# Matrices, moments and quadrature with applications

Gérard MEURANT

February 29 2008

# A little bit of history


G. Dahlquist, S.C. Eisenstat and G.H. Golub,
*Bounds for the error of linear systems of equations using the theory of moments,*
J. Math. Anal. Appl., v 37, (1972), pp 151–166

G. Dahlquist, G.H. Golub and S.G. Nash,
*Bounds for the error in linear systems.*
In Proc. of the Workshop on Semi–Infinite Programming,
R. Hettich Ed., Springer (1978), pp 154–172

The first paper considers a sequence of Krylov vectors $r^{i+1} = Ar^i,\ i = 0, 1, \ldots k - 1$ (which may not be such a good idea numerically) and looks at the moments

$$(r^i, r^j) = (A^{i+j} r^0, r^0) = \mu_{i+j}$$

Assume $A$ is SPD, given $\mu_i,\ i = 0, \ldots, 2k$ how do we compute bounds for $\mu_{-2} = (A^{-2} r^0, r^0) = \|e\|^2$ ?

where $Ax = b,\ x = \xi + e,\ r^0 = b - A\xi$

If we know the eigenvalues of $A$ we can use linear programming, otherwise (as you will see) we can use Gauss quadrature

This paper applies this to the Jacobi iterative method

The second paper reviews the same material, uses the Lanczos algorithm to compute the bounds and proposes to use CG to improve the estimate $\xi$

This paper also gives a (complicated) proof of an identity for a norm of the CG error $\|e^k\|_A^2$ (that we will see later)

In these 2 papers, we have the basis for 30 years of research !

There are beautiful relationships between matrices, moments, orthogonal polynomials, quadrature, . . .

# Examples of applications

Solve

$$Ax = b$$

Let $r^k = b - Ax^k$ be the residual vector
Since $\|r^k\|$ is often misleading for stopping iterative methods, it is of interest to obtain bounds or estimates of norms of the error

$$\epsilon^k = x - x^k$$

We have

$$A\epsilon^k = r^k$$

Therefore,

$$\|\epsilon^k\|_A^2 = (A\epsilon^k, \epsilon^k) = (A^{-1}r^k, r^k) = (r^k)^T A^{-1} r^k$$

$$\|\epsilon^k\|^2 = (r^k)^T A^{-2} r^k$$

We have to consider bilinear forms $u^T f(A) u$ with $u = r^k$ and $f(x) = 1/x$ or $1/x^2$

# Examples of applications 2

Ill-posed problems

We want to solve

$$Ax = y$$

where $A$ ($m \times n$ matrix) arises from the discretization of an inverse problem (Fredholm integral equation of the first kind)

Generally, the right hand side is corrupted with (an unknown) noise

$$y = \hat{y} + e$$

The matrix $A$ has very small singular values

Tikhonov regularization

$$(A^T A + \nu I)x = A^T y$$

How to choose the regularization parameter $\nu$ ?

Generalized Cross Validation (GCV) :
see G.H. Golub, M. Heath and G. Wahba (1979)

find the minimum of

$$G(\nu) = \frac{\frac{1}{m}\|(I - A(A^T A + \nu I)^{-1} A^T)y\|^2}{(\frac{1}{m}\mathrm{tr}(I - A(A^T A + \nu I)^{-1} A^T))^2}$$

L-curve : find the “corner” of $\log(\|x(\nu)\|)$ as a function of $\log(\|y - Ax(\nu)\|)$

Easy to solve if we know the SVD of $A$, not feasible if the matrix is large in these methods and others we need to compute

$$y^T A(A^T A + \nu I)^{-p} A^T y$$

and/or

$$y^T (AA^T + \nu I)^{-p} y$$

$p = 1, 2, 3, 4,$ for given $\nu$ and $y$

# Examples of applications 3

Total least squares (TLS)
see G.H. Golub and C. Van Loan (1980)

$$(A + E)x = b + r, \quad A : m \times n$$

$$\begin{pmatrix} A & b \end{pmatrix} \begin{pmatrix} x \\ -1 \end{pmatrix} + \begin{pmatrix} E & r \end{pmatrix} \begin{pmatrix} x \\ -1 \end{pmatrix} = 0$$

