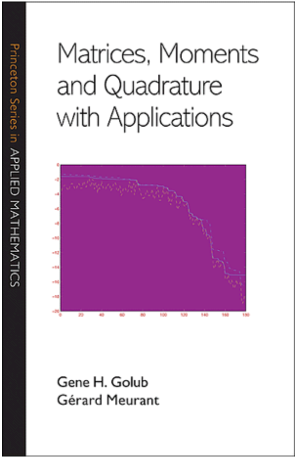
# Matrices, moments and quadrature with applications (I)

Gérard MEURANT

October 2010

This series of lectures is based on a book written in collaboration with Gene H. Golub started in 2005

published by Princeton University Press in 2010



Unfortunately Gene Golub passed away in November 2007

G.H Golub (1932-2007)

# Introduction

The aim of these lectures is to describe numerical algorithms to compute bounds or estimates of bilinear forms

$$u^T f(A) v$$

where $A$ is a square non singular real symmetric matrix, $f$ is a smooth function and $u$ and $v$ are given vectors

Typically $A$ will be large and sparse and we do not want (or cannot) compute $f(A)$

$f$ will be $1/x, exp(x), \sqrt{x}, \ldots$

If you want to compute all the elements of $f(A)$ , see the book by N. Higham, Functions of matrices: theory and computation, SIAM, 2008

# Applications

In many problems we may want to compute some elements of $f(A)$, then we take $u = e^i, v = e^j$ ($e^i$ is the $i$th column of the identity matrix)

$$f(A)_{i,j} = (e^i)^T f(A) e^j$$

For instance, if f(x)=1/x this will give entries of the inverse of $A$

In this case using the techniques we will describe will be more efficient than solving $Ax = e^j$ and taking $x_i$

Moreover, more generally, if $i = j$ we could obtain upper and lower bounds for the exact value
If $i \neq j$, we just obtain estimates

Another application is to compute norms of the error when solving linear systems

$$Ax = b$$

Assume that we have an approximate solution $\hat{x}$. Then the error is $e = x - \hat{x}$ and the residual is $r = b - A\hat{x}$. $r$ is directly computable, but not $e$

We have the relationship

$$Ae = A(x - \hat{x}) = b - A\hat{x} = r$$

Solving this system is as expensive as solving the initial one. However,

$$\|e\|^2 = e^T e = (A^{-1} r)^T A^{-1} r = r^T A^{-2} r$$

If $A$ is positive definite we can define $\|e\|_A^2 = e^T Ae$. Then

$$\|e\|_A^2 = r^T A^{-1} r$$

# Another example

Assume that we know the eigenvalues of a symmetric matrix $A$ and we would like to compute the eigenvalues of a rank-one modification of $A$

$$Ax = \lambda x$$

We know the eigenvalues $\lambda$ and we want to compute $\mu$ such that

$$(A + cc^T)y = \mu y$$

where $c$ is a given vector (not orthogonal to an eigenvector of $A$)

Then

$$y = -(A - \mu I)^{-1} cc^T y$$

Multiplying by $c^T$

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

Finally, we have to solve

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

This is called a secular equation and for solving we have to evaluate quadratic forms

Bilinear (or quadratic) forms arise in many other applications

- Estimates of $\det(A)$ or $\mathrm{trace}(A^{-1})$
- Least squares problems (estimates of the backward error)
- Total least squares
- Tikhonov regularization of discrete ill–posed problems (estimation of the regularization parameter)
- ...

The main technique is to write a quadratic form

$$u^T f(A) u$$

as a Riemann-Stieltjes integral and to use Gauss quadrature to obtain an estimate (or a bound in some cases) of the integral

# Ingredients

Along our journey we will use

- Orthogonal polynomials
- Tridiagonal matrices
- Quadrature rules
- The Lanczos and conjugate gradient methods

In this lecture, we look at orthogonal polynomials and Gauss quadrature

The next lecture will consider the Lanczos and conjugate gradient algorithms, tridiagonal matrices and inverse problems

Next we will look at applications to practical problems

# Quadratic forms

$$u^T f(A) u$$

Since $A$ is symmetric

$$A = Q \Lambda Q^T$$

where $Q$ is the orthonormal matrix whose columns are the normalized eigenvectors of $A$ and $\Lambda$ is a diagonal matrix whose diagonal elements are the eigenvalues $\lambda_i$. Then

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

In fact this is a definition of $f(A)$ when $A$ is symmetric
Of course, usually we don't know $Q$ and $\Lambda$. That's what makes the problem interesting!

$$
\begin{aligned}
u^T f(A) u &= u^T Q f(\Lambda) Q^T u \\
&= \gamma^T f(\Lambda) \gamma \\
&= \sum_{i=1}^{n} f(\lambda_i) \gamma_i^2
\end{aligned}
$$

This last sum can be considered as a Riemann–Stieltjes integral

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

where the measure $\alpha$ is piecewise constant and defined by

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

# Riemann-Stieltjes integrals

$[a, b]$ = finite or infinite interval of the real line

## Definition

A Riemann–Stieltjes integral of a real valued function $f$ of a real variable with respect to a real function $\alpha$ is denoted by

$$\int_a^b f(\lambda)\, d\alpha(\lambda) \tag{1}$$

and is defined to be the limit (if it exists), as the mesh size of the partition $\pi$ of the interval $[a, b]$ goes to zero, of the sums

$$\sum_{\{\lambda_i\}\in\pi} f(c_i)(\alpha(\delta_{i+1}) - \alpha(\delta_i))$$

where $c_i \in [\delta_i, \delta_{i+1}]$

Thomas Jan Stieltjes (1856-1894)

- if $f$ is continuous and $\alpha$ is of bounded variation on $[a, b]$ then the integral exists
- $\alpha$ is of bounded variation if it is the difference of two nondecreasing functions
- The integral exists if $f$ is continuous and $\alpha$ is nondecreasing

In many cases Riemann–Stieltjes integrals are directly written as

$$\int_a^b f(\lambda)\, w(\lambda) d\lambda$$

where $w$ is called the weight function

# Moments and inner product

Let $\alpha$ be a nondecreasing function on the interval $(a, b)$ having finite limits at $\pm\infty$ if $a = -\infty$ and/or $b = +\infty$

## Definition

The numbers

$$\mu_i = \int_a^b \lambda^i \, d\alpha(\lambda), \;\; i = 0, 1, \ldots \tag{2}$$

are called the **moments** related to the measure $\alpha$

## Definition

Let $\mathcal{P}$ be the space of real polynomials, we define an inner product (related to the measure $\alpha$) of two polynomials $p$ and $q \in \mathcal{P}$ as

$$\langle p, q \rangle = \int_a^b p(\lambda) q(\lambda) \, d\alpha(\lambda) \tag{3}$$

The norm of $p$ is defined as

$$\|p\| = \left( \int_a^b p(\lambda)^2 \, d\alpha(\lambda) \right)^{\frac{1}{2}} \tag{4}$$

We will consider also discrete inner products as

$$\langle p, q \rangle = \sum_{j=1}^{m} p(t_j) q(t_j) w_j^2 \tag{5}$$

The values $t_j$ are referred as points or nodes and the values $w_j^2$ are the weights

We will use the fact that the sum in equation (5) can be seen as an approximation of the integral (3)

Conversely, it can be written as a Riemann–Stieltjes integral for a measure $\alpha$ which is piecewise constant and has jumps at the nodes $t_j$ (that we assume to be distinct for simplicity), see Atkinson; Dahlquist, Eisenstat and Golub; Dahlquist, Golub and Nash

$$\alpha(\lambda) = \begin{cases} 0 & \text{if } \lambda < t_1 \\ \sum_{j=1}^{i} [w_j]^2 & \text{if } t_i \leq \lambda < t_{i+1} \; i = 1, \ldots, m-1 \\ \sum_{j=1}^{m} [w_j]^2 & \text{if } t_m \leq \lambda \end{cases}$$

There are different ways to normalize polynomials:

A polynomial $p$ of exact degree $k$ is said to be **monic** if the coefficient of the monomial of highest degree is 1, that is $p(\lambda) = \lambda^k + c_{k-1}\lambda^{k-1} + \dots$

### Definition

- The polynomials $p$ and $q$ are said to be **orthogonal** with respect to inner products (3) or (5), if $\langle p, q \rangle = 0$
- The polynomials $p$ in a set of polynomials are **orthonormal** if they are mutually orthogonal and if $\langle p, p \rangle = 1$
- Polynomials in a set are said to be **monic orthogonal** polynomials if they are orthogonal, monic and their norms are strictly positive

The inner product $\langle \cdot, \cdot \rangle$ is said to be **positive definite** if $\|p\| > 0$ for all nonzero $p \in \mathcal{P}$

A necessary and sufficient condition for having a positive definite inner product is that the determinants of the Hankel moment matrices are positive

$$\det \begin{pmatrix} \mu_0 & \mu_1 & \cdots & \mu_{k-1} \\ \mu_1 & \mu_2 & \cdots & \mu_k \\ \vdots & \vdots & & \vdots \\ \mu_{k-1} & \mu_k & \cdots & \mu_{2k-2} \end{pmatrix} > 0, \ \ k = 1, 2, \ldots$$

where $\mu_i$ are the moments of definition (2)

# Existence of orthogonal polynomials

**Theorem**

*If the inner product $\langle \cdot, \cdot \rangle$ is positive definite on $\mathcal{P}$, there exists a unique infinite sequence of monic orthogonal polynomials related to the measure $\alpha$*

See Gautschi

We have defined orthogonality relative to an inner product given by a Riemann–Stieltjes integral but, more generally, orthogonal polynomials can be defined relative to a linear functional $L$ such that $L(\lambda^k) = \mu_k$

Two polynomials $p$ and $q$ are said to be orthogonal if $L(pq) = 0$
One obtains the same kind of existence result, see the book by Brezinski

# Three-term recurrences

The main ingredient is the following property for the inner product

$$\langle \lambda p, q \rangle = \langle p, \lambda q \rangle$$

## Theorem

*For monic orthogonal polynomials, there exist sequences of coefficients* $\alpha_k$, $k = 1, 2, \ldots$ *and* $\gamma_k$, $k = 1, 2, \ldots$ *such that*

$$p_{k+1}(\lambda) = (\lambda - \alpha_{k+1}) p_k(\lambda) - \gamma_k p_{k-1}(\lambda), \;\; k = 0, 1, \ldots \qquad (6)$$

$$p_{-1}(\lambda) \equiv 0, \;\; p_0(\lambda) \equiv 1.$$

*where*

$$\alpha_{k+1} = \frac{\langle \lambda p_k, p_k \rangle}{\langle p_k, p_k \rangle}, \; k = 0, 1, \ldots$$

$$\gamma_k = \frac{\langle p_k, p_k \rangle}{\langle p_{k-1}, p_{k-1} \rangle}, \; k = 1, 2, \ldots$$

*Proof.*

A set of monic orthogonal polynomials $p_j$ is linearly independent
Any polynomial $p$ of degree $k$ can be written as

$$p = \sum_{j=0}^{k} \omega_j p_j,$$

for some real numbers $\omega_j$
$p_{k+1} - \lambda p_k$ is of degree $\leq k$

$$p_{k+1} - \lambda p_k = -\alpha_{k+1} p_k - \gamma_k p_{k-1} + \sum_{j=0}^{k-2} \delta_j p_j \tag{7}$$

Taking the inner product of equation (7) with $p_k$

$$\langle \lambda p_k, p_k \rangle = \alpha_{k+1} \langle p_k, p_k \rangle$$

Multiplying equation (7) by $p_{k-1}$

$$\langle \lambda p_k, p_{k-1} \rangle = \gamma_k \langle p_{k-1}, p_{k-1} \rangle$$

But, using equation (7) for the degree $k-1$

$$\langle \lambda p_k, p_{k-1} \rangle = \langle p_k, \lambda p_{k-1} \rangle = \langle p_k, p_k \rangle$$

we multiply equation (7) with $p_j$, $j < k-1$

$$\langle \lambda p_k, p_j \rangle = \delta_j \langle p_j, p_j \rangle$$

The left hand side of the last equation vanishes
For this, the property $\langle \lambda p_k, p_j \rangle = \langle p_k, \lambda p_j \rangle$ is crucial
Since $\lambda p_j$ is of degree $< k$, the left hand side is 0 and it implies
$\delta_j = 0$, $j = 0, \ldots, k-2$ □

There is a converse to this theorem

It is is attributed to J. Favard whose paper was published in 1935, although this result had also been obtained by J. Shohat at about the same time and it was known earlier to Stieltjes

### Theorem

*If a sequence of monic polynomials* $p_k$, $k = 0, 1, \ldots$ *satisfies a three–term recurrence relation such as equation (6) with real coefficients and* $\gamma_k > 0$, *then there exists a positive measure* $\alpha$ *such that the sequence* $p_k$ *is orthogonal with respect to an inner product defined by a Riemann–Stieltjes integral for the measure* $\alpha$

# Orthonormal polynomials

**Theorem**

*For orthonormal polynomials, there exist sequences of coefficients* $\alpha_k,\ k = 1, 2, \dots$ *and* $\beta_k,\ k = 1, 2, \dots$ *such that*

$$\sqrt{\beta_{k+1}} p_{k+1}(\lambda) = (\lambda - \alpha_{k+1}) p_k(\lambda) - \sqrt{\beta_k} p_{k-1}(\lambda),\ \ k = 0, 1, \dots \tag{8}$$

$$p_{-1}(\lambda) \equiv 0,\ \ p_0(\lambda) \equiv 1/\sqrt{\beta_0},\ \ \beta_0 = \int_a^b d\alpha$$

*where*

$$\alpha_{k+1} = \langle \lambda p_k, p_k \rangle,\ k = 0, 1, \dots$$

*and* $\beta_k$ *is computed such that* $\|p_k\| = 1$

# Relations between monic and orthonormal polynomials

Assume that we have a system of monic polynomials $p_k$ satisfying a three-term recurrence (6), then we can obtain orthonormal polynomials $\hat{p}_k$ by normalization

$$\hat{p}_k(\lambda) = \frac{p_k(\lambda)}{\langle p_k,\, p_k \rangle^{1/2}}$$

Using equation (6)

$$\|p_{k+1}\|\hat{p}_{k+1} = \left(\lambda\|p_k\| - \frac{\langle \lambda p_k, p_k \rangle}{\|p_k\|}\right)\hat{p}_k - \frac{\|p_k\|^2}{\|p_{k-1}\|}\hat{p}_{k-1}$$

After some manipulations

$$\frac{\|p_{k+1}\|}{\|p_k\|}\hat{p}_{k+1} = (\lambda - \langle \lambda\hat{p}_k, \hat{p}_k \rangle)\hat{p}_k - \frac{\|p_k\|}{\|p_{k-1}\|}\hat{p}_{k-1}$$

Note that

$$\langle \lambda \hat{p}_k, \hat{p}_k \rangle = \frac{\langle \lambda p_k, p_k \rangle}{\|p_k\|^2}$$

and

$$\sqrt{\beta_{k+1}} = \frac{\|p_{k+1}\|}{\|p_k\|}$$

Therefore the coefficients $\alpha_k$ are the same and $\beta_k = \gamma_k$

If we have the coefficients of monic orthogonal polynomials we just have to take the square root of $\gamma_k$ to obtain the coefficients of the corresponding orthonormal polynomials

# Jacobi matrices

If the orthonormal polynomials exist for all $k$, there is an infinite symmetric tridiagonal matrix $J_\infty$ associated with them

$$J_\infty = \begin{pmatrix} \alpha_1 & \sqrt{\beta_1} & & & \\ \sqrt{\beta_1} & \alpha_2 & \sqrt{\beta_2} & & \\ & \sqrt{\beta_2} & \alpha_3 & \sqrt{\beta_3} & \\ & & \ddots & \ddots & \ddots \end{pmatrix}$$

Since it has positive subdiagonal elements, the matrix $J_\infty$ is called an infinite Jacobi matrix

Its leading principal submatrix of order $k$ is denoted as $J_k$

Orthogonal polynomials are fully described by their Jacobi matrices

# Properties of zeros

Let

$$P_k(\lambda) = \begin{pmatrix} p_0(\lambda) & p_1(\lambda) & \dots & p_{k-1}(\lambda) \end{pmatrix}^T$$

In matrix form, the three-term recurrence is written as

$$\lambda P_k = J_k P_k + \eta_k p_k(\lambda) e^k \tag{9}$$

where $J_k$ is the Jacobi matrix of order $k$ and $e^k$ is the last column of the identity matrix ($\eta_k = \sqrt{\beta_k}$)

Theorem

*The zeros $\theta_j^{(k)}$ of the orthonormal polynomial $p_k$ are the eigenvalues of the Jacobi matrix $J_k$*

*Proof.* If $\theta$ is a zero of $p_k$, from equation (9) we have

$$\theta P_k(\theta) = J_k P_k(\theta)$$

This shows that $\theta$ is an eigenvalue of $J_k$ and $P_k(\theta)$ is a corresponding (unnormalized) eigenvector □

$J_k$ being a symmetric tridiagonal matrix, its eigenvalues (the zeros of the orthogonal polynomial $p_k$) are real and distinct

### Theorem

*The zeros of the orthogonal polynomials $p_k$ associated with the measure $\alpha$ on $[a, b]$ are real, distinct and located in the interior of $[a, b]$*

see Szegö

# Examples of orthogonal polynomials

For classical orthogonal polynomials (Chebyshev, Legendre, Laguerre, Hermite, ...) the coefficients of the recurrence are explicitly known

Jacobi polynomials

$$d\alpha(\lambda) = w(\lambda)\, d\lambda$$

$$a = -1,\ b = 1, \quad w(\lambda) = (1-\lambda)^{\delta}(1+\lambda)^{\beta},\ \delta, \beta > -1$$

Special cases:

Chebyshev polynomials of the first kind: $\delta = \beta = -1/2$

$$C_k(\lambda) = \cos(k \arccos \lambda)$$

They satisfy

$$C_0(\lambda) \equiv 1,\ C_1(\lambda) \equiv \lambda, \quad C_{k+1}(\lambda) = 2\lambda C_k(\lambda) - C_{k-1}(\lambda)$$

The zeros of $C_k$ are

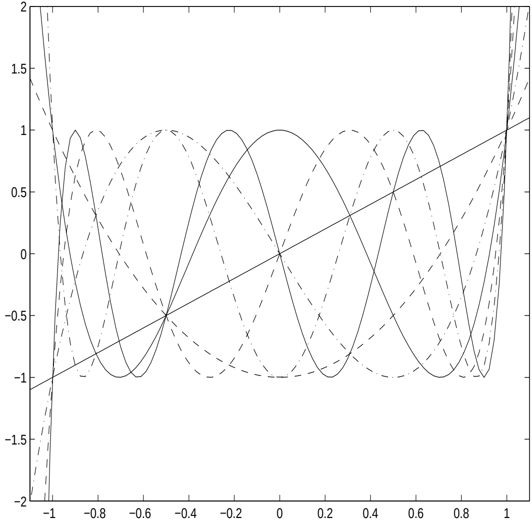
$$\lambda_{j+1} = \cos\left(\frac{2j+1}{k}\frac{\pi}{2}\right),\ j = 0, 1, \dots k-1$$

The polynomial $C_k$ has $k+1$ extremas in $[-1, 1]$

$$\lambda'_j = \cos\left(\frac{j\pi}{k}\right),\ j = 0, 1, \dots, k$$

and $C_k(\lambda'_j) = (-1)^j$
For $k \geq 1$, $C_k$ has a leading coefficient $2^{k-1}$

$$< C_i, C_j >_\alpha = \begin{cases} 0 & i \neq j \\ \frac{\pi}{2} & i = j \neq 0 \\ \pi & i = j = 0 \end{cases}$$




Chebyshev polynomials (first kind) $C_k,\ k = 1, \ldots, 7$ on $[-1.1, 1.1]$

Let $\pi_n^1 = \{$ poly. of degree $n$ in $\lambda$ whose value is 1 for $\lambda = 0$ $\}$

Chebyshev polynomials provide the solution of the minimization problem

$$\min_{q_n \in \pi_n^1} \max_{\lambda \in [a,b]} |q_n(\lambda)|$$

The solution is written as

$$\min_{q_n \in \pi_n^1} \max_{\lambda \in [a,b]} |q_n(\lambda)| = \max_{\lambda \in [a,b]} \left| \frac{C_n\left(\frac{2\lambda-(a+b)}{b-a}\right)}{C_n\left(\frac{a+b}{b-a}\right)} \right| = \left| \frac{1}{C_n\left(\frac{a+b}{b-a}\right)} \right|$$

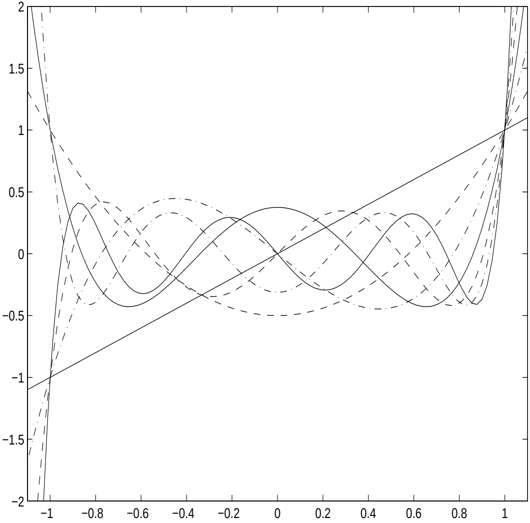
see Dahlquist and Björck

# Legendre polynomials

$$a = -1,\ b = 1, \quad \delta = \beta = 0, \quad w(\lambda) \equiv 1$$

$$(k+1)P_{k+1}(\lambda) = (2k+1)\lambda P_k(\lambda) - kP_{k-1}(\lambda),\ P_0(\lambda) \equiv 1,\ P_1(\lambda) \equiv \lambda$$

The Legendre polynomial $P_k$ is bounded by 1 on $[-1, 1]$




Legendre polynomials $P_k$, $k = 1, \ldots, 7$ on $[-1.1, 1.1]$

# Variable-signed weight functions

What happens if the weight function $w$ is not positive?

