n is the number of degrees of freedom
    - m is the number of kinematic constraints
- Eigenvalues and eigenvectors $(\omega_\theta, \varphi_\theta)$ satisfy

$$(K(\theta) - \omega_\theta^2 M(\theta))\varphi_\theta = 0, \varphi_\theta \neq 0$$

- Experimental modal basis from test campaign $(\omega_{exp}, \phi_{exp})$
    - s is the number of sensors



The objective function and the constraints

The solution fields and the constraints

- φ is the best estimation of φ_θ, minimizing the distance with the ϕ_{exp} at the pulsation ω_{exp}
- ψ is the stiffness error in the mode, such that

$$K(\theta)\psi = (K(\theta) - \omega_{exp}^2 M(\theta))\varphi$$

The objective function and the constraints

The solution fields and the constraints

- φ is the best estimation of φ_θ, minimizing the distance with the ϕ_{exp} at the pulsation ω_{exp}
- ψ is the stiffness error in the mode, such that

$$K(\theta)\psi = (K(\theta) - \omega_{exp}^2 M(\theta))\varphi$$

- **Kinematic constraints**

$$C\varphi = 0, \qquad C\psi = 0$$

  where C represents m linear relations coming from the kinematic boundary conditions and constraints.

The objective function and the constraints

The solution fields and the constraints

- φ is the best estimation of φ_θ, minimizing the distance with the ϕ_{exp} at the pulsation ω_{exp}
- ψ is the stiffness error in the mode, such that

$$K(\theta)\psi = (K(\theta) - \omega_{exp}^2 M(\theta))\varphi$$

- **Kinematic constraints**

$$C\varphi = 0, \qquad C\psi = 0$$

  where C represents m linear relations coming from the kinematic boundary conditions and constraints.

The objective function

- Quadratic cost function ➜ Model error + numerical/experimental distance

$$e_\omega(\varphi, \psi, \theta) = \frac{1}{2}\psi^T K(\theta)\psi + \frac{r}{2(1-r)}(\Pi\varphi - \phi_{exp})^T K_r(\Pi\varphi - \phi_{exp})$$

General formulation

Mathematical formulation of the constrained quadratic optimization problem

- Minimizing the cost function

$$e_\omega(\varphi, \psi, \theta) = \frac{1}{2}\psi^T K(\theta)\psi + \frac{r}{2(1-r)}(\Pi\varphi - \phi_{exp})^T K_r(\Pi\varphi - \phi_{exp})$$

subject to

$$\begin{cases} C\varphi = 0 \\ C\psi = 0 \\ K(\theta)\psi = (K(\theta) - \omega_{exp}^2 M(\theta))\varphi \end{cases}$$

Lagrange multipliers method

- convert the constrained problem into an unconstrained one

$$F_\omega(\varphi, \psi, \lambda, \lambda_1, \lambda_2, \theta) = e_\omega(\varphi, \psi, \theta) + c_\omega(\varphi, \psi, \lambda, \lambda_1, \lambda_2, \theta)$$

- c_ω is a Lagrangian functional with Lagrange multipliers λ, λ_1, and λ_2 such that

$$c_\omega(\varphi, \psi, \lambda, \lambda_1, \lambda_2, \theta) = \lambda^T((K(\theta) - \omega_{exp}^2 M(\theta))\varphi - K(\theta)\psi) - \lambda_1^T C\psi + \lambda_2^T(C\psi - C\varphi)$$

The linear system structure

The saddle-point linear system

- Minimizing the unconstrained cost function yields the following saddle-point linear system

$$\begin{pmatrix} -K(\theta) & K(\theta) - \omega_{exp}^2 M(\theta) & -C^T & C^T \\ K(\theta) - \omega_{exp}^2 M(\theta) & \frac{r}{1-r}\Pi^T K_r \Pi & C^T & 0 \\ -C & C & 0 & 0 \\ C & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \psi \\ \varphi \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{r}{1-r}\Pi^T K_r \phi_{exp} \\ 0 \\ 0 \end{pmatrix}$$

The linear system structure

The saddle-point linear system

- Minimizing the unconstrained cost function yields the following saddle-point linear system

$$\begin{pmatrix} -K(\theta) & K(\theta)-\omega_{exp}^2 M(\theta) & -C^T & C^T \\ K(\theta)-\omega_{exp}^2 M(\theta) & \frac{r}{1-r}\Pi^T K_r \Pi & C^T & 0 \\ -C & C & 0 & 0 \\ C & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \psi \\ \varphi \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{r}{1-r}\Pi^T K_r \phi_{exp} \\ 0 \\ 0 \end{pmatrix}$$

$$\mathcal{A} = \begin{pmatrix} \widetilde{E} & \widetilde{C}^T \\ \widetilde{C} & 0 \end{pmatrix} = \begin{pmatrix} -A & B^T & -C^T & C^T \\ B & T & C^T & 0 \\ -C & C & 0 & 0 \\ C & 0 & 0 & 0 \end{pmatrix}$$

➜ Notation

- $A = K(\theta)$
- $B = K(\theta) - \omega_{exp}^2 M(\theta)$
- $T = \frac{r}{1-r}\Pi^T K_r \Pi.$

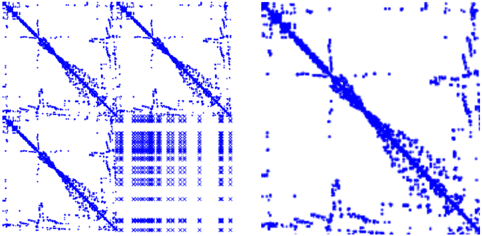
➜ Assumptions

- $C \in \mathbb{R}^{m \times n}$ of full rank m
- $\ker(A) \cap \ker(C) = \{0\}$

The system properties

Coefficient matrix properties

- symmetric sparse matrix
- saddle-point matrix [1]
- nonsingular
- some numerical issues : Large bandwidth, bad fill-in ratio, highly indefinite, poorly conditioned ...



[1] **M. Benzi, G.H. Golub, J. Liesen**. Numerical solution of saddle point problems. Acta numerica. 2005 May ;14 :1-37.

Existing direct solution approaches and contributions

Direct solution methods existing in mechanical softwares

- Direct methods : multifrontal methods (inhouse solvers, external solver MUMPS[1])
  ➜ global approaches

[1] http ://mumps.enseeiht.fr

Existing direct solution approaches and contributions

Direct solution methods existing in mechanical softwares

- Direct methods : multifrontal methods (inhouse solvers, external solver MUMPS[1])
  ➜ global approaches

Thesis contributions – direct solution methods

- Implementation of a code to compute a fill-in reducing ordering without pivoting
- Implementation of a symmetric sparse 2-by-2 block factorization approach adapted to the studied saddle-point systems

  ➜ significant gain in term of fill-in reduction up to
  - 50% in comparison with symmetric sparse solvers
  - 90% in comparison with unsymmetric sparse solvers

  ➜ not presented here

[1] http ://mumps.enseeiht.fr

Existing iterative solution approaches and contributions

Iterative solution methods existing in mechanical softwares

- Iterative methods : Krylov subspace methods + global preconditioners (incomplete factorization, multigrid)[1]
  ➜ global approaches

[1] **Y. Saad**. Iterative methods for sparse linear systems. Society for Industrial and Applied Mathematics ; 2003 Jan 1.

Existing iterative solution approaches and contributions

Iterative solution methods existing in mechanical softwares

- Iterative methods : Krylov subspace methods + global preconditioners (incomplete factorization, multigrid)[1]
  ➜ global approaches

Thesis contributions – Double projection approach

- Hybrid approach : Double explicit-implicit projection approach onto the nullspace of constraints (chapter 4, thesis)

[1] **Y. Saad**. Iterative methods for sparse linear systems. Society for Industrial and Applied Mathematics ; 2003 Jan 1.

Different constraints

Two constraints

- kinematic constraints
- sensors constraints

$$\widetilde{\mathcal{A}} = \begin{pmatrix} -A_{ss} & -A_{st} & B_{ss}^T & B_{ts}^T & -C_{ss}^T & -C_{ts}^T & C_{ss}^T & C_{ts}^T \\ -A_{ts} & -A_{tt} & B_{st}^T & B_{tt}^T & -C_{st}^T & -C_{tt}^T & C_{st}^T & C_{tt}^T \\ B_{ss} & B_{st} & T_{ss} & 0 & C_{ss}^T & C_{ts}^T & 0 & 0 \\ B_{ts} & B_{tt} & 0 & 0 & C_{st}^T & C_{tt}^T & 0 & 0 \\ -C_{ss} & -C_{st} & C_{ss} & C_{st} & 0 & 0 & 0 & 0 \\ -C_{ts} & -C_{tt} & C_{ts} & C_{tt} & 0 & 0 & 0 & 0 \\ C_{ss} & C_{st} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{ts} & C_{tt} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Different constraints

Two constraints

- kinematic constraints
- sensors constraints

$$\widetilde{\mathcal{A}} = \begin{pmatrix} -A_{ss} & -A_{st} & B_{ss}^T & B_{ts}^T & -C_{ss}^T & -C_{ts}^T & C_{ss}^T & C_{ts}^T \\ -A_{ts} & -A_{tt} & B_{st}^T & B_{tt}^T & -C_{st}^T & -C_{tt}^T & C_{st}^T & C_{tt}^T \\ B_{ss} & B_{st} & T_{ss} & 0 & C_{ss}^T & C_{ts}^T & 0 & 0 \\ B_{ts} & B_{tt} & 0 & 0 & C_{st}^T & C_{tt}^T & 0 & 0 \\ -C_{ss} & -C_{st} & C_{ss} & C_{st} & 0 & 0 & 0 & 0 \\ -C_{ts} & -C_{tt} & C_{ts} & C_{tt} & 0 & 0 & 0 & 0 \\ C_{ss} & C_{st} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{ts} & C_{tt} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} T_{ss} & 0 \\ 0 & 0 \end{pmatrix} \equiv \frac{r}{1-r} \Pi^T K_r \Pi$$