$$\left(C + F\right) z = 0$$

Find

$$\min_z \frac{\|Cz\|_2}{\|z\|_2}$$

We have $C^T C z = \sigma_{n+1} z$ ($\sigma_{n+1}$ smallest singular value of $(A \quad b)$) and the secular equation

$$\sigma_{n+1}^2 = b^T b - b^T A(A^T A - \sigma_{n+1}^2 I)^{-1} A^T b$$

This is the same type of function as in GCV

# Other examples

## Rank one change for eigenvalues

$$Ax = \lambda x, \quad (A + c^T c)y = \mu y$$

Secular equation :

$$1 + c^T (A - \mu I)^{-1} c = 0$$

## Quadratic constraint

$$A = A^T, \quad \min_x x^T A x - 2b^T x, \; x^T x = \alpha^2$$

Secular equation :

$$b^T (A - \mu I)^{-2} b = \alpha^2$$

# Relation to quadrature

Suppose $A$ is symmetric positive definite

$$A = Q\Lambda Q^T, \quad Q^T Q = I$$

Consider

$$u^T f(A)u, \quad f(A) = Qf(\Lambda)Q^T$$

We write the quadratic form as a Riemann–Stieltjes integral

$$I[f] = u^T f(A)u = \int_a^b f(\lambda)\, d\alpha(\lambda)$$

the measure $\alpha$ is piecewise constant
if $y = Q^T u$

$$\alpha(\lambda) = \begin{cases} 0 & \text{if } \lambda < a = \lambda_1, \\ \sum_{j=1}^{i} y_j^2 & \text{if } \lambda_i \le \lambda < \lambda_{i+1}, \\ \sum_{j=1}^{n} y_j^2 & \text{if } b = \lambda_n \le \lambda \end{cases}$$

$\lambda_i$ are the eigenvalues of $A$ that we usually don't know

We would like to approximate or to bound the integral by using Gauss quadrature rules

# Gauss quadrature

$$I[f] = \int_a^b f(\lambda)\, d\alpha(\lambda) = \sum_{j=1}^{N} w_j f(t_j) + \sum_{k=1}^{M} v_k f(z_k) + R[f]$$

the weights $[w_j]_{j=1}^N, [v_k]_{k=1}^M$ and the nodes $[t_j]_{j=1}^N$ are unknowns and the nodes $[z_k]_{k=1}^M$ are prescribed

$$R[f] = \frac{f^{(2N+M)}(\eta)}{(2N+M)!} \int_a^b \prod_{k=1}^{M}(\lambda - z_k) \left[\prod_{j=1}^{N}(\lambda - t_j)\right]^2 d\alpha(\lambda)$$

$$a < \eta < b$$

Gauss rule : $M = 0$ no prescribed nodes

Suppose $f^{(2n)}(\xi) > 0, \ \forall n, \ \forall \xi, \ a < \xi < b$, then

$$L_G[f] = \sum_{j=1}^{N} w_j^G f(t_j^G)$$

$$L_G[f] \leq I[f]$$

Gauss–Radau rule : $M = 1, z_1 = a$ or $z_1 = b$

Suppose $f^{(2n+1)}(\xi) < 0, \forall n, \ \forall \xi, a < \xi < b$, then

$$U_{GR}[f] = \sum_{j=1}^{N} w_j^a f(t_j^a) + v_1^a f(a), \quad z_1 = a$$

$$L_{GR}[f] = \sum_{j=1}^{N} w_j^b f(t_j^b) + v_1^b f(b), \quad z_1 = b$$

$$L_{GR}[f] \le I[f] \le U_{GR}[f],$$

Gauss–Lobatto rule : $M = 2, z_1 = a, z_2 = b$

Suppose $f^{(2n)}(\xi) > 0$, $\forall n$, $\forall \xi, a < \xi < b$, then

$$U_{GL}[f] = \sum_{j=1}^{N} w_j^{GL} f(t_j^{GL}) + v_1 f(a) + v_2 f(b)$$

$$I[f] \leq U_{GL}[f],$$

# Computation of nodes and weights

Relation to orthogonal polynomials, see G.H. Golub and J.H. Welsch (1969)

$$\int_a^b p_i(\lambda) p_j(\lambda)\ d\alpha(\lambda) = \begin{cases} 1 & \textit{if}\ \text{i=j} \\ 0 & \textit{otherwise} \end{cases}$$