**Theorem**

*Assume that all the moments exist and are finite*
*For any $k > 0$, there exists a polynomial $p_k$ of degree at most $k$ such that $p_k$ is orthogonal to all polynomials of degree $\leq k-1$ with respect to $w$*

see G.W. Struble

The important words in this result are: "of degree at most $k$"
In some cases the polynomial $p_k$ can be of degree less than $k$

$C(k)$ = set of polynomials of degree $\leq k$ orthogonal to all polynomials of degree $\leq k-1$
$C(k)$ is called **degenerate** if it contains polynomials of degree less than $k$
If $C(k)$ is **non-degenerate** it contains one unique polynomial (up to a multiplicative constant)

Theorem
*Let $C(k)$ be non-degenerate with a polynomial $p_k$*
*Assume $C(k+n)$, $n > 0$ is the next non-degenerate set. Then $p_k$ is the unique (up to a multiplicative constant) polynomial of lowest degree in $C(k+m)$, $m = 1, \ldots, n-1$*

$$p_k(\lambda) = (\alpha_k \lambda^{d_k - d_{k-1}} + \sum_{i=0}^{d_k - d_{k-1} - 1} \beta_{k,i} \lambda^i) p_{k-1}(\lambda) - \gamma_{k-1} p_{k-2}(\lambda), \;\; k = 2, \tag{10}$$

$$p_0(\lambda) \equiv 1, \quad p_1(\lambda) = (\alpha_1 \lambda^{d_1} + \sum_{i=0}^{d_1 - 1} \beta_{1,i} \lambda^i) p_0(\lambda)$$

The coefficient of $p_{k-1}$ contains powers of $\lambda$ depending on the difference of the degrees of the polynomials in the non-degenerate cases

The coefficients $\alpha_k$ and $\gamma_{k-1}$ have to be nonzero

# Matrix orthogonal polynomials

We would like to have matrices as coefficients of the polynomials
For our purposes we just need $2 \times 2$ matrices

### Definition

For $\lambda$ real, a matrix polynomial $p_i(\lambda)$, which is a $2 \times 2$ matrix, is defined as

$$p_i(\lambda) = \sum_{j=0}^{i} \lambda^j C_j^{(i)}$$

where the coefficients $C_j^{(i)}$ are given $2 \times 2$ real matrices
If the leading coefficient is the identity matrix, the matrix polynomial is said to be monic

The "measure" $\alpha(\lambda)$ is a matrix of order 2 that we suppose to be symmetric and positive semi–definite

We assume that the **(matrix) moments**

$$M_k = \int_a^b \lambda^k \, d\alpha(\lambda) \tag{11}$$

exist for all $k$

The "inner product" of two matrix polynomials $p$ and $q$ is defined as

$$\langle p, q \rangle = \int_a^b p(\lambda) \, d\alpha(\lambda) q(\lambda)^T \tag{12}$$

Two matrix polynomials in a sequence $p_k, \ k = 0, 1, \ldots$ are said to be **orthonormal** if

$$< p_i, p_j >= \delta_{i,j} I_2 \tag{13}$$

where $\delta_{i,j}$ is the Kronecker symbol and $I_2$ the identity matrix of order 2

**Theorem**

*Sequences of matrix orthonormal polynomials satisfy a block three–term recurrence*

$$p_j(\lambda)\Gamma_j = \lambda p_{j-1}(\lambda) - p_{j-1}(\lambda)\Omega_j - p_{j-2}(\lambda)\Gamma_{j-1}^T \tag{14}$$

$$p_0(\lambda) \equiv I_2, \quad p_{-1}(\lambda) \equiv 0$$

*where* $\Gamma_j$, $\Omega_j$ *are* $2 \times 2$ *matrices and the matrices* $\Omega_j$ *are symmetric*

The block three-term recurrence can be written in matrix form as

$$\lambda[p_0(\lambda), \dots, p_{k-1}(\lambda)] = [p_0(\lambda), \dots, p_{k-1}(\lambda)] J_k + [0, \dots, 0, p_k(\lambda)\Gamma_k] \quad (15)$$

where

$$J_k = \begin{pmatrix} \Omega_1 & \Gamma_1^T & & & \\ \Gamma_1 & \Omega_2 & \Gamma_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & \Gamma_{k-2} & \Omega_{k-1} & \Gamma_{k-1}^T \\ & & & \Gamma_{k-1} & \Omega_k \end{pmatrix}$$

is a block tridiagonal matrix of order $2k$ with $2 \times 2$ blocks

Let $P(\lambda) = [p_0(\lambda), \dots, p_{k-1}(\lambda)]^T$

We have the matrix relation

$$J_k P(\lambda) = \lambda P(\lambda) - [0, \dots, 0, p_k(\lambda)\Gamma_k]^T$$

These matrix polynomials will be useful to estimate $u^T f(A) v$ when $u \neq v$

# Quadrature rules

Given a measure $\alpha$ on the interval $[a, b]$ and a function $f$, a quadrature rule is a relation

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

$R[f]$ is the remainder which is usually not known exactly

The real numbers $t_j$ are the nodes and $w_j$ the weights

The rule is said to be of exact degree $d$ if $R[p] = 0$ for all polynomials $p$ of degree $d$ and there are some polynomials $q$ of degree $d + 1$ for which $R[q] \neq 0$

- Quadrature rules of degree $N-1$ can be obtained by interpolation
- Such quadrature rules are called interpolatory
- Newton–Cotes formulas are defined by taking the nodes to be equally spaced
- A popular choice for the nodes is the zeros of the Chebyshev polynomial of degree $N$. This is called the Fejér quadrature rule
- Another interesting choice is the set of extrema of the Chebyshev polynomial of degree $N-1$. This gives the Clenshaw–Curtis quadrature rule

## Theorem

*Let $k$ be an integer, $0 \leq k \leq N$. The quadrature rule has degree $d = N - 1 + k$ if and only if it is interpolatory and*

$$\int_a^b \prod_{j=1}^{N} (\lambda - t_j) p(x)\, d\alpha = 0, \quad \forall p \text{ polynomial of degree } \leq k - 1.$$

see Gautschi

If the measure is positive, $k = N$ is maximal for interpolatory quadrature since if $k = N + 1$ the condition in the last theorem would give that the polynomial

$$\prod_{j=1}^{N} (\lambda - t_j)$$

is orthogonal to itself which is impossible

# Gauss quadrature rules

The optimal quadrature rule of degree $2N - 1$ is called a Gauss quadrature

It was introduced by C.F. Gauss at the beginning of the nineteenth century

The general formula for a Riemann–Stieltjes integral is

$$I[f] = \int_a^b f(\lambda)\, d\alpha(\lambda) = \sum_{j=1}^{N} w_j f(t_j) + \sum_{k=1}^{M} v_k f(z_k) + R[f], \quad (16)$$

where 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

see Davis and Rabinowitz; Gautschi; Golub and Welsch

Carl Friedrich Gauss (1777-1855)

- If $M = 0$, this is the **Gauss** rule with no prescribed nodes
- If $M = 1$ and $z_1 = a$ or $z_1 = b$ we have the **Gauss–Radau** rule
- If $M = 2$ and $z_1 = a, z_2 = b$, this is the **Gauss–Lobatto** rule

The term $R[f]$ is the remainder which generally cannot be explicitly computed
If the measure $\alpha$ is a positive non–decreasing function

$$R[f] = \frac{f^{(2N+M)}(\eta)}{(2N+M)!} \int_a^b \prod_{k=1}^{M} (\lambda - z_k) \left[ \prod_{j=1}^{N} (\lambda - t_j) \right]^2 d\alpha(\lambda), \quad a < \eta < b \tag{17}$$

Note that for the Gauss rule, the remainder $R[f]$ has the sign of $f^{(2N)}(\eta)$
see Stoer and Bulirsch

Before the 1960s mathematicians were publishing books containing tables giving the nodes and weights for some given distribution functions
See the book by Stroud and Secrest

With the advent of computers, routines appear to compute the nodes and weights

At the beginning people were solving non linear equations for these computations

# The Gauss rule

How do we compute the nodes $t_j$ and the weights $w_j$?

- One way to compute the nodes and weights is to use $f(\lambda) = \lambda^i$, $i = 0, \dots, 2N-1$ and to solve the non linear equations expressing the fact that the quadrature rule is exact
- Use of the orthogonal polynomials associated with the measure $\alpha$ (if we know them)

$$\int_a^b p_i(\lambda) p_j(\lambda) \, d\alpha(\lambda) = \delta_{i,j}$$

$$P(\lambda) = [p_0(\lambda)\ p_1(\lambda) \cdots\ p_{N-1}(\lambda)]^T, \quad e^N = (0\ 0\ \cdots\ 0\ 1)^T$$

$$\lambda P(\lambda) = J_N P(\lambda) + \gamma_N p_N(\lambda) e^N$$

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

$J_N$ is a Jacobi matrix, its eigenvalues are real, simple and located in $[a, b]$

# References

F.V. ATKINSON, *Discrete and continuous boundary problems*, Academic Press, (1964)

C. BREZINSKI, *Biorthogonality and its applications to numerical analysis*, Marcel Dekker, (1992)

T.S. CHIHARA, *An introduction to orhogonal polynomials*, Gordon and Breach, (1978)

G. DAHLQUIST AND A. BJÖRCK, *Numerical methods in scientific computing, volume I*, SIAM, (2008)

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

P.J. Davis and P. Rabinowitz, *Methods of numerical integration*, Second Edition, Academic Press, (1984)

W. Gautschi, *Orthogonal polynomials: computation and approximation*, Oxford University Press, (2004)

G.H. Golub and G. Meurant, *Matrices, moments and quadrature*, in Numerical Analysis 1993, D.F. Griffiths and G.A. Watson eds., Pitman Research Notes in Mathematics, v 303, (1994), pp 105–156

G.H. Golub and J.H. Welsch, *Calculation of Gauss quadrature rules*, Math. Comp., v 23, (1969), pp 221–230

D.P. Laurie, *Anti–Gaussian quadrature formulas*, Math. Comp., v 65 n 214, (1996), pp 739–747

J. Stoer and R. Bulirsch, *Introduction to numerical analysis*, second edition, Springer Verlag, (1983)

G.W. Struble, *Orthogonal polynomials: variable–signed weight functions*, Numer. Math., v 5, (1963), pp 88–94

G. Szegö, *Orthogonal polynomials*, Third Edition, American Mathematical Society, (1974)

# Matrices, moments and quadrature with applications
# (II)

Gérard MEURANT

October 2010

# Previous episode

We wrote the quadratic form

$$u^T f(A) u$$

as a Riemann-Stieltjes integral involving an unknown measure $\alpha$

Then, we were looking for a Gauss quadrature approximation to this integral(assuming for the moment that we know the orthogonal polynomials associated to $\alpha$; that is, the Jacobi matrix)

# The Gauss rule

**Theorem**

*The eigenvalues of $J_N$ (the so–called Ritz values $\theta_j^{(N)}$ which are also the zeros of $p_N$) are the nodes $t_j$ of the Gauss quadrature rule. The weights $w_j$ are the squares of the first elements of the normalized eigenvectors of $J_N$*

*Proof.*

The monic polynomial $\prod_{j=1}^{N}(\lambda - t_j)$ is orthogonal to all polynomials of degree less than or equal to $N-1$. Therefore, (up to a multiplicative constant) it is the orthogonal polynomial associated to $\alpha$ and the nodes of the quadrature rule are the zeros of the orthogonal polynomial, that is the eigenvalues of $J_N$

The vector $P(t_j)$ is an unnormalized eigenvector of $J_N$ corresponding to the eigenvalue $t_j$

If $q$ is an eigenvector with norm 1, we have $P(t_j) = \omega q$ with a scalar $\omega$. From the Christoffel–Darboux relation (which I didn't state)

$$w_j P(t_j)^T P(t_j) = 1, \, j = 1, \dots, N$$

Then

$$w_j P(t_j)^T P(t_j) = w_j \omega^2 \|q\|^2 = w_j \omega^2 = 1$$

Hence, $w_j = 1/\omega^2$. To find $\omega$ we can pick any component of the eigenvector $q$, for instance, the first one which is different from zero $\omega = p_0(t_j)/q_1 = 1/q_1$. Then, the weight is given by

$$w_j = q_1^2$$

If the integral of the measure is not 1

$$w_j = q_1^2 \mu_0 = q_1^2 \int_a^b d\alpha(\lambda)$$

□

The knowledge of the Jacobi matrix and of the first moment allows to compute the nodes and weights of the Gauss quadrature rule

Golub and Welsch showed how the squares of the first components of the eigenvectors can be computed without having to compute the other components with a QR–like method

$$I[f] = \int_a^b f(\lambda)\, d\alpha(\lambda) = \sum_{j=1}^{N} w_j^G f(t_j^G) + R_G[f]$$

with

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

The monic polynomial $\prod_{j=1}^{N}(t_j^G - \lambda)$ which is the determinant $\chi_N$ of $J_N - \lambda I$ can be written as $\gamma_1 \cdots \gamma_{N-1} p_N(\lambda)$

### Theorem

*Assume* $f$ *is such that* $f^{(2n)}(\xi) > 0,\ \forall n,\ \forall \xi,\ a < \xi < b$, *and let*

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

*The Gauss rule is exact for polynomials of degree less than or equal to* $2N-1$ *and*

$$L_G[f] \le I[f]$$

*Moreover* $\forall N,\ \exists \eta \in [a, b]$ *such that*

$$I[f] - L_G[f] = (\gamma_1 \cdots \gamma_{N-1})^2 \frac{f^{(2N)}(\eta)}{(2N)!}$$

To summarize:

if we know the Jacobi matrix of the coefficients of the orthogonal polynomials associated to the measure $\alpha$, we can compute an estimate (or bound) of the Riemann-Stieltjes integral

If we know the Jacobi matrix associated with our piecewise constant measure, then we can obtain estimates (or bounds - depending on $f$) for our quadratic form $u^T f(A) u$

We will see later how we can compute this Jacobi matrix

# The Gauss–Radau rule

To obtain the Gauss–Radau rule, we have to extend the matrix $J_N$ in such a way that it has one prescribed eigenvalue $z_1 = a$ or $b$

Assume $z_1 = a$. We wish to construct $p_{N+1}$ such that $p_{N+1}(a) = 0$

$$0 = \gamma_{N+1} p_{N+1}(a) = (a - \omega_{N+1}) p_N(a) - \gamma_N p_{N-1}(a)$$

This gives

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

Note that

$$(J_N - aI)P(a) = -\gamma_N p_N(a) e^N$$

Let $\delta(a) = [\delta_1(a), \cdots, \delta_N(a)]^T$ with

$$\delta_l(a) = -\gamma_N \frac{p_{l-1}(a)}{p_N(a)} \quad l = 1, \ldots, N$$

This gives $\omega_{N+1} = a + \delta_N(a)$ and $\delta(a)$ satisfies

$$(J_N - aI)\delta(a) = \gamma_N^2 e^N$$

- we generate $\gamma_N$
- we solve the tridiagonal system for $\delta(a)$, this gives $\delta_N(a)$
- we compute $\omega_{N+1} = a + \delta_N(a)$

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

gives the nodes and the weights of the Gauss–Radau quadrature rule

### Theorem

*Assume* $f$ *is such that* $f^{(2n+1)}(\xi) < 0, \forall n, \ \forall \xi, a < \xi < b$. *Let*

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

$w_j^a, v_1^a, t_j^a$ *being the weights and nodes computed with* $z_1 = a$ *and let* $L_{GR}$

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

$w_j^b, v_1^b, t_j^b$ *being the weights and nodes computed with* $z_1 = b$.
*The Gauss–Radau rule is exact for polynomials of degree less than or equal to* $2N$ *and we have*

$$L_{GR}[f] \leq I[f] \leq U_{GR}[f]$$

### Theorem (end)

*Moreover* $\forall N \ \exists \ \eta_U, \eta_L \in [a, b]$ *such that*

$$I[f] - U_{GR}[f] = \frac{f^{(2N+1)}(\eta_U)}{(2N+1)!} \int_a^b (\lambda - a) \left[ \prod_{j=1}^{N} (\lambda - t_j^a) \right]^2 d\alpha(\lambda)$$

$$I[f] - L_{GR}[f] = \frac{f^{(2N+1)}(\eta_L)}{(2N+1)!} \int_a^b (\lambda - b) \left[ \prod_{j=1}^{N} (\lambda - t_j^b) \right]^2 d\alpha(\lambda)$$

# The Gauss–Lobatto rule

We would like to have

$$p_{N+1}(a) = p_{N+1}(b) = 0$$

Using the recurrence relation

$$\begin{pmatrix} p_N(a) & p_{N-1}(a) \\ p_N(b) & p_{N-1}(b) \end{pmatrix} \begin{pmatrix} \omega_{N+1} \\ \gamma_N \end{pmatrix} = \begin{pmatrix} a\ p_N(a) \\ b\ p_N(b) \end{pmatrix}$$

Let

$$\delta_l = -\frac{p_{l-1}(a)}{\gamma_N p_N(a)}, \quad \mu_l = -\frac{p_{l-1}(b)}{\gamma_N p_N(b)}, \ l = 1, \ldots, N$$

then

$$(J_N - aI)\delta = e^N, \quad (J_N - bI)\mu = e^N$$

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

- we solve the tridiagonal systems for $\delta$ and $\mu$, this gives $\delta_N$ and $\mu_N$
- we compute $\omega_{N+1}$ and $\gamma_N$

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

### Theorem

*Assume* $f$ *is such that* $f^{(2n)}(\xi) > 0,\ \forall n,\ \forall \xi,\ a < \xi < b$ *and let*

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

$t_j^{GL}$, $w_j^{GL}$, $v_1^{GL}$ *and* $v_2^{GL}$ *being the nodes and weights computed with* $a$ *and* $b$ *as prescribed nodes. The Gauss–Lobatto rule is exact for polynomials of degree less than or equal to* $2N+1$ *and*

$$I[f] \le U_{GL}[f]$$

*Moreover* $\forall N\ \exists\ \eta \in [a, b]$ *such that*

$$I[f] - U_{GL}[f] = \frac{f^{(2N+2)}(\eta)}{(2N+2)!} \int_a^b (\lambda - a)(\lambda - b) \left[\prod_{j=1}^{N} (\lambda - t_j^{GL})\right]^2 d\alpha(\lambda)$$

# Computation of the Gauss rules

The weights $w_i$ are given by the squares of the first components of the eigenvectors $w_i = (z_1^i)^2 = ((e^1)^T z^i)^2$

**Theorem**

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

*Proof.*

$$\begin{aligned}
\sum_{l=1}^{N} w_l f(t_l) &= \sum_{l=1}^{N} (e^1)^T z^l f(t_l) (z^l)^T e^1 \\
&= (e^1)^T \left( \sum_{l=1}^{N} z^l f(t_l) (z^l)^T \right) e^1 \\
&= (e^1)^T Z_N f(\Theta_N) Z_N^T e^1 \\
&= (e^1)^T f(J_N) e^1
\end{aligned}$$

This result means that we do not necessarily have to compute the nodes and weights (that is, the eigenvalues and first entries of the eigenvectors) if we know how to compute the $(1,1)$ element of $f(J_N)$ where $J_N$ is the Jacobi matrix

