Null space computation of sparse singular matrices with MUMPS

Xavier Vasseur (CERFACS)

In collaboration with
Patrick Amestoy (INPT-IRIT, University of Toulouse and ENSEEIHT),
Serge Gratton (INPT-IRIT, University of Toulouse and ENSEEIHT),
Jean-Yves L'Excellent (INRIA Rhône-Alpes, Laboratoire de l'Informatique du Parallélisme, ENS Lyon)

ENSEEIHT, Toulouse

MUMPS User Group Meeting, April 16th 2010

Outline

Outline

Problem setting

Within a sparse multifrontal code

- Estimate the deficiency d of the singular matrix A and compute a null space basis $Z \in \mathbb{C}^{n \times d}$ such that $A\,Z = 0_{n \times d}$ where $A \in \mathbb{C}^{n \times n}$ is a large sparse singular matrix.

Properties of the null space detection

- Reliable for sparse matrices with small to high deficiency
- Efficient also possibly in a parallel distributed memory environment

Analysis of the solution phase (not discussed today)

- Exploit the sparsity of the multiple right-hand side problems in an out-of-core framework (addressed in Slavova's PhD thesis [2009, Section 11.2])

Outline

Motivations

Increased interest when designing advanced iterative methods

- Solution methods for saddle point problems [Benzi et al., 2005, Section 6]
- Constrained optimization
- Domain decomposition solvers (singular problems at the subdomain level) [Toselli et al., 2004]

Increased interest in real-life applications

- Electromagnetics (constrained eigenvalue solvers)
- Fluid and/or structural mechanics (null space of discrete operators)

Some recent related software for sparse rank-deficient matrices

- LUSOL [http://www.stanford.edu/group/SOL/software/lusol.html]
- spnrank [http://www.math.sjsu.edu/singular/matrices/software/]
- SuiteSparseQR [http://www.cise.ufl.edu/research/sparse/SPQR/]

Outline

Related references (of course incomplete !)

Sparse rank-revealing LU or orthogonal decompositions

- Sparse multifrontal rank-revealing QR factorization [Pierce and Lewis, 1997]
- Sparse LU with null pivot detection [Farhat and Géradin, 1998]
- Sparse symmetric rank-revealing decompositions VSV^T [Bratland and Frimodt, 2002]
- Sparse rank-revealing LU based on threshold rook pivoting or threshold complete pivoting [Gill et al, 2005, Sections 4 and 5]
- Sparse LU method with inverse power method to compute the null space of the triangular factor [Gotsman and Toledo, 2008]

Limitations that were found

- Orthogonal methods lead to usually severe fill-in for sparse problems
- Limited problem size
- No parallel implementation

Outline

Main governing idea

Determination of the null space of the singular upper triangular factor

- Analysis phase: consider the preprocessed matrix $\tilde{A} = P_s A P_c P_s^T$ where P_s corresponds to a permutation that aims at minimizing the fill-in during factorization and P_c is a column permutation to obtain a zero-free diagonal.
- Derive null space informations by inspecting only $U \in \mathbb{C}^{n \times n}$, the singular upper triangular factor obtained after numerical factorization of the preprocessed matrix $\tilde{A} = LU$
- Factorization phase: determine accurately the deficiency of U noted d and the matrix $Z_U \in \mathbb{C}^{n \times d}$ defined as

$$U\, Z_U = 0_{n \times d}.$$

A basis of the subspace spanned by the columns of Z_U will represent the right nullspace of U.

Outline

Proposed numerical pivoting strategy

Threshold partial pivoting in the non singular case

- At the j-th step of the Gaussian elimination the set of eligible pivots S_{ep} is usually defined as:

$$S_{ep} = \{q \mid |F_{SV}(q,j)| \geq u \max_i |F_{SV}(i,j)|\}$$

  where F_{SV} is the block corresponding to the fully summed variables of the frontal matrix [Duff and Reid, 84] and $u \in \mathbb{R}$ a threshold between 0 and 1 that balances sparsity and numerical stability.
- Among the set of eligible pivots one preserving sparsity is usually selected to minimize the fill-in. We will denote p this pivot and define the set of tentative pivot $S_{tp} = \{p\}$

Numerical pivoting strategy in the singular case

First set: set of null pivot rows

- $S_{nr} = \{i \mid \|F_{SV}(i,:)\|_2 \leq \tau_A\}$
- τ_A is a positive real-valued threshold parameter such as $\tau_A = v\ \varepsilon$ with $1 \leq v \leq 1000$ and ε the machine precision
- S_{nr} allows to detect the so called null pivot rows

Goal of the first set

- Those rows are modified to continue the factorization; nonzero elements are replaced by zero and the diagonal element is set to a certain fixation value.
- This modification is of course an arbitrary decision that will define one particular solution of the singular system of equations.