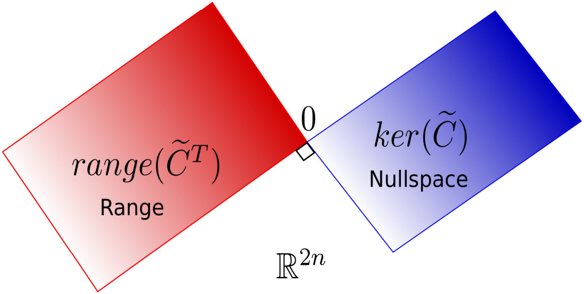
Implicit versus explicit projection

- Alternative ways to describe the projection operator

$$\widetilde{Z} \in \mathbb{R}^{2n\times 2(n-m)} \text{ such that } \operatorname{span}(\widetilde{Z}) = \ker(\widetilde{C})$$

$$\widetilde{Y} \in \mathbb{R}^{2n\times 2m} \text{ such that } \operatorname{span}(\widetilde{Y}) = \operatorname{range}(\widetilde{C}^T)$$

$$x = \widetilde{Y} x_Y + \widetilde{Z} x_Z$$



Projection approaches

Two constraints

- **kinematic constraints**
- **sensors constraints**

$$\widetilde{\mathcal{A}} = \begin{pmatrix} -A_{ss} & -A_{st} & B_{ss}^T & B_{ts}^T & -C_{ss}^T & -C_{ts}^T & C_{ss}^T & C_{ts}^T \\ -A_{ts} & -A_{tt} & B_{st}^T & B_{tt}^T & -C_{st}^T & -C_{tt}^T & C_{st}^T & C_{tt}^T \\ B_{ss} & B_{st} & T_{ss} & 0 & C_{ss}^T & C_{ts}^T & 0 & 0 \\ B_{ts} & B_{tt} & 0 & 0 & C_{st}^T & C_{tt}^T & 0 & 0 \\ -C_{ss} & -C_{st} & C_{ss} & C_{st} & 0 & 0 & 0 & 0 \\ -C_{ts} & -C_{tt} & C_{ts} & C_{tt} & 0 & 0 & 0 & 0 \\ C_{ss} & C_{st} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{ts} & C_{tt} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Elimination of kinematic constraints in the finite element matrices

New saddle point structure

- The saddle-point coefficient matrix is different from the literature

$$\widetilde{\mathcal{A}} = \begin{pmatrix} \widetilde{E} & \widetilde{C}^T \\ \widetilde{C} & 0 \end{pmatrix}$$

The nullspace of kinematic constraints

- Compute a nullspace basis

$$\widetilde{Z} \in \mathbb{R}^{2n \times 2(n-m)} \text{ such that } \operatorname{span}(\widetilde{Z}) = \ker(\widetilde{C}).$$

- Some criteria for $\widetilde{Z}$: sparse, well-conditioned, easy to apply

Construction of a sparse nullspace basis of the kinematic constraint matrix $\widetilde{C}$

Augmented nullspace basis

Let the matrix $\widetilde{C} = \begin{pmatrix} -C & C \\ C & 0 \end{pmatrix} \in \mathbb{R}^{2m\times 2n}$ be the constraint matrix.

If $Z \in \mathbb{R}^{n\times(n-m)}$ is a nullspace basis of C then $\widetilde{Z} = \begin{pmatrix} Z & 0 \\ 0 & Z \end{pmatrix}$ is a nullspace basis of $\widetilde{C}$.

Construction of a sparse nullspace basis of the kinematic constraint matrix $\widetilde{C}$

Augmented nullspace basis

Let the matrix $\widetilde{C} = \begin{pmatrix} -C & C \\ C & 0 \end{pmatrix} \in \mathbb{R}^{2m\times 2n}$ be the constraint matrix.

If $Z \in \mathbb{R}^{n\times(n-m)}$ is a nullspace basis of C then $\widetilde{Z} = \begin{pmatrix} Z & 0 \\ 0 & Z \end{pmatrix}$ is a nullspace basis of $\widetilde{C}$.

Computational benefits

- The computational cost is cut by half
- A minimum threshold of sparsity : two null $n \times (n-m)$ blocks
- Easy to apply due to similar blocks Z

The projected system

Projected system through the nullspace basis

- The projected system to solve is

$$\widetilde{Z}^T \widetilde{E} \widetilde{Z} x_Z = \widetilde{Z}^T f$$

which yields

$$\begin{pmatrix} -A_Z & B_Z \\ B_Z & T_Z \end{pmatrix} \begin{pmatrix} x_{Z_1} \\ x_{Z_2} \end{pmatrix} = \begin{pmatrix} 0 \\ Z^T f \end{pmatrix}.$$

where

$$A_Z = Z^T A Z$$
$$B_Z = Z^T B Z$$
$$T_Z = Z^T T Z$$

Explicit projection conclusions

- Low cost for computing the nullspace basis
- Generation of a new regularized saddle-point system where the block (1,1) is definite
- The nullspace basis is computed once for all generated saddle-point system

Implicit projection through constraint preconditioned Krylov subspace method

Implicit nullspace projection

- we apply a Krylov subspace method preconditioned with a constraint preconditioner[1]

$$\underbrace{\mathcal{A}_Z = \begin{pmatrix} -A_Z & B_Z \\ B_Z & T_Z \end{pmatrix}}_{\text{The coefficient matrix}} \in \mathbb{R}^{2(n-m)\times 2(n-m)} \text{ and } \underbrace{\mathcal{G}_Z = \begin{pmatrix} -G_Z & B_Z \\ B_Z & T_Z \end{pmatrix}}_{\text{The constraint preconditioner}} \in \mathbb{R}^{2(n-m)\times 2(n-m)}.$$

to solve

$$\begin{pmatrix} -A_Z & B_Z \\ B_Z & T_Z \end{pmatrix} \begin{pmatrix} x_{Z_1} \\ x_{Z_2} \end{pmatrix} = \begin{pmatrix} 0 \\ Z^T f \end{pmatrix}.$$

[1] **N.I. Gould, M.E. Hribar, J. Nocedal** . On the solution of equality constrained quadratic programming problems arising in optimization. SIAM Journal on Scientific Computing. 2001 ;23(4) :1376-95.

Implicit projection through constraint preconditioned Krylov subspace method

Implicit nullspace projection

- we apply a Krylov subspace method preconditioned with a constraint preconditioner[1]

$$\underbrace{\mathcal{A}_Z = \begin{pmatrix} \boxed{-A_Z} & B_Z \\ B_Z & T_Z \end{pmatrix}}_{\text{The coefficient matrix}} \in \mathbb{R}^{2(n-m)\times 2(n-m)} \text{ and } \underbrace{\mathcal{G}_Z = \begin{pmatrix} \boxed{-G_Z} & B_Z \\ B_Z & T_Z \end{pmatrix}}_{\text{The constraint preconditioner}} \in \mathbb{R}^{2(n-m)\times 2(n-m)}.$$

to solve

$$\begin{pmatrix} -A_Z & B_Z \\ B_Z & T_Z \end{pmatrix} \begin{pmatrix} x_{Z_1} \\ x_{Z_2} \end{pmatrix} = \begin{pmatrix} 0 \\ Z^T f \end{pmatrix}.$$

[1] **N.I. Gould, M.E. Hribar, J. Nocedal** . On the solution of equality constrained quadratic programming problems arising in optimization. SIAM Journal on Scientific Computing. 2001 ;23(4) :1376-95.

Spectral characterization

Iterative solution method of the projected system

$$\underbrace{\begin{pmatrix} -G_Z & B_Z \\ B_Z & T_Z \end{pmatrix}^{-1} \begin{pmatrix} -A_Z & B_Z \\ B_Z & T_Z \end{pmatrix}}_{\text{The preconditioned matrix } \mathcal{G}_Z^{-1}\mathcal{A}_Z} \begin{pmatrix} x_{Z_1} \\ x_{Z_2} \end{pmatrix} = \begin{pmatrix} -G_Z & B_Z \\ B_Z & T_Z \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ Z^T f \end{pmatrix}.$$

Spectral characterization (Section 4.2.1, thesis)

Let $\mathcal{G}_Z$ and $\mathcal{A}_Z$ be as defined before. The matrix $\mathcal{G}_Z^{-1}\mathcal{A}_Z$ has

- an eigenvalue at 1 with multiplicity $2(n-m)-s$,
- a generalized symmetric positive definite eigenproblem of dimension equal to the number of sensors s

Physics-based approximation of the constraint preconditioner

Schur complement decomposition of the constraint preconditioner

$$\begin{pmatrix} -G_Z & B_Z^T \\ B_Z & T_Z \end{pmatrix}^{-1} = \begin{pmatrix} -G_Z^{-1} & 0 \\ 0 & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{I} & -B_Z^T \\ 0 & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ 0 & (T_Z + B_Z G_Z^{-1} B_Z^T)^{-1} \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ B_Z G_Z^{-1} & \mathbb{I} \end{pmatrix}$$

Physics-based approximation of the constraint preconditioner

Schur complement decomposition of the constraint preconditioner

$$\begin{pmatrix} -G_Z & B_Z^T \\ B_Z & T_Z \end{pmatrix}^{-1} = \begin{pmatrix} -G_Z^{-1} & 0 \\ 0 & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{I} & -B_Z^T \\ 0 & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ 0 & (T_Z + B_Z G_Z^{-1} B_Z^T)^{-1} \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ B_Z G_Z^{-1} & \mathbb{I} \end{pmatrix}$$