For $f(x) = 1/x$ we have to compute

$$(J_N^{-1})_{1,1}$$

for a symmetric tridiagonal matrix $J_N$ and this is easy to do

# Nonsymmetric Gauss quadrature rules

The following will be useful for $u \neq v$

We consider the case where the measure $\alpha$ can be written as

$$\alpha(\lambda) = \sum_{k=1}^{l} \alpha_k \delta_k, \quad \lambda_l \leq \lambda < \lambda_{l+1},\ l = 1, \ldots, N-1$$

where $\alpha_k \neq \delta_k$ and $\alpha_k \delta_k \geq 0$
We assume that there exists two sequences of mutually orthogonal (sometimes called bi–orthogonal) polynomials $p$ and $q$ such that

$$\begin{aligned} \gamma_j p_j(\lambda) &= (\lambda - \omega_j) p_{j-1}(\lambda) - \beta_{j-1} p_{j-2}(\lambda), \quad p_{-1}(\lambda) \equiv 0, \quad p_0(\lambda) \equiv 1 \\ \beta_j q_j(\lambda) &= (\lambda - \omega_j) q_{j-1}(\lambda) - \gamma_{j-1} q_{j-2}(\lambda), \quad q_{-1}(\lambda) \equiv 0, \quad q_0(\lambda) \equiv 1 \end{aligned}$$

with $\langle p_i, q_j \rangle = 0,\ i \neq j$

Let

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

$$Q(\lambda)^T = [q_0(\lambda)\ q_1(\lambda) \cdots\ q_{N-1}(\lambda)]$$

and

$$J_N = \begin{pmatrix} \omega_1 & \gamma_1 & & & \\ \beta_1 & \omega_2 & \gamma_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{N-2} & \omega_{N-1} & \gamma_{N-1} \\ & & & \beta_{N-1} & \omega_N \end{pmatrix}$$

In matrix form

$$\begin{aligned} \lambda P(\lambda) &= J_N P(\lambda) + \gamma_N p_N(\lambda) e^N \\ \lambda Q(\lambda) &= J_N^T Q(\lambda) + \beta_N q_N(\lambda) e^N \end{aligned}$$

## Proposition

$$p_j(\lambda) = \frac{\beta_j \cdots \beta_1}{\gamma_j \cdots \gamma_1} q_j(\lambda)$$

Hence, $q_N$ is a multiple of $p_N$ and the polynomials have the same roots which are also the common real eigenvalues of $J_N$ and $J_N^T$
We define the quadrature rule as

$$\int_a^b f(\lambda)\, d\alpha(\lambda) = \sum_{j=1}^{N} f(\theta_j) s_j t_j + R[f]$$

where $\theta_j$ is an eigenvalue of $J_N$, $s_j$ is the first component of the eigenvector $u_j$ of $J_N$ corresponding to $\theta_j$ and $t_j$ is the first component of the eigenvector $v_j$ of $J_N^T$ corresponding to the same eigenvalue, normalized such that $v_j^T u_j = 1$

**Theorem**

*Assume that $\gamma_j \beta_j \neq 0$, then the nonsymmetric Gauss quadrature rule is exact for polynomials of degree less than or equal to $2N-1$*

The remainder is characterized as

$$R[f] = \frac{f^{(2N)}(\eta)}{(2N)!} \int_a^b p_N(\lambda)^2 \, d\alpha(\lambda)$$

The extension of the Gauss–Radau and Gauss–Lobatto rules to the nonsymmetric case is almost identical to the symmetric case

# The block Gauss quadrature rules

Also useful for the case $u \neq v$

The integral $\int_a^b f(\lambda) d\alpha(\lambda)$ is now a $2 \times 2$ symmetric matrix. The most general quadrature formula is of the form

$$\int_a^b f(\lambda) d\alpha(\lambda) = \sum_{j=1}^{N} W_j f(T_j) W_j + R[f]$$

where $W_j$ and $T_j$ are symmetric $2 \times 2$ matrices. This can be reduced to

$$\sum_{j=1}^{2N} f(t_j) u^j (u^j)^T$$

where $t_j$ is a scalar and $u^j$ is a vector with two components

There exist orthogonal matrix polynomials related to $\alpha$ such that

$$\lambda p_{j-1}(\lambda) = p_j(\lambda)\Gamma_j + p_{j-1}(\lambda)\Omega_j + p_{j-2}(\lambda)\Gamma_{j-1}^T$$

$$p_0(\lambda) \equiv I_2, \quad p_{-1}(\lambda) \equiv 0$$

This can be written as

$$\lambda[p_0(\lambda), \ldots, p_{N-1}(\lambda)] = [p_0(\lambda), \ldots, p_{N-1}(\lambda)]J_N + [0, \ldots, 0, p_N(\lambda)\Gamma_N]$$

where

$$J_N = \begin{pmatrix} \Omega_1 & \Gamma_1^T & & & \\ \Gamma_1 & \Omega_2 & \Gamma_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & \Gamma_{N-2} & \Omega_{N-1} & \Gamma_{N-1}^T \\ & & & \Gamma_{N-1} & \Omega_N \end{pmatrix}$$

is a symmetric block tridiagonal matrix of order $2N$

The nodes $t_j$ are the zeros of the determinant of the matrix orthogonal polynomials that is the eigenvalues of $J_N$ and $u_i$ is the vector consisting of the two first components of the corresponding eigenvector

However, the eigenvalues may have a multiplicity larger than 1

Let $\theta_i, i = 1, \dots, l$ be the set of distinct eigenvalues and $n_i$ their multiplicities. The quadrature rule is then

$$\sum_{i=1}^{l} \left( \sum_{j=1}^{n_i} (w_i^j)(w_i^j)^T \right) f(\theta_i)$$

The block Gauss quadrature rule is exact for polynomials of degree less than or equal to $2N - 1$ but the proof is rather involved

Skip Radau and Lobatto

# The block Gauss–Radau rule

We would like $a$ to be a double eigenvalue of $J_{N+1}$

$$J_{N+1}P(a) = aP(a) - [0, \ldots, 0, p_{N+1}(a)\Gamma_{N+1}]^T$$

$$ap_N(a) - p_N(a)\Omega_{N+1} - p_{N-1}(a)\Gamma_N^T = 0$$

If $p_N(a)$ is non singular

$$\Omega_{N+1} = aI_2 - p_N(a)^{-1}p_{N-1}(a)\Gamma_N^T$$

But

$$(J_N - aI)\begin{pmatrix} -p_0(a)^T p_N(a)^{-T} \\ \vdots \\ -p_{N-1}(a)^T p_N(a)^{-T} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ \Gamma_N^T \end{pmatrix}$$

- We first solve

$$(J_N - aI)\begin{pmatrix}\delta_0(a)\\ \vdots \\ \delta_{N-1}(a)\end{pmatrix} = \begin{pmatrix}0\\ \vdots \\ \Gamma_N^T\end{pmatrix}$$

- We compute

$$\Omega_{N+1} = aI_2 + \delta_{N-1}(a)^T \Gamma_N^T$$

# The block Gauss–Lobatto rule

The generalization of the Gauss–Lobatto construction to the block case is a little more difficult
We would like to have $a$ and $b$ as double eigenvalues of the matrix $J_{N+1}$
It gives

$$\begin{pmatrix} I_2 & p_N^{-1}(a)p_{N-1}(a) \\ I_2 & p_N^{-1}(b)p_{N-1}(b) \end{pmatrix} \begin{pmatrix} \Omega_{N+1} \\ \Gamma_N^T \end{pmatrix} = \begin{pmatrix} aI_2 \\ bI_2 \end{pmatrix}$$

Let $\delta(\lambda)$ be the solution of

$$(J_N - \lambda I)\delta(\lambda) = (0 \ldots 0 \;\, I_2)^T$$

Then, as before

$$\delta_{N-1}(\lambda) = -p_{N-1}(\lambda)^T p_N(\lambda)^{-T} \Gamma_N^{-T}$$

Solving the $4 \times 4$ linear system we obtain

$$\Gamma_N^T \Gamma_N = (b-a)(\delta_{N-1}(a) - \delta_{N-1}(b))^{-1}$$

Thus, $\Gamma_N$ is given as a Cholesky factorization of the right hand side matrix which is positive definite because $\delta_{N-1}(a)$ is a diagonal block of the inverse of $(J_N - aI)^{-1}$ which is positive definite and $-\delta_{N-1}(b)$ is the negative of a diagonal block of $(J_N - bI)^{-1}$ which is negative definite

From $\Gamma_N$, we compute

$$\Omega_{N+1} = aI_2 + \Gamma_N \delta_{N-1}(a) \Gamma_N^T$$

# Computation of the block Gauss rules

Theorem

$$\sum_{i=1}^{2N} f(t_i) u_i u_i^T = e^T f(J_N) e$$

*where* $e^T = (I_2 \; 0 \ldots 0)$

Here we need the $2 \times 2$ principal matrix of $f(J_N)$ where $J_N$ is a block tridiagonal matrix

How do we generate the Jacobi matrix corresponding to the measure $\alpha$ which is unknown?

The answer is to use the Lanczos algorithm

# The Lanczos algorithm

Let $A$ be a real symmetric matrix of order $n$

The Lanczos algorithm constructs an orthogonal basis of a Krylov subspace spanned by the columns of

$$K_k = \left(v, \quad Av, \quad \cdots, \quad A^{k-1}v\right)$$

Gram–Schmidt orthogonalization (Arnoldi) $v^1 = v$

$$h_{i,j} = (Av^j, v^i), \quad i = 1, \ldots, j$$

$$\bar{v}^j = Av^j - \sum_{i=1}^{j} h_{i,j} v^i$$

$$h_{j+1,j} = \|\bar{v}^j\|, \quad \text{if } h_{j+1,j} = 0 \text{ then stop}$$

$$v^{j+1} = \frac{\bar{v}^j}{h_{j+1,j}}$$

Aleksei N. Krylov (1863-1945)

$$AV_k = V_k H_k + h_{k+1,k} v^{k+1} (e^k)^T$$

$H_k$ is an upper Hessenberg matrix with elements $h_{i,j}$
Note that $h_{i,j} = 0, j = 1, \ldots, i-2, \; i > 2$

$$H_k = V_k^T A V_k$$

If $A$ is symmetric, $H_k$ is symmetric and therefore tridiagonal

$$H_k = J_k$$

We also have $AV_n = V_n J_n$, if no $v^j$ is zero before step $n$ since $v^{n+1} = 0$ because $v^{n+1}$ is a vector orthogonal to a set of $n$ orthogonal vectors in a space of dimension $n$
Otherwise there exists an $m < n$ for which $AV_m = V_m J_m$ and the algorithm has found an invariant subspace of $A$, the eigenvalues of $J_m$ being eigenvalues of $A$

starting from a vector $\tilde{v}^1 = v/\|v\|$

$$\alpha_1 = (Av^1, v^1), \tilde{v}^2 = Av^1 - \alpha_1 v^1$$

and then, for $k = 2, 3, \ldots$

$$\eta_{k-1} = \|\tilde{v}^k\|$$

$$v^k = \frac{\tilde{v}^k}{\eta_{k-1}}$$

$$\alpha_k = (v^k, Av^k) = (v^k)^T A v^k$$

$$\tilde{v}^{k+1} = Av^k - \alpha_k v^k - \eta_{k-1} v^{k-1}$$

Cornelius Lanczos (1893-1974)

A variant of the Lanczos algorithm has been proposed by Chris Paige to improve the local orthogonality in finite precision computations

$$\alpha_k = (v^k)^T (Av^k - \eta_{k-1} v^{k-1})$$

$$\tilde{v}^{k+1} = (Av^k - \eta_{k-1} v^{k-1}) - \alpha_k v^k$$

Since we can suppose that $\eta_i \neq 0$, the tridiagonal Jacobi matrix $J_k$ has real and simple eigenvalues which we denote by $\theta_j^{(k)}$

They are known as the Ritz values and are the approximations of the eigenvalues of $A$ given by the Lanczos algorithm

**Theorem**

*Let $\chi_k(\lambda)$ be the determinant of $J_k - \lambda I$ (which is a monic polynomial), then*

$$v^k = p_k(A)v^1, \quad p_k(\lambda) = (-1)^{k-1}\frac{\chi_{k-1}(\lambda)}{\eta_1 \cdots \eta_{k-1}}$$

*The polynomials $p_k$ of degree $k-1$ are called the normalized Lanczos polynomials*

The polynomials $p_k$ satisfy a scalar three–term recurrence

$$\eta_k p_{k+1}(\lambda) = (\lambda - \alpha_k)p_k(\lambda) - \eta_{k-1}p_{k-1}(\lambda), \; k = 1, 2, \ldots$$

with initial conditions, $p_0 \equiv 0, \; p_1 \equiv 1$

### Theorem

*Consider the Lanczos vectors* $v^k$*. There exists a measure* $\alpha$ *such that*

$$(v^k, v^l) = \langle p_k, p_l \rangle = \int_a^b p_k(\lambda) p_l(\lambda) d\alpha(\lambda)$$

*where* $a \leq \lambda_1 = \lambda_{min}$ *and* $b \geq \lambda_n = \lambda_{max}$, $\lambda_{min}$ *and* $\lambda_{max}$ *being the smallest and largest eigenvalues of* $A$

*Proof.*

Let $A = Q \Lambda Q^T$ be the spectral decomposition of $A$
Since the vectors $v^j$ are orthonormal and $p_k(A) = Q p_k(\Lambda) Q^T$, we have

$$\begin{aligned}
(v^k, v^l) &= (v^1)^T p_k(A)^T p_l(A) v^1 \\
&= (v^1)^T Q p_k(\Lambda) Q^T Q p_l(\Lambda) Q^T v^1 \\
&= (v^1)^T Q p_k(\Lambda) p_l(\Lambda) Q^T v^1 \\
&= \sum_{j=1}^{n} p_k(\lambda_j) p_l(\lambda_j) [\hat{v}_j]^2,
\end{aligned}$$

where $\hat{v} = Q^T v^1$

The last sum can be written as an integral for a measure $\alpha$ which is piecewise constant

$$\alpha(\lambda) = \begin{cases} 0 & \text{if } \lambda < \lambda_1 \\ \sum_{j=1}^{i} [\hat{v}_j]^2 & \text{if } \lambda_i \le \lambda < \lambda_{i+1} \\ \sum_{j=1}^{n} [\hat{v}_j]^2 & \text{if } \lambda_n \le \lambda \end{cases}$$

The measure $\alpha$ has a finite number of points of increase at the (unknown) eigenvalues of $A$

If you remember the first lecture, this is precisely the measure we need. Hence we can generate the Jacobi matrix for our (unknown) measure $\alpha$ by the Lanczos algorithm

The Lanczos algorithm can also be used to solve linear systems $Ax = c$ when $A$ is symmetric and $c$ is a given vector

Let $x^0$ be a given starting vector and $r^0 = c - Ax^0$ be the corresponding residual

Let $v = v^1 = r^0/\|r^0\|$

$$x^k = x^0 + V_k y^k$$

We request the residual $r^k = c - Ax^k$ to be orthogonal to the Krylov subspace of dimension $k$

$$V_k^T r^k = V_k^T c - V_k^T Ax^0 - V_k^T AV_k y^k = V_k^T r^0 - J_k y^k = 0$$

But, $r^0 = \|r^0\| v^1$ and $V_k^T r^0 = \|r^0\| e^1$

$$J_k y^k = \|r^0\| e^1$$

# The nonsymmetric Lanczos algorithm

When the matrix $A$ is not symmetric we cannot generally construct a vector $v^{k+1}$ orthogonal to all the previous basis vectors by only using the two previous vectors $v^k$ and $v^{k-1}$

Construct bi-orthogonal sequences using $A^T$

choose two starting vectors $v^1$ and $\tilde{v}^1$ with $(v^1, \tilde{v}^1) \neq 0$ normalized such that $(v^1, \tilde{v}^1) = 1$. We set $v^0 = \tilde{v}^0 = 0$. Then for $k = 1, 2, \ldots$

$$
\begin{aligned}
z^k &= Av^k - \omega_k v^k - \eta_{k-1} v^{k-1} \\
w^k &= A^T \tilde{v}^k - \omega_k \tilde{v}^k - \tilde{\eta}_{k-1} \tilde{v}^{k-1}
\end{aligned}
$$

$$
\omega_k = (\tilde{v}^k, Av^k), \quad \eta_k \tilde{\eta}_k = (z^k, w^k)
$$

$$
v^{k+1} = \frac{z^k}{\tilde{\eta}_k}, \quad \tilde{v}^{k+1} = \frac{w^k}{\eta_k}
$$

$$J_k = \begin{pmatrix} \omega_1 & \eta_1 & & & \\ \tilde{\eta}_1 & \omega_2 & \eta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \tilde{\eta}_{k-2} & \omega_{k-1} & \eta_{k-1} \\ & & & \tilde{\eta}_{k-1} & \omega_k \end{pmatrix}$$

and

$$V_k = [v^1 \ \cdots v^k], \quad \tilde{V}_k = [\tilde{v}^1 \cdots \tilde{v}^k]$$

Then, in matrix form

$$\begin{aligned} AV_k &= V_k J_k + \tilde{\eta}_k v^{k+1} (e^k)^T \\ A^T \tilde{V}_k &= \tilde{V}_k J_k^T + \eta_k \tilde{v}^{k+1} (e^k)^T \end{aligned}$$

Theorem

*If the nonsymmetric Lanczos algorithm does not break down with $\eta_k \tilde{\eta}_k$ being zero, the algorithm yields biorthogonal vectors such that*

$$(\tilde{v}^i, v^j) = 0,\ i \neq j, \quad i, j = 1, 2, \ldots$$

*The vectors $v^1, \ldots, v^k$ span $\mathcal{K}_k(A, v^1)$ and $\tilde{v}^1, \ldots, \tilde{v}^k$ span $\mathcal{K}_k(A^T, \tilde{v}^1)$. The two sequences of vectors can be written as*

$$v^k = p_k(A)v^1, \quad \tilde{v}^k = \tilde{p}_k(A^T)\tilde{v}^1$$

*where $p_k$ and $\tilde{p}_k$ are polynomials of degree $k-1$*

$$\tilde{\eta}_k p_{k+1} = (\lambda - \omega_k)p_k - \eta_{k-1}p_{k-1}$$

$$\eta_k \tilde{p}_{k+1} = (\lambda - \omega_k)\tilde{p}_k - \tilde{\eta}_{k-1}\tilde{p}_{k-1}$$

The algorithm breaks down if at some step we have $(z^k, w^k) = 0$

Either

- a) $z^k = 0$ and/or $w^k = 0$
  If $z^k = 0$ we can compute the eigenvalues or the solution of the linear system $Ax = c$. If $z^k \neq 0$ and $w^k = 0$, the only way to deal with this situation is to restart the algorithm
- b) The more dramatic situation ("serious breakdown") is when $(z^k, w^k) = 0$ with $z^k$ and $w^k \neq 0$
  Need to use look–ahead strategies or restart

For our purposes we will use the nonsymmetric Lanczos algorithm with a symmetric matrix!