Numerical pivoting strategy in the singular case

Second set: set of delayed pivots

- $S_{dp} = \{p \in S_{tp} \mid |p| \leq \tau_B \|\tilde{A}\|_\infty\}$
- τ_B is a positive real-valued threshold parameter
- S_{dp} consists of the set of delayed pivots.

Goal of the second set

- The corresponding fully summed variables are not eliminated because of numerical issues.
- They are instead included in the Schur complement matrix of the frontal matrix and their elimination is postponed to the parent node or latter - potentially up to the root of the elimination tree.

Numerical pivoting strategy in the singular case

Summary: modified numerical pivoting strategy

- The modified numerical pivoting strategy leads to

$$P\tilde{A} = LU = L\begin{pmatrix} U_{11} & U_{12} \\ 0_{m\times(n-m)} & M \end{pmatrix}$$

- P is a permutation matrix of order n that corresponds to the considered modified pivoting strategy
- $U_{11} \in \mathbb{C}^{(n-m)\times(n-m)}$ is non singular
- The root matrix $M \in \mathbb{C}^{m\times m}$ is the contribution block related to the delayed pivots that have been postponed up to the root of the elimination tree.
- The next step is to analyse the deficiency of the root matrix M.

Outline

Rank-revealing algorithms applied to the root matrix M

Root matrix M

- Depending on τ_B, $M \in \mathbb{C}^{m \times m}$ can be less sparse than $\tilde{A}$ and is of reduced size ($m << n$)

Truncated rank-revealing method

- Truncated QR factorization with column permutation [Foster and Kommu, 2006]

$$M\Pi = QR = Q \begin{pmatrix} R_{11} & R_{12} \\ 0_{(m-k)\times k} & 0_{(m-k)\times(m-k)} \end{pmatrix}.$$

- k, the order of $R_{11} \in \mathbb{C}^{k \times k}$, is the effective rank of M
- Its cost is $4m^2k - 2k^2m - \frac{2}{3}k^3$ [Foster and Kommu, 2006]
- If M is a low rank matrix, this cost is thus of order $O(m^2k)$ which is significantly less than the cost required by a SVD factorization ($O(m^3)$).

Outline

Computation of a basis of the null space [I]

Null space related to the delayed pivots $Z_{RR} \in \mathbb{C}^{n\times(m-k)}$

- Determination of the null space of U

$$P\tilde{A} = LU = L\begin{pmatrix} U_{11} & U_{12} \\ 0_{m\times(n-m)} & M \end{pmatrix}$$

- Null space of the root matrix $Z_M \in \mathbb{C}^{m\times(m-k)}$

$$M\Pi\begin{pmatrix} R_{11}^{-1}R_{12} \\ -I_{m-k} \end{pmatrix} = 0_{m\times(m-k)} \quad i.e. \quad Z_M = \Pi\begin{pmatrix} R_{11}^{-1}R_{12} \\ -I_{m-k} \end{pmatrix}$$

- Null space related to the delayed pivots $Z_{RR} \in \mathbb{C}^{n\times(m-k)}$

$$\begin{pmatrix} U_{11} & U_{12} \\ 0_{m\times(n-m)} & M \end{pmatrix}\begin{pmatrix} -U_{11}^{-1}U_{12}Z_M \\ Z_M \end{pmatrix} = 0_{n\times(m-k)}$$

Computation of a basis of the null space [II]

Null space related to the null pivot rows $Z_{NP} \in \mathbb{C}^{n \times l}$

- Computation of Z_{NP} according to the row modifications made during the factorization phase

$$\begin{pmatrix} U_{11} & U_{12} \\ 0_{m \times (n-m)} & M \end{pmatrix} Z_{NP} = \begin{pmatrix} E_{(n-m) \times l} \\ 0_{m \times l} \end{pmatrix}$$

  where the columns of $E_{(n-m) \times l}$ are the i-th Cartesian basis vector of $\mathbb{R}^{(n-m)}$ where $i \in S_{nr}$.