Choosing approximations

- The block (1,1) G_Z as the Cholesky decomposition of A_Z

$$G_Z = L_A L_A^T$$

- The Schur complement $S_Z = T_Z + B_Z G_Z^{-1} B_Z^T$ as a Cholesky decomposition of a sparse matrix

$$S_Z = T_Z + B_Z A_Z^{-1} B_Z = \underbrace{T_Z + K_Z}_{\substack{\text{Approx.} \\ \widetilde{S_Z} = L_S L_S^T}} - 2\omega_{exp}^2 M_Z + \omega_{exp}^4 M_Z K_Z^{-1} M_Z$$

or

$$S_Z = T_Z + B_Z A_Z^{-1} B_Z = \underbrace{T_Z + K_Z - 2\omega_{exp}^2 M_Z}_{\substack{\text{Approx.} \\ \widetilde{S_Z} = L_S L_S^T}} + \omega_{exp}^4 M_Z K_Z^{-1} M_Z$$

- we call this constraint preconditioner $\mathcal{P}_{Chol}$

Algorithm of implicit projection onto the nullspace of sensors constraints

Step 1 : Build the constraint preconditioner $\mathcal{G}_Z$ for the inner solution methods

$$\mathcal{G}_Z^{-1} = \begin{pmatrix} PCG(-G_Z^{-1}, -\hat{G}_Z^{-1}) & 0 \\ 0 & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{I} & -B_Z^T \\ 0 & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ 0 & PCG(S_Z^{-1}, \hat{S}_Z^{-1}) \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ B_Z PCG(-G_Z^{-1}, -\hat{G}_Z^{-1}) & \mathbb{I} \end{pmatrix}$$

– Choose G_Z the approximation of the (1,1) block A_Z.
– Choose S_Z the approximation of the Schur complement.

Algorithm of implicit projection onto the nullspace of sensors constraints

Step 2 : Applying the outer solution method

$$\underbrace{\begin{pmatrix} -G_Z & B_Z \\ B_Z & T_Z \end{pmatrix}^{-1} \begin{pmatrix} -A_Z & B_Z \\ B_Z & T_Z \end{pmatrix}}_{\text{The preconditioned matrix } \mathcal{G}_Z^{-1}\mathcal{A}_Z} \begin{pmatrix} x_{Z_1} \\ x_{Z_2} \end{pmatrix} = \begin{pmatrix} -G_Z & B_Z \\ B_Z & T_Z \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ Z^T f \end{pmatrix}.$$

Solve the projected system using the preconditioned GMRES with the constraint preconditioner $\mathcal{G}_Z$ in the outer loop.

Implementation

Code_Aster

– a routine to retrieve matrix blocks by identifying the nature of each degree of freedom from the global ordering :

- K Stiffness matrix
- M Mass matrix
- C Kin. con. matrix
- Π Observation matrix
- K_r Norm matrix
- rhs Right hand side

SuperLU[1] + SuiteSparse[2]

PETSc[3]

[1] **http ://crd-legacy.lbl.gov/ xiaoye/SuperLU/**

[2] **http ://faculty.cse.tamu.edu/davis/suitesparse.html**

[3] **https ://www.mcs.anl.gov/petsc**

Implementation

Code_Aster

– a routine to retrieve matrix blocks by identifying the nature of each degree of freedom from the global ordering :

- K Stiffness matrix
- M Mass matrix
- C Kin. con. matrix
- Π Observation matrix
- K_r Norm matrix
- rhs Right hand side

SuperLU[1] + SuiteSparse[2]

- a routine to compute the nullspace basis Z
- we use an interface developed in Fortran 90 to the direct solver SuperLU
- we use the routine `amd` from package SuiteSparse to perform minimum degree ordering to get sparse factors

PETSc[3]

[1] **http ://crd-legacy.lbl.gov/ xiaoye/SuperLU/**

[2] **http ://faculty.cse.tamu.edu/davis/suitesparse.html**

[3] **https ://www.mcs.anl.gov/petsc**

Implementation

Code_Aster

– a routine to retrieve matrix blocks by identifying the nature of each degree of freedom from the global ordering :

- K Stiffness matrix
- M Mass matrix
- C Kin. con. matrix
- Π Observation matrix
- K_r Norm matrix
- *rhs* Right hand side

SuperLU[1] + SuiteSparse[2]

- a routine to compute the nullspace basis Z
- we use an interface developed in Fortran 90 to the direct solver SuperLU
- we use the routine `amd` from package SuiteSparse to perform minimum degree ordering to get sparse factors

PETSc[3]

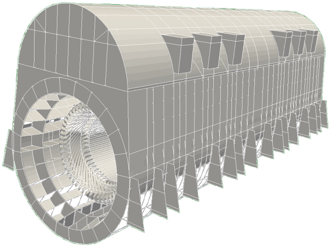
- C code to solve the projected system through the constraint preconditioned GMRES
- use the saddle-point block structure
- developed in a distributed parallel environment (MPI)

[1] **http ://crd-legacy.lbl.gov/ xiaoye/SuperLU/**

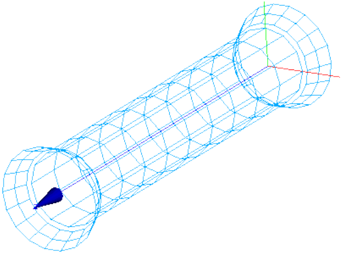
[2] **http ://faculty.cse.tamu.edu/davis/suitesparse.html**

[3] **https ://www.mcs.anl.gov/petsc**

Large scale industrial application

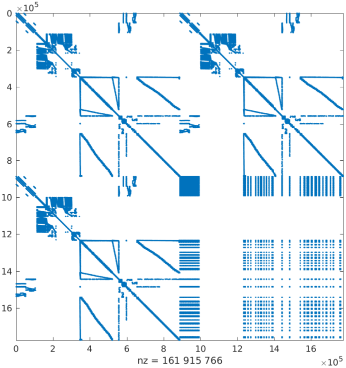


- $n = 887,601$ degrees of freedom
- $m = 44,292$ kinematic constraints



- experimental mesh using $s = 684$ sensors

The saddle-point system



$$\begin{pmatrix} -\widetilde{K}(\theta) & \widetilde{K}(\theta) - \omega_{exp}^2 \widetilde{M}(\theta) \\ \widetilde{K}(\theta) - \omega_{exp}^2 \widetilde{M}(\theta) & \frac{r}{1-r}\widetilde{\Pi}^T \widetilde{K}_r \widetilde{\Pi} \end{pmatrix}$$

Size	nnz
1.81 million	162 million

Condition number 10^{11}

Explicit projection onto the nullspace of kinematic constraints

The constraint matrix C	$22{,}000 \times 890{,}000$ $nnz = 321,000$	The nullspace basis Z	$890{,}000 \times 865{,}000$ $nnz = 1.23$ million

- The initial system

$$\text{size} \ \approx 1.81 \text{ million}$$
$$\text{nnz} \ \approx 162 \text{ million}$$

- The projected system

$$\text{size} \ \approx 1.77 \text{ million}$$
$$\text{nnz} \ \approx 167 \text{ million } (+3\% \text{ of additional fill-in})$$

Explicit projection onto the nullspace of kinematic constraints

The constraint matrix C	$22{,}000 \times 890{,}000$ $nnz = 321,000$	The nullspace basis Z	$890{,}000 \times 865{,}000$ $nnz = 1.23$ million

- The initial system

$$\text{size } \approx 1.81 \text{ million}$$

$$\text{nnz } \approx 162 \text{ million}$$

- The projected system

$$\text{size } \approx 1.77 \text{ million}$$

$$\text{nnz } \approx 167 \text{ million } (+3\% \text{ of additional fill-in})$$

Statistics

- CPU Time < 30 seconds
- Memory < 20 MB

Implicit projection onto the nullspace of sensors constraints – constraint preconditioner

Solution methods for the industrial test case

- Solving the saddle-point system using the direct solver MUMPS[1] with METIS[2] ordering
- Solving the saddle-point system using the double projection approach with two constraint preconditioners :

$$\begin{pmatrix} -G_Z & B_Z^T \\ B_Z & T_Z \end{pmatrix}^{-1} = \begin{pmatrix} -G_Z^{-1} & 0 \\ 0 & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{I} & -B_Z^T \\ 0 & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ 0 & S_Z^{-1} \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ B_Z G_Z^{-1} & \mathbb{I} \end{pmatrix}$$

  - we consider SIMPLE constraint preconditioner $\mathcal{P}_{SIMPLE}$ such that

    The block (1,1) $G_Z = A_Z$

    the shur complement $S_Z \approx T_Z + B_Z Diag(A_Z)^{-1} B_Z^T$

  - we recall the proposed constraint preconditioner $\mathcal{P}_{Chol}$ such that

    The block (1,1) $G_Z = A_Z$

    the shur complement $S_Z \approx T_Z + K_Z$

  - GMRES in the outer loop (Maximum iterations $= 10,000$, relative tolerance $= 10^{-9}$)

[1] http ://mumps.enseeiht.fr

[2] http ://glaros.dtc.umn.edu/gkhome/metis/metis/overview

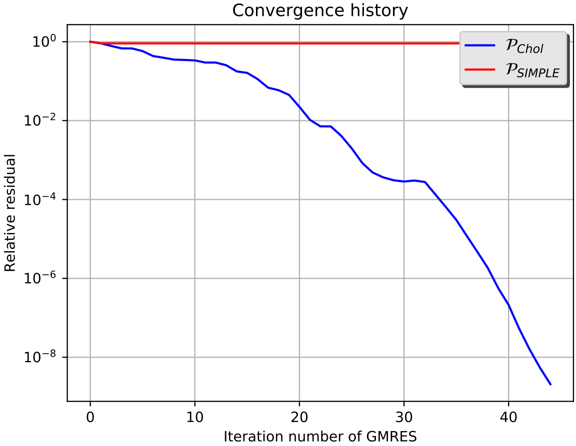
The double projection approach outperforms direct solver MUMPS – 16 procs/1 node [1]

Solver		Iterations	CPU Time (sec)	Memory (MB)
MUMPS		$\times$	5,812	36,000
$\mathcal{P}_{SIMPLE}$	Implicit projection	8,624	41,520	724
	Explicit projection	$\times$	29	19
	Total	$\times$	41,549	**743**
$\mathcal{P}_{Chol}$	Implicit projection	45	2,516	393
	Explicit projection	$\times$	29	19
	Total	$\times$	**2,545**	**412**

➜ A saving of up to 56% in terms of CPU time using $\mathcal{P}_{Chol}$
➜ Limited amount of memory (up to 100% gain)
➜ High precision of 10^{-9}

[1] Computation is carried out on Aster5 cluster, a IBM IDATAPLEX computer, Physical memory available on a given node (24 cores) of Aster5 ranges from 64 GB to 1 TB.

