Computation of the smallest singular value

Bernard Philippe

INRIA-IRISA
Rennes, FRANCE

The decomposition : For $A \in \mathbb{R}^{n\times n}$, there exist two orthogonal matrices U, $V \in \mathbb{R}^{n\times n}$ such that

$$\Sigma = V^T A U$$

is a diagonal matrix with non negative entries.

Notation : $\Sigma = diag(\sigma_1, \cdots, \sigma_n)$ with

$$0 \leq \sigma_1 \leq \cdots \leq \sigma_n.$$

The eigenvalues of $A^T A$ are : ${\sigma_1}^2 \leq \cdots \leq {\sigma_n}^2$.

The eigenvalues of $\begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix}$ are :

$$-\sigma_n \leq \cdots \leq -\sigma_1 \leq 0 \leq \sigma_1 \leq \cdots \leq \sigma_n.$$

For linear systems :
$\sigma_1(A)$ = distance of A to the set of singular matrices

(With the 2-norm)

$$\|A^{-1}\| = \frac{1}{\sigma_1},$$

$\exists \Delta \in \mathbb{R}^{n \times n}$ such that $\|\Delta\| = \sigma_1$ and $A + \Delta$ is singular

$\forall \Delta \in \mathbb{R}^{n \times n}$, $(\|\Delta\| < \sigma_1 \Rightarrow A + \Delta$ is non singular)

For eigenvalue problems : pseudospectrum of $\|A\|$:

$$\begin{aligned} \Lambda_\epsilon(A) &= \{z \in \mathbb{C} : z \text{ is an eigenvalue of } A + \Delta;\ \|\Delta\| \le \epsilon \|A\|\} \\ &= \{z \in \mathbb{C} : \sigma_1(A - zI) \le \epsilon \|A\|\}. \end{aligned}$$

A singular value problem is a symmetric eigenvalue problem.

For all the singular values : $i = 1, \cdots, n$

$$|\sigma_i(A + \Delta) - \sigma_i(A)| \leq \|\Delta\|.$$

If $\|\Delta\| = \epsilon \|A\|$, we get

$$\frac{|\sigma_1(A + \Delta) - \sigma_1(A)|}{\sigma_1(A)} \leq \epsilon \chi(A).$$

For a complete study, see *Matrix Perturbation Theory* [Stewart-Sun]

For a dense matrix on a sequential computer : LAPACK

The computation consists of three steps :

1. Find U_0 and V_0 orthogonal matrices such that $B = {V_0}^T A U_0$ is a bidiagonal matrix (by householder transformations),
2. Compute the singular values of B (from the eigenvalues of $B^T B$) by the implicit QR algorithm or from Sturm sequences,
3. Back transform the singular vectors when necessary.

The total number of operations varies

$$\text{from } 3n^3 + O(n^2) \text{ to } 11n^3 + O(n^2).$$

First approach : SCALAPACK

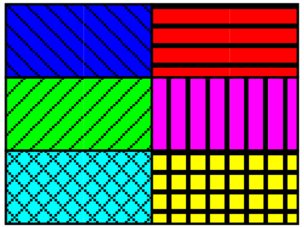
The matrix is distributed by blocks on a grid of processors : tradeoff to be found for the block size :

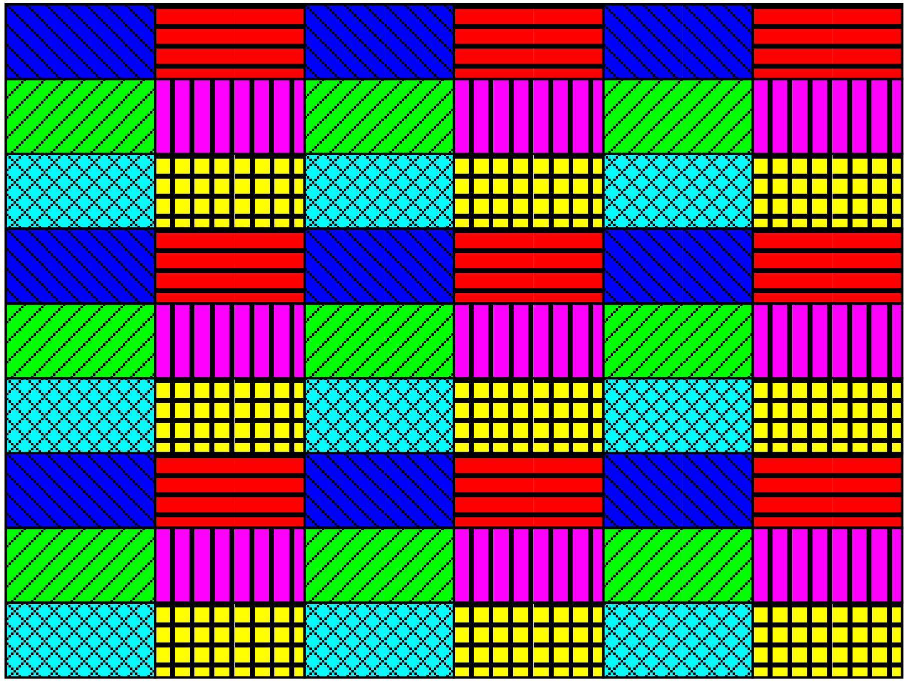
- increasing the size of the blocks reduces the price to pay for communications,
- small size for the blocks improves load balancing.