They satisfy a three–term recurrence

$$\gamma_j p_j(\lambda) = (\lambda - \omega_j) p_{j-1}(\lambda) - \gamma_{j-1} p_{j-2}(\lambda), \quad j = 1, 2, \ldots, N$$

In matrix form

$$\lambda P(\lambda) = J_N P(\lambda) + \gamma_N p_N(\lambda) e_N$$

$$P(\lambda)^T = [p_0(\lambda)\ p_1(\lambda) \cdots\ p_{N-1}(\lambda)]$$

$$J_N = \begin{pmatrix} \omega_1 & \gamma_1 & & & \\ \gamma_1 & \omega_2 & \gamma_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \gamma_{N-2} & \omega_{N-1} & \gamma_{N-1} \\ & & & \gamma_{N-1} & \omega_N \end{pmatrix}$$

The nodes of the Gauss rule are the eigenvalues of $J_N$ and the weights are the squares of the first elements of the normalized eigenvectors

To obtain the Gauss–Radau rule ($M = 1$), we extend the matrix $J_N$ such that it has one prescribed eigenvalue ($a$ or $b$)

This is an inverse eigenvalue problem, see G.H. Golub (1973)

$$\hat{J}_{N+1} = \begin{pmatrix} J_N & \gamma_N e_N \\ \gamma_N (e_N)^T & \omega_{N+1} \end{pmatrix}$$

We compute $\omega_{N+1}$ by

$$\omega_{N+1} = a - \gamma_N \frac{p_{N-1}(a)}{p_N(a)}$$

$$(J_N - aI)\delta(a) = \gamma_N^2 e_N, \quad \omega_{N+1} = a + \delta_N(a)$$

We do something similar for Gauss–Lobatto

$$(J_N - aI)\delta = e_N, \quad (J_N - bI)\mu = e_N$$

$$\begin{pmatrix} 1 & -\delta_N \\ 1 & -\mu_N \end{pmatrix} \begin{pmatrix} \omega_{N+1} \\ \gamma_N^2 \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}$$

We can compute the nodes and weights by using Golub and Welsch, but this is not always necessary

We have

$$\sum_{l=1}^{N} w_l f(t_l) = (e_1)^T f(J_N) e_1$$

Sometimes, we can compute the (1,1) element of $f(J_N)$ efficiently (example : $f(x) = 1/x$)

# Computation of the orthogonal polynomials

Suppose $\|u\| = 1$ then the matrix $J_N$ is computed by $N$ iterations of the Lanczos algorithm starting from $v^1 = u$

$$\gamma_k = \|\tilde{v}^k\|,$$

$$v^k = \frac{\tilde{v}^k}{\eta_k},$$

$$\omega_k = (v^k, Av^k) = (v^k)^T Av^k,$$

$$\tilde{v}^{k+1} = Av^k - \omega_k v^k - \gamma_k v^{k-1}.$$

# The algorithm to compute bounds

Suppose the derivatives of $f$ have constant signs, then

- do Lanczos iterations from $u/\|u\|$ to compute $J_i$
- compute $e_1^T f(J_i) e_1$ or $e_1^T f(\hat{J}_i) e_1$ to obtain bounds

# Bilinear forms

To estimate $u^T f(A) v$ when $u \neq v$ we can use

- $u^T f(A) v = [(u+v)^T f(A)(u+v) - (u-v)^T f(A)(u-v)]/4$
- the non-symmetric Lanczos algorithm
- the block Lanczos algorithm

see G.H. Golub and G. Meurant (1994)

Reprinted in

Milestones in Matrix Computations, the selected works of Gene H. Golub with commentaries, R.H. Chan, C. Greif and D.P. O'Leary Eds, Oxford University Press, (2007)

# The conjugate gradient algorithm

What to do for CG ?