Comparison between constraint preconditioners $\mathcal{P}_{Chol}$ and $\mathcal{P}_{SIMPLE}$



Numerical and theoretical results

Numerical results

- Efficiency : solving large scale industrial applications
- Robustness : study for different mesh sizes, with similar gains (see thesis, Section 4.3.3)
- Precision : high precision 10^{-9}

Theoretical results

- Proof of nonsingularity of the studied saddle-point system
- Proposition of an augmented form of the nullspace basis
- Proposition of a constraint preconditioner with a problem-dependent approximation of Schur complement
- Spectral characterization

Contributions – the direct solution methods

Academic contributions

- Article is being submitted on the sparse block-wise factorization for saddle point matrices in November 2017

Industrial contributions – direct solution methods

- Computing a fill-in reducing ordering without pivoting
- Computing an adapted symmetric sparse 2-by-2 block factorization approach

➜ significant gain in term of fill-in reduction up to

- 50% in comparison with symmetric sparse solvers
- 90% in comparison with unsymmetric sparse solvers

Contributions – the double projection approach

Academic contributions

- Article is being submitted on the double explicit-implicit projection approach in November 2017

Industrial contributions – double projection approach

- Implementation of a code to compute a sparse nullspace basis
- Standalone C code in PETSc
  - Incorporation of a new saddle-point structure
  - Definition and application of constraint preconditioners
- Implicit projection using constraint preconditioners is a general result

Perspectives

Double projection approach

- Choosing different sets of constraints for either explicit or implicit projection
- Testing the class of implicit-factorization constraint preconditioners
- Application of the constraint preconditioner using low-rank corrections
- More integrated developments in Code_Aster ®

Towards industrial applications

- Update large scale finite element models from test data
- Identify unknown parameters

Thanks for your attention

Efficient solvers for constrained optimization in parameter identification problems

Naoufal NIFA

CentraleSupélec, Université Paris-Saclay
Laboratoire de mécanique des sols, structures et matériaux

Directeur de thèse : Denis AUBRY
Responsable industriel : Mathieu CORUS

Soutenance de thèse – 24 novembre 2017

Industrial framework

Context

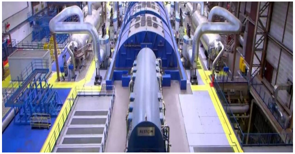
- EDF is an operator and not a manufacturer of energy structures
- Exposition to high levels of vibration due to ageing installations and maintenance tasks

Vibration issues

- Reduction of structural integrity
- Impact on the quality of service and performance

Industrial needs

- To understand the mechanical behavior of equipment
- Predictive model to asses vibration issues



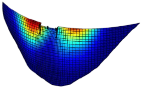
Building a predictive model : the hybrid model

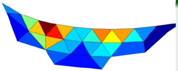
Context

- The hybrid model = the numerical model + the experimental model
- Using the experimental data to model-updating ➜ an inverse problem
- Least-squares methods is used in industrial context

The energy-based functional approach [1,2,3] (minimization of energy residual)

- Origin : an error indicator to check the quality of finite element solutions[4]
- We seek to extend the solutions identified experimentally on the numerical model





Good properties

- Locate erroneously modeled regions in space
- Robust even in presence of noisy data
- Good convexity properties of cost functions

[1] **E. Balmès**. Review and evaluation of shape expansion methods. 2000

[2] **M. Reynier**. Sur le contrôle de modélisations par EF : recalage à partir d'essais dynamiques. Thèse de doctorat Paris VI. 1990

[3] **A. Kuczkowiak et al.**. Robust expansion of experimental mode shapes, Proceedings of the 32nd IMAC, 3 (2014), pp. 419–427.

[4] **P. Ladevèze**. Comparaison des modèles des milieux continus. Thèse de doctorat UPMC. 1975

The energy-based functional approach numerical issues

Why it is not implemented for industrial purposes ?

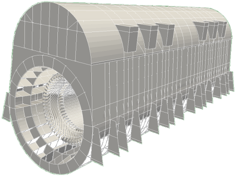
- The resulting linear system is a difficult challenge for mechanical softwares
- High cost in CPU time and memory when using direct solvers
- The repeated use of the approach in mechanical applications

Why not within model reduction framework ?[1]

- Modal Reduced basis downgrades the error localization properties
- Enriched basis : normalization problems, numerical conditioning problems

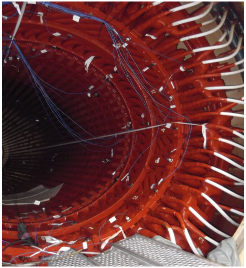
[1] **A. Bobillot and E. Balmès**. Solving minimum dynamic residual expansion and using results for error localization, In Proceeding of IMAC XIX, 4359 (2001), pp. 179–185.

The energy-based functional approach applied to a large scale problem



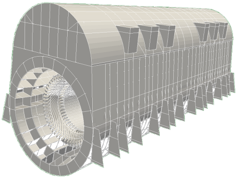
- Numerical model with more than 10^6 dofs
- 3D elements, shell elements, quadrangles, straight beams ...
- Experimental mesh using 684 sensors

→ Robust expansion of shape modes[1]



[1] **A. Kuczkowiak, S. Cogan, M. Ouisse, E. Foltete, and M. Corus**. Robust expansion of experimental mode shapes under epistemic uncertainties, Proceedings of the 32nd IMAC, A Conference and Exposition on Structural Dynamics, 3 (2014), pp. 419–427.

The energy-based functional approach applied to a large scale problem



- Numerical model with more than 10^6 dofs
- 3D elements, shell elements, quadrangles, straight beams ...
- Experimental mesh using 684 sensors

➜ Robust expansion of shape modes[1]

Computational cost[1]

- *MD Nastran*® direct solver provides a huge computation cost for a single calculation
  ➜ 5h for one mode shape expansion, memory 36GB – cluster Intel 64 bit 2GHz
- Some options (ordering / shift trick) reduce computational time from 5h to 3h

[1] **A. Kuczkowiak, S. Cogan, M. Ouisse, E. Foltete, and M. Corus**. Robust expansion of experimental mode shapes under epistemic uncertainties, Proceedings of the 32nd IMAC, A Conference and Exposition on Structural Dynamics, 3 (2014), pp. 419–427.

Objectives

Academic objective

- Specific linear system structure
- Recent research work devoted to solving comparable linear systems[1,2,3,4,5]
- Develop proper solution methods for this specific structure

Industrial objectives

- Solve the linear systems on a **standard desktop computer**
- Develop powerful solution approach using **standard packages**
- Transfer efficient tools to engineering through the mechanical software Code_Aster ®

[1] **M. Benzi, G.H. Golub, J. Liesen**. Numerical solution of saddle point problems. Acta numerica. 2005 May ;14 :1-37.

[2] **N.I. Gould, M.E. Hribar, J. Nocedal** . On the solution of equality constrained quadratic programming problems arising in optimization. SIAM Journal on Scientific Computing. 2001 ;23(4) :1376-95.

[3] **M. Benzi, G.H. Golub**. A preconditioner for generalized saddle point problems. SIAM Journal on Matrix Analysis and Applications. 2004 ;26(1) :20-41.

[4] **Y. Little, Y. Saad**. Block preconditioners for saddle point problems. Numerical Algorithms. 2003 Aug 1 ;33(1-4) :367-79.

[5] **C. Keller, N.I. Gould, A.J. Wathen**. Constraint preconditioning for indefinite linear systems. SIAM Journal on Matrix Analysis and Applications. 2000 ;21(4) :1300-17.

Overview

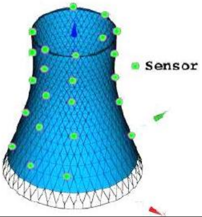
Numerical and experimental data

Ingredients

- Unknown model parameters θ
- FE model of the structure $(M(\theta), K(\theta))$ such that
  - n is the number of degrees of freedom
  - m is the number of kinematic constraints
- Eigenvalues and eigenvectors $(\omega_\theta, \varphi_\theta)$ satisfy

$$(K(\theta) - \omega_\theta^2 M(\theta))\varphi_\theta = 0, \varphi_\theta \neq 0$$

- Experimental modal basis from test campaign $(\omega_{exp}, \phi_{exp})$
  - s is the number of sensors



The objective function and the constraints

The solution fields and the constraints

- φ is the best estimation of φ_θ, minimizing the distance with the ϕ_{exp} at the pulsation ω_{exp}
- ψ is the stiffness error in the mode, such that

$$K(\theta)\psi = (K(\theta) - \omega_{exp}^2 M(\theta))\varphi$$

- **Kinematic constraints**

$$C\varphi = 0, \qquad C\psi = 0$$

  where C represents m linear relations coming from the kinematic boundary conditions and constraints.