There exist divide-and-conquer versions of SVD.

block-cyclic distribution - $M \times N$ matrix

grid of $P = P_r \times P_c$ processors





optimizing parameters : block size $r \times c$

Second approach : Jacobi algorithms

- one-sided : implicitely equivalent to the two-sided Jacobi algorithms on $A^T A$; one sweep : $4n^3 + O(n^2)$ flops. Easy to parallelize.

- two-sided : generalization of the classical Jacobi algorithm to the non symmetric case (distinct rotations on left and right sides) ; one sweep : $6n^3 + O(n^2)$ flops.

The one-sided algorithm is better suited to parallelism than the two-sided. It might be competitive for a very large number of processors (of the order of n) (???)

Results of PORTRAIT project :

Research leaded in the framework of a European project (Copernicus project, PORTRAIT CP940682)

1. One-sided Jacobi (Brent and Lu ordering) [Robert]
2. Block two-sided block Jacobi (different orderings) [Bečka,Vajteršic]
3. Bidiagonalization (SCALAPACK) and bisection [Malyshev]

Tests on Intel iPSC2 :

Jacobi algorithms : $nproc = 50$, $n = 1000$:

(1) One-sided Jacobi : 245 s ;

(2) Block two-sided Jacobi : 1072 s ;

SCALAPACK : $nproc = 56$:

(3) $n = 1000$: 18 s ; $n = 2000$: 63 s ; $n = 4000$: 323 s.

Methods based on the matrix-vector multiplication

The basic method is the Lanczos algorithm which provides estimation of the extreme eigenvalues of a symmetric matrix.

Applying Lanczos on the augmented matrix

$$B_a = \begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix}$$

is not efficient since the sought eigenvalues lie in the interior of the spectrum :

$$\cdots \leq -\sigma_p \leq \cdots \leq -\sigma_1 \leq \sigma_1 \leq \cdots \leq \sigma_p \leq \cdots$$

Lanczos can be applied to $B_n = A^T A$ to compute the largest singular values, but not the smallest (lack of precision).

- Lanczos on the augmented matrix built from A^{-1},

- Davidson methods (acceleration of Lanczos which needs a pre-conditioner.

- Trace minimization method (acceleration of the simultaneous iteration),

1. The smallest singular value of A is the inverse of the largest eigenvalue of

$$B = \begin{pmatrix} 0 & A^{-1} \\ A^{-T} & 0 \end{pmatrix} \quad \text{or} \quad \tilde{B} = \begin{pmatrix} 0 & R^{-1} \\ R^{-T} & 0 \end{pmatrix}$$

   where $A = QR$ is the QR decomposition of A

2. Use the Lanczos algorithm to compute the largest eigenvalue of $\tilde{B}$.

3. Use a multifrontal QR decomposition [C. Puglisi]

4. Use a minimum degree reordering or a nested dissection reordering of A to reduce the fill-ins in step 3.

Choose $v_1 \in \mathbb{R}^{2n}$; $v_1 = v_1/||v_1||$
$\beta_1 = 0$
$v_0 = [0]$
For $j = 1...m$

$$\begin{bmatrix} v_{j,1} \\ v_{j,2} \end{bmatrix} = v_j \text{ where } v_{j*} \in \mathbb{R}^n$$