We can choose

$$\eta_k = \pm\tilde{\eta}_k = \pm\sqrt{|(z^k, w^k)|}$$

with for instance, $\eta_k \geq 0$ and $\tilde{\eta}_k = \mathrm{sgn}[(z^k, w^k)]\, \eta_k$. Then

$$\tilde{p}_k = \pm p_k$$

# The block Lanczos algorithm

See Golub and Underwood

We consider only $2 \times 2$ blocks

Let $X_0$ be an $n \times 2$ given matrix, such that $X_0^T X_0 = I_2$. Let $X_{-1} = 0$ be an $n \times 2$ matrix. Then, for $k = 1, 2, \ldots$

$$\Omega_k = X_{k-1}^T A X_{k-1}$$

$$R_k = A X_{k-1} - X_{k-1}\Omega_k - X_{k-2}\Gamma_{k-1}^T$$

$$X_k \Gamma_k = R_k$$

The last step is the QR decomposition of $R_k$ such that $X_k$ is $n \times 2$ with $X_k^T X_k = I_2$

We obtain a block tridiagonal matrix

- The matrix $R_k$ can eventually be rank deficient and in that case $\Gamma_k$ is singular
- One of the columns of $X_k$ can be chosen arbitrarily
- To complete the algorithm, we choose this column to be orthogonal with the previous block vectors $X_j$

The block Lanczos algorithm generates a sequence of matrices such that

$$X_j^T X_i = \delta_{ij} I_2$$

## Proposition

$$X_i = \sum_{k=0}^{i} A^k X_0 C_k^{(i)}$$

*where* $C_k^{(i)}$ *are* $2 \times 2$ *matrices*

## Theorem

*The matrix valued polynomials* $p_k$ *satisfy*

$$p_k(\lambda)\Gamma_k = \lambda p_{k-1}(\lambda) - p_{k-1}(\lambda)\Omega_k - p_{k-2}(\lambda)\Gamma_{k-1}^T$$

$$p_{-1}(\lambda) \equiv 0, \quad p_0(\lambda) \equiv I_2$$

*where* $\lambda$ *is a scalar and* $p_k(\lambda) = \sum_{j=0}^{k} \lambda^j X_0 C_j^{(k)}$

$$\lambda[p_0(\lambda), \dots, p_{N-1}(\lambda)] = [p_0(\lambda), \dots, p_{N-1}(\lambda)]J_N + [0, \dots, 0, p_N(\lambda)\Gamma_N]$$

and as $P(\lambda) = [p_0(\lambda), \dots, p_{N-1}(\lambda)]^T$

$$J_N P(\lambda) = \lambda P(\lambda) - [0, \dots, 0, p_N(\lambda)\Gamma_N]^T$$

where $J_N$ is block tridiagonal

## Theorem

*Considering the matrices $X_k$, there exists a matrix measure $\alpha$ such that*

$$X_i^T X_j = \int_a^b p_i(\lambda)^T d\alpha(\lambda) p_j(\lambda) = \delta_{ij} I_2$$

*where $a \leq \lambda_1 = \lambda_{min}$ and $b \geq \lambda_n = \lambda_{max}$*

*Proof.*

$$
\begin{aligned}
\delta_{ij} I_2 = X_i^T X_j &= \left(\sum_{k=0}^{i} (C_k^{(i)})^T X_0^T A^k\right) \left(\sum_{l=0}^{j} A^l X_0 C_l^{(j)}\right) \\
&= \sum_{k,l} (C_k^{(i)})^T X_0^T Q \Lambda^{k+l} Q^T X_0 C_l^{(j)} \\
&= \sum_{k,l} (C_k^{(i)})^T \hat{X} \Lambda^{k+l} \hat{X}^T C_l^{(j)} \\
&= \sum_{k,l} (C_k^{(i)})^T \left(\sum_{m=1}^{n} \lambda_m^{k+l} \hat{X}_m \hat{X}_m^T\right) C_l^{(j)} \\
&= \sum_{m=1}^{n} \left(\sum_k \lambda_m^k (C_k^{(i)})^T\right) \hat{X}_m \hat{X}_m^T \left(\sum_l \lambda_m^l C_l^{(j)}\right)
\end{aligned}
$$

where $\hat{X}_m$ are the columns of $\hat{X} = X_0^T Q$ which is a $2 \times n$ matrix

Hence

$$X_i^T X_j = \sum_{m=1}^{n} p_i(\lambda_m)^T \hat{X}_m \hat{X}_m^T \, p_j(\lambda_m)$$

The sum in the right hand side can be written as an integral for a $2 \times 2$ matrix measure

$$\alpha(\lambda) = \begin{cases} 0 & \text{if } \lambda < \lambda_1 \\ \sum_{j=1}^{i} \hat{X}_j \hat{X}_j^T & \text{if } \lambda_i \leq \lambda < \lambda_{i+1} \\ \sum_{j=1}^{n} \hat{X}_j \hat{X}_j^T & \text{if } \lambda_n \leq \lambda \end{cases}$$

Then

$$X_i^T X_j = \int_a^b p_i(\lambda)^T \, d\alpha(\lambda) \, p_j(\lambda)$$

□

# The conjugate gradient algorithm

The conjugate gradient (CG) algorithm is an iterative method to solve linear systems $Ax = c$ where the matrix $A$ is symmetric positive definite (Hestenes and Stiefel 1952)

It can be obtained from the Lanczos algorithm by using the $LU$ factorization of $J_k$

starting from a given $x^0$ and $r^0 = c - Ax^0$:

for $k = 0, 1, \ldots$ until convergence do

$$\beta_k = \frac{(r^k, r^k)}{(r^{k-1}, r^{k-1})}, \ \beta_0 = 0$$

$$p^k = r^k + \beta_k p^{k-1}$$

$$\gamma_k = \frac{(r^k, r^k)}{(Ap^k, p^k)}$$

$$x^{k+1} = x^k + \gamma_k p^k$$

$$r^{k+1} = r^k - \gamma_k Ap^k$$

Magnus Hestenes (1906-1991)

Eduard Stiefel (1909-1978)

In exact arithmetic the residuals $r^k$ are orthogonal and

$$v^{k+1} = (-1)^k r^k / \|r^k\|$$

Moreover

$$\alpha_k = \frac{1}{\gamma_{k-1}} + \frac{\beta_{k-1}}{\gamma_{k-2}}, \ \beta_0 = 0, \ \gamma_{-1} = 1$$

$$\eta_k = \frac{\sqrt{\beta_k}}{\gamma_{k-1}}$$

The iterates are given by

$$x^{k+1} = x^0 + s_k(A) r^0$$

where $s_k$ is a polynomial of degree $k$

Let

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

be the $A$-norm of the error $\epsilon^k = x - x^k$

## Theorem

*Consider all the iterative methods that can be written as*

$$\overline{x}^{k+1} = \overline{x}^0 + q_k(A)\overline{r}^0, \quad \overline{x}^0 = x^0, \quad \overline{r}^0 = c - A\overline{x}^0$$

*where $q_k$ is a polynomial of degree $k$*
*Of all these methods, CG is the one which minimizes $\|\epsilon^k\|_A$ at each iteration*

As a consequence

**Theorem**

$$\|\epsilon^{k+1}\|_A^2 \leq \max_{1 \leq i \leq n} (t_{k+1}(\lambda_i))^2 \|\epsilon^0\|_A^2$$

*for all polynomials* $t_{k+1}$ *of degree* $k+1$ *such that* $t_{k+1}(0) = 1$

**Theorem**

$$\|\epsilon^k\|_A \leq 2 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \|\epsilon^0\|_A$$

*where* $\kappa = \frac{\lambda_n}{\lambda_1}$ *is the condition number of* $A$

This bound is usually overly pessimistic. This is why it is useful to be able to compute estimates (or bounds) for $\|e^k\|_A$

# Computing $u^T f(A)u$

When $u = v$, we remark that $\alpha$ is an increasing positive function

The algorithm is the following:

- normalize $u$ if necessary to obtain $v^1$
- run $k$ iterations of the Lanczos algorithm with $A$ starting from $v^1$, compute the Jacobi matrix $J_k$
- if we use the Gauss–Radau or Gauss–Lobatto rules, modify $J_k$ to $\tilde{J}_k$ accordingly. For the Gauss rule $\tilde{J}_k = J_k$
- if this is feasible, compute $(e^1)^T f(\tilde{J}_k)e^1$. Otherwise, compute the eigenvalues and the first components of the eigenvectors using the Golub and Welsch algorithm to obtain the approximations from the Gauss, Gauss–Radau and Gauss–Lobatto quadrature rules

Let $n$ be the order of the matrix $A$ and $V_k$ be the $n \times k$ matrix whose columns are the Lanczos vectors

If $A$ has distinct eigenvalues, after $n$ Lanczos iterations we have
$AV_n = V_n J_n$

If $Q$ (resp. $Z$) is the matrix of the eigenvectors of $A$ (resp. $J_n$) we have the relation $V_n Z = Q$. If $\|u\| = 1$

$$u^T f(A) u = (e^1)^T V_n^T Q f(\Lambda) Q^T V_n e^1 = (e^1)^T Z^T f(\Lambda) Z e^1 = (e^1)^T f(J_n) e^1$$

$$R[f] = (e^1)^T f(J_n) e^1 - (e^1)^T f(J_k) e^1$$

The convergence of the Gauss quadrature approximation to the integral depends on the convergence of the Ritz values to the eigenvalues of $A$

# Preconditioning

The convergence rate can be improved in some cases by preconditioning

If we are interested in $u^T A^{-1} u$ and if we have a preconditioner $M = LL^T$ for $A$

$$u^T A^{-1} u = u^T L^{-T} (L^{-1} A L^{-T})^{-1} L^{-1} u$$

$L^{-1} A L^{-T}$ is the preconditioned matrix to which we apply the Lanczos algorithm with the vector $L^{-1} u$

# Example of computations of an element of the inverse

2D Poisson problem, GL, $n = 900$, $A^{-1}_{150,150} = 0.3602$

| $k$ | G | G–R $b_L$ | G–R $b_U$ | G–L |
|---|---|---|---|---|
| 10 | 0.3578 | 0.3581 | 0.3777 | 0.3822 |
| 20 | 0.3599 | 0.3599 | 0.3608 | 0.3609 |
| 30 | 0.3601 | 0.3601 | 0.3602 | 0.3602 |
| 40 | 0.3602 | 0.3602 | 0.3602 | 0.3602 |

We will see more examples next time. . .

# References

W.E. Arnoldi, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Quarterly of Appl. Math., v 9, (1951), pp 17–29

F.V. Atkinson, *Discrete and continuous boundary problems*, Academic Press, (1964)

G. Dahlquist and A. Björck, *Numerical methods in scientific computing, volume I*, SIAM, (2008)

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

P.J. Davis and P. Rabinowitz, *Methods of numerical integration*, Second Edition, Academic Press, (1984) G.H. Golub and G. Meurant, *Matrices, moments and quadrature*, in Numerical Analysis 1993, D.F. Griffiths and G.A. Watson eds., Pitman Research Notes in Mathematics, v 303, (1994), pp 105–156

G.H. Golub and R. Underwood, *The block Lanczos method for computing eigenvalues*, in Mathematical Software III, J. Rice Ed., (1977), pp 361–377

G.H. Golub and J.H. Welsch, *Calculation of Gauss quadrature rules*, Math. Comp., v 23, (1969), pp 221–230

M.R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Nat. Bur. Stand., v 49 n 6, (1952), pp 409–436

C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Standards, v 45, (1950), pp 255–282

- C. Lanczos, *Solution of systems of linear equations by minimized iterations*, J. Res. Nat. Bur. Standards, v 49, (1952), pp 33–53
- G. Meurant, *Computer solution of large linear systems*, North–Holland, (1999)
- G. Meurant, *The Lanczos and Conjugate Gradient algorithms, from theory to finite precision computations*, SIAM, (2006)
- G. Meurant and Z. Strakoš, *The Lanczos and conjugate gradient algorithms in finite precision arithmetic*, Acta Numerica, (2006)
- J. Stoer and R. Bulirsch, *Introduction to numerical analysis*, second edition, Springer Verlag, (1983)

# Matrices, moments and quadrature with applications
# (III)

Gérard MEURANT

October 2010

# Previous episodes

We wrote the quadratic form

$$u^T f(A) u$$

as a Riemann-Stieltjes integral involving an unknown measure $\alpha$

We were looking for a Gauss quadrature approximation to this integral

Then, we have seen that we can generate the orthogonal polynomials associated to $\alpha$; that is, the Jacobi matrix by using the Lanczos algorithm

# The case $u \neq v$

A first possibility is to use the (so-called polarization) identity

$$u^T f(A)v = [(u+v)^T f(A)(u+v) - (u-v)^T f(A)(u-v)]/4$$

Another possibility is to apply the nonsymmetric Lanczos algorithm to the symmetric matrix $A$

The framework of the algorithm is the same as for the case $u = v$.

However, the algorithm may break down

A way to get around the breakdown problem is to introduce a parameter $\delta$ and use $v^1 = u/\delta$ and $\tilde{v}^1 = \delta u + v$. This will give an estimate of $u^T f(A)v/\delta + u^T f(A)u$

# The block case

A third possibility is to use the block Lanczos algorithm

$$I_B[f] = W^T f(A) W = \int_a^b f(\lambda)\ d\alpha(\lambda)$$

However, we have seen that we have to start the algorithm from an $n \times 2$ matrix $X_0$ such that $X_0^T X_0 = I_2$

Considering the bilinear form $u^T f(A) v$ we would like to use $X_0 = [u\ v]$ but this does not fulfill the condition on the starting matrix

We have to orthogonalize the pair $[u\ v]$ before starting the algorithm. Let $u$ and $v$ be independent vectors and $n_u = \|u\|$

$$\tilde{u} = \frac{u}{n_u}, \quad \bar{v} = v - \frac{u^T v}{n_u^2} u, \quad n_v = \|\bar{v}\|, \quad \tilde{v} = \frac{\bar{v}}{n_v},$$

and we set $X_0 = [\tilde{u}\ \tilde{v}]$

Let $J^1$ be the leading $2 \times 2$ submatrix of the matrix $f(J_k)$

$$u^T f(A) v \approx (u^T v) J^1_{1,1} + n_u n_v J^1_{1,2}$$

Moreover

$$u^T f(A) u \approx n_u^2 J^1_{1,1}$$

$$v^T f(A) v \approx n_v^2 J^1_{2,2} + 2(u^T v)\frac{n_u}{n_v} J^1_{1,2} + \frac{(u^T v)^2}{n_u^2} J^1_{1,1}$$

# Extensions to nonsymmetric matrices

- nonsymmetric Lanczos algorithm ( Saylor and Smolarski)
- Arnoldi algorithm (Calvetti, Kim and Reichel)
- Generalized LSQR (Golub, Stoll and Wathen)
- Vorobyev moment problem (Strakoš and Tichý)

# Analytic bounds for elements of functions of matrices

Performing analytically one or two Lanczos iterations, we are able to obtain bounds for the entries of $A^{-1}$

## Theorem

*Let $A$ be a symmetric positive definite matrix. Let*

$$s_i^2 = \sum_{j \neq i} a_{ji}^2, \quad i = 1, \dots, n$$

*Using the Gauss, Gauss–Radau and Gauss–Lobatto rules*

$$\frac{\sum_{k \neq i} \sum_{l \neq i} a_{k,i} a_{k,l} a_{l,i}}{a_{i,i} \sum_{k \neq i} \sum_{l \neq i} a_{k,i} a_{k,l} a_{l,i} - \left( \sum_{k \neq i} a_{k,i}^2 \right)^2} \leq (A^{-1})_{i,i}$$

$$\frac{a_{i,i} - b + \frac{s_i^2}{b}}{a_{i,i}^2 - a_{i,i} b + s_i^2} \leq (A^{-1})_{i,i} \leq \frac{a_{i,i} - a + \frac{s_i^2}{a}}{a_{i,i}^2 - a_{i,i} a + s_i^2}$$

$$(A^{-1})_{i,i} \leq \frac{a + b - a_{ii}}{ab}$$

Compute analytically $\alpha_1, \eta_1, \alpha_2$, the inverse of

$$J_2 = \begin{pmatrix} \alpha_1 & \eta_1 \\ \eta_1 & \alpha_2 \end{pmatrix}$$

is

$$J_2^{-1} = \frac{1}{\alpha_1\alpha_2 - \eta_1^2} \begin{pmatrix} \alpha_2 & -\eta_1 \\ -\eta_1 & \alpha_1 \end{pmatrix}$$

For Gauss–Radau we have to modify the $(2, 2)$ element of $J_2$

Using the nonsymmetric Lanczos algorithm

### Theorem

*Let $A$ be a symmetric positive definite matrix and*

$$t_i = \sum_{k \neq i} a_{k,i}(a_{k,i} + a_{k,j}) - a_{i,j}(a_{i,j} + a_{i,i})$$

*For $(A^{-1})_{i,j} + (A^{-1})_{i,i}$ we have the two following estimates*

$$\frac{a_{i,i} + a_{i,j} - a + \frac{t_i}{a}}{(a_{i,i} + a_{i,j})^2 - a(a_{i,i} + a_{i,j}) + t_i}, \quad \frac{a_{i,i} + a_{i,j} - b + \frac{t_i}{b}}{(a_{i,i} + a_{i,j})^2 - b(a_{i,i} + a_{i,j}) + t_i}$$

If $t_i \geq 0$, the first expression with $a$ gives an upper bound and the second one with $b$ a lower bound

# Other functions

We have to compute $f(J)$ for

$$J = \begin{pmatrix} \alpha & \eta \\ \eta & \xi \end{pmatrix}$$

## Proposition

*Let* $\delta = (\alpha - \xi)^2 + 4\eta^2$

$$\gamma = \exp\left(\frac{1}{2}(\alpha + \xi - \sqrt{\delta})\right), \quad \omega = \exp\left(\frac{1}{2}(\alpha + \xi + \sqrt{\delta})\right)$$

*The* $(1,1)$ *element of the exponential of* $J$ *is*

$$\frac{1}{2}\left[\gamma + \omega + \frac{\omega - \gamma}{\sqrt{\delta}}(\alpha - \xi)\right]$$

### Theorem

*Let*

$$\lambda_+ = \frac{1}{2}(\alpha + \xi + \sqrt{\delta}), \quad \lambda_- = \frac{1}{2}(\alpha + \xi - \sqrt{\delta})$$

*The* $(1,1)$ *element of* $f(J)$ *is*

$$\frac{1}{2\sqrt{\delta}}\left[(\alpha - \xi)(f(\lambda_+) - f(\lambda_-)) + \sqrt{\delta}(f(\lambda_+) + f(\lambda_-))\right]$$

We can obtain analytic bounds for the $(i,i)$ element of $f(A)$ for any function for which we can compute $f(\lambda_+)$ and $f(\lambda_-)$

# Examples

**Example F1**

This is an example of dimension 10

$$A = \frac{1}{11}\begin{pmatrix} 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 9 & 18 & 16 & 14 & 12 & 10 & 8 & 6 & 4 & 2 \\ 8 & 16 & 24 & 21 & 18 & 15 & 12 & 9 & 6 & 3 \\ 7 & 14 & 21 & 28 & 24 & 20 & 16 & 12 & 8 & 4 \\ 6 & 12 & 18 & 24 & 30 & 25 & 20 & 15 & 10 & 5 \\ 5 & 10 & 15 & 20 & 25 & 30 & 24 & 18 & 12 & 6 \\ 4 & 8 & 12 & 16 & 20 & 24 & 28 & 21 & 14 & 7 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 & 16 & 8 \\ 2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 & 18 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{pmatrix}$$

This matrix was chosen since. . .

The inverse of $A$ is a tridiagonal matrix

$$A^{-1} = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}$$

**Example F3**

This is an example proposed by Z. Strakoš. Let $\Lambda$ be a diagonal matrix

$$\lambda_i = \lambda_1 + \left(\frac{i-1}{n-1}\right)(\lambda_n - \lambda_1)\rho^{n-i}, \; i = 1, \ldots, n$$

Let $Q$ be the orthogonal matrix of the eigenvectors of the tridiagonal matrix $(-1, 2, -1)$. Then the matrix is

$$A = Q^T \Lambda Q$$

We will use $\lambda_1 = 0.1, \lambda_n = 100$ and $\rho = 0.9$

**Example F4**

The matrix is arising from the 5–point finite difference approximation of the Poisson equation in a unit square with an $m \times m$ mesh

This gives a linear system $Ax = c$ of order $m^2$

$$A = \begin{pmatrix} T & -I & & & \\ -I & T & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & T & -I \\ & & & -I & T \end{pmatrix}$$

Each block is of order $m$ and

$$T = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix}$$

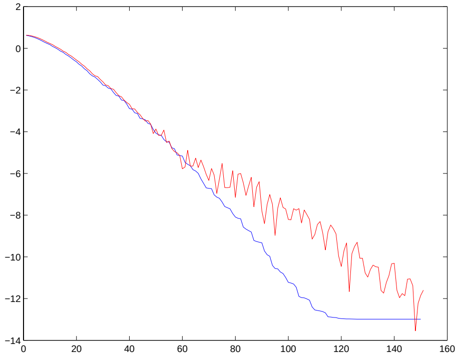
# Diagonal elements

Example F1, GL, $A^{-1}_{5,5} = 2$

| rule | Nit=1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| G | 0.3667 | 1.3896 | 1.7875 | 1.9404 | 1.9929 | 1.9993 | 2 |
| G–R $b_L$ | 1.3430 | 1.7627 | 1.9376 | 1.9926 | 1.9993 | 2.0000 | 2 |
| G–R $b_U$ | 3.0330 | 2.2931 | 2.1264 | 2.0171 | 2.0020 | 2.0001 | 2 |
| G–L | 3.1341 | 2.3211 | 2.1356 | 2.0178 | 2.0021 | 2.0001 | 2 |

Example F3, GL, $n = 100$, $A^{-1}_{50,50} = 4.2717$

| Nit | G | G–R $b_L$ | G–R $b_U$ | G–L |
|---|---|---|---|---|
| 10 | 2.7850 | 3.0008 | 5.1427 | 5.1664 |
| 20 | 4.0464 | 4.0505 | 4.4262 | 4.4643 |
| 30 | 4.2545 | 4.2553 | 4.2883 | 4.2897 |
| 40 | 4.2704 | 4.2704 | 4.2728 | 4.2733 |
| 50 | 4.2716 | 4.2716 | 4.2718 | 4.2718 |
| 60 | 4.2717 | 4.2717 | 4.2717 | 4.2717 |




Error in $A^{-1}_{50,50}$, Gauss (blue), CG (red)

## Example F4, GL, $n = 900$, $A^{-1}_{150,150} = 0.3602$

| Nit | G | G–R $b_L$ | G–R $b_U$ | G–L |
|---|---|---|---|---|
| 10 | 0.3578 | 0.3581 | 0.3777 | 0.3822 |
| 20 | 0.3599 | 0.3599 | 0.3608 | 0.3609 |
| 30 | 0.3601 | 0.3601 | 0.3602 | 0.3602 |
| 40 | 0.3602 | 0.3602 | 0.3602 | 0.3602 |

# Non–diagonal elements with the nonsymmetric Lanczos algorithm

Example F1, GNS, $A^{-1}_{2,2} + A^{-1}_{2,1} = 1$

| rule | Nit=1 | 2 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| G | 0.4074 | 0.6494 | 0.9512 | 0.9998 | 1.0004 | 1 |
| G–R $b_L$ | 0.6181 | 0.8268 | 0.9998 | 1.0004 | 1.0001 | 1 |
| G–R $b_U$ | 2.6483 | 1.4324 | 1.0035 | 1.0012 | 0.9994 | 1 |
| G–L | 3.2207 | 1.4932 | 1.0036 | 1.0012 | 0.9993 | 0.9994 |