It does not make sense to do Lanczos iterations starting from $r^k/\|r^k\|$

However, since $A\epsilon^k = r^k = r^0 - AV_k y^k$ and $J_k y^k = \|r^0\|^2 e^1$

$$\|\epsilon^k\|_A^2 = \|r^0\|^2[(J_n^{-1}e^1, e^1) - (J_k^{-1}e^1, e^1)]$$

Note that $\|r^0\|^2(J_n^{-1}e^1, e^1) = (A^{-1}r^0, r^0)$

Hence, $\|\epsilon^k\|_A^2$ is the remainder of Gauss quadrature for the Riemann–Stieltjes integral $(A^{-1}r^0, r^0)$

$$\|\epsilon^k\|_A^2 = \frac{1}{\xi_k^{2k+1}} \sum_{i=1}^{n} \left[ \prod_{j=1}^{k} (\lambda_i - \theta_j^{(k)})^2 \right] (r^0, q^i)^2,$$

where $q^i$ is the $i$th eigenvector of $A$ corresponding to $\lambda_i$, $\theta_j^{(k)}$ Ritz values (eigenvalues of $J_k$), $a \le \xi \le b$

The formula for $\|\epsilon^k\|_A^2$ is equivalent to a formula proved in Hestenes and Stiefel (1952)

$$\|\epsilon^k\|_A^2 = \sum_{j=k}^{n-1} \gamma_j \|r^k\|^2$$

$\gamma_j$ is a CG parameter

# Approximation of the norm of the error

Of course, we do not know $(J_n^{-1}e^1, e^1)$

Let $d$ be a positive integer, at iteration $k$ we use

$$\|\epsilon^{k-d}\|_A^2 \simeq \|r^0\|^2[(J_k^{-1}e^1, e^1) - (J_{k-d}^{-1}e^1, e^1)]$$

or

$$\|\epsilon^{k-d}\|_A^2 \simeq \sum_{j=k-d+1}^{k} \gamma_j \|r^k\|^2$$

The first formula can also be used with Gauss–Radau or Gauss–Lobatto rules to obtain upper bounds

If we want a lower bound for $\|e^{k-d}\|_A^2$ we use the H-S formula

If we have an estimate of the smallest eigenvalue, we compute $(\hat{J}_k^{-1})_{1,1}$ incrementally by using the Sherman–Morrison formula to obtain an upper bound, see Meurant (1997, 1999)

Strakŏs and Tichý (2002) have proved that these formulas work also in finite precision arithmetic

Arioli (2004) and Arioli, Loghin and Wathen (2005) have used these techniques to provide reliable stopping criteria for finite element problems

# Elements of $f(A)$

Finite difference approximation of the Poisson equation on a $16 \times 16$ mesh

We look for $(A^{-1})_{125,125}$ whose value is 0.5604

| rule | Nit=2 | 4 | 6 | 8 | 10 | 20 |
|---|---|---|---|---|---|---|
| G | 0.3333 | 0.4337 | 0.4920 | 0.5201 | 0.5378 | 0.5600 |
| G–R $b_L$ | 0.3639 | 0.4514 | 0.5006 | 0.5255 | 0.5414 | 0.5601 |
| G–R $b_U$ | 1.5208 | 0.8154 | 0.6518 | 0.5925 | 0.5730 | 0.5604 |
| G–L | 2.1011 | 0.8983 | 0.6803 | 0.6012 | 0.5760 | 0.5604 |

Block Lanczos, $m = 6, n = 36$
$(A^{-1})_{2,1} = 0.1040$

| rule | Nit=2 | 4 | 6 | 8 |
|---|---|---|---|---|
| G | 0.0894 | 0.1008 | 0.1033 | 0.1040 |
| G–R $b_L$ | 0.0931 | 0.1017 | 0.1035 | 0.1040 |
| G–R $b_U$ | 0.1257 | 0.1059 | 0.1042 | 0.1040 |
| G–L | 0.1600 | 0.1079 | 0.1041 | 0.1041 |

# Larger example

Block Lanczos, $m = 30$, $n = 900$

$(A^{-1})_{1,1} = 0.302346$, $(A^{-1})_{2,2} = 0.344408$, $(A^{-1})_{2,1} = 0.104693$

Results after 10 block iterations for Gauss :

$$\begin{pmatrix} 0.3021799137963044 & 0.1043616568803480 \\ 0.1043616568803480 & 0.3437475221129595 \end{pmatrix}$$

Results after 10 block iterations for Gauss–Radau with exact eigenvalues :

$$\begin{pmatrix} 0.3022010722636479 & 0.1044036770842950 \\ 0.1.044036770842950 & 0.3438314340061286 \end{pmatrix}$$

$$\begin{pmatrix} 0.3039414302035057 & 0.1078375193911064 \\ 0.1078375193911064 & 0.3506698361080970 \end{pmatrix}$$