$w_{j,1} = R^{-1} v_{j,2} - \beta_j v_{j-1,1}$

$w_{j,2} = R^{-T} v_{j,1} - \beta_j v_{j-1,2}$

$$w_j = \begin{bmatrix} w_{j,1} \\ w_{j,2} \end{bmatrix}$$

$H_{j,j} = \alpha_j = w_j^t v_j$

$w_j = w_j - \alpha_j v_j$

$H_{j,j+1} = \beta_{j+1} = ||w_j||_2$

$v_{j+1} = w_j/\beta_{j+1}$

End For

$(\lambda_{max}, y) = eig(H_m)$ where $H_m y = \lambda_{max} y$;
$x = Vy$ with $V = [v_1, \cdots, v_m]$

Build the elimination tree
for $k = 1...n$
  if k is a leaf **then**
   F_k is the set of all the rows of the original matrix having their leading element in column k
  else
   F_k is composed by appending the contributing blocks coming from the sons to the rows of the original matrix having their leading element in column k
  End if
  Perform a Householder reflection on F_k
  (The first row of F_k is the kth row of R, the remaining sub-matrix is the contribution block which is sent to the father)
End for

Tests on a workstation :

Matrix dimension	Multifrontal QR	Lanczos	$\sigma_{\min}(A - zI)$
1890×1890	3.7 84%	0.7 16%	4.4
3906×3906	41.8 95%	2.1 5%	43.9
32130×32130	2353 61%	1480 39%	3832

Running Time (s) and percentage of the total time .

We must consider parallel implementations of the numerical algorithms for large matrices.

To compute an eigenpair of a hermitian matrix $B \in \mathbb{R}^{n\times n}$, the Davidson method builds a sequence of subspaces $\{V_m\}$ (${V_m}^T V_m = I$) and approximates the eigenpairs by Ritz pairs obtained from V_m.

$V_m \hookrightarrow V_{m+1}$ by correction of the sought eigenvector. Periodic restarting.

Correction of an approximated eigenvector u : (we assume $u = e_1$)

$$B = \begin{pmatrix} \lambda & \tilde{r}^T \\ \tilde{r} & C \end{pmatrix}$$

We look for y such that : $B \begin{pmatrix} 1 \\ y \end{pmatrix} = (\lambda + \epsilon) \begin{pmatrix} 1 \\ y \end{pmatrix}$

This is equivalent to compute ϵ and y such that $\begin{cases} (C - (\lambda+\epsilon)I)y = -\tilde{r} \\ \epsilon = \tilde{r}^T y \end{cases}$

Near convergence : $\epsilon = O(\|y\|^2)$ and the system becomes $(C - \lambda I)y = -\tilde{r}$ with $\tilde{r}^T y = 0$.

First idea : **Davidson :** [Davidson '75,Crouzeix-BP-Sadkane '94]

Let $\lambda = u^H Bu$ and $r = Bu - \lambda u = \begin{pmatrix} 0 \\ \tilde{r} \end{pmatrix}$

Approximate solution of $(B - \lambda I)\ t_D = -r$

Second approach : **Jacobi-Davidson :** [Sleijpen-Van der Vorst '96]

Approximate solution of $(I - uu^T)(B - \lambda I)(I - uu^T)t_J = -r$
with the constraint $t_J \perp u$.

Algorithm of Davidson (or Jacobi-Davidson) applied to $B = A^T A$ for computing the smallest singular values of A (Block version).

$A \in \mathbb{R}^{n*n}$ and l denotes the block-size.
Choose an initial orthonormal matrix $V_1 = [v_1, \cdots, v_l] \in \mathbb{R}^{n*l}$;

For $k = 1, \cdots$
- **1:** Compute the matrix $U_k = AV_k$;
- **2:** Compute the matrix $W_k = A^T U_k$;
- **3:** Compute the Rayleigh matrix $H_k = V_k^T W_k$;
- **4:** Compute the l smallest eigenpairs $(\nu_{k,i}^2, y_{k,i})_{1 \le i \le l}$ of H_k;
- **5:** Compute the vectors $x_{k,i} = V_k y_{k,i} \quad i = 1, \cdots, l$;
  Let $X_k = [x_{k,1}, \cdots, x_{k,l}]$;
- **6:** Compute the residuals $r_{k,i} = W_k y_{k,i} - \nu_{k,i}^2 x_{k,i}$ for $i = 1, \cdots, l$;
  If convergence then **exit**
- **7:** Compute the new directions $t_{k,i} = C_{k,i}(r_{k,i}) \quad i = 1, \cdots, l$;
- **8:** **If** $dim(V_k) \le m - l$
  then $V_{k+1} = [V_k, t_{k,1}, \cdots, t_{k,l}]$
  where $[t_{k,1}, \cdots, t_{k,l}] = MGS((I - V_k^T V_k)[t_{k,1}, \cdots, t_{k,l}])$;
  else $V_{k+1} = [X_k, t_{k,1}, \cdots, t_{k,l}]$
  where $[t_{k,1}, \cdots, t_{k,l}] = MGS((I - X_k^T X_k)[t_{k,1}, \cdots, t_{k,l}])$;

endfor

1st choice : fixed preconditioning. $C_{k,i} = M^{-1}$ where M is an approximation of $A^T A$ which might be :