- One of the possible solutions has thus the following simple form

$$Z_{NP} = \begin{pmatrix} U_{11}^{-1} E_{(n-m) \times l} \\ 0_{m \times l} \end{pmatrix}$$

  which requires the solution of a sparse upper triangular system with sparse multiple right-hand sides [Slavova, 2009].

Computation of a basis of the null space [III]

Null space of the preprocessed matrix $\tilde{A}$

- Part [I]: null space related to the delayed pivots $Z_{RR} \in \mathbb{C}^{n\times(m-k)}$
- Part [II]: null space related to the null pivot rows $Z_{NP} \in \mathbb{C}^{n\times l}$
- The null space of U is thus $Z_U = [Z_{NP}\, Z_{RR}]$

Null space of the original matrix A

- Since $\tilde{A} = P_s A P_c P_s^T$, $Z = P_c P_s^T Z_U$ is such that $A\,Z = 0_{n\times d}$
- The deficiency of A is equal to $d = l + m - k$

Outline

Sketch of the algorithm

Sketch of the algorithm

- Analysis phase $\tilde{A} = P_s A P_c P_s^T$
- Detection of null pivot rows and delay of pivots during the factorization of $\tilde{A}$. At the end the following decomposition is obtained
$$P\tilde{A} = L\,U = L \begin{pmatrix} U_{11} & U_{12} \\ 0_{m\times(n-m)} & M \end{pmatrix}$$
where $U_{11} \in \mathbb{C}^{(n-m)\times(n-m)}$ is nonsingular, $U_{12} \in \mathbb{C}^{(n-m)\times m}$ and $M \in \mathbb{C}^{m\times m}$.
- A l-dimensional set of row indices noted S_{nr} corresponding to null pivot rows has been determined
- M refers to the contribution block gathering pivots that have been delayed up to the root of the elimination tree.
- Null space computation $Z = P_c P_s^T [Z_{NP}\, Z_{RR}]$.

Use in MUMPS [I]

Control parameters related to $S_{nr} = \{i \mid \|F_{SV}(i,:)\|_2 \leq \tau_A\}$

- ICNTL(24) = 1: detection of null pivots
- CNTL(3) τ_A threshold for null pivot rows detection

  $\tau_A = CNTL(3)$ if $CNTL(3) > 0$

  $\tau_A = \varepsilon \times 10^{-5} \times \|A\|$ if $CNTL(3) \leq 0$

Control parameters related to $S_{dp} = \{p \in S_{tp} \mid |p| \leq \tau_B \|\tilde{A}\|_\infty\}$

- ICNTL(16) = 1: postpone delayed pivots to root node
- CNTL(6): τ_B threshold for postponing pivots

  $\tau_B = CNTL(6)$ if $CNTL(6) > 0$

  $\tau_B = \varepsilon$ if $CNTL(6) \leq 0$

Use in MUMPS [II]

Output related to null space

- INFOG(28): estimated deficiency
- ICNTL(25)=-1 computes the complete null space basis
- ICNTL(25)=i, where $1 \leq i \leq INFOG(28)$ returns the i-th vector of the null space basis
- PIVNUL_LIST list of row indices corresponding to null pivots

Features

- Implementation available in [s,d,c,z] arithmetics
- Detection of null pivots available in parallel [official release]
- Compatible with OOC version
- Interface to both MATLAB and Scilab
- Root node processed serially [restricted release]

Outline

1. Problem setting and motivations
   - Problem setting
   - Motivations
   - Related references
2. Proposed algorithm
   - Main governing idea
   - Proposed numerical pivoting strategy
   - Rank-revealing algorithms applied to the root matrix M
   - Computation of a basis of the nullspace
   - Sketch of the algorithm and use in MUMPS
3. Numerical experiments
   - Overview and goals
   - Electromagnetism applications
   - Structural mechanics applications
4. Conclusions

Numerical results: focus on real-life applications

ANR Benchmark matrices

- Concrete applications in structural mechanics [EDF]
- Real and symmetric semi-definite matrices
- Double precision arithmetics

Goals

- Reliability: can the null space detection fail ?
- Accuracy of the null space basis $Z = [Z_1 \; Z_2 \; \cdots \; Z_d]$ of the original matrix A
- Error analysis [componentwise scaled residuals and normwise backward error for each null space vector Z_i]

$$\text{Nullspace normwise backward error} = \max_{1 \le i \le d} \frac{\|A\,Z_i\|_\infty}{\|A\|_\infty \, \|Z_i\|_\infty}$$