## Example F3, GNS, $n = 100$, $A^{-1}_{50,50} + A^{-1}_{50,49} = 1.4394$

| Nit | G | G–R $b_L$ | G–R $b_U$ | G–L |
|---|---|---|---|---|
| 10 | 0.8795 | 0.9429 | 2.2057 | 2.2327 |
| 20 | 1.3344 | 1.3362 | 1.5535 | 1.5839 |
| 30 | 1.4301 | 1.4308 | 1.4510 | 1.4516 |
| 40 | 1.4386 | 1.4387 | 1.4404 | 1.4404 |
| 50 | 1.4394 | 1.4394 | 1.4395 | 1.4395 |
| 60 | 1.4394 | 1.4394 | 1.4394 | 1.4394 |

## Example F4, GNS, $n = 900$, $A^{-1}_{150,150} + A^{-1}_{150,50} = 0.3665$

| Nit | G | G–R $b_L$ | G–R $b_U$ | G–L |
|---|---|---|---|---|
| 10 | 0.3611 | 0.3615 | 0.3917 | 0.3979 |
| 20 | 0.3656 | 0.3657 | 0.3678 | 0.3680 |
| 30 | 0.3663 | 0.3664 | 0.3666 | 0.3666 |
| 40 | 0.3665 | 0.3665 | 0.3665 | 0.3665 |

# Non–diagonal elements with the block Lanczos algorithm

Let $(J_k^{-1})_{1,1}$ the $2 \times 2$ $(1,1)$ block of the inverse of $J_k$ with

$$J_k = \begin{pmatrix} \Omega_1 & \Gamma_1^T & & & \\ \Gamma_1 & \Omega_2 & \Gamma_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & \Gamma_{k-2} & \Omega_{k-1} & \Gamma_{k-1}^T \\ & & & \Gamma_{k-1} & \Omega_k \end{pmatrix}$$

$$\Delta_1 = \Omega_1, \quad \Delta_i = \Omega_i - \Gamma_{i-1}\Omega_{i-1}^{-1}\Gamma_{i-1}^T, \; i = 2, \ldots, k$$

$$C_k = \Delta_1^{-1}\Gamma_1^T \Delta_2^{-1}\Gamma_2^T \cdots \Delta_{k-1}^{-1}\Gamma_{k-1}^T \Delta_k^{-1}\Gamma_k^T$$

$$(J_{k+1}^{-1})_{1,1} = (J_k^{-1})_{1,1} + C_k \Delta_{k+1}^{-1} C_k^T$$

Going from step $k$ to step $k+1$ we compute $C_{k+1}$ incrementally

Note that we can reuse $C_k \Delta_{k+1}^{-1}$ to compute $C_{k+1}$

## Example F3, GB, $n = 100$, $A^{-1}_{2,1} = -3.2002$

| Nit | G | G–R $b_L$ | G–R $b_U$ | G–L |
|---|---|---|---|---|
| 2 | -3.0808 | -3.0948 | -3.9996 | -4.1691 |
| 3 | -3.1274 | -3.1431 | -3.5655 | -3.6910 |
| 4 | -3.2204 | -3.2187 | -3.2637 | -3.5216 |
| 5 | -3.2015 | -3.2001 | -3.1974 | -3.2473 |
| 6 | -3.1969 | -3.1966 | -3.1964 | -3.1969 |
| 7 | -3.1970 | -3.1972 | -3.1995 | -3.1994 |
| 8 | -3.1993 | -3.1995 | -3.2008 | -3.1999 |
| 9 | -3.2001 | -3.2001 | -3.2005 | -3.2008 |
| 10 | -3.2002 | -3.2002 | -3.2002 | -3.2004 |

We see that we obtain good approximations but not always bounds
As a bonus we also obtain estimates of $A^{-1}_{1,1}$ and $A^{-1}_{2,2}$

Example F4, GB, $n = 900$, $A^{-1}_{400,100} = 0.0597$

| Nit | G | G–R $b_L$ | G–R $b_U$ | G–L |
|---|---|---|---|---|
| 10 | 0.0172 | 0.0207 | 0.0632 | 0.0588 |
| 20 | 0.0527 | 0.0532 | 0.0616 | 0.0621 |
| 30 | 0.0590 | 0.0591 | 0.0597 | 0.0597 |
| 40 | 0.0597 | 0.0597 | 0.0597 | 0.0597 |

Note that for this problem the Gauss rule gives a lower bound, Gauss–Radau a lower and an upper bound

# Dependence on the eigenvalue estimates

We take Example F4 with $m = 6$

We look at the number of Lanczos iterations needed to obtain an upper bound for the element $(18, 18)$ with four exact digits

Example F4, GL, $n = 36$

| a=$10^{-4}$ | $10^{-2}$ | 0.1 | 0.3 | 0.4 | 1 | 6 |
|---|---|---|---|---|---|---|
| 15 | 13 | 11 | 11 | 8 | 8 | 9 |

With the exact eigenvalue $a = 0.3961$ we need 9 Lanczos iterations

Note that it works even when $a > \lambda_{min}$

# Bounds for the elements of the exponential

Example F3, GL, $n = 100$, $exp(A)_{50,50} = 5.3217\ 10^{41}$. Results $\times 10^{-41}$

| Nit | G | G–R $b_L$ | G–R $b_L$ | G–L |
|---|---|---|---|---|
| 2 | 0.0000 | 0.0000 | 7.0288 | 8.8014 |
| 3 | 0.0075 | 0.2008 | 5.6649 | 6.0776 |
| 4 | 1.0322 | 2.5894 | 5.3731 | 5.4565 |
| 5 | 3.9335 | 4.7779 | 5.3270 | 5.3385 |
| 6 | 5.1340 | 5.2680 | 5.3235 | 5.3232 |
| 7 | 5.3070 | 5.3178 | 5.3218 | 5.3219 |
| 8 | 5.3203 | 5.3209 | 5.3218 | 5.3218 |
| 9 | 5.3212 | 5.3213 | 5.3217 | 5.3217 |
| 10 | 5.3215 | 5.3217 | 5.3217 | 5.3217 |
| 11 | 5.3217 | 5.3217 | 5.3217 | 5.3217 |

Convergence is faster than with $A^{-1}$

Example F4, GNS, $n = 900$, $exp(A)_{50,50} + exp(A)_{50,49} = 83.8391$

| rule | Nit=2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| G | 63.4045 | 81.4124 | 83.6607 | 83.8318 | 83.8389 | 83.8391 |
| G–R $b_L$ | 108.0918 | 86.3239 | 83.8796 | 83.8420 | 83.8392 | 83.8391 |
| G–R $b_U$ | 76.1266 | 83.7668 | 83.7781 | 83.8383 | 83.8391 | 83.8391 |
| G–L | 163.8043 | 90.9304 | 84.1878 | 83.8530 | 83.8395 | 83.8391 |

Convergence is quite fast

# Bounds for the elements of the square root

Example F4, GL, $n = 900$, $(\sqrt{A})_{50,50} = 1.9189$

| Nit | G | G–R $b_L$ | G–R $b_U$ | G–L |
|---|---|---|---|---|
| 2 | 1.9319 | 1.8945 | 1.9255 | 1.8697 |
| 3 | 1.9220 | 1.9112 | 1.9209 | 1.9038 |
| 4 | 1.9201 | 1.9160 | 1.9197 | 1.9140 |
| 5 | 1.9195 | 1.9176 | 1.9193 | 1.9169 |
| 6 | 1.9192 | 1.9183 | 1.9191 | 1.9180 |
| 7 | 1.9191 | 1.9186 | 1.9190 | 1.9185 |
| 8 | 1.9190 | 1.9187 | 1.9190 | 1.9187 |
| 9 | 1.9190 | 1.9188 | 1.9190 | 1.9188 |
| 10 | 1.9190 | 1.9189 | 1.9190 | 1.9189 |
| 11 | 1.9190 | 1.9189 | 1.9190 | 1.9189 |
| 12 | 1.9190 | 1.9189 | 1.9189 | 1.9189 |
| 13 | 1.9189 | 1.9189 | 1.9189 | 1.9189 |

# Jacobi matrices

For our application to compute $u^T f(A) u$ we know how to compute the Jacobi matrix from the Lanczos algorithm

When computing quadrature rules for classical weight functions (Legendre, Chebyshev, Laguerre, Hermite,...) we know explicitly the Jacobi matrices

But, more generally, how can we compute the Jacobi matrix (the coefficients of the three-term recurrence of orthogonal polynomials) ?

We may assume that we know either the measure $\alpha$ or the moments $\mu_k$

# The Stieltjes procedure

Computation from the measure

With a discrete inner product, sums like

$$\langle p, q\rangle = \sum_{j=1}^{m} p(t_j)q(t_j)w_j^2$$

are trivial to compute given the nodes $t_j$ and the weights $w_j^2$
The coefficients of the three–term recurrence are given by

$$\alpha_{k+1} = \frac{\langle \lambda p_k, p_k\rangle}{\langle p_k, p_k\rangle}, \quad \gamma_k = \frac{\langle p_k, p_k\rangle}{\langle p_{k-1}, p_{k-1}\rangle}$$

for a monic polynomial

- $p_0 \equiv 1 \to \alpha_1$
- $\alpha_1 \to p_1(t_j)$ (three-term recurrence)
- $p_1(t_j), w_j \to \gamma_1, \alpha_2$
- $\gamma_1, \alpha_2 \to p_2(t_j)$ (three-term recurrence)
- $\ldots$

For a continuous measure, discretize first, then apply Stieltjes

# Computation from the moments

see Szegö or Gautschi
Let

$$\Delta_0 = 1, \quad \Delta_k = \det(H_k), \quad H_k = \begin{pmatrix} \mu_0 & \mu_1 & \cdots & \mu_{k-1} \\ \mu_1 & \mu_2 & \cdots & \mu_k \\ \vdots & \vdots & & \vdots \\ \mu_{k-1} & \mu_k & \cdots & \mu_{2k-2} \end{pmatrix}, \; k = 1, 2, .$$

and

$$\Delta'_0 = 0, \; \Delta'_1 = \mu_1, \; \Delta'_k = \det \begin{pmatrix} \mu_0 & \mu_1 & \cdots & \mu_{k-2} & \mu_k \\ \mu_1 & \mu_2 & \cdots & \mu_{k-1} & \mu_{k+1} \\ \vdots & \vdots & & \vdots & \vdots \\ \mu_{k-1} & \mu_k & \cdots & \mu_{2k-3} & \mu_{2k-1} \end{pmatrix}, \; k = 2, 3$$

### Theorem

*The monic orthogonal polynomial $\pi_k$ of degree $k$ associated with the moments $\mu_j$, $j = 0, \ldots, 2k-1$ is*

$$\pi_k(\lambda) = \frac{1}{\Delta_k} \det \begin{pmatrix} \mu_0 & \mu_1 & \cdots & \mu_k \\ \mu_1 & \mu_2 & \cdots & \mu_{k+1} \\ \vdots & \vdots & & \vdots \\ \mu_{k-1} & \mu_k & \cdots & \mu_{2k-1} \\ 1 & \lambda & \cdots & \lambda^k \end{pmatrix}, \quad k = 1, 2, \ldots$$

### Theorem

*The recursion coefficients of the three–term recurrence for the polynomial $\pi_k$*

$$\pi_{k+1}(\lambda) = (\lambda - \alpha_{k+1})\pi_k(\lambda) - \gamma_k \pi_{k-1}(\lambda), \quad \pi_{-1}(\lambda) = 0,\ \pi_0(\lambda) = 1$$

*are given by*

$$\alpha_{k+1} = \frac{\Delta'_{k+1}}{\Delta_{k+1}} - \frac{\Delta'_k}{\Delta_k}, \;\; k = 0, 1, \ldots$$

$$\gamma_k = \frac{\Delta_{k+1}\Delta_{k-1}}{\Delta_k^2}, \;\; k = 1, 2, \ldots$$

The map moments $\longrightarrow$ coefficients is badly conditioned (see Gautschi)

Gautschi used the Cholesky factorization of the Hankel matrix

$$H_k = R_k^T R_k$$

to obtain the coefficients of the Jacobi matrix

## Theorem

*Let $H_k = R_k^T R_k$ be the Cholesky factorization of the moment matrix. The coefficients of the orthonormal polynomial are given by*

$$\eta_j = \frac{r_{j+1,j+1}}{r_{j,j}}, \; j = 1, \ldots, k-1 \quad \alpha_1 = r_{1,2}, \; \alpha_j = \frac{r_{j,j+1}}{r_{j,j}} - \frac{r_{j-1,j}}{r_{j-1,j-1}}, \; j = 2, ..$$

# The modified Chebyshev algorithm

Using the moments $\mu_k$ to compute the recurrence coefficients of $\pi_k$ is not be numerically safe (see Gautschi)

Remedy: use another family of known orthogonal polynomials (Wheeler; Sack and Donovan)

The **modified moments** (using orthogonal known polynomials $p_k$) are

$$m_k = \int_a^b p_k(\lambda)\, d\alpha$$

$$p_{k+1}(\lambda) = (\lambda - a_{k+1})p_k(\lambda) - c_k p_{k-1}(\lambda), \quad p_{-1}(\lambda) \equiv 0,\ p_0(\lambda) \equiv 1$$

The **mixed moments** related to $p_l$ and $\alpha$ are

$$\sigma_{k,l} = \int_a^b \pi_k(\lambda) p_l(\lambda)\, d\alpha(\lambda)$$

$$\sigma_{0,l} = m_l, \;\; l = 0, \ldots, 2m-1$$

By orthogonality, we have $\sigma_{k,l} = 0,\; k > l$ and

$$\sigma_{k,k} = \int_a^b \pi_k(\lambda)\lambda p_{k-1}(\lambda)\, d\alpha(\lambda) = \int_a^b \pi_k^2(\lambda)\, d\alpha(\lambda)$$

Algorithm: compute recursively the mixed moments and the coefficients of $\pi_k$

The mixed moments at level $k$ are given by

$$\sigma_{k,l} = \sigma_{k-1,l+1} - (\alpha_k - a_{l+1})\sigma_{k-1,l} - \eta_{k-1}\sigma_{k-2,l} + c_l\sigma_{k-1,l-1}$$

$$(k-2,l),(k-1,l-1),(k-1,l),(k-1,l+1) \rightarrow (k,l)$$

The modified Chebyshev algorithm is

$$\sigma_{-1,l} = 0,\ l = 1,\dots,2m-2, \quad \sigma_{0,l} = m_l,\ \ l = 0,1,\dots,2m-1$$

$$\alpha_1 = a_1 + \frac{m_1}{m_0}$$

and for $k = 1,\dots,m-1$

$$\sigma_{k,l} = \sigma_{k-1,l+1} + (a_{l+1} - \alpha_k)\sigma_{k-1,l} \quad + \quad c_l\sigma_{k-1,l-1} - \eta_{k-1}\sigma_{k-2,l}$$

$$l = k,\dots,2m-k-1$$

$$\alpha_{k+1} = a_{k+1} + \frac{\sigma_{k,k+1}}{\sigma_{k,k}} - \frac{\sigma_{k-1,k}}{\sigma_{k-1,k-1}}$$

$$\eta_k = \frac{\sigma_{k,k}}{\sigma_{k-1,k-1}}$$

# Inverse eigenvalue problem

The following problem is related to ours:

Given its eigenvalues and the first components of its eigenvectors, construct a Jacobi matrix $J_k$

- which means, given the nodes and the weights of a quadrature rule can we recover the orthogonal polynomials?

see De Boor and Golub; Gragg and Harrod; Reichel; Laurie

# Solution by the Lanczos algorithm

Take $A = \Lambda$ diagonal matrix of the eigenvalues $t_j$, then

$$v^k = p_k(A)v^1, \quad v^1 = v$$

and

$$(v^i, v^j) = (p_j(\Lambda_m)v, p_i(\Lambda_m)v) = \sum_{l=1}^{m} p_j(t_l)p_i(t_l)v_l^2 = \delta_{i,j}$$

Hence, if the initial vector $v$ is chosen as the vector of the first components, the Lanczos polynomials are orthogonal for the given discrete inner product and the Jacobi matrix which is sought is the tridiagonal matrix generated by the Lanczos algorithm

# Solution using rotations

## Gragg and Harrod; Reichel

Let $d$ be a vector whose elements are $\beta_0$ times the given first components

Assume that

$$\begin{pmatrix} 1 & \\ & Q^T \end{pmatrix} \begin{pmatrix} \alpha_0 & d^T \\ d & \Lambda \end{pmatrix} \begin{pmatrix} 1 & \\ & Q \end{pmatrix} = \begin{pmatrix} \alpha_0 & \beta_0 (e^1)^T \\ \beta_0 e^1 & J_n \end{pmatrix}$$

with $Q$ an orthogonal matrix

We construct $Q$ incrementally. Let us add $(\delta, \lambda)$ to $(d, \Lambda)$

$$\begin{pmatrix} 1 & & \\ & Q^T & \\ & & 1 \end{pmatrix} \begin{pmatrix} \alpha_0 & d^T & \delta \\ d & \Lambda & 0 \\ \delta & 0 & \lambda \end{pmatrix} \begin{pmatrix} 1 & & \\ & Q & \\ & & 1 \end{pmatrix} = \begin{pmatrix} \alpha_0 & \beta_0 (e^1)^T & \delta \\ \beta_0 e^1 & J_n & 0 \\ \delta & 0 & \lambda \end{pmatrix}$$

To tridiagonalize the matrix in the right hand side, we use rotations to chase the element $\delta$ in the last row and column towards the diagonal

The Kahan–Pal–Walker version of this algorithm is the most efficient one: it squares some equations to update the squares of most of the involved quantities

$\gamma_0^2 = 1$, $\beta_n^2 = \sigma_0^2 = \tau_0 = 0$, $\alpha_{n+1} = \lambda$, $\pi_0^2 = \delta^2$

for $k = 1, \ldots, n+1$

$\rho_k^2 = \beta_{k-1}^2 + \pi_{k-1}^2$, $\bar{\beta}_{k-1}^2 = \gamma_{k-1}^2 \rho_k^2$

if $\rho_k^2 = 0$ then $\gamma_k^2 = 1$, $\sigma_k^2 = 0$ else

$\gamma_k^2 = \beta_{k-1}^2 / \rho_k^2$, $\sigma_k^2 = \pi_{k-1}^2 / \rho_k^2$

$\tau_k = \sigma_k^2(\alpha_k - \lambda) - \gamma_k^2 \tau_{k-1}$

$\bar{\alpha}_k = \alpha_k - (\tau_k - \tau_{k-1})$

if $\sigma_k^2 = 0$ then $\pi_k^2 = \sigma_{k-1}^2 \beta_{k-1}^2$ else $\pi_k^2 = \tau_k^2 / \sigma_k^2$

end

Note that if $\xi_1 = \alpha_1 - \lambda$ and

$$\xi_k = \alpha_k - \lambda - \frac{\beta_k^2}{\xi_{k-1}}$$

(which are the diagonal elements of the Cholesky–like factorization) then $\tau_k = \sigma_k^2 \xi_k$ and $\pi_k^2 = \tau_k \xi_k$

# Solution using the QD algorithm

The basic QD algorithm (Rutishauser) given the Cholesky decomposition $J_k = L_k^T L_k$ computes the Cholesky decomposition $\hat{L}_k \hat{L}_k^T$ of $\hat{J}_k = L_k^T L_k$

Variations of this algorithm were used by Laurie to solve the inverse problem with his algorithm pftoqd

# Modifications of weight functions

How to obtain the coefficients of the three-term recurrences of orthogonal polynomials related to a weight function $r(\lambda)w(\lambda)$ when knowing the coefficients of the orthogonal polynomials related to $w$?

When $r$ is a rational function

$$r(\lambda) = q(\lambda) + \sum_i \frac{a_i}{\lambda - t_i} + \sum_j \frac{b_j \lambda + c_j}{(\lambda - x_j)^2 + y_j^2}$$

where $q$ is a real polynomial, $t_i$, $i = 1, 2, \dots$ and $z_j = x_j \pm \imath y_j$, $\imath = \sqrt{-1}$, $j = 1, 2, \dots$ are the real and complex roots of the denominator of $r$

Hence, we just have to consider multiplication and division by linear and quadratic factors as well as addition of measures

This was done by Fischer and Golub; Gautschi; Kautsky and Golub; Golub and Fischer; Elhay and Kautsky

The most difficult one is the division algorithm

# Error norms in solving linear systems

Let $A$ be an SPD matrix of order $n$ and $\tilde{x}$ an approximate solution of

$$Ax = c$$

The residual $r$ is defined as

$$r = c - A\tilde{x}$$

The error $\epsilon$ being defined as $\epsilon = x - \tilde{x}$

$$\epsilon = A^{-1}r$$

The $A$–norm of the error is

$$\|\epsilon\|_A^2 = \epsilon^T A \epsilon = r^T A^{-1} A A^{-1} r = r^T A^{-1} r$$

and the $l_2$ norm is $\|\epsilon\|^2 = r^T A^{-2} r$

$$I[A,r] = r^T A^{-i} r = \int_a^b \lambda^{-i} \, d\alpha(\lambda)$$

Bounds can be obtained by running $N$ iterations of the Lanczos algorithm

$$\|r\|^2 (e^1)^T (J_N)^{-i} e^1$$

However

$$\text{CG} \equiv \text{Lanczos}$$

therefore, it does not make to much sense to run Lanczos to bound the error norm of CG!