- $M = U^T L^T L U$ where $A = LU + E_1$ is obtained by an incomplete LU-factorization of A,
- $M = R^T R$ where $A = QR + E_2$ is obtained by an incomplete QR-factorization of A,
- $M = LL^T$ where $A^T A = LL^T + E_3$ is obtained by an incomplete Cholesky-factorization of $A^T A$.

2nd approach : iterative preconditioning. Fixed number of CG iterations on the following systems :

- Davidson : $A^T A t_D = -r$,
- Jacobi-Davidson : $(I - uu^T)(A^T A - \lambda I)(I - uu^T) t_J = -r$.

(Block size = 1)

Theoretical situation: $M = A^T A$

Davidson $\equiv$ **Lanczos on** $(A^T A)^{-1}$ since they generate the same Krylov subspaces.

(this would be an expensive way to apply Lanczos)

When $M \approx A^T A$:

An estimation of the convergence rate is given by [LAA 97] :

Notations :

σ	:	smallest singular value of A ;
ν_k	:	approximation of σ at step k (without restarting)
θ_k	:	angle between the exact right singular vector and its approx. ;
$\sigma^{'}$	:	second singular value of A ;
κ	:	condition number of A.

Let $\epsilon_k = \frac{\|(M^{-1}-(A^T A)^{-1})r_k\|_2}{\kappa^2 \sin\theta_k}$ be the parameter measuring the quality of the preconditionner.

If $\frac{\sigma^{'2}}{\sigma^2}\epsilon_k \kappa^2 \leq 1$ then

$$0 \leq \frac{\nu_{k+1}^2 - \sigma^2}{\nu_k^2 - \sigma^2} \leq \frac{\sigma^4}{\sigma^{'2}(\sigma^{'2} - \sigma^2)} \left(1 + \tan^2\theta_k\right) \left(1 + O(\frac{\sigma^{'2}}{\sigma^2}\epsilon_k \kappa^4)\right)$$

Characteristics of the test matrices

Matrix	No. of rows	No. of columns	No. of of entries	Description
ADI	374	82	1343	Seismic tomography
ILLC1033	1033	320	4732	Least squares problems in surveying
WELL1850	1850	712	8758	Least squares problems in surveying
HOR131	434	434	4710	Flow network problem
SHERMAN1	1000	1000	3750	3D-Simulation of black oil
PORES3	532	532	3474	Reservoir modelling problem

BLSVD : Block Lanczos [Berry],

DAVID : Davidson with diagonal preconditionning,

DAVIDLU : Davidson with incomplete LU factorization of A,

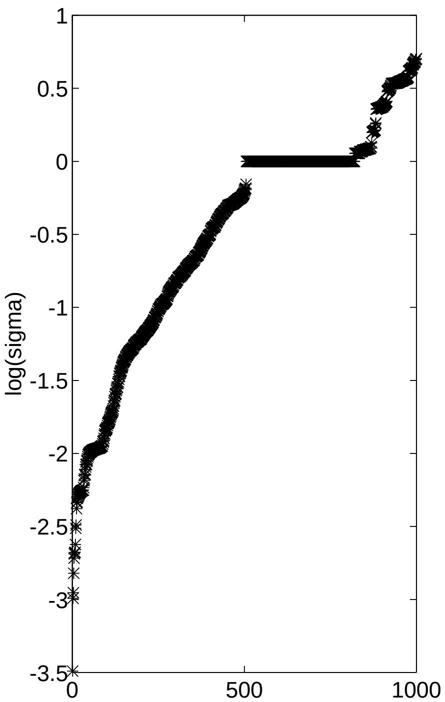
DAVIDIC : Davidson with incomplete Cholesky factorization of A^TA,