The objective function

- Quadratic cost function ➜ Model error + numerical/experimental distance

$$e_\omega(\varphi, \psi, \theta) = \frac{1}{2}\psi^T K(\theta)\psi + \frac{r}{2(1-r)}(\Pi\varphi - \phi_{exp})^T K_r(\Pi\varphi - \phi_{exp})$$

General formulation

Mathematical formulation of the constrained quadratic optimization problem

- Minimizing the cost function

$$e_\omega(\varphi,\psi,\theta) = \frac{1}{2}\psi^T K(\theta)\psi + \frac{r}{2(1-r)}(\Pi\varphi - \phi_{exp})^T K_r(\Pi\varphi - \phi_{exp})$$

subject to

$$\begin{cases} C\varphi = 0 \\ C\psi = 0 \\ K(\theta)\psi = (K(\theta) - \omega_{exp}^2 M(\theta))\varphi \end{cases}$$

Lagrange multipliers method

- convert the constrained problem into an unconstrained one

$$F_\omega(\varphi,\psi,\lambda,\lambda_1,\lambda_2,\theta) = e_\omega(\varphi,\psi,\theta) + c_\omega(\varphi,\psi,\lambda,\lambda_1,\lambda_2,\theta)$$

- c_ω is a Lagrangian functional with Lagrange multipliers λ, λ_1, and λ_2 such that

$$c_\omega(\varphi,\psi,\lambda,\lambda_1,\lambda_2,\theta) = \lambda^T((K(\theta) - \omega_{exp}^2 M(\theta))\varphi - K(\theta)\psi) - \lambda_1^T C\psi + \lambda_2^T(C\psi - C\varphi)$$

The linear system structure

The saddle-point linear system

- Minimizing the unconstrained cost function yields the following saddle-point linear system

$$\begin{pmatrix} -K(\theta) & K(\theta)-\omega_{exp}^2 M(\theta) & -C^T & C^T \\ K(\theta)-\omega_{exp}^2 M(\theta) & \frac{r}{1-r}\Pi^T K_r \Pi & C^T & 0 \\ -C & C & 0 & 0 \\ C & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \psi \\ \varphi \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{r}{1-r}\Pi^T K_r \phi_{exp} \\ 0 \\ 0 \end{pmatrix}$$

$$\mathcal{A} = \begin{pmatrix} \widetilde{E} & \widetilde{C}^T \\ \widetilde{C} & 0 \end{pmatrix} = \begin{pmatrix} -A & B^T & -C^T & C^T \\ B & T & C^T & 0 \\ -C & C & 0 & 0 \\ C & 0 & 0 & 0 \end{pmatrix}$$

➜ Notation

- $A = K(\theta)$
- $B = K(\theta) - \omega_{exp}^2 M(\theta)$
- $T = \frac{r}{1-r}\Pi^T K_r \Pi.$

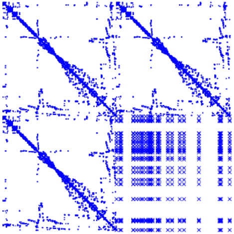
➜ Assumptions

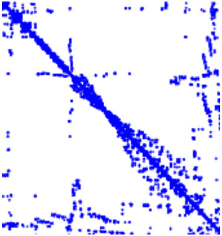
- $C \in \mathbb{R}^{m\times n}$ of full rank m
- $\ker(A) \cap \ker(C) = \{0\}$

The system properties

Coefficient matrix properties

- symmetric sparse matrix
- saddle-point matrix [1]
- nonsingular
- some numerical issues : Large bandwidth, bad fill-in ratio, highly indefinite, poorly conditioned ...





[1] **M. Benzi, G.H. Golub, J. Liesen**. Numerical solution of saddle point problems. Acta numerica. 2005 May ;14 :1-37.

Existing direct solution approaches and contributions

Direct solution methods existing in mechanical softwares

- Direct methods : multifrontal methods (inhouse solvers, external solver MUMPS[1])
  ➜ global approaches

Thesis contributions – direct solution methods

- Implementation of a code to compute a fill-in reducing ordering without pivoting
- Implementation of a symmetric sparse 2-by-2 block factorization approach adapted to the studied saddle-point systems

  ➜ significant gain in term of fill-in reduction up to
  - 50% in comparison with symmetric sparse solvers
  - 90% in comparison with unsymmetric sparse solvers

  ➜ not presented here

[1] http ://mumps.enseeiht.fr

Existing iterative solution approaches and contributions

Iterative solution methods existing in mechanical softwares

- Iterative methods : Krylov subspace methods + global preconditioners (incomplete factorization, multigrid)[1]
  ➜ global approaches

Thesis contributions – Double projection approach

- Hybrid approach : Double explicit-implicit projection approach onto the nullspace of constraints (chapter 4, thesis)

[1] **Y. Saad**. Iterative methods for sparse linear systems. Society for Industrial and Applied Mathematics ; 2003 Jan 1.

Different constraints

Two constraints

- kinematic constraints
- sensors constraints

$$\widetilde{\mathcal{A}} = \begin{pmatrix} -A_{ss} & -A_{st} & B_{ss}^T & B_{ts}^T & -C_{ss}^T & -C_{ts}^T & C_{ss}^T & C_{ts}^T \\ -A_{ts} & -A_{tt} & B_{st}^T & B_{tt}^T & -C_{st}^T & -C_{tt}^T & C_{st}^T & C_{tt}^T \\ B_{ss} & B_{st} & T_{ss} & 0 & C_{ss}^T & C_{ts}^T & 0 & 0 \\ B_{ts} & B_{tt} & 0 & 0 & C_{st}^T & C_{tt}^T & 0 & 0 \\ -C_{ss} & -C_{st} & C_{ss} & C_{st} & 0 & 0 & 0 & 0 \\ -C_{ts} & -C_{tt} & C_{ts} & C_{tt} & 0 & 0 & 0 & 0 \\ C_{ss} & C_{st} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{ts} & C_{tt} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Different constraints

Two constraints

- kinematic constraints
- sensors constraints

$$\widetilde{\mathcal{A}} = \begin{pmatrix} -A_{ss} & -A_{st} & B_{ss}^T & B_{ts}^T & -C_{ss}^T & -C_{ts}^T & C_{ss}^T & C_{ts}^T \\ -A_{ts} & -A_{tt} & B_{st}^T & B_{tt}^T & -C_{st}^T & -C_{tt}^T & C_{st}^T & C_{tt}^T \\ B_{ss} & B_{st} & T_{ss} & 0 & C_{ss}^T & C_{ts}^T & 0 & 0 \\ B_{ts} & B_{tt} & 0 & 0 & C_{st}^T & C_{tt}^T & 0 & 0 \\ -C_{ss} & -C_{st} & C_{ss} & C_{st} & 0 & 0 & 0 & 0 \\ -C_{ts} & -C_{tt} & C_{ts} & C_{tt} & 0 & 0 & 0 & 0 \\ C_{ss} & C_{st} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{ts} & C_{tt} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} T_{ss} & 0 \\ 0 & 0 \end{pmatrix} \equiv \frac{r}{1-r} \Pi^T K_r \Pi$$

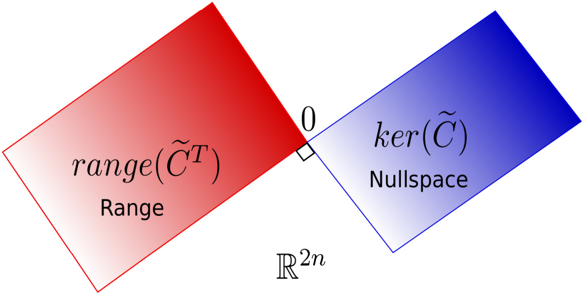
Implicit versus explicit projection

- Alternative ways to describe the projection operator

$$\widetilde{Z} \in \mathbb{R}^{2n\times 2(n-m)} \text{ such that } \operatorname{span}(\widetilde{Z}) = \ker(\widetilde{C})$$

$$\widetilde{Y} \in \mathbb{R}^{2n\times 2m} \text{ such that } \operatorname{span}(\widetilde{Y}) = \operatorname{range}(\widetilde{C}^T)$$

$$x = \widetilde{Y}x_Y + \widetilde{Z}x_Z$$



Projection approaches

Two constraints

- **kinematic constraints**
- **sensors constraints**

$$\widetilde{\mathcal{A}} = \begin{pmatrix} -A_{ss} & -A_{st} & B_{ss}^T & B_{ts}^T & -C_{ss}^T & -C_{ts}^T & C_{ss}^T & C_{ts}^T \\ -A_{ts} & -A_{tt} & B_{st}^T & B_{tt}^T & -C_{st}^T & -C_{tt}^T & C_{st}^T & C_{tt}^T \\ B_{ss} & B_{st} & T_{ss} & 0 & C_{ss}^T & C_{ts}^T & 0 & 0 \\ B_{ts} & B_{tt} & 0 & 0 & C_{st}^T & C_{tt}^T & 0 & 0 \\ -C_{ss} & -C_{st} & C_{ss} & C_{st} & 0 & 0 & 0 & 0 \\ -C_{ts} & -C_{tt} & C_{ts} & C_{tt} & 0 & 0 & 0 & 0 \\ C_{ss} & C_{st} & 0 & 0 & 0 & 0 & 0 & 0 \\ C_{ts} & C_{tt} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Elimination of kinematic constraints in the finite element matrices

New saddle point structure

- The saddle-point coefficient matrix is different from the literature

$$\widetilde{\mathcal{A}} = \begin{pmatrix} \widetilde{E} & \widetilde{C}^T \\ \widetilde{C} & 0 \end{pmatrix}$$

The nullspace of kinematic constraints

- Compute a nullspace basis

$$\widetilde{Z} \in \mathbb{R}^{2n\times 2(n-m)} \text{ such that } \operatorname{span}(\widetilde{Z}) = \ker(\widetilde{C}).$$

- Some criteria for $\widetilde{Z}$: sparse, well-conditioned, easy to apply

Construction of a sparse nullspace basis of the kinematic constraint matrix $\widetilde{C}$

Augmented nullspace basis

Let the matrix $\widetilde{C} = \begin{pmatrix} -C & C \\ C & 0 \end{pmatrix} \in \mathbb{R}^{2m \times 2n}$ be the constraint matrix.