What can we do for CG?

# Formulas for the $A$–norm of the error in CG

Theorem

*The square of the $A$–norm of the error at CG iteration $k$ is given by*

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

*where $n$ is the order of the matrix $A$ and $J_k$ is the Jacobi matrix of the Lanczos algorithm whose coefficients can be computed from those of CG. Moreover*

$$\|\epsilon^k\|_A^2 = \|r^0\|^2 \left[\sum_{j=1}^{n} \frac{[(z_{(n)}^j)_1]^2}{\lambda_j} - \sum_{j=1}^{k} \frac{[(z_{(k)}^j)_1]^2}{\theta_j^{(k)}}\right]$$

*where $z_{(k)}^j$ is the $j$th normalized eigenvector of $J_k$ corresponding to the eigenvalue $\theta_j^{(k)}$*

*Proof.*
We have $A\epsilon^k = r^k = r^0 - AV_k y^k$ where $V_k$ is the matrix of the Lanczos vectors and $y^k$ is the solution of $J_k y^k = \|r^0\| e^1$

$$\|\epsilon^k\|_A^2 = (A\epsilon^k, \epsilon^k) = (A^{-1}r^0, r^0) - 2(r^0, V_k y^k) + (AV_k y^k, V_k y^k)$$

But $A^{-1}V_n = V_n J_n^{-1}$

$$r^0 = \|r^0\| v^1 = \|r^0\| V_n e^1$$

Therefore

$$A^{-1}r^0 = \|r^0\| A^{-1} V_n e^1 = \|r^0\| V_n J_n^{-1} e^1$$

and

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

Since $r^0 = \|r^0\| v^1 = \|r^0\| V_k e^1$

$$(r^0, V_k y^k) = \|r^0\|^2 (e^1, J_k^{-1} e^1)$$

Finally

$$(AV_k y^k, V_k y^k) = (V_k^T A V_k y^k, y^k) = (J_k y^k, y^k) = \|r^0\|^2 (J_k^{-1} e^1, e^1)$$

The second relation is obtained by using the spectral decomposition of $J_n$ and $J_k$ □

This formula is the link between CG and Gauss quadrature

It shows that the square of the $A$–norm of the error is the remainder of a Gauss quadrature rule for computing $(A^{-1} r^0, r^0)$

# Estimates of the $A$–norm of the error

At CG iteration $k$ we do not know $(J_n^{-1})_{1,1}$!
Let $d$ be a given delay integer, an approximation of the $A$–norm of the error at iteration $k-d$ is obtained by

$$\|\epsilon^{k-d}\|_A^2 \approx \|r^0\|^2((J_k^{-1})_{(1,1)} - (J_{k-d}^{-1})_{(1,1)})$$

– This can also be understood as writing

$$\|\epsilon^{k-d}\|_A^2 - \|\epsilon^k\|_A^2 = \|r^0\|^2((J_k^{-1})_{(1,1)} - (J_{k-d}^{-1})_{(1,1)})$$

and supposing that $\|\epsilon^k\|_A$ is negligible against $\|\epsilon^{k-d}\|_A$
– Another interpretation is to consider that having a Gauss rule with $k-d$ nodes at iteration $k-d$, we use another more precise Gauss quadrature with $k$ nodes to estimate the error of the quadrature rule

We have to be careful in computing $(J_k^{-1})_{(1,1)} - (J_{k-d}^{-1})_{(1,1)}$
Let $j_k = J_k^{-1} e^k$ be the last column of the inverse of $J_k$; Using the Sherman–Morrison formula

$$(J_{k+1}^{-1})_{1,1} = (J_k^{-1})_{1,1} + \frac{\eta_{k+1}^2 (j_k j_k^T)_{1,1}}{\alpha_{k+1} - \eta_{k+1}^2 (j_k)_k}$$

Using the Cholesky factorization of $J_k$ whose diagonal elements are $\delta_1 = \alpha_1$ and

$$\delta_i = \alpha_i - \frac{\eta_i^2}{\delta_{i-1}}, \quad i = 2, \dots, k$$

Then

$$(j_k)_1 = (-1)^{k-1} \frac{\eta_2 \cdots \eta_k}{\delta_1 \cdots \delta_k}, \quad (j_k)_k = \frac{1}{\delta_k}$$

Let $b_k = (J_k^{-1})_{1,1}$

$$b_k = b_{k-1} + f_k, \quad f_k = \frac{\eta_k^2 c_{k-1}^2}{\delta_{k-1}(\alpha_k \delta_{k-1} - \eta_k^2)} = \frac{c_k^2}{\delta_k}$$

where

$$c_k = \frac{\eta_2 \cdots \eta_{k-1}}{\delta_1 \cdots \delta_{k-2}} \frac{\eta_k}{\delta_{k-1}} = c_{k-1} \frac{\eta_k}{\delta_{k-1}}$$

Since $J_k$ is positive definite, $f_k > 0$
Moreover

$$c_k = \frac{\eta_2 \cdots \eta_k}{\delta_1 \cdots \delta_{k-1}} = \frac{\|r^{k-1}\|}{\|r^0\|}$$

and $\gamma_{k-1} = 1/\delta_k$ where $\gamma_{k-1}$ is the CG parameter ($=(r^{k-1}, r^{k-1})/(p^{k-1}, Ap^{k-1})$)

Therefore

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

This gives a lower bound of the error norm

Other bounds can be obtained with the Gauss–Radau and Gauss–Lobatto quadrature rules

Gauss–Radau gives an upper bound of the error norm if we know a lower bound of the smallest eigenvalue

# Algorithm CGQL

Let $x^0$ be given, $r^0 = b - Ax^0$, $p^0 = r^0$, $\beta_0 = 0$, $\alpha_{-1} = 1$, $c_1 = 1$
For $k = 1, \ldots$ until convergence

$$\gamma_{k-1} = \frac{(r^{k-1}, r^{k-1})}{(p^{k-1}, Ap^{k-1})}$$

$$\alpha_k = \frac{1}{\gamma_{k-1}} + \frac{\beta_{k-1}}{\gamma_{k-2}}$$

# CGQL (2)

if $k = 1$

$$f_1 = \frac{1}{\alpha_1}$$

$$\delta_1 = \alpha_1$$

$$\bar{\delta}_1 = \alpha_1 - \lambda_m$$

$$\underline{\delta}_1 = \alpha_1 - \lambda_M$$

else

$$c_k = c_{k-1}\frac{\eta_k}{\delta_{k-1}} = \frac{\|r^{k-1}\|}{\|r^0\|}$$

$$\delta_k = \alpha_k - \frac{\eta_k^2}{\delta_{k-1}} = \frac{1}{\gamma_{k-1}}$$

$$f_k = \frac{\eta_k^2 c_{k-1}^2}{\delta_{k-1}(\alpha_k \delta_{k-1} - \eta_k^2)} = \gamma_{k-1} c_k^2$$

# CGQL (3)

$$\bar{\delta}_k = \alpha_k - \lambda_m - \frac{\eta_k^2}{\bar{\delta}_{k-1}} = \alpha_k - \bar{\alpha}_{k-1}$$

$$\underline{\delta}_k = \alpha_k - \lambda_M - \frac{\eta_k^2}{\underline{\delta}_{k-1}} = \alpha_k - \underline{\alpha}_{k-1}$$

end

$$x^k = x^{k-1} + \gamma_{k-1} p^{k-1}$$

$$r^k = r^{k-1} - \gamma_{k-1} A p^{k-1}$$

$$\beta_k = \frac{(r^k, r^k)}{(r^{k-1}, r^{k-1})}$$

$$\eta_{k+1} = \frac{\sqrt{\beta_k}}{\gamma_{k-1}}$$

$$p^k = r^k + \beta_k p^{k-1}$$

# CGQL (4)

$$\bar{\alpha}_k = \lambda_m + \frac{\eta_{k+1}^2}{\bar{\delta}_k}$$

$$\underline{\alpha}_k = \lambda_M + \frac{\eta_{k+1}^2}{\underline{\delta}_k}$$

$$\breve{\alpha}_k = \frac{\bar{\delta}_k \underline{\delta}_k}{\underline{\delta}_k - \bar{\delta}_k} \left( \frac{\lambda_M}{\bar{\delta}_k} - \frac{\lambda_m}{\underline{\delta}_k} \right)$$

$$\breve{\eta}_{k+1}^2 = \frac{\bar{\delta}_k \underline{\delta}_k}{\underline{\delta}_k - \bar{\delta}_k} (\lambda_M - \lambda_m)$$

$$\bar{f}_k = \frac{\eta_{k+1}^2 c_k^2}{\delta_k (\bar{\alpha}_k \delta_k - \eta_{k+1}^2)}$$

$$\underline{f}_k = \frac{\eta_{k+1}^2 c_k^2}{\delta_k (\underline{\alpha}_k \delta_k - \eta_{k+1}^2)}$$

$$\breve{f}_k = \frac{\breve{\eta}_{k+1}^2 c_k^2}{\delta_k (\breve{\alpha}_k \delta_k - \breve{\eta}_{k+1}^2)}$$

# CGQL (5)

if $k > d$

$$g_k = \sum_{j=k-d+1}^{k} f_j$$

$$s_{k-d} = \|r^0\|^2 g_k$$

$$\bar{s}_{k-d} = \|r^0\|^2 (g_k + \bar{f}_k)$$

$$\underline{s}_{k-d} = \|r^0\|^2 (g_k + \underline{f}_k)$$

$$\check{s}_{k-d} = \|r^0\|^2 (g_k + \check{f}_k)$$

end

### Proposition

*Let $J_k$, $\underline{J}_k$, $\bar{J}_k$ and $\breve{J}_k$ be the tridiagonal matrices of the Gauss, Gauss–Radau (with $b$ and $a$ as prescribed nodes) and the Gauss–Lobatto rules*

*Then, if $0 < a = \lambda_m \leq \lambda_{min}(A)$ and $b = \lambda_M \geq \lambda_{max}(A)$, $\|r^0\|(J_k^{-1})_{1,1}$, $\|r^0\|(\underline{J}_k^{-1})_{1,1}$ are lower bounds of $\|e^0\|_A^2 = r^0 A^{-1} r^0$, $\|r^0\|(\bar{J}_k^{-1})_{1,1}$ and $\|r^0\|(\breve{J}_k^{-1})_{1,1}$ are upper bounds of $r^0 A^{-1} r^0$*

### Theorem

*At iteration number $k$ of CGQL, $s_{k-d}$ and $\underline{s}_{k-d}$ are lower bounds of $\|\epsilon^{k-d}\|_A^2$, $\bar{s}_{k-d}$ and $\breve{s}_{k-d}$ are upper bounds of $\|\epsilon^{k-d}\|_A^2$*

# Preconditioned CG

For the preconditioned CG algorithm, the formula to consider is

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

where $Mz^0 = r^0$, $M$ being the preconditioner, a symmetric positive definite matrix that is chosen to speed up the convergence

The Gauss rule estimate is

$$\|\epsilon^{k-d}\|_A^2 \approx \sum_{j=k-d}^{k-1} \gamma_j (z^j, r^j)$$

where

$$Mz^j = r^j$$

# Estimates of the $l_2$ norm of the error

**Theorem**

$$\begin{aligned}\|\epsilon^k\|^2 &= \|r^0\|^2[(e^1, J_n^{-2}e^1) - (e^1, J_k^{-2}e^1)] \\ &+ (-1)^k 2\eta_{k+1}\frac{\|r^0\|}{\|r^k\|}(e^k, J_k^{-2}e^1)\|\epsilon^k\|_A^2\end{aligned}$$

**Corollary**

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

This can be computed introducing a delay and using a QR factorization of $J_k$

# Relation with finite element problems

Suppose we want to solve a PDE

$$\mathcal{L}u = f \quad \text{in } \Omega$$

$\Omega$ being a two or three–dimensional bounded domain, with appropriate boundary conditions on $\Gamma$ the boundary of $\Omega$
As a simple example, consider the PDE

$$-\Delta u = f, \quad u|_\Gamma = 0$$

This problem is naturally formulated in the Hilbert space $H_0^1(\Omega)$

$$a(u, v) = (f, v), \ \forall v \in V = H_0^1(\Omega)$$

where $a(u, v)$ is a self–adjoint bilinear form

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v \; dx$$

and

$$(f,v) = \int_\Omega fv \, dx$$

There is a unique solution $u \in H_0^1(\Omega)$

The approximate solution is sought in a finite dimensional subspace $V_h \subset V$ as

$$a(u_h, v_h) = (f, v_h), \;\; \forall v_h \in V_h$$

The simplest method triangulates the domain $\Omega$ (with triangles or tetrahedrons of maximal diameter $h$) and uses functions which are linear on each element

Using basis functions $\phi_i$ which are piecewise linear and have a value 1 at vertex $i$ and 0 at all the other vertices

$$v_h(x) = \sum_{j=1}^{n} v_j \phi_j(x)$$

The approximated problem is equivalent to a linear system $Au = c$, where

$$[A]_{i,j} = a(\phi_i, \phi_j), \quad c_i = (f, \phi_i)$$

The matrix $A$ is symmetric and positive definite. The solution of the finite dimensional problem is

$$u_h(x) = \sum_{j=1}^{n} u_j \phi_j(x)$$

We use CG to solve the linear system

We have two sources of errors, the difference between the exact and approximate solution $u - u_h$ and $u_h - u_h^{(k)}$, the difference between the approximate solution and its CG computed value (not speaking of rounding errors)

Of course, we desire the components of $u - u_h^{(k)}$ to be small. This depends on $h$ and on the CG stopping criterion
The problem of finding an appropriate stopping criterion has been studied by Arioli and al (on these topics, see also Jiranek, Strakoš and Vohralik)
Let $\|v\|_a^2 = a(v, v)$ and $u_h^* \in V_h$ be such that

$$\|u_h - u_h^*\|_a^2 \le h^{2t} \|u_h\|_a^2$$

Then

$$\begin{aligned} \|u - u_h^*\|_a &\le \|u - u_h\|_a + \|u_h - u_h^*\| \\ &\le h^t \|u\|_a + (1 + h^t) \|u - u_h\|_a \end{aligned}$$

If $t > 0$ and $h < 1$

$$\|u - u_h^*\|_a \leq h^t \|u\|_a + 2\|u - u_h\|_a$$

Therefore, if $u_h^* = u_h^{(k)}$ and we choose $\|u_h - u_h^*\|_a$ such that $h^t\|u\|_a$ is of the same order as $\|u - u_h\|_a$ we have

$$\|u - u_h^*\|_a \approx \|u - u_h\|_a$$

We have

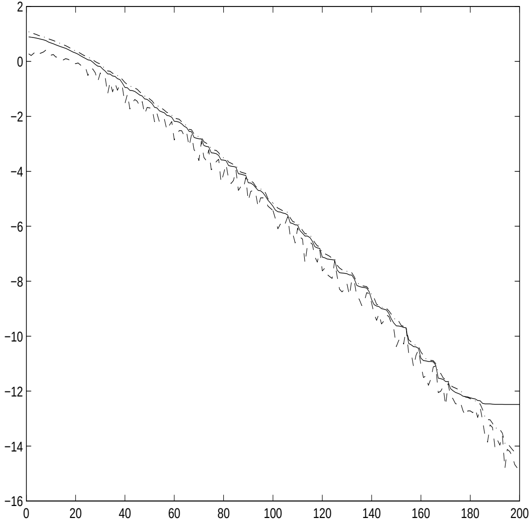
$$\|v_h^{(k)}\|_a = \|v^k\|_A$$

Let $\zeta_k$ be an estimate of $\|\varepsilon^k\|_A^2$, Arioli's stopping test is

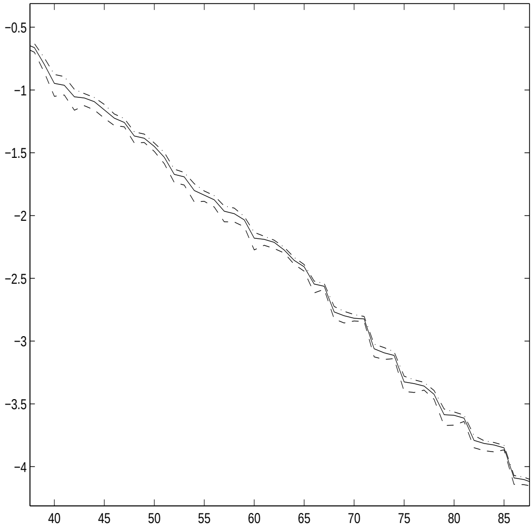
$$\text{If } \zeta_k \leq \eta^2((u^k)^T r^0 + c^T u^0) \text{ then stop}$$

The parameter $\eta$ is chosen as $h$ or $\eta^2$ as the maximum area of the triangles in 2D

# Numerical experiments

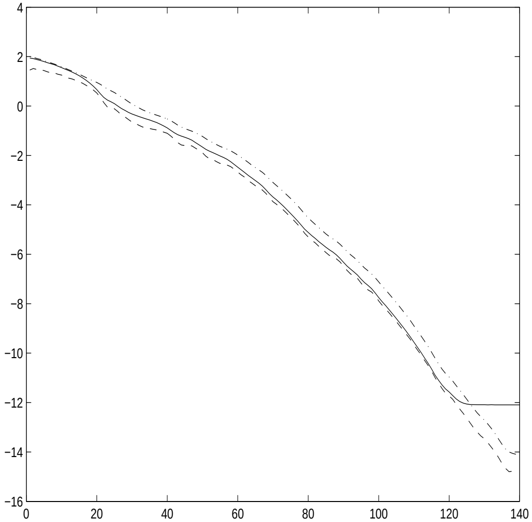



F3, $d = 1$, $\log_{10}$ of the $A$–norm of the error (plain), Gauss (dashed), Gauss–Radau($\lambda_{min}$) (dot–dashed)



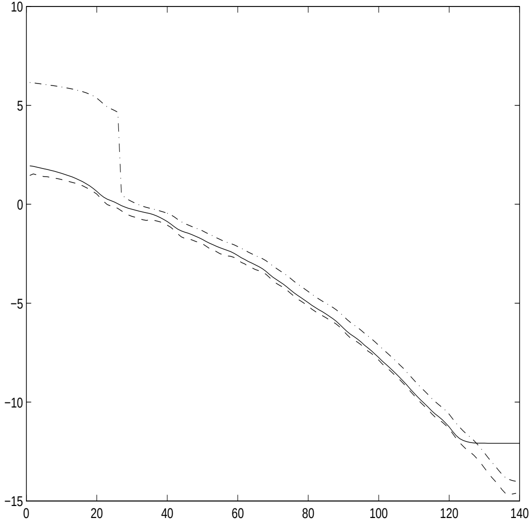

F3, $d = 5$, zoom of $\log_{10}$ of the $A$–norm of the error (plain), Gauss (dashed), Gauss–Radau (dot–dashed)

For the Gauss–Radau upper bound we use a value of $a = 0.02$ whence the smallest eigenvalue is $\lambda_{min} = 0.025$




F4, $n = 900$, $d = 1$, $\log_{10}$ of the $A$–norm of the error (plain), Gauss (dashed), Gauss–Radau (dot–dashed)

# Adaptive algorithm for the smallest eigenvalue




F4, $n = 900$, $d = 1$, est. of $\lambda_{min}$, $\log_{10}$ of the $A$–norm of the error (plain), Gauss (dashed), Gauss–Radau (dot–dashed)

# Another example (CG2)

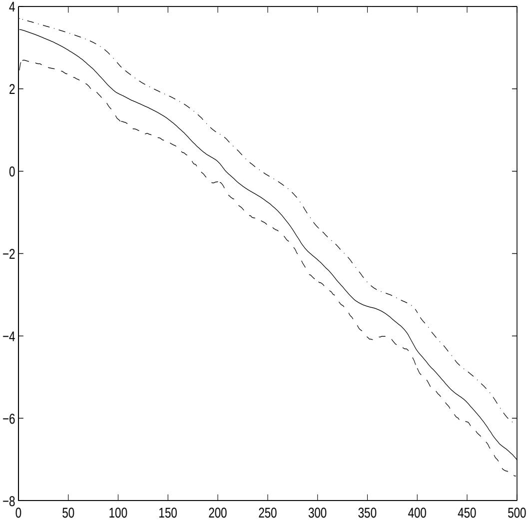
$$-\mathrm{div}(\lambda(x,y)\nabla u) = f, \quad u|_\Gamma = 0$$