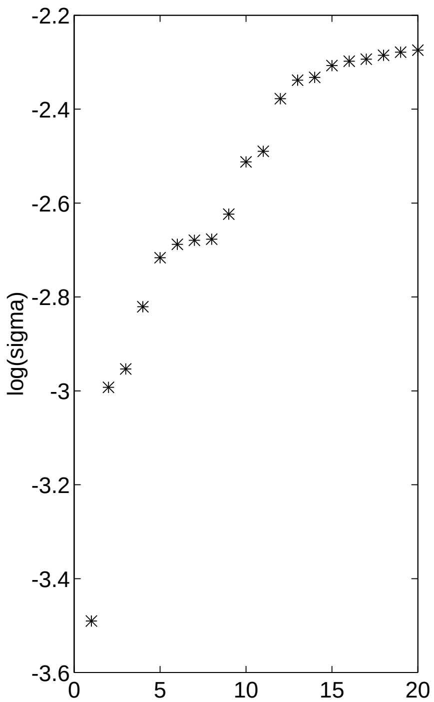
DAVIDQR : Davidson with incomplete QR factorization of A.

Number of matrix-vector multiplications before convergence

Matrix	BLSVD	DAVID	DAVIDLU	DAVIDIC	DAVIDQR
ADI	2180	1634		560-634	554-628
ILLC1033	F	F		504-2440	320-1816
WELL1850	F	F		760-848	760-848
HOR131	F	F	324-1110	218-1248	F
SHERMAN1	F	F	938-1106	468-1836	550-850
PORES3	F	F	1186	F	F

F : failure





Singular values of SHERMAN1 (left). The twenty smallest ones (right)

Algorithm	tol	fill-in	mat-vec
BLSVD			*
DAVID			*
DAVIDLU	1E-4	18204	938
DAVIDLU	5E-4	10473	978
DAVIDLU	1E-3	8304	1106
DAVIDIC	1E-5	41068	468
DAVIDIC	1E-4	32431	550
DAVIDIC	1E-3	24588	1836
DAVIDQR	1E-5	41065	550
DAVIDQR	1E-4	32130	850

Performances,

nvalues = 20,

$*$: maxit > 150

[Sameh-Wisniewski '82 ; Sameh-Tong '00] Acceleration of the simultaneous method : To compute the smallest singular values of a positive definite matrix A.

At step k, the following problem is solved : given x_k

$$\begin{array}{l}\text{Minimize } trace[(X_k - \Delta_k)^T A (X_k - \Delta_k)] \\ \text{subject to } (X_k)^T \Delta_k = 0,\end{array}$$

which implies

$$trace[(X_{k+1})^T A (X_{k+1})] \leq trace[(X_k)^T A (X_k)].$$

When a shifting strategy is considered, the resulting correction is close to the Jacobi-Davidson strategy.

Algorithm of Trace Minimization applied to $B = A^T A$ for computing the smallest singular values of A.

$A \in \mathbb{R}^{n*n}$ and s denotes the block-size.
Choose an initial orthonormal matrix $V_1 = [v_1, \cdots, v_s] \in \mathbb{R}^{n*l}$;

For $k = 1, \cdots$

1: Compute the matrix $U_k = AV_k$;
2: Compute the matrix $W_k = A^T U_k$;
3: Compute the Rayleigh matrix $H_k = V_k^T W_k$;
4: Compute the eigenpairs (Y_k, Θ_k) of H_k;
5: Compute the Ritz vectors $X_k = V_k Y_k$;
6: Compute the residuals $R_k = W_k y_k - X_k {\Theta_k}^2$;
If convergence then **exit**
7: Solve the system (1)
8: $V_{k+1} =$MGS$(X_k - \Delta_k)$;

endfor

$$(PBP)\Delta_k = PBX, \quad {X_k}^T \Delta_k = 0 \text{ where } P = I - X_k {X_k}^T. \tag{1}$$

Matrix allocation :

$$A = \begin{bmatrix} A^0 \\ \hline A^1 \\ \hline \vdots \quad \vdots \quad \vdots \quad \vdots \\ \hline A^{p-1} \end{bmatrix} \qquad v = \begin{bmatrix} v_0 \\ \hline v_1 \\ \hline \vdots \\ \hline v_{p-1} \end{bmatrix} \qquad \begin{array}{l} \longrightarrow Node\ 0 \\ \longrightarrow Node\ 1 \\ \longrightarrow \vdots \\ \longrightarrow Node\ (p-1) \end{array}$$