If $Z \in \mathbb{R}^{n \times (n-m)}$ is a nullspace basis of C then $\widetilde{Z} = \begin{pmatrix} Z & 0 \\ 0 & Z \end{pmatrix}$ is a nullspace basis of $\widetilde{C}$.

Computational benefits

- The computational cost is cut by half
- A minimum threshold of sparsity : two null $n \times (n-m)$ blocks
- Easy to apply due to similar blocks Z

The projected system

Projected system through the nullspace basis

- The projected system to solve is

$$\widetilde{Z}^T \widetilde{E} \widetilde{Z} x_Z = \widetilde{Z}^T f$$

which yields

$$\begin{pmatrix} -A_Z & B_Z \\ B_Z & T_Z \end{pmatrix} \begin{pmatrix} x_{Z_1} \\ x_{Z_2} \end{pmatrix} = \begin{pmatrix} 0 \\ Z^T f \end{pmatrix}.$$

where

$$A_Z = Z^T A Z$$
$$B_Z = Z^T B Z$$
$$T_Z = Z^T T Z$$

Explicit projection conclusions

- Low cost for computing the nullspace basis
- Generation of a new regularized saddle-point system where the block (1,1) is definite
- The nullspace basis is computed once for all generated saddle-point system

Implicit projection through constraint preconditioned Krylov subspace method

Implicit nullspace projection

- we apply a Krylov subspace method preconditioned with a constraint preconditioner[1]

$$\underbrace{\mathcal{A}_Z = \begin{pmatrix} -A_Z & B_Z \\ B_Z & T_Z \end{pmatrix}}_{\text{The coefficient matrix}} \in \mathbb{R}^{2(n-m)\times 2(n-m)} \text{ and } \underbrace{\mathcal{G}_Z = \begin{pmatrix} -G_Z & B_Z \\ B_Z & T_Z \end{pmatrix}}_{\text{The constraint preconditioner}} \in \mathbb{R}^{2(n-m)\times 2(n-m)}.$$

to solve

$$\begin{pmatrix} -A_Z & B_Z \\ B_Z & T_Z \end{pmatrix} \begin{pmatrix} x_{Z_1} \\ x_{Z_2} \end{pmatrix} = \begin{pmatrix} 0 \\ Z^T f \end{pmatrix}.$$

[1] **N.I. Gould, M.E. Hribar, J. Nocedal** . On the solution of equality constrained quadratic programming problems arising in optimization. SIAM Journal on Scientific Computing. 2001 ;23(4) :1376-95.

Implicit projection through constraint preconditioned Krylov subspace method

Implicit nullspace projection

- we apply a Krylov subspace method preconditioned with a constraint preconditioner[1]

$$\underbrace{\mathcal{A}_Z = \begin{pmatrix} \boxed{-A_Z} & B_Z \\ B_Z & T_Z \end{pmatrix}}_{\text{The coefficient matrix}} \in \mathbb{R}^{2(n-m)\times 2(n-m)} \text{ and } \underbrace{\mathcal{G}_Z = \begin{pmatrix} \boxed{-G_Z} & B_Z \\ B_Z & T_Z \end{pmatrix}}_{\text{The constraint preconditioner}} \in \mathbb{R}^{2(n-m)\times 2(n-m)}.$$

to solve

$$\begin{pmatrix} -A_Z & B_Z \\ B_Z & T_Z \end{pmatrix} \begin{pmatrix} x_{Z_1} \\ x_{Z_2} \end{pmatrix} = \begin{pmatrix} 0 \\ Z^T f \end{pmatrix}.$$

[1] **N.I. Gould, M.E. Hribar, J. Nocedal** . On the solution of equality constrained quadratic programming problems arising in optimization. SIAM Journal on Scientific Computing. 2001 ;23(4) :1376-95.

Spectral characterization

Iterative solution method of the projected system

$$\underbrace{\begin{pmatrix} -G_Z & B_Z \\ B_Z & T_Z \end{pmatrix}^{-1} \begin{pmatrix} -A_Z & B_Z \\ B_Z & T_Z \end{pmatrix}}_{\text{The preconditioned matrix } \mathcal{G}_Z^{-1}\mathcal{A}_Z} \begin{pmatrix} x_{Z_1} \\ x_{Z_2} \end{pmatrix} = \begin{pmatrix} -G_Z & B_Z \\ B_Z & T_Z \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ Z^T f \end{pmatrix}.$$

Spectral characterization (Section 4.2.1, thesis)

Let $\mathcal{G}_Z$ and $\mathcal{A}_Z$ be as defined before. The matrix $\mathcal{G}_Z^{-1}\mathcal{A}_Z$ has

- an eigenvalue at 1 with multiplicity $2(n-m)-s$,
- a generalized symmetric positive definite eigenproblem of dimension equal to the number of sensors s

Physics-based approximation of the constraint preconditioner

Schur complement decomposition of the constraint preconditioner

$$\begin{pmatrix} -G_Z & B_Z^T \\ B_Z & T_Z \end{pmatrix}^{-1} = \begin{pmatrix} -G_Z^{-1} & 0 \\ 0 & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{I} & -B_Z^T \\ 0 & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ 0 & (T_Z + B_Z G_Z^{-1} B_Z^T)^{-1} \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ B_Z G_Z^{-1} & \mathbb{I} \end{pmatrix}$$

Choosing approximations

- The block (1,1) G_Z as the Cholesky decomposition of A_Z

$$G_Z = L_A L_A^T$$

- The Schur complement $S_Z = T_Z + B_Z G_Z^{-1} B_Z^T$ as a Cholesky decomposition of a sparse matrix

$$S_Z = T_Z + B_Z A_Z^{-1} B_Z = \underbrace{T_Z + K_Z}_{\substack{\text{Approx.} \\ \widetilde{S_Z} = L_S L_S^T}} - 2\omega_{exp}^2 M_Z + \omega_{exp}^4 M_Z K_Z^{-1} M_Z$$

or

$$S_Z = T_Z + B_Z A_Z^{-1} B_Z = \underbrace{T_Z + K_Z - 2\omega_{exp}^2 M_Z}_{\substack{\text{Approx.} \\ \widetilde{S_Z} = L_S L_S^T}} + \omega_{exp}^4 M_Z K_Z^{-1} M_Z$$

- we call this constraint preconditioner $\mathcal{P}_{Chol}$

Algorithm of implicit projection onto the nullspace of sensors constraints

Step 1 : Build the constraint preconditioner $\mathcal{G}_Z$ for the inner solution methods

$$\mathcal{G}_Z^{-1} = \begin{pmatrix} PCG(-G_Z^{-1}, -\hat{G}_Z^{-1}) & 0 \\ 0 & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{I} & -B_Z^T \\ 0 & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ 0 & PCG(S_Z^{-1}, \hat{S}_Z^{-1}) \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ B_Z PCG(-G_Z^{-1}, -\hat{G}_Z^{-1}) & \mathbb{I} \end{pmatrix}$$

- Choose G_Z the approximation of the (1,1) block A_Z.
- Choose S_Z the approximation of the Schur complement.

Algorithm of implicit projection onto the nullspace of sensors constraints

Step 2 : Applying the outer solution method

$$\underbrace{\begin{pmatrix} -G_Z & B_Z \\ B_Z & T_Z \end{pmatrix}^{-1} \begin{pmatrix} -A_Z & B_Z \\ B_Z & T_Z \end{pmatrix}}_{\text{The preconditioned matrix } \mathcal{G}_Z^{-1}\mathcal{A}_Z} \begin{pmatrix} x_{Z_1} \\ x_{Z_2} \end{pmatrix} = \begin{pmatrix} -G_Z & B_Z \\ B_Z & T_Z \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ Z^T f \end{pmatrix}.$$

Solve the projected system using the preconditioned GMRES with the constraint preconditioner $\mathcal{G}_Z$ in the outer loop.

Implementation

Code_Aster

– a routine to retrieve matrix blocks by identifying the nature of each degree of freedom from the global ordering :

- K Stiffness matrix
- M Mass matrix
- C Kin. con. matrix
- Π Observation matrix
- K_r Norm matrix
- *rhs* Right hand side

SuperLU[1] + SuiteSparse[2]

- a routine to compute the nullspace basis Z
- we use an interface developed in Fortran 90 to the direct solver SuperLU
- we use the routine `amd` from package SuiteSparse to perform minimum degree ordering to get sparse factors

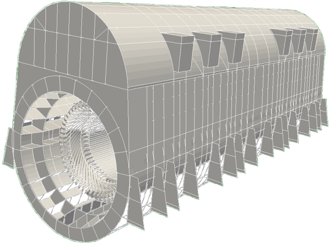
PETSc[3]

- C code to solve the projected system through the constraint preconditioned GMRES
- use the saddle-point block structure
- developed in a distributed parallel environment (MPI)

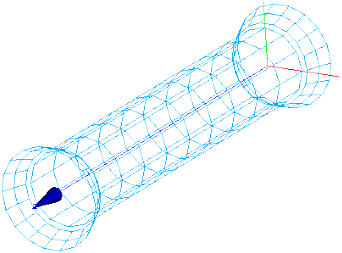
[1] **http ://crd-legacy.lbl.gov/ xiaoye/SuperLU/**
[2] **http ://faculty.cse.tamu.edu/davis/suitesparse.html**
[3] **https ://www.mcs.anl.gov/petsc**

1. **Mathematical framework**
2. **Double explicit implicit projection approach onto the nullspace of constraints**
3. **Solving a large scale industrial problem : the power plant alternator**
   - Test case presentation
   - Performance of the double projection approach
4. **Conclusions and perspectives**