# Exponential of $A$

Finite difference approximation of the Poisson equation on a $30 \times 30$ mesh

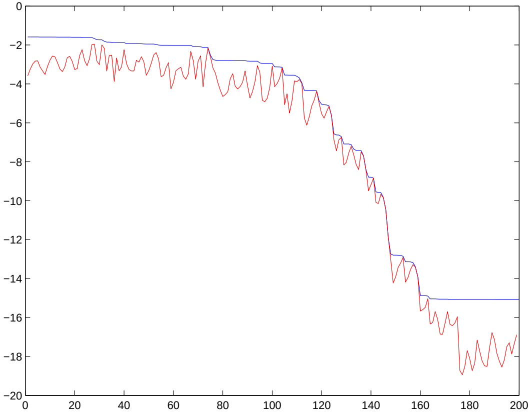
We look for $(exp(A))_{18,18}$ whose value is 197.9724768113708 using Gauss quadrature

After 5 iterations : 197.9|599617609761

After 10 iterations : 197.9724768113|530

# CG error norm

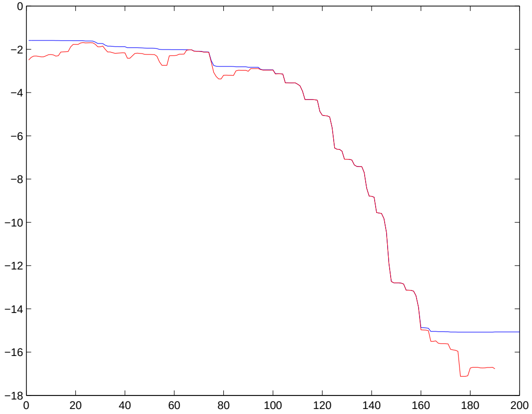
Matrix Bcsstk01, $n = 48$




$A$ norm (blue), Gauss quad $d = 1$ (red)

# CG error norm 2

Matrix Bcsstk01, $n = 48$



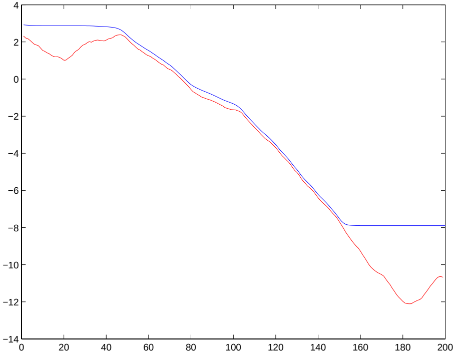

$A$ norm (blue), Gauss quad $d = 10$ (red)

# PCG error norm

Elliptic problem, diff. coeff.=

$$\frac{1}{(2 + p \sin \frac{x}{\eta})(2 + p \sin \frac{y}{\eta})}$$

$p = 1.99$ and $\eta = 0.01$, mesh $100 \times 100$, IC(0)




$A$ norm (blue), Gauss quad $d = 1$ (red)

# Ill–posed problems

The matrix to consider is $B = A^T A$ or $B = AA^T$

We use the Golub–Kahan bidiagonalization algorithm (1965) which produces a lower bidiagonal matrix $C_k$

We have to compute quantities like

$$I[C_k] = (e^1)^T (C_k^T C_k + \nu I)^{-p} e^1$$

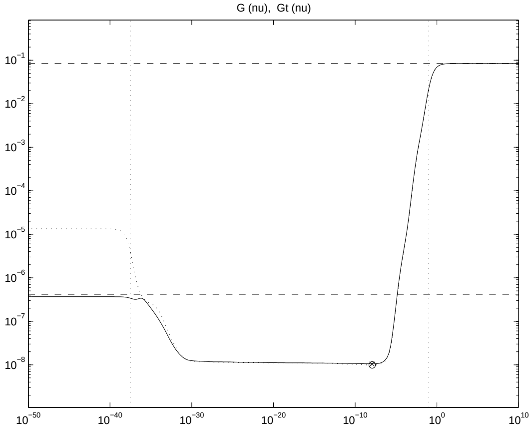
This can be done by solving least squares problems or by using the SVD of $C_k$

For computing the trace we use a result of Hutchinson (1989)