Data partition for the multiplications by A and A^T :

$$A = \left[\begin{array}{c|c|c|c|c} A_{loc0} & A^1_{ext0} & \cdots & \cdots & A^{p-1}_{ext0} \\ \hline A^0_{ext1} & A_{loc1} & \cdots & \cdots & A^{p-1}_{ext1} \\ \hline \vdots & & & \vdots & \vdots \\ \hline \vdots & & & \vdots & \vdots \\ \hline A^0_{ext(p-1)} & A^1_{ext(p-1)} & \cdots & \cdots & A_{loc(p-1)} \end{array}\right] \qquad v = \begin{bmatrix} v^0_{loc} \\ \hline v^1_{loc} \\ \hline \vdots \\ \hline \vdots \\ \hline v^{p-1}_{loc} \end{bmatrix} \qquad \begin{array}{l} \longrightarrow Node\ 0 \\ \longrightarrow Node\ 1 \\ \longrightarrow \vdots \\ \longrightarrow \vdots \\ \longrightarrow Node\ (p-1) \end{array}$$

k = processor number ; p = number of processors.

Matrix-by-vector multiplication Av :

Step 1: Send the needed components of v^k_{loc} to other processors.
Step 2: Compute $w^k = A_{lock} v^k_{loc}$.
Step 3: Receive the needed components of v^j_{loc} for $j \neq k$ from other processors.
Step 4: Compute $w^k = w^k + \sum_{j \neq k} A^j_{ext} . v^j_{loc}$.

Matrix transpose-by-vector multiplication $A^T v$:

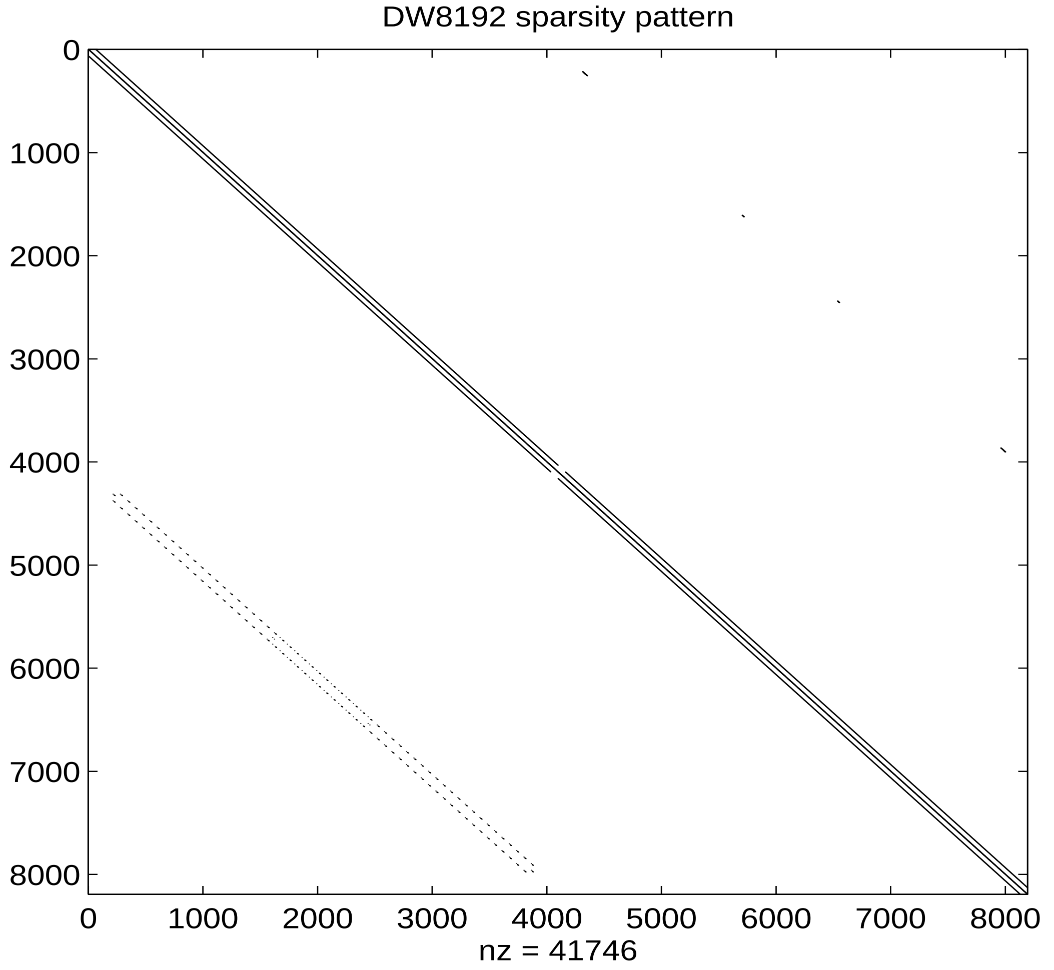
Step 1: Compute $y^j = A^{jT}_{ext} v^k_{loc}$ for $j \neq k$.
Step 2: Send y^j for $j \neq k$ to node number j.
Step 3: Compute $w^k = A^T_{lock} v^k_{loc}$.
Step 4: Receive $y^{j'}$ with $j' \neq k$ from other processors.
step 5: Compute $w^k = w^k + \sum_{j' \neq k} y^{j'}$.