Large scale industrial application

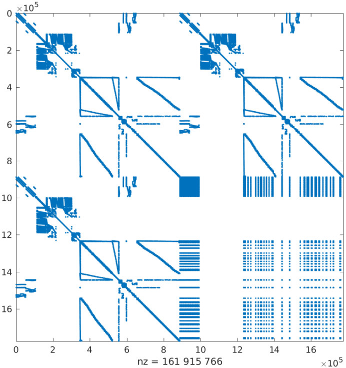


- $n = 887,601$ degrees of freedom
- $m = 44,292$ kinematic constraints



- experimental mesh using $s = 684$ sensors

The saddle-point system



$$\begin{pmatrix} -\widetilde{K}(\theta) & \widetilde{K}(\theta) - \omega_{exp}^2 \widetilde{M}(\theta) \\ \widetilde{K}(\theta) - \omega_{exp}^2 \widetilde{M}(\theta) & \frac{r}{1-r}\widetilde{\Pi}^T \widetilde{K}_r \widetilde{\Pi} \end{pmatrix}$$

Size	nnz
1.81 million	162 million

Condition number 10^{11}

Explicit projection onto the nullspace of kinematic constraints

The constraint matrix C	$22{,}000 \times 890{,}000$ $nnz = 321,000$	The nullspace basis Z	$890{,}000 \times 865{,}000$ $nnz = 1.23$ million

- The initial system

$$\text{size} \;\approx 1.81 \text{ million}$$

$$\text{nnz} \;\approx 162 \text{ million}$$

- The projected system

$$\text{size} \;\approx 1.77 \text{ million}$$

$$\text{nnz} \;\approx 167 \text{ million } (+3\% \text{ of additional fill-in})$$

Statistics

- CPU Time < 30 seconds
- Memory < 20 MB

Implicit projection onto the nullspace of sensors constraints – constraint preconditioner

Solution methods for the industrial test case

- Solving the saddle-point system using the direct solver MUMPS[1] with METIS[2] ordering
- Solving the saddle-point system using the double projection approach with two constraint preconditioners :

$$\begin{pmatrix} -G_Z & B_Z^T \\ B_Z & T_Z \end{pmatrix}^{-1} = \begin{pmatrix} -G_Z^{-1} & 0 \\ 0 & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{I} & -B_Z^T \\ 0 & \mathbb{I} \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ 0 & S_Z^{-1} \end{pmatrix} \begin{pmatrix} \mathbb{I} & 0 \\ B_Z G_Z^{-1} & \mathbb{I} \end{pmatrix}$$

  - we consider SIMPLE constraint preconditioner $\mathcal{P}_{SIMPLE}$ such that

    The block (1,1) $G_Z = A_Z$

    the shur complement $S_Z \approx T_Z + B_Z Diag(A_Z)^{-1} B_Z^T$
  - we recall the proposed constraint preconditioner $\mathcal{P}_{Chol}$ such that

    The block (1,1) $G_Z = A_Z$

    the shur complement $S_Z \approx T_Z + K_Z$
  - GMRES in the outer loop (Maximum iterations $= 10,000$, relative tolerance $= 10^{-9}$)

[1] http ://mumps.enseeiht.fr

[2] http ://glaros.dtc.umn.edu/gkhome/metis/metis/overview

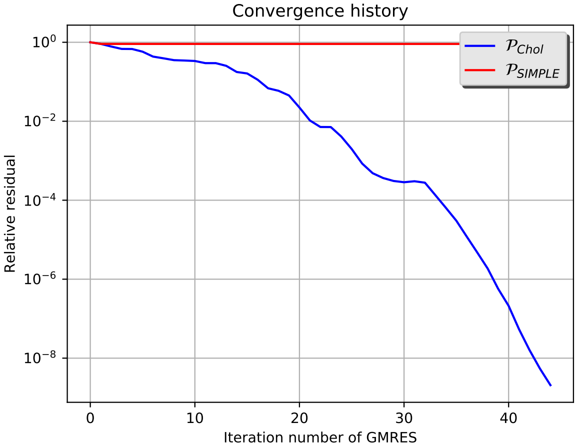
The double projection approach outperforms direct solver MUMPS – 16 procs/1 node [1]

Solver		Iterations	CPU Time (sec)	Memory (MB)
MUMPS		×	5,812	36,000
$\mathcal{P}_{SIMPLE}$	Implicit projection	8,624	41,520	724
	Explicit projection	×	29	19
	Total	×	41,549	**743**
$\mathcal{P}_{Chol}$	Implicit projection	45	2,516	393
	Explicit projection	×	29	19
	Total	×	**2,545**	**412**

➜ A saving of up to 56% in terms of CPU time using $\mathcal{P}_{Chol}$
➜ Limited amount of memory (up to 100% gain)
➜ High precision of 10^{-9}

[1] Computation is carried out on Aster5 cluster, a IBM IDATAPLEX computer, Physical memory available on a given node (24 cores) of Aster5 ranges from 64 GB to 1 TB.

Comparison between constraint preconditioners $\mathcal{P}_{Chol}$ and $\mathcal{P}_{SIMPLE}$



Numerical and theoretical results

Numerical results

- Efficiency : solving large scale industrial applications
- Robustness : study for different mesh sizes, with similar gains (see thesis, Section 4.3.3)
- Precision : high precision 10^{-9}

Theoretical results

- Proof of nonsingularity of the studied saddle-point system
- Proposition of an augmented form of the nullspace basis
- Proposition of a constraint preconditioner with a problem-dependent approximation of Schur complement
- Spectral characterization

Contributions – the direct solution methods

Academic contributions

- Article is being submitted on the sparse block-wise factorization for saddle point matrices in November 2017

Industrial contributions – direct solution methods

- Computing a fill-in reducing ordering without pivoting
- Computing an adapted symmetric sparse 2-by-2 block factorization approach

➜ significant gain in term of fill-in reduction up to

- 50% in comparison with symmetric sparse solvers
- 90% in comparison with unsymmetric sparse solvers

Contributions – the double projection approach

Academic contributions

- Article is being submitted on the double explicit-implicit projection approach in November 2017

Industrial contributions – double projection approach

- Implementation of a code to compute a sparse nullspace basis
- Standalone C code in PETSc
  - Incorporation of a new saddle-point structure
  - Definition and application of constraint preconditioners
- Implicit projection using constraint preconditioners is a general result

Perspectives

Double projection approach

- Choosing different sets of constraints for either explicit or implicit projection
- Testing the class of implicit-factorization constraint preconditioners
- Application of the constraint preconditioner using low-rank corrections
- More integrated developments in Code_Aster ®

Towards industrial applications

- Update large scale finite element models from test data
- Identify unknown parameters

Thanks for your attention

Construction of a sparse nullspace basis of the kinematic constraint matrix $\widetilde{C}$

Algorithm of construction of the nullspace basis Z

Step 1 : perform LU on the skinny matrix C^T

$$PC^TQ = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} U_1,$$

where P used for stability, Q used for sparsity, and $L_1 \in \mathbb{R}^{m\times m}$ is invertible.

Step 2 : solve $n-m$ triangular linear systems $-L_1^{-T}L_2^T$

Step 3 : consider the nullspace basis such as

$$Z = P^T \begin{bmatrix} -L_1^{-T}L_2^T \\ \mathbb{I}_{n-m} \end{bmatrix}.$$

The projected system

Equivalent system through the nullspace basis

- we reduce the saddle-point system

$$\begin{pmatrix} \widetilde{E} & \widetilde{C}^T \\ \widetilde{C} & 0 \end{pmatrix} \begin{pmatrix} x \\ w \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}$$

- using the fundamental nullspace basis[1]

$$\begin{pmatrix} \widetilde{Y} & \widetilde{Z} & 0 \\ 0 & 0 & \mathbb{I}_{2m} \end{pmatrix},$$

where

$$\widetilde{Y} \in \mathbb{R}^{2n \times 2m} \text{ such that } \operatorname{span}(\widetilde{Y}) = \operatorname{range}(\widetilde{C}^T)$$

- to get the block triangular form

$$\begin{pmatrix} \widetilde{C}\widetilde{E}\widetilde{C}^T & \widetilde{C}\widetilde{E}\widetilde{Z} & \widetilde{C}\widetilde{C}^T \\ \widetilde{Z}^T\widetilde{E}\widetilde{C}^T & \widetilde{Z}^T\widetilde{E}\widetilde{Z} & 0 \\ \widetilde{C}\widetilde{C}^T & 0 & 0 \end{pmatrix} \begin{pmatrix} x_Y \\ x_Z \\ w \end{pmatrix} = \begin{pmatrix} \widetilde{Y}^T f \\ \widetilde{Z}^T f \\ 0 \end{pmatrix}.$$

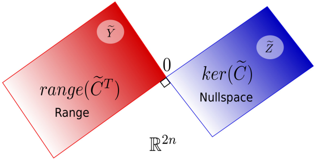
Notation

$$x = \begin{pmatrix} \psi \\ \varphi \end{pmatrix} \text{ solution fields}$$

$$w = \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} \text{ Lagrange multipliers}$$

$$f = \begin{pmatrix} 0 \\ \frac{r}{1-r}\Pi^T[K_r]\phi_{exp} \end{pmatrix} \text{ right hand side}$$

$$x = \widetilde{Y}x_Y + \widetilde{Z}x_Z$$



[1] **M. Benzi, G.H. Golub, J. Liesen**. Numerical solution of saddle point problems. Acta numerica. 2005 May ;14 :1-37.