Numerical results with MUMPS

Available strategies

- ICNTL(24) = 1 only
- ICNTL(24) = 1 and ICNTL(16) = 1

Control parameters

- Threshold for numerical pivoting CNTL(1)=10^{-2}
- Threshold for null pivots detection CNTL(3)=10^{-9}
- Fixation for null pivots CNTL(5)=10^{6}
- Threshold for postponing pivots CNTL(6)=10^{-4}

Outline

Benchmark "Box cavities" matrices

PDE problem based on Maxwell's equations [Geus, 2002]

- Computation of resonance modes in closed cavities
- Use of Nédélec vector finite elements
- Indefinite eigenvalue problems with constraints

$$\begin{aligned} Ax &= \lambda Mx \\ Cx = Y^T Mx &= 0 \end{aligned}$$

  where $A \in \mathbb{R}^{n \times n}$ is symmetric positive semi-definite [discretization of **curl curl** operators], $M \in \mathbb{R}^{n \times n}$ symmetric positive definite
- $AY = 0$, Y is generally known [$\mathbf{curl}\, \nabla = \mathbf{0}$]
- **Goal A**: investigate the influence of the ordering on the null space computation
- **Goal B**: compute the null space Z and compare with Y

Box cavities $[0., 5.2] \times [0., 3.3] \times [0., 0.77]$

Goal A: Influence of the ordering on the rank detection of $\tilde{A} = P_s A P_c P_s^T$

- $Box\_8\_5\_3 \in \mathbb{R}^{619 \times 619}$ of deficiency 56
- Single: ICNTL(24) = 1
- Combined: ICNTL(24) = 1 and ICNTL(16) = 1

	Single	Combined			
Ordering	Def.	Null piv. r.	Root size	Root def.	Def.
AMD	56	36	73	20	56
AMF	56	49	27	7	56
PORD	56	49	38	7	56
METIS	56	41	68	15	56
QAMD	56	36	73	20	56

- Correct estimation of the deficiency independently of the ordering strategy

Box cavities $[0., 5.2] \times [0., 3.3] \times [0., 0.77]$

Goal A: Influence of the ordering on the rank detection of $\tilde{A} = P_s A P_c P_s^T$

- $Box\_30\_20\_4 \in \mathbb{R}^{14454 \times 14454}$ of deficiency 1653
- Single: ICNTL(24) = 1
- Combined: ICNTL(24) = 1 and ICNTL(16) = 1

	Single	Combined			
Ordering	Def.	Null piv. r.	Root size	Root def.	Def.
AMD	1653	1515	470	138	1653
AMF	1653	1617	114	36	1653
PORD	1653	1604	178	49	1653
METIS	1653	1563	337	90	1653
QAMD	1653	1515	470	138	1653

- Correct estimation of the deficiency independently of the ordering strategy

Box cavities $[0., 5.2] \times [0., 3.3] \times [0., 0.77]$

Box cavities $[0., 5.2] \times [0., 3.3] \times [0., 0.77]$

Matrix	Size	Def.	Null piv. r.	Root size	Root def.	Def.
*Box*_8_5_3	619	56	49	27	7	56
*Box*_16_10_3	2675	270	260	31	10	270
*Box*_20_13_3	4419	456	446	31	10	456
*Box*_30_20_4	14454	1653	1563	337	90	1653
*Box*_40_27_5	33627	4056	3876	649	180	4056
*Box*_50_34_6	64878	8085	7794	1012	291	8085
*Box*_60_41_7	111147	14160	13812	1185	348	14160

- Correct estimation of the deficiency on all matrices with automatic choice of the ordering made during analysis
- Benchmark matrices used in Slavova's PhD [2009]

Box cavities $[0., 5.2] \times [0., 3.3] \times [0., 0.77]$

Goal B: Null space basis

- Computation of the null space Z and compare with the true null space Y
- Distance between subspaces $sin(\Theta(Y,Z)) = \|Y\,Y^T - Z\,Z^T\|_2$
- $\Theta(Y,Z)$ small means that the two spaces are nearly linearly dependent

Matrix			Null space basis		
Name	Size	Def.	$\Theta(Y,Z)$	Nz(Y)	Nz(Z)
Box_8_5_3	619	56	3.75e-11	784	18074
Box_16_10_3	2675	270	8.30e-11	3780	268174
Box_20_13_3	4419	456	3.94e-10	6384	597464

- Accurate computation of the null space on this benchmark problem

Outline

Applications provided by EDF - ANR SOLSTICE project (2007-2010)