DW8192 : $n = 8192$, $n_z = 41746$;

It comes from the discretization of the Helmotz equations :

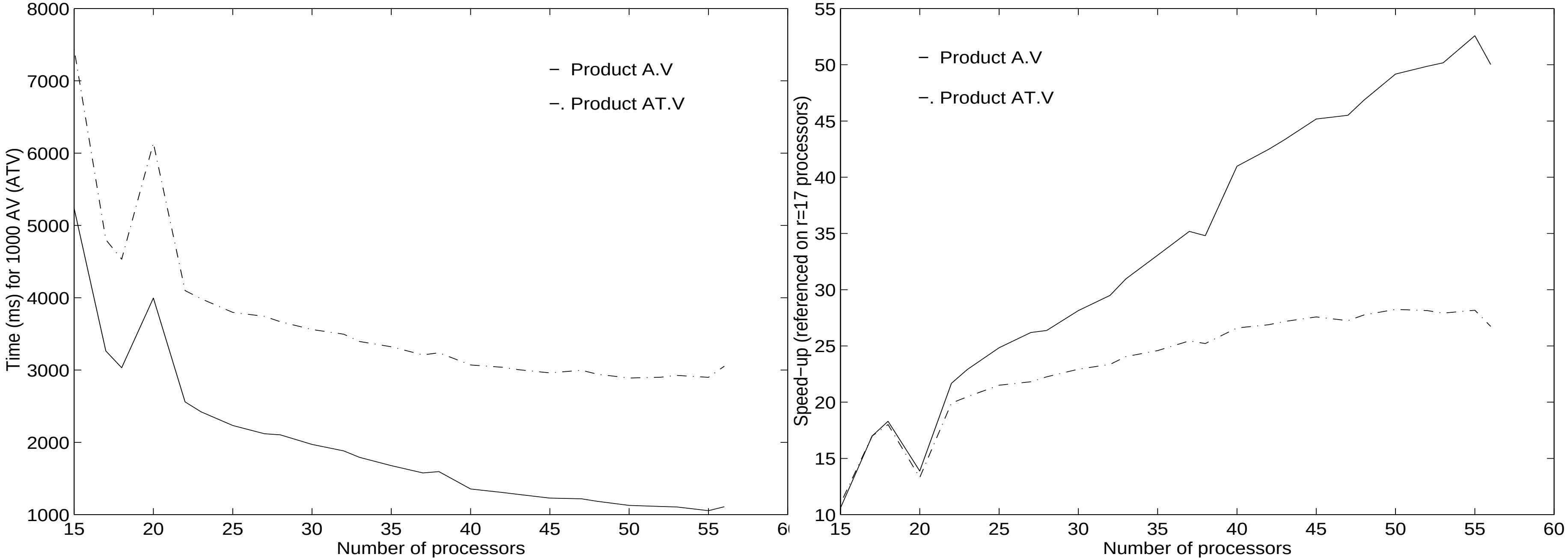
$$\begin{aligned}\nabla^2 H_x + k^2 n^2(x,y) H_x &= \beta^2 H_x,\\ \nabla^2 H_y + k^2 n^2(x,y) H_y &= \beta^2 H_y.\end{aligned}$$

Test matrix : DW8192 [Bai and al.]



Structure of DW8192

Computer = Intel iPSC2



Time for 1000 $Av/A^H v$

Speed-up for $Av/A^H v$

We consider now a parallel computation of the smallest singular value of a given operator.