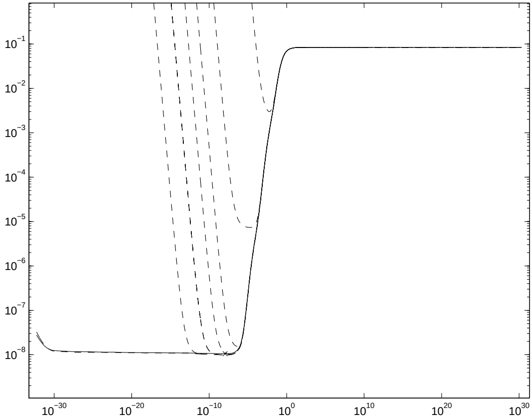
$$tr[(AA^T + \nu I)^{-1}] \approx \frac{1}{q} \sum_{i=1}^{q} (u^i)^T (AA^T + \nu I)^{-1} u^i$$

where $u^i$ are random vectors. In practice, $i = 1$

# Problem : Baart, Regutools (Hansen), $n = 100$




functions $G$ and $\tilde{G}$, `Baart`, $m = n = 100, \|e\| = 10^{-3}$




functions $G$, $\tilde{G}$ and upper bound, Baart, $m = n = 100, \|e\| = 10^{-3}$

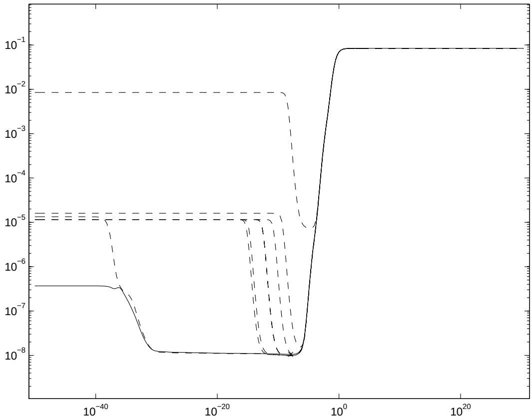
Golub et Von Matt (1997) compute also a lower bound and stop when the maximum of the lower bound is larger than the minimum of the upper bound after $\lceil 3 \log \min(m, n) \rceil$ iterations

Notice that

- We do not want to compute the bounds for too many points $\nu_i$
- It would be nice to know that the upper bound has "converged" before looking for the minimum
- The upper bound does not have the right asymptotic behavior when $\nu \to 0$

We modify the function for the upper bound, instead of $p(\nu)/q(\nu)^2$, we consider

$$\frac{p(\nu)}{q(\nu)^2 + \|y\|^2}$$




functions $G$, $\tilde{G}$ and modified upper bound, Baart,
$m = n = 100, \|e\| = 10^{-3}$

- We test the convergence of the upper bounds for a small value of $\nu$
- We compute the minimum
- We test its convergence
- Functions values are computed using SVDs of $C_k$

We compare to the Von Matt's implementation

# Comparisons

Baart

| | $\|e\|$ | nb it | $\nu/m$ | $f$ min | $f$ max | $f$ total |
|---|---|---|---|---|---|---|
| GVM | $10^{-7}$ | 17 | $9.6482\ 10^{-15}$ | 494 | 613 | 1107 |
| | $10^{-5}$ | 14 | $9.7587\ 10^{-12}$ | 125 | 132 | 257 |
| | $10^{-3}$ | 14 | $1.2018\ 10^{-8}$ | 130 | 123 | 253 |
| | $10^{-1}$ | 14 | $1.0336\ 10^{-7}$ | 128 | 126 | 254 |
| | 10 | 14 | $8.8817\ 10^{-8}$ | 127 | 119 | 246 |
| GM | $10^{-7}$ | 12 | $1.0706\ 10^{-14}$ | 436 | 0 | 436 |
| | $10^{-5}$ | 12 | $1.0581\ 10^{-11}$ | 437 | 0 | 437 |
| | $10^{-3}$ | 8 | $1.3077\ 10^{-8}$ | 293 | 0 | 293 |
| | $10^{-1}$ | 7 | $1.1104\ 10^{-7}$ | 294 | 0 | 294 |
| | 10 | 7 | $9.1683\ 10^{-8}$ | 294 | 0 | 294 |