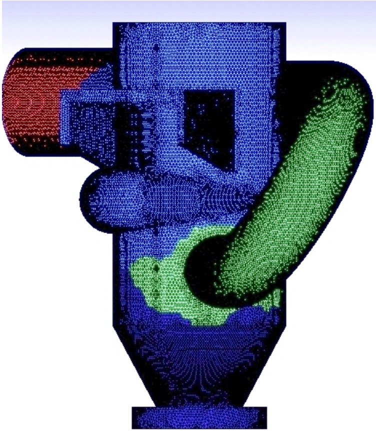
Structural mechanics

- FETI domain decomposition algorithm [Farhat and Roux, 1991] implemented in Code_Aster
- Subdomain matrices $A^{(i)}$ are singular on some subdomains
- $A^{(i)}$ are real symmetric positive semidefinite matrices
- The exact deficiency of $A^{(i)}$ is known due to structural mechanics arguments: it is equal to 6 in the two cases
- The computation of the null space of $A^{(i)}$ is required inside the FETI algorithm

Linear elasticity test cases provided by O. Boiteau [EDF]

- Realistic problems on complicated three-dimensional geometries
- Fixed partition in 10 subdomains for the two test cases [METIS]

Pump test case (linear elasticity)

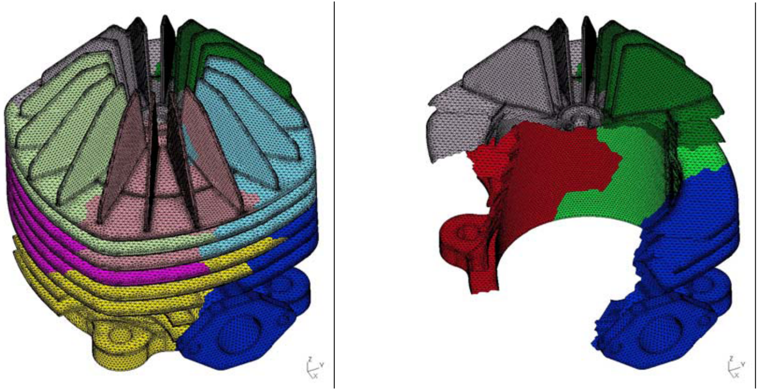


- Tetrahedral mesh (261 520 nodes)
- Global problem has 803 352 degrees of freedom

Pump test case (linear elasticity)

Matrix			MUMPS	
Size	Nz	Def.	Def.	Back. error
84018	2911020	6	6	0.512e-15
65646	2584782	6	6	0.758e-15
83196	2895633	6	6	0.886e-15
83373	2902116	6	6	0.659e-15
84165	2911854	6	6	0.576e-15
82434	2883390	6	6	0.151e-15
75450	2757588	6	6	0.287e-15
81138	2856795	6	6	0.404e-15

Carter test case (linear elasticity)



- Tetrahedral mesh (179 463 nodes)
- Global problem has 530 121 degrees of freedom

Carter test case (linear elasticity)

Matrix			MUMPS	
Size	Nz	Def.	Def.	Back. error
53364	930849	6	6	2.05e-15
53316	930546	6	6	0.493e-15
52647	918957	6	6	0.744e-15
55866	965553	6	6	0.484e-15
53622	935802	6	6	0.410e-15
53403	932916	6	6	0.573e-15
54003	939255	6	6	0.328e-15
53361	931959	6	6	0.425e-15
53712	938718	6	6	0.492e-15

Summary on the numerical results

On these sets of benchmark matrices...

- Reliable (deficiency correctly detected)
- Accurate (backward error close to machine precision)
- Efficient for low and high deficiency

...but can fail !

- On rank-deficient matrices with no large gap in the ratio s_k/s_{k+1} if k is the numerical rank of the matrix and s_k the k-th singular value
- This requires to combine (block) iterative methods with the proposed null space detection algorithm [Master thesis of Bonnement, 2008]

Conclusions

Null space detection in MUMPS

- Experimental option for rank detection and computation of the null space basis (already early developments in 1998)
- Detection of null pivots during numerical factorization
- Delay of small pivots to the root matrix

Singular benchmark matrices

- Singular Matrix Database (references, codes and matrices) [http://www.math.sjsu.edu/singular/matrices/]
- ANR SOLSTICE test matrices in TLSE [about 70 singular sparse matrices, http://www.gridtlse.org/]

- **This work was partially supported by ”Agence Nationale de la Recherche”, through SOLSTICE project ANR-06-CIS6-010**