Spectral characterization

Spectral characterization

Let $\mathcal{G}_Z$ and $\mathcal{A}_Z$ be as defined before. The matrix $\mathcal{G}_Z^{-1}\mathcal{A}_Z$ has

- an eigenvalue at 1 with multiplicity $2(n-m)-s$,
- s eigenvalues which are defined by the generalized eigenvalue problem

$$N^T(A_Z + B_Z^T LD^{-1}L^T B_Z)Nv = \lambda N^T(G_Z + B_Z^T LD^{-1}L^T B_Z)Nv.$$

where

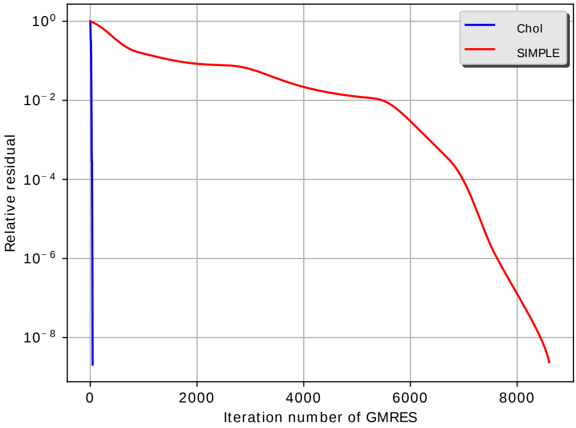
$$s = rank(T_Z) \text{ is the number of sensors degrees of freedom}$$

$$T_Z = LDL^T \text{such that} L \in \mathbb{R}^{(n-m)\times s} \text{ and } D \in \mathbb{R}^{s\times s} \text{ is SPD}$$

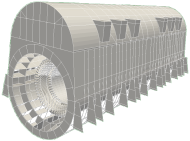
$$\text{range}(Q) = \ker(T_Z) \text{ such that } Q \in \mathbb{R}^{(n-m)\times(n-m-s)}$$

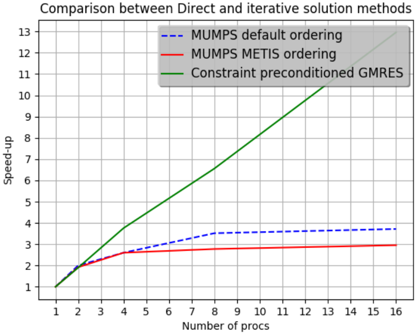
$$\text{range}(N) = \ker(Q^T B_Z) \text{ such that } N \in \mathbb{R}^{(n-m)\times s}$$

Comparison between both constraint preconditioners $\mathcal{P}_{Chol}$ and $\mathcal{P}_{SIMPLE}$

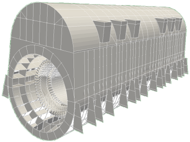


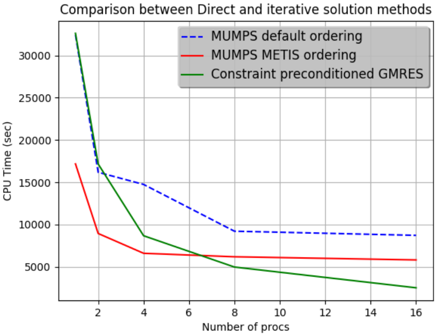
Speed-up




Comparison between Direct and iterative solution methods
MUMPS default ordering
MUMPS METIS ordering
Constraint preconditioned GMRES
Speed-up
Number of procs
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Speed-up




Comparison between Direct and iterative solution methods
MUMPS default ordering
MUMPS METIS ordering
Constraint preconditioned GMRES
CPU Time (sec)
30000
25000
20000
15000
10000
5000
2
4
6
8
10
12
14
16
Number of procs

Cluster information

Cluster Athos/Aster5

- 2720 nodes, 2 CPU (12 cores each Intel e5-2600 2.7 GHz)
- Between 64 GB and 1 TB of RAM on each node (24 cores)
- Performance : real 352.7 Tflops / 391.9 Tflops

Sparse 2-by-2 block factorization – thesis, chapter 3

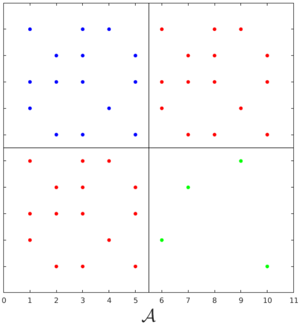


Figure – The matrix $\widetilde{\mathcal{A}}$

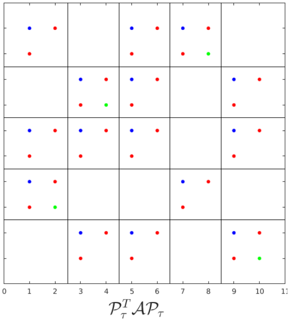


Figure – The matrix $\mathcal{P}_\tau^T \mathcal{A} \mathcal{P}_\tau$

$$\tau = \begin{pmatrix} 1 & 2 & 3 & \cdots & 2(n+m)-1 & 2(n+m) \\ 1 & n+m+1 & 2 & \cdots & n+m & 2(n+m) \end{pmatrix}$$

Sparse 2-by-2 block factorization – thesis, chapter 3

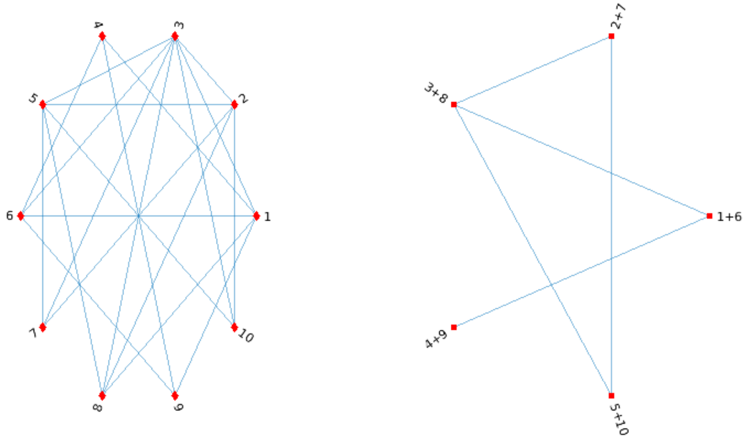
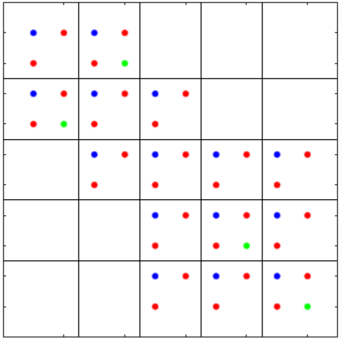


Figure – The graph associated with matrix $\widetilde{\mathcal{A}}$ (left) and the compressed graph associated with matrix $\mathcal{P}_\tau^T \mathcal{A} \mathcal{P}_\tau$ (right)

Sparse 2-by-2 block factorization – thesis, chapter 3



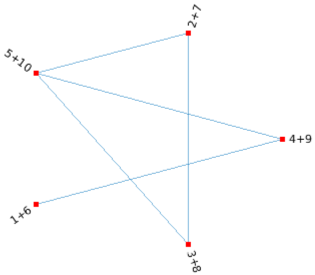


Figure – The matrix $\mathcal{P}_\pi^T \mathcal{A} \mathcal{P}_\pi$ and its associated compressed graph

$$\pi = \begin{pmatrix} 1 & 2 & 3 & \cdots & 2(n+m)-1 & 2(n+m) \\ r(1) & n+m+r(1) & r(2) & \cdots & r(n+m) & (n+m)+r(n+m) \end{pmatrix}.$$

Sparse 2-by-2 block factorization – thesis, chapter 3

Choosing pivots of order 2

$$\begin{pmatrix} -A & B_1^T \\ B_2 & C \end{pmatrix} = \begin{pmatrix} \mathbb{I} & 0 \\ B_2 A^{-1} & \mathbb{I} \end{pmatrix} \begin{pmatrix} A & 0 \\ 0 & C + B_2 A^{-1} B_1^T \end{pmatrix} \begin{pmatrix} \mathbb{I} & A^{-1} B_1^T \\ 0 & \mathbb{I} \end{pmatrix}.$$

➜ Comparison with symmetric indefinite solvers

	Matrix	$FE1$	$FE2$	$FE3$	$FE4$
	$2(n+m)$ nnz	2,410 14,427	4,104 24,608	6,504 39,660	10,000 60,054
SBlock	*Fill-in* *Err*	100,112 1.94e-10	266,800 1.75e-10	612,522 2.48e-10	1,216,652 1.22e-09
$MA57 - LDL^T$ *Matlab*	*Fill-in* *Err*	217,136 5.55e-09	624,756 1.62e-08	1,701,536 5.54e-09	3,583,097 1.69e-08
$MUMPS - LDL$	*Fill-in* *Err*	303,299 1.51e-09	1,016,383 2.54e-09	2,860,465 2.48e-09	6,602,797 4.18e-08

➜ 50% in comparison with symmetric sparse solvers

Sparse 2-by-2 block factorization – thesis, chapter 3

Choosing pivots of order 2

$$\begin{pmatrix} -A & B_1^T \\ B_2 & C \end{pmatrix} = \begin{pmatrix} \mathbb{I} & 0 \\ B_2 A^{-1} & \mathbb{I} \end{pmatrix} \begin{pmatrix} A & 0 \\ 0 & C + B_2 A^{-1} B_1^T \end{pmatrix} \begin{pmatrix} \mathbb{I} & A^{-1} B_1^T \\ 0 & \mathbb{I} \end{pmatrix}.$$

➜ Comparison with unsymmetric indefinite solvers

	Matrix	$FE1$	$FE2$	$FE3$	$FE4$
	$2(n+m)$ nnz	2,410 14,427	4,104 24,608	6,504 39,660	10,000 60,054
SBlock	*Fill-in* *Err*	100,112 1.94e-10	266,800 1.75e-10	612,522 2.48e-10	1,216,652 1.22e-09
UMFPACK – LU Matlab	*Fill-in* *Err*	774,108 9.90e-13	2,817,520 1.75e-12	7,165,751 2.48e-12	15,218,709 2.83e-12
MUMPS – LU	*Fill-in* *Err*	574,624 2.94e-12	1,823,314 4.18e-12	4,800,112 3.48e-12	10,365,088 8.60e-12

➜ 90% in comparison with unsymmetric sparse solvers