# Comparisons 2

## ILaplace

| | $\|e\|$ | nb it | $\nu/m$ | $f$ min | $f$ max | $f$ total |
|---|---|---|---|---|---|---|
| GVM | $10^{-7}$ | 112 | $2.1520\ 10^{-15}$ | 12438 | 10216 | 22654 |
| | $10^{-5}$ | 47 | $5.2329\ 10^{-12}$ | 4242 | 3428 | 7670 |
| | $10^{-3}$ | 18 | $2.2111\ 10^{-8}$ | 620 | 541 | 1161 |
| | $10^{-1}$ | 14 | $1.9484\ 10^{-5}$ | 120 | 125 | 245 |
| | 10 | 14 | $6.5983\ 10^{-3}$ | 124 | 126 | 250 |
| GM | $10^{-7}$ | 58 | $4.2396\ 10^{-14}$ | 5239 | 0 | 5239 |
| | $10^{-5}$ | 28 | $5.4552\ 10^{-11}$ | 1453 | 0 | 1453 |
| | $10^{-3}$ | 17 | $2.3046\ 10^{-8}$ | 440 | 0 | 440 |
| | $10^{-1}$ | 15 | $2.0896\ 10^{-5}$ | 293 | 0 | 293 |
| | 10 | 10 | $6.8436\ 10^{-3}$ | 296 | 0 | 296 |

# Comparisons 3

ILaplace

| | $\|e\|$ | $\nu/m$ | $\|y - Ax\|$ | $\|x - x_0\|$ | t(s) |
|---|---|---|---|---|---|
| GVM | $10^{-7}$ | $2.1520\ 10^{-15}$ | $9.5132\ 10^{-8}$ | $1.4909\ 10^{-2}$ | 10.06 |
| | $10^{-5}$ | $5.2329\ 10^{-12}$ | $9.6965\ 10^{-6}$ | $6.8646\ 10^{-2}$ | 2.37 |
| | $10^{-3}$ | $2.2111\ 10^{-8}$ | $9.7215\ 10^{-4}$ | $1.9890\ 10^{-1}$ | 0.35 |
| | $10^{-1}$ | $1.9484\ 10^{-5}$ | $9.8196\ 10^{-2}$ | $3.4627\ 10^{-1}$ | 0.22 |
| | 10 | $6.5983\ 10^{-3}$ | 9.9095 | $8.8165\ 10^{-1}$ | 0.12 |
| GM | $10^{-7}$ | $4.2396\ 10^{-14}$ | $1.1004\ 10^{-7}$ | $2.7130\ 10^{-2}$ | 2.03 |
| | $10^{-5}$ | $5.4552\ 10^{-11}$ | $1.0560\ 10^{-5}$ | $9.6771\ 10^{-2}$ | 0.53 |
| | $10^{-3}$ | $2.3046\ 10^{-8}$ | $9.7243\ 10^{-4}$ | $1.9937\ 10^{-1}$ | 0.29 |
| | $10^{-1}$ | $2.0896\ 10^{-5}$ | $9.8235\ 10^{-2}$ | $3.4634\ 10^{-1}$ | 0.09 |
| | 10 | $6.8436\ 10^{-3}$ | 9.9115 | $8.8791\ 10^{-1}$ | 0.14 |

$x_0$ is the noise free exact solution

# Total least squares

Example :

$$A = U_s \Sigma_s V_s^T, \quad U_s = I - 2\frac{u_s u_s^T}{\|u_s\|^2}, \quad V_s = I - 2\frac{v_s v_s^T}{\|v_s\|^2},$$

$u_s$ and $v_s$ are random vectors, $\Sigma_s$ is an $m \times n$ diagonal matrix
$x_s$ is a vector whose $i$th component is $1/i$ and $c_s = A_s x_s$

The singular values are $\sigma_i = \exp(-3\ 10^{-8} i)$, $i = 1, \dots, n$ and then randomly perturbed

- We monitor the convergence of the smallest singular value of $A$
- It is computed by solving a secular equation using a third order rational approximation
- When it has converged we compute the solution of the secular equations for TLS

For secular equations solvers, see Melman (1997)

$m = 10000$ and $n = 500$, noise=0.3, $\varepsilon = 10^{-6}$

| Meth. | Lanc. it. | trid sec | secul. it. | solution |
|---|---|---|---|---|
| - | 92 | 260 | | |
| Gauss | | | 3 | 0.2379928280610862 |
| G–R | | | 3 | 0.2379902725954008 |
| G–R | | | 3 | 0.2379902838087289 |

The exact smallest singular value of $(A\ \ b)$ is 0.2379927701875446

# Good bye Gene