Finite differences in the unit square

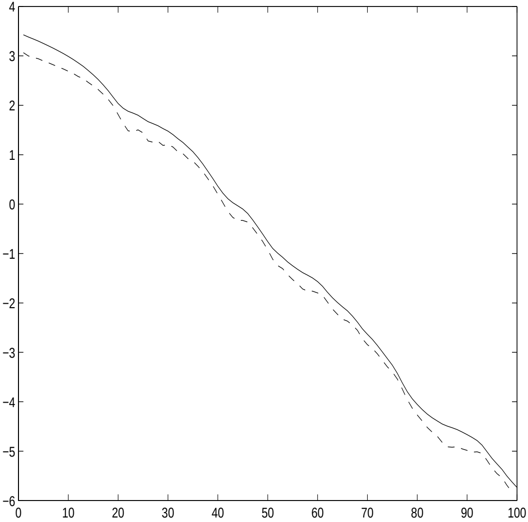
$$\lambda(x,y) = \frac{1}{(2 + p\sin\frac{x}{\eta})(2 + p\sin\frac{y}{\eta})}$$

We use $p = 1.8$ and $\eta = 0.1$

We compute $f$ such that the solution is $u(x,y) = \sin(\pi x)\sin(\pi y)$




CG2, $d = 1$, $n = 10000$, $\log_{10}$ of the $A$–norm of the error (plain), Gauss (dashed), Gauss–Radau (dot–dashed), $a = 10^{-4}$, $\lambda_{min} = 2.3216\ 10^{-4}$




CG2, $d = 1$, $n = 10000$, IC(0), $\log_{10}$ of the $A$–norm of the error (plain), Gauss (dashed)

# Stopping criterion

Since we are using finite differences and we have multiplied the right hand side by $h^2$, we modify the Arioli's criteria to

$$\text{If } \zeta_k \le 0.1 * (1/n)^2((x^k)^T r^0 + c^T x^0) \text{ then stop}$$

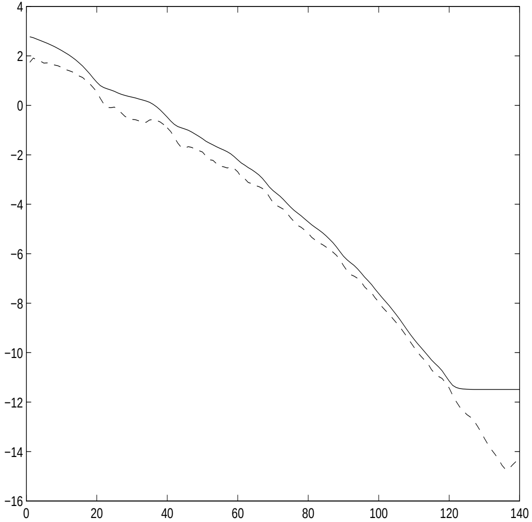
where $\zeta_k$ is an estimate of $\|\epsilon^k\|_A^2$

When using $n = 10000$, the $A$–norm of the difference between the "exact" solution of the linear system (obtained by Gaussian elimination) and the discretization of $u$ is $n_u = 5.6033\ 10^{-5}$

With the previous stopping criterion, we do 226 iterations and we have $n_x = 9.5473\ 10^{-5}$

Using an incomplete Cholesky preconditioner IC(0) we do 47 iterations and obtain $n_x = 5.6033\ 10^{-5}$

# Bound of the $l_2$ norm of the error




F4, $d = 1$, $n = 900$, $\log_{10}$ of the $l_2$ norm of the error (plain), Gauss (dashed)

C. de Boor and G.H. Golub, *The numerically stable reconstruction of a Jacobi matrix from spectral data*, Linear Alg. Appl., v 21, (1978), pp 245–260

D. Calvetti, L. Reichel and F. Sgallari, *Application of anti–Gauss rules in linear algebra*, in Applications and Computation of Orthogonal Polynomials, W. Gautschi, G.H. Golub and G. Opfer Eds, Birkhauser, (1999), pp 41–56

D. Calvetti, S. Morigi, L. Reichel and F. Sgallari, *Computable error bounds and estimates for the conjugate gradient method*, Numer. Algo., v 25, (2000), pp 79–88

D. Calvetti, Sun–Mi Kim and L. Reichel, *Quadrature rules based on the Arnoldi process*, SIAM J. Matrix Anal. Appl., v 26 n 3, (2005), pp 765–781

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

S. Elhay and J. Kautsky, *Jacobi matrices for measures modified by a rational factor*, Numer. Algo., v 6, (1994), pp 205–227

K.V. Fernando and B.N. Parlett, *Accurate singular values and differential qd algorithms*, Num. Math., v 67, (1994), pp 191–229

B. Fischer and G.H. Golub, *On generating polynomials which are orthogonal over several intervals*, Math. Comp., v 56 n 194, (1991), pp 711–730

B. Fischer and G.H. Golub, *On the error computation for polynomial based iteration methods*, in Recent advances in iterative methods, A. Greenbaum and M. Luskin Eds., Springer, (1993)

W. Gautschi, *Orthogonal polynomials: computation and approximation*, Oxford University Press, (2004)

G. H. Golub and B. Fischer, *How to generate unknown orthogonal polynomials out of known orthogonal polynomials*, J. Comp. Appl. Math., v 43, (1992), pp 99–115

G.H. Golub and G. Meurant, *Matrices, moments and quadrature*, in Numerical Analysis 1993, D.F. Griffiths and G.A. Watson eds., Pitman Research Notes in Mathematics, v 303, (1994), pp 105–156

G.H. Golub and G. Meurant, *Matrices, moments and quadrature II or how to compute the norm of the error in iterative methods*, BIT, v 37 n 3, (1997), pp 687–705

G.H. Golub, M. Stoll and A. Wathen, *Approximation of the scattering amplitude*, Elec. Trans. Numer. Anal., v 31, (2008), pp 178–203.

G.H. Golub and Z. Strakŏs, *Estimates in quadratic formulas*, Numer. Algo., v 8, n II–IV, (1994) G.H. Golub and J.H. Welsch, *Calculation of Gauss quadrature rules*, Math. Comp., v 23, (1969), pp 221–230

W.B. Gragg and W.J. Harrod, *The numerically stable reconstruction of Jacobi matrices from spectral data*, Numer. Math., v 44, (1984), pp 317–335

M.R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Nat. Bur. Stand., v 49 n 6, (1952), pp 409–436

P. Jiranek, Z. Strakoš and M. Vohralik, *A posteriori error estimates including algebraic error and stopping criteria for iterative solvers*, SIAM J. Sci. Comput., v 32, (2010), pp 1567–1590

J. Kautsky and G.H. Golub, *On the calculation of Jacobi matrices*, Linear Alg. Appl., v 52/53, (1983), pp 439–455

D.P. Laurie, *Accurate recovery of recursion coefficients from Gaussian quadrature formulas*, J. Comp. Appl. Math., v 112, (1999), pp 165–180

G. Meurant, *The computation of bounds for the norm of the error in the conjugate gradient algorithm*, Numer. Algo., v 16, (1997), pp 77–87

G. Meurant, *Numerical experiments in computing bounds for the norm of the error in the preconditioned conjugate gradient algorithm*, Numer. Algo., v 22, (1999), pp 353–365

G. Meurant, *Estimates of the $l_2$ norm of the error in the conjugate gradient algorithm*, Numer. Algo., v 40 n 2, (2005), pp 157–169

G. Meurant, *The Lanczos and Conjugate Gradient algorithms, from theory to finite precision computations*, SIAM, (2006)

L. Reichel, *Construction of polynomials that are orthogonal with respect to a discrete bilinear form*, Adv. Comput. Math., v1, (1993), pp 241–258

H. Rutishauser, *Der Quotienten-Differenzen-Algorithmus*, Zeitschrift für Angewandte Mathematik und Physik (ZAMP), v 5 n 3, (1954), pp 233-251

R.A. Sack and A. Donovan, *An algorithm for Gaussian quadrature given modified moments*, Numer. Math., v 18 n 5, (1972), pp 465–478

P.E. Saylor and D.C. Smolarski, *Why Gaussian quadrature in the complex plane?*, Numer. Algo., v 26, (2001), pp 251–280

P.E. Saylor and D.C. Smolarski, *Addendum to: Why Gaussian quadrature in the complex plane?*, Numer. Algo., v 27, (2001), pp 215–217

Z. Strakoš, *Model reduction using the Vorobyev moment problem*, Numer. Algo., v 51, (2009), pp 363–379

Z. Strakoš and P. Tichý, *On error estimates in the conjugate gradient method and why it works in finite precision* *computations*, Elec. Trans. Numer. Anal., v 13, (2002), pp 56–80

Z. Strakoš and P. Tichý, *Error estimation in preconditioned conjugate gradients*, BIT Numerical Mathematics, v 45, (2005), pp 789–817

Z. Strakoš and P. Tichý, *On efficient numerical approximation of the bilinear form* $c^*A^{-1}b$, submitted to SIAM J. Sci. Comput., (2008)

G. Szegö, *Orthogonal polynomials*, Third Edition, American Mathematical Society, (1974)

J.C. Wheeler, *Modified moments and Gaussian quadrature*, in Proceedings of the international conference on Padé approximants, continued fractions and related topics, Univ. Colorado, Boulder, Rocky Mtn. J. Math., v 4 n 2, (1974), pp 287–296

# Matrices, moments and quadrature with applications
# (IV)

Gérard MEURANT

November 2010

# Previous episodes

We have seen how to compute bounds or estimates of

$$u^T f(A) u \quad \text{or } u^T f(A) v$$

when $A$ is symmetric positive definite using the Lanczos algorithm

# Introduction to ill–posed problems

We speak of a discrete ill–posed problem (DIP) when the solution is sensitive to perturbations of the data

Example:

$$A = \begin{pmatrix} 0.15 & 0.1 \\ 0.16 & 0.1 \\ 2.02 & 1.3 \end{pmatrix}, \quad c + \Delta c = A \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0.01 \\ -0.032 \\ 0.01 \end{pmatrix}$$

The solution of the perturbed least squares problem (rounded to 4 decimals) using the QR factorization of $A$ is

$$x_{QR} = \begin{pmatrix} -2.9977 \\ 7.2179 \end{pmatrix}$$

Why is it so?

The SVD of $A$ is

$$U = \begin{pmatrix} -0.0746 & 0.7588 & -0.6470 \\ -0.0781 & -0.6513 & -0.7548 \\ -0.9942 & -0.0058 & 0.1078 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} 2.4163 & 0 \\ 0 & 0.0038 \\ 0 & 0 \end{pmatrix}$$

$$V = \begin{pmatrix} -0.8409 & -0.5412 \\ -0.5412 & 0.8409 \end{pmatrix}$$

The component $(u^2)^T \Delta c/\sigma_2$ ($u^2$ being the second column of $U$) corresponding to the smallest nonzero singular value is large being 6.2161

This gives the large change in the solution

$$Ax \approx c = \bar{c} - e$$

where $A$ is a matrix of dimension $m \times n$, $m \geq n$ and the right hand side $\bar{c}$ is contaminated by a (generally) unknown noise vector $e$

- The standard solution of the least squares problem $\min \|c - Ax\|$ (even using backward stable methods like QR) may give a vector $x$ severely contaminated by noise
- This may seems hopeless
- The solution is to modify the problem by regularization
- We have to find a balance between obtaining a problem that we can solve reliably and obtaining a solution which is not too far from the solution without noise

# Examples of ill-posed problems

These examples were obtained with the Regutools Matlab toolbox from Per-Christian Hansen

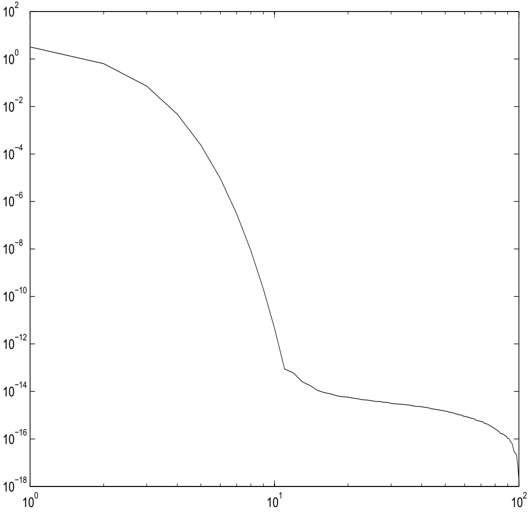
The Baart problem arises from the discretization of a first-kind Fredholm integral equation

$$\int_0^1 K(s,t)f(t)\,dt = g(s) + e(s)$$

with kernel $K$ and right-hand side $g$ given by

$$K(s,t) = \exp(s\cos(t)), \quad g(s) = 2\sinh(s)/s$$

and with integration intervals $s \in [0, \pi/2], t \in [0, \pi]$
The solution is given by $f(t) = \sin(t)$
The square dense matrix $A$ of order 100 is dense and its smallest and largest singular values are $1.7170\ 10^{-18}$ and 3.2286




Singular values for the Baart problem, $m = n = 100$

The Phillips problem arises from the discretization of a first-kind Fredholm integral equation devised by D. L. Phillips
Let

$$\phi(x) = 1 + \cos(x\pi/3) \text{ for } |x| < 3, \; 0 \text{ for } |x| >= 3$$

The kernel $K$, the solution $f$ and the right-hand side $g$ are given by

$$K(s,t) = \phi(s-t), \quad f(t) = \phi(t)$$

$$g(s) = (6 - |s|)(1 + 0.5\cos(s\pi/3)) + 9/(2\pi)\sin(|s|\pi/3)$$

The integration interval is $[-6, 6]$
The square matrix $A$ of order 200 is banded and its smallest and largest singular values are 1.3725 $10^{-7}$ and 5.8029.

# Tikhonov regularization

Replace the LS problem by

$$\min_x\{\|c - Ax\|^2 + \mu\|x\|^2\}$$

where $\mu \geq 0$ is a regularization parameter to be chosen
For some problems (particularly in image restoration) it is better to consider

$$\min_x\{\|c - Ax\|^2 + \mu\|Lx\|^2\}$$

where $L$ is typically the discretization of a derivative operator of first or second order
The solution $x_\mu$ of the problem solves the linear system

$$(A^T A + \mu I)x = A^T c$$

# The Golub–Kahan bidiagonalization algorithm

It is a special case of the Lanczos for $A^T A$

The first algorithm (LB1) reduces $A$ to upper bidiagonal form

Let $q^0 = c/\|c\|$, $r^0 = Aq^0$, $\delta_1 = \|r^0\|$, $p^0 = r^0/\delta_1$
then for $k = 1, 2, \ldots$

$$u^k = A^T p^{k-1} - \delta_k q^{k-1}$$

$$\gamma_k = \|u^k\|$$

$$q^k = u^k/\gamma_k$$

$$r^k = Aq^k - \gamma_k p^{k-1}$$

$$\delta_{k+1} = \|r^k\|$$

$$p^k = r^k/\delta_{k+1}$$

If

$$P_k = \begin{pmatrix} p^0 & \cdots & p^{k-1} \end{pmatrix}, \quad Q_k = \begin{pmatrix} q^0 & \cdots & q^{k-1} \end{pmatrix}$$

and

$$B_k = \begin{pmatrix} \delta_1 & \gamma_1 & & \\ & \ddots & \ddots & \\ & & \delta_{k-1} & \gamma_{k-1} \\ & & & \delta_k \end{pmatrix}$$

then $P_k$ and $Q_k$, which is an orthogonal matrix, satisfy the equations

$$\begin{aligned} AQ_k &= P_k B_k \\ A^T P_k &= Q_k B_k^T + \gamma_k q^k (e^k)^T \end{aligned}$$

and therefore

$$A^T A Q_k = Q_k B_k^T B_k + \gamma_k \delta_k q^k (e^k)^T$$

The second algorithm (LB2) reduces $A$ to lower bidiagonal form

Let $p^0 = c/\|c\|$, $u^0 = A^T p^0$, $\gamma_1 = \|u^0\|$, $q^0 = u^0/\gamma_1$,
$r^1 = Aq^0 - \gamma_1 p^0$, $\delta_1 = \|r^1\|$, $p^1 = r^1/\delta_1$
then for $k = 2, 3, \ldots$

$$u^{k-1} = A^T p^{k-1} - \delta_{k-1} q^{k-2}$$

$$\gamma_k = \|u^{k-1}\|$$

$$q^{k-1} = u^{k-1}/\gamma_k$$

$$r^k = Aq^{k-1} - \gamma_k p^{k-1}$$

$$\delta_k = \|r^k\|$$

$$p^k = r^k/\delta_k$$

If

$$P_{k+1} = \begin{pmatrix} p^0 & \cdots & p^k \end{pmatrix}, \quad Q_k = \begin{pmatrix} q^0 & \cdots & q^{k-1} \end{pmatrix}$$

and

$$C_k = \begin{pmatrix} \gamma_1 & & & \\ \delta_1 & \ddots & & \\ & \ddots & \ddots & \\ & & \ddots & \gamma_k \\ & & & \delta_k \end{pmatrix}$$

a $k+1$ by $k$ matrix, then $P_k$ and $Q_k$, which is an orthogonal matrix, satisfy the equations

$$\begin{aligned} AQ_k &= P_{k+1}C_k \\ A^T P_{k+1} &= Q_k C_k^T + \gamma_{k+1} q^k (e^{k+1})^T \end{aligned}$$

Of course, by eliminating $P_{k+1}$ in these equations we obtain

$$A^T A Q_k = Q_k C_k^T C_k + \gamma_{k+1} \delta_k q^k (e^k)^T$$

and

$$C_k^T C_k = B_k^T B_k = J_k$$

$B_k$ is the Cholesky factor of $J_k$ and $C_k^T C_k$

$J_k$ is the tridiagonal Jacobi matrix for $A^T A$

The main problem in Tikhonov regularization is to choose $\mu$

- If $\mu$ is too small the solution is contaminated by the noise in the right hand side
- if $\mu$ is too large the solution is a poor approximation of the original problem
- Many methods have been devised for choosing $\mu$
- Most of these methods lead to the evaluation of bilinear forms with different matrices

# Some methods for choosing $\mu$

- Morozov's discrepancy principle
  Ask for the norm of the residual to be equal to the norm of the noise vector (if it is known)

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

- The Gfrerer/Raus method

$$\mu^3 c^T (AA^T + \mu I)^{-3} c = \|e\|^2$$

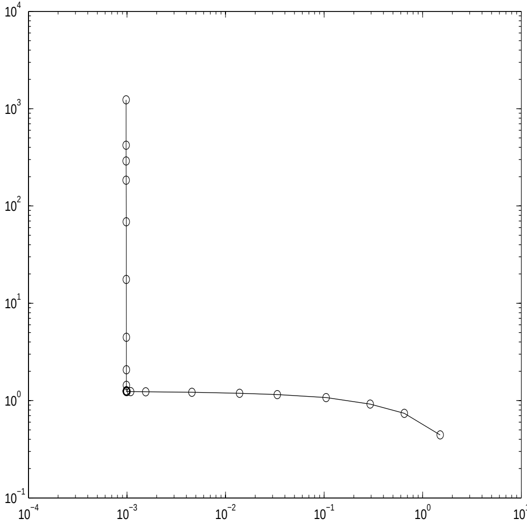
- The quasi–optimality criterion

$$\min[\mu^2 c^T A(A^T A + \mu I)^{-4} A^T c]$$

# The L–curve criterion

- plot in log–log scale the curve $(\|x_\mu\|, \|b - Ax_\mu\|)$ obtained by varying the value of $\mu \in [0, \infty)$ where $x_\mu$ is the regularized solution
- In most cases this curve is shaped as an "L"
- Lawson and Hanson proposed to choose the value $\mu_L$ corresponding to the "corner" of the L–curve (the point of maximal curvature (see also Hansen; Hansen and O'Leary)
- This is done to have a balance between $\mu$ being too small and the solution contaminated by the noise, and $\mu$ being too large giving a poor approximation of the solution. The "vertex" of the L–curve gives an average value between these two extremes

# An example of L–curve




The L-curve for the `Baart` problem, $m = n = 100$, $noise = 10^{-3}$

# How to locate the corner of the L–curve?

see Hansen and al.