The method which is used is the method already introduced :

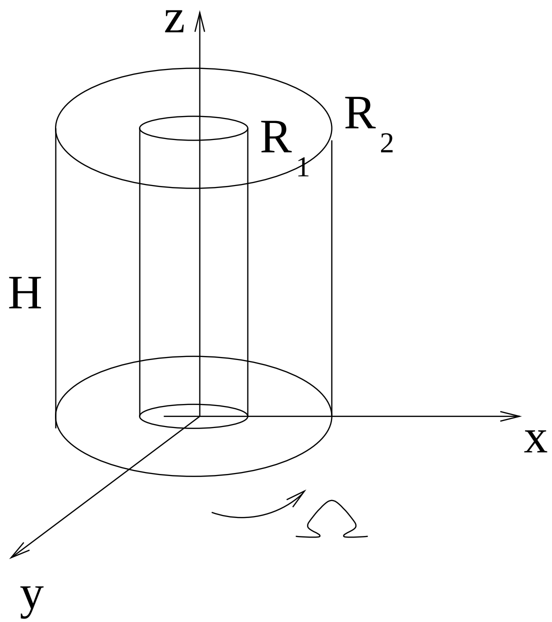
Lanczos on the augmented matrix built from the inverse of the triangular factor in the QR factorization of A.

Parallelism occurs in the QR factorization in the triangular factorization and in the Lanczos process.

The experiment is done on a network of workstations.

The problem has been defined and studied by O. Daube, P. Le Quéré and L. Tuckermann (LIMSI, Orsay).

We consider the geometrical configuration :



Fluid between the two rotating cylinders.

Under a compact form, the Navier-Stokes equation is :

$$\frac{\partial X}{\partial t} = (N + D)X$$

where N is the non-linear part of the operator, D the diffusion terms.

Let $X_0 = (\omega_0, v_{\phi_0})^T$ a steady state :

$$(N + D)X_0 = 0$$

We denote $\tilde{N}$ the linearization of operator N. For a given perturbation x such that $X = X_0 + x$, we consider the stability of the ODE :

$$\frac{\partial x}{\partial t} = (\tilde{N} + D)x$$

$\longrightarrow$ Are there eigenvalues with positive real parts ?

$$\begin{pmatrix} (\tilde{N}+D)_\omega \\ \hline (\tilde{N}+D)_{v_\phi} \end{pmatrix} = \left[\left(\begin{array}{c|c} A_1 L & A_2 \\ \hline 0 & A_3 \end{array} \right) + \left(\begin{array}{c|c} A_1 B + V_1 & 0 \\ \hline V_2 & 0 \end{array} \right) \right] \left(\begin{array}{c|c} L^{-1} & 0 \\ \hline 0 & I \end{array} \right)$$

Where

L	:	velocity potential ($\omega = L\psi$)
V_1, V_2	:	velocity components
B	:	boundary conditions
$A_{1,2,3}$	:	diffusion terms

Matrix	Dimension	Shape	$N_z(A)$	$N_z(L)$
Hydro1	1890×1890	AL^{-1}	23078	5514
Hydro2	3906×3906	AL^{-1}	49030	11530
Hydro3	32130×32130	AL^{-1}	411590	95754

Machines	Single PentiumIII processor running at 600MHz
Memory	128MB + 600MB Swap space
Internal Cache	512KB
OS	Red Hat 6.0, kernel 2.2.5-15
Compiler	gcc-2.95.2
Libraries	mpich-1.2.0
	f2c translated LAPACK 1.2
	lsblaspii.2b_11.99.a (`http://www.cs.utk.edu/~ghenry/distrib`)
Network card	3Com 10-100BaseT
HUB	3Com 10BaseT

A low cost general purpose machine configuration; a low bandwidth network.

Matrix	1 Proc.	3 Procs	Speedup	Efficiency
	Multifrontal QR decomposition			
Hydro1	3.7	3.62	1.02	0.34
Hydro2	41.8	30.83	1.35	0.45
Hydro3	2353	1410	1.66	0.55
	Lanczos algorithm			
Hydro1	0.7	5.24	0.13	0.04
Hydro2	2.1	15.62	0.14	0.05
Hydro3	1479	647	2.29	0.76
	$\sigma_{\text{min}}(A - zI)$			
Hydro1	4.4	8.86	0.49	0.16
Hydro2	43.9	46.45	0.94	0.32
Hydro3	3822	2057	1.85	0.61

1 Worker = 3 Procs = 1 Group Master + 2 Slaves

There exist already efficient parallel codes to compute the smallest singular value of a matrix.

However, improvements are needed for considering very large matrices.

How to speed-up the convergence ? How to obtain reliable codes with no exact factorization ?

Davidson, Jacobi-Davidson and Trace Minimization might be the methods of choice.