- Easy if we know the SVD of $A$
- Otherwise compute points on the L–curve and use interpolation
- However, computing a point on the L–curve is expensive
- Alternative, L–ribbon approximation (Calvetti, Golub and Reichel)

# The L–ribbon

$$\|x_\mu\|^2 = c^T A(A^T A + \mu I)^{-2} A^T c$$

and

$$\begin{aligned}\|c - Ax_\mu\|^2 &= c^T c + c^T A(A^T A + \mu I)^{-1} A^T A(A^T A + \mu I)^{-1} A^T c \\ &\quad -2c^T A(A^T A + \mu I)^{-1} A^T c\end{aligned}$$

By denoting $K = A^T A$ and $d = A^T c$

$$\|c - Ax_\mu\|^2 = c^T c + d^T K(K + \mu I)^{-2} d - 2d^T (K + \mu I)^{-1} d$$

Define

$$\begin{aligned}\phi_1(t) &= (t+\mu)^{-2} \\ \phi_2(t) &= t(t+\mu)^{-2} - 2(t+\mu)^{-1}\end{aligned}$$

we are interested in $s_i = d^T \phi_i(K) d$, $i = 1, 2$

We can obtain bounds using the Golub–Kahan bidiagonalization algorithm

At iteration $k$, the algorithm computes a Jacobi matrix $J_k = B_k^T B_k$ and the Gauss rule gives

$$I_k^G(\phi_i) = \|d\|^2 (e^1)^T \phi_i(J_k) e^1$$

We can also use the Gauss–Radau rule with a prescribed node $a = 0$

$$I_k^{GR}(\phi_i) = \|d\|^2 (e^1)^T \phi_i(\hat{J}_k) e^1$$

$\hat{J}_k = \hat{B}_k^T \hat{B}_k$ where $\hat{B}_k$ is obtained from $B_k$ by setting the last diagonal element $\delta_k = 0$

## Theorem

$$I_k^G(\phi_1) \leq s_1 \leq I_k^{GR}(\phi_1)$$

*where*

$$\begin{aligned} I_k^G(\phi_1) &= \|d\|^2 (e^1)^T (B_k^T B_k + \mu I)^{-2} e^1 \\ I_k^{GR}(\phi_1) &= \|d\|^2 (e^1)^T (\hat{B}_k^T \hat{B}_k + \mu I)^{-2} e^1 \end{aligned}$$

$$I_k^{GR}(\phi_2) \leq s_2 \leq I_k^G(\phi_2)$$

*where*

$$I_k^G(\phi_2) = \|d\|^2 [(e^1)^T B_k^T B_k (B_k^T B_k + \mu I)^{-2} e^1 - 2(e^1)^T (B_k^T B_k + \mu I)^{-1} e^1]$$

$$I_k^{GR}(\phi_2) = \|d\|^2 [(e^1)^T \hat{B}_k^T \hat{B}_k (\hat{B}_k^T \hat{B}_k + \mu I)^{-2} e^1 - 2(e^1)^T (\hat{B}_k^T \hat{B}_k + \mu I)^{-1} e^1]$$

$$x^-(\mu) = \sqrt{I_k^G(\phi_1)}, \quad x^+(\mu) = \sqrt{I_k^{GR}(\phi_1)}$$

$$y^-(\mu) = \sqrt{c^T c + I_k^{GR}(\phi_2)}, \quad y^+(\mu) = \sqrt{c^T c + I_k^G(\phi_2)}$$

For a given value of $\mu > 0$ the bounds are

$$x^-(\mu) \le \|x_\mu\| \le x^+(\mu), \quad y^-(\mu) \le \|c - Ax_\mu\| \le y^+(\mu)$$

Calvetti, Golub and Reichel defined the L–ribbon as the union of rectangles for all $\mu > 0$

$$\bigcup_{\mu>0} \{ \{x(\mu), y(\mu)\} : x^-(\mu) \le x(\mu) \le x^+(\mu), \, y^-(\mu) \le y(\mu) \le y^+(\mu)\}$$

Then, we have to select a point (a value of $\mu$) inside the L–ribbon

Note that the Golub–Kahan iterations are independent of $\mu$

# The L–curvature

Another possibility is to obtain bounds of the curvature (in log–log scale) and to look for the maximum

$$C_\mu = 2\frac{\rho''\eta' - \rho'\eta''}{((\rho')^2 + (\eta')^2)^{3/2}}$$

where $'$ denotes differentiation with respect to $\mu$ and

$$\begin{aligned}\rho(\mu) &= \frac{1}{2}\log\|c - Ax_\mu\| = \log \mu^2 c^T \phi(AA^T)c \\ \eta(\mu) &= \frac{1}{2}\log\|x_\mu\| = \log c^T A\phi(A^T A)A^T c\end{aligned}$$

where $\phi(t) = (t+\mu)^{-2}$

The first derivatives can be computed as

$$\begin{aligned}\rho'(\mu) &= \frac{c^T A(A^T A + \mu I)^{-3} A^T c}{\mu c^T (AA^T + \mu I)^{-2} c} \\ \eta'(\mu) &= -\frac{c^T A(A^T A + \mu I)^{-3} A^T c}{c^T A(A^T A + \mu I)^{-2} A^T c}\end{aligned}$$

The numerator is more complicated

$$\begin{aligned}\rho'\eta'' - \rho''\eta' = \quad & \left(\frac{c^T A(A^T A + \mu I)^{-3} A^T c}{\mu c^T (AA^T + \mu I)^{-2} c \cdot c^T A(A^T A + \mu I)^{-2} A^T c}\right)^2 \\ & (c^T (AA^T + \mu I)^{-2} c \cdot c^T A(A^T A + \mu I)^{-2} A^T c \\ & + 2\mu c^T (AA^T + \mu I)^{-3} c \cdot c^T A(A^T A + \mu I)^{-2} A^T c \\ & - 2\mu c^T (AA^T + \mu I)^{-2} c \cdot c^T A(A^T A + \mu I)^{-3} A^T c)\end{aligned}$$

# Locating the corner of the L–curve

There are many possibilities

- Using the SVD (Hansen): `lc`
- Pruning algorithm (Hansen, Jensen and Rodriguez): `lp`
- Rotating the L–curve (GM): `lc1`
- Finding an interval where $\log \|x_\mu\|$ and $\log \|c - Ax_\mu\|$ are almost constant (GM): `lc2`

## L-curve algorithms, Baart problem, $n = 100$

| noise | meth | $\mu$ | $\|c - Ax\|$ | $\|x - x_0\|$ |
|---|---|---|---|---|
| $10^{-3}$ | opt | $2.4990\ 10^{-8}$ | $9.8720\ 10^{-4}$ | $1.5080\ 10^{-1}$ |
| | lc | $4.5414\ 10^{-9}$ | $9.8524\ 10^{-4}$ | $1.6030\ 10^{-1}$ |
| | lp | $8.2364\ 10^{-9}$ | $9.8545\ 10^{-4}$ | $1.5454\ 10^{-1}$ |
| | lc1 | $6.3232\ 10^{-9}$ | $9.8534\ 10^{-4}$ | $1.5669\ 10^{-1}$ |
| | lc2 | $5.8203\ 10^{-12}$ | $9.8463\ 10^{-4}$ | $4.1492\ 10^{-1}$ |
| | | $4.1297\ 10^{-8}$ | $9.8996\ 10^{-4}$ | $1.5153\ 10^{-1}$ |

opt is the point with (almost) smallest error

## L-curve algorithms, `Phillips` problem, $n = 200$

| noise | meth | $\mu$ | $\|c - Ax\|$ | $\|x - x_0\|$ |
|---|---|---|---|---|
| $10^{-3}$ | `opt` | $8.5392\ 10^{-7}$ | $9.9864\ 10^{-4}$ | $7.3711\ 10^{-3}$ |
| | `lc` | $7.1966\ 10^{-10}$ | $8.5111\ 10^{-4}$ | $5.3762\ 10^{-1}$ |
| | `lp` | $4.5729\ 10^{-10}$ | $8.3869\ 10^{-4}$ | $6.8849\ 10^{-1}$ |
| | `lc1` | $3.6084\ 10^{-10}$ | $8.3172\ 10^{-4}$ | $7.8603\ 10^{-1}$ |
| | `lc2` | $1.0250\ 10^{-9}$ | $8.6013\ 10^{-4}$ | $4.4563\ 10^{-1}$ |
| | | $2.9147\ 10^{-7}$ | $9.7098\ 10^{-4}$ | $1.3595\ 10^{-2}$ |

## L-ribbon

| Ex | noise | nb it | $\mu$ | nb it no reorth. |
|---|---|---|---|---|
| Baart | $10^{-7}$ | 11 | $6.0889\ 10^{-17}$ | 40 |
| | $10^{-5}$ | 9 | $6.1717\ 10^{-13}$ | 19 |
| | $10^{-3}$ | 8 | $6.3232\ 10^{-9}$ | 10 |
| | $10^{-1}$ | 6 | $7.2928\ 10^{-5}$ | 6 |
| | 10 | 5 | $3.260\ 10^{-2}$ | 5 |

With and without reorthogonalization

# Generalized cross–validation

GCV comes from statistics (Golub, Heath and Wahba)

The regularized problem is written as

$$\min\{\|c - Ax\|^2 + m\mu\|x\|^2\}$$

where $\mu \geq 0$ is the regularization parameter and the matrix $A$ is $m$ by $n$

The GCV estimate of the parameter $\mu$ is the minimizer of

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

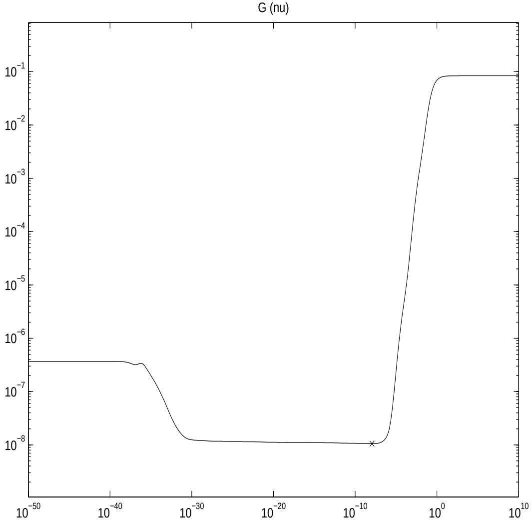
If we know the SVD of $A$ and $m \geq n$ this can be computed as

$$G(\nu) = \frac{m\left\{\sum_{i=1}^{r} d_i^2 \left(\frac{\nu}{\sigma_i^2+\nu}\right)^2 + \sum_{i=r+1}^{m} d_i^2\right\}}{[m - n + \sum_{i=1}^{r} \frac{\nu}{\sigma_i^2+\nu}]^2}$$

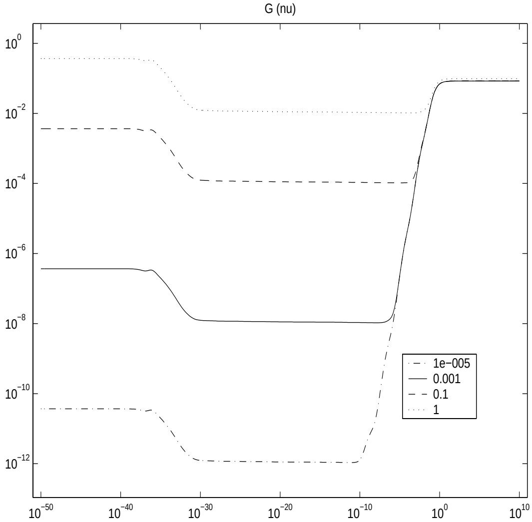
where $\nu = m\mu$

- $G$ is almost constant when $\nu$ is very small or large, at least in log–log scale
- When $\nu \to \infty$, $G(\nu) \to \|c\|^2/m$
- When $\nu \to 0$ the situation is different wether $m = n$ or not

# An example of GCV function




GCV function for the `Baart` problem, $m = n = 100$, $noise = 10^{-3}$




GCV functions for the Baart problem, $m = n = 100$ for different noise levels

The main problem is that the GCV function is usually quite flat near the minimum

For large problems we cannot use the SVD

- First we approximate the trace in the denominator $\rightarrow \tilde{G}$
- Then using the Golub–Kahan bidiagonalization algorithms we can obtain bounds of all the terms in $\tilde{G}$
- Finally we have to locate the minimum of the lower and/or upper bounds

# Approximation of the trace

## Proposition (Hutchinson)

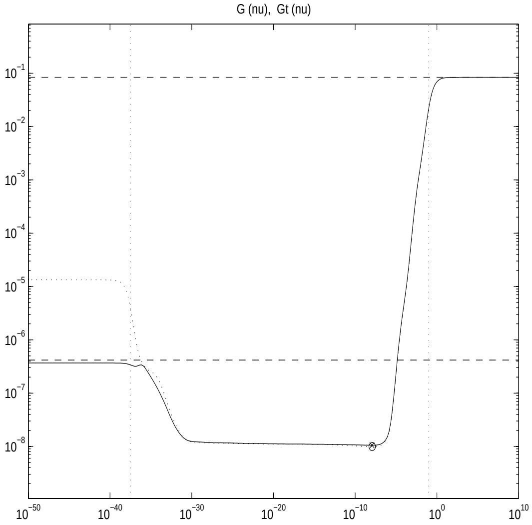
*Let* $B$ *be a symmetric matrix of order* $n$ *with* $\mathrm{tr}(B) \neq 0$
*Let* $\mathcal{Z}$ *be a discrete random variable with values* $1$ *and* $-1$ *with equal probability* $0.5$ *and let* $z$ *be a vector of* $n$ *independent samples from* $\mathcal{Z}$
*Then* $z^T Bz$ *is an unbiased estimator of* $\mathrm{tr}(B)$

$$E(z^T Bz) = \mathrm{tr}(B)$$

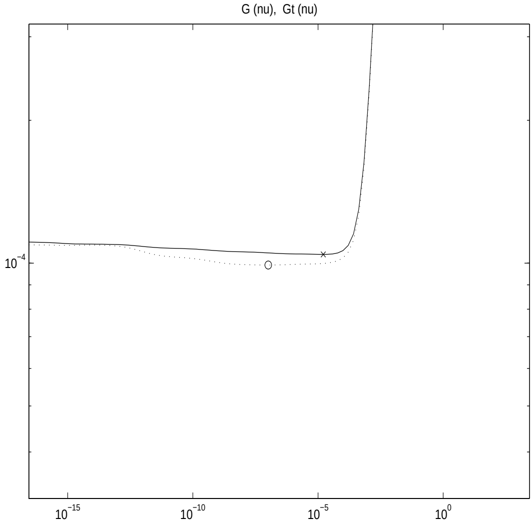
$$\mathrm{var}(z^T Bz) = 2 \sum_{i \neq j} b_{i,j}^2$$

*where* $E(\cdot)$ *denotes the expected value and* $\mathrm{var}$ *denotes the variance*

For GCV we just use one vector $z$




$G$ (plain) and $\tilde{G}$ (dotted) functions for the Baart problem, $m = n = 100$, $noise = 10^{-3}$




$G$ (plain) and $\tilde{G}$ (dotted) functions for the Baart problem,
$m = n = 100, noise = 10^{-1}$

# The Golub and Von Matt algorithm

Let $s_z(\nu) = z^T(A^TA + \nu I)^{-1}z$, where $z$ is a random vector
Using Gauss and Gauss–Radau we can obtain

$$g_z(\nu) \le s_z(\nu) \le r_z(\nu)$$

We can also bound
$s_c^{(p)}(\nu) = c^TA(A^TA + \nu I)^pA^Tc, \quad p = -1, -2$ satisfying

$$g_c^{(p)}(\nu) \le s_c^{(p)}(\nu) \le r_c^{(p)}(\nu)$$

We want to compute approximations of the minimum of

$$\tilde{G}(\mu) = m\frac{c^Tc - s_c^{(-1)}(\nu) - \nu s_c^{(-2)}(\nu)}{(m - n + \nu s_z(\nu))^2}$$

We define

$$L_0(\nu) = m \frac{c^T c - r_c^{(-1)}(\nu) - \nu r_c^{(-2)}(\nu)}{(m - n + \nu r_z(\nu))^2}$$

$$U_0(\nu) = m \frac{c^T c - g_c^{(-1)}(\nu) - \nu g_c^{(-2)}(\nu)}{(m - n + \nu g_z(\nu))^2}$$

These quantities $L_0$ and $U_0$ are lower and upper bounds for the estimate of $G(\mu)$

We can also compute estimates of the derivatives of $L_0$ and $U_0$

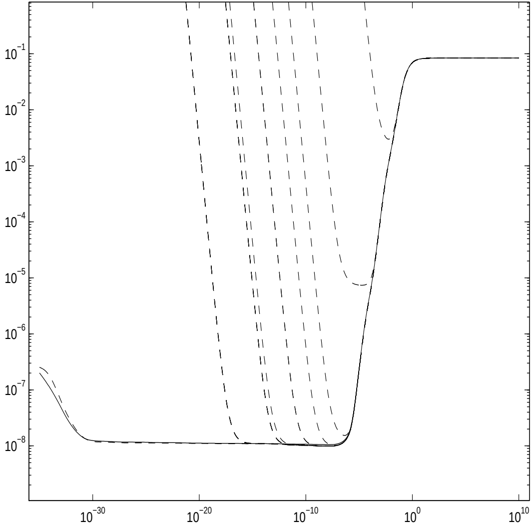
These bounds improve with the number of Lanczos iterations

- They first do $k_{min} = \lceil 3 \log \min(m, n) \rceil$ Lanczos iterations
- Then the global minimizer $\hat{\nu}$ of $U_0(\nu)$ is computed
- If one can find a $\nu$ such that $0 < \nu < \hat{\nu}$ and $L_0(\nu) > L_0(\hat{\nu})$, the algorithm stops and return $\hat{\nu}$
- Otherwise, the algorithm executes one more Lanczos iteration and repeats the convergence test

Von Matt computed the minimum of the upper bound:

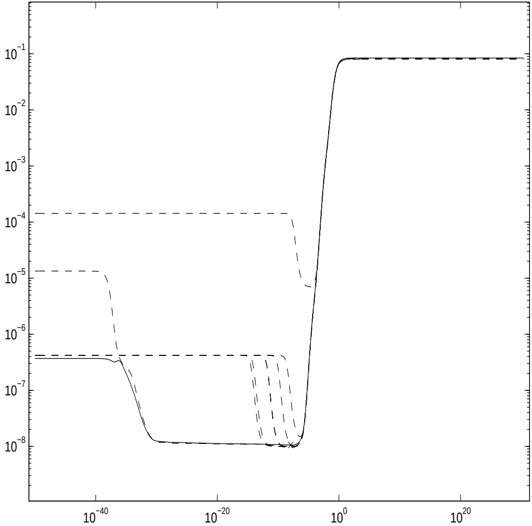
- By sampling the function on 100 points with an exponential distribution
- If the neighbors of the minimum do not have the same values, he looked at the derivative and sought for a local minimum in either the left or right interval depending on the sign of the derivative
- The local minimum is found by using bisection

The upper bound does not have the right asymptotic behavior when $m = n$ and $\nu \to 0$




$G$ (plain) and $\tilde{G}$ (dashed) functions and upper bounds for the Baart problem, $m = n = 100$, $noise = 10^{-3}$

To obtain a better behavior we add a term $\|c\|^2$ to the denominator




$G$ (plain) and $\tilde{G}$ (dashed) functions and upper bounds for the `Baart` problem, $m = n = 100$, $noise = 10^{-3}$

# Optimization of the algorithm

- We choose a (small) value of $\nu$ (denoted as $\nu_0$)
- When

$$\left| \frac{U_k^0(\nu_0) - U_{k-1}^0(\nu_0)}{U_{k-1}^0(\nu_0)} \right| \leq \epsilon_0$$

  we start computing the minimum of the upper bound

The algorithm for finding the minimum is modified as follows

- We work in log–log scale and compute only a minimizer of the upper bound
- We evaluate the numerator of the approximation by computing the SVD of $B_k$ once per iteration
- We compute 50 samples of the function on a regular mesh
- We locate the minimum, say the point $k$, we then compute again 50 samples in the interval $[k-1\ \ k+1]$

- We use the Von Matt algorithm for computing a local minimum in this interval
- After locating a minimum $\nu_k$ with a value of the upper bound $U_k^0$ at iteration $k$, the stopping criteria is

$$\left| \frac{\nu_k - \nu_{k-1}}{\nu_{k-1}} \right| + \left| \frac{U_k^0 - U_{k-1}^0}{U_{k-1}^0} \right| \leq \epsilon$$

## GCV algorithms, Baart problem

| | noise | $\mu$ | $\|c - Ax\|$ | $\|x - x_0\|$ | t (s) |
|---|---|---|---|---|---|
| vm | $10^{-7}$ | $9.6482\ 10^{-15}$ | $9.8049\ 10^{-8}$ | $5.9424\ 10^{-2}$ | 0.38 |
| | $10^{-5}$ | $9.7587\ 10^{-12}$ | $9.8566\ 10^{-6}$ | $6.5951\ 10^{-2}$ | 0.18 |
| | $10^{-3}$ | $1.2018\ 10^{-8}$ | $9.8573\ 10^{-4}$ | $1.5239\ 10^{-1}$ | 0.16 |
| | $10^{-1}$ | $1.0336\ 10^{-7}$ | $9.8730\ 10^{-2}$ | 1.6614 | – |
| gm-opt | $10^{-7}$ | $1.0706\ 10^{-14}$ | $9.8058\ 10^{-8}$ | $5.9519\ 10^{-2}$ | 0.18 |
| | $10^{-5}$ | $1.0581\ 10^{-11}$ | $9.8588\ 10^{-6}$ | $6.5957\ 10^{-2}$ | 0.27 |
| | $10^{-3}$ | $1.3077\ 10^{-8}$ | $9.8582\ 10^{-4}$ | $1.5205\ 10^{-1}$ | 0.14 |
| | $10^{-1}$ | $1.1104\ 10^{-7}$ | $9.8736\ 10^{-2}$ | 1.6227 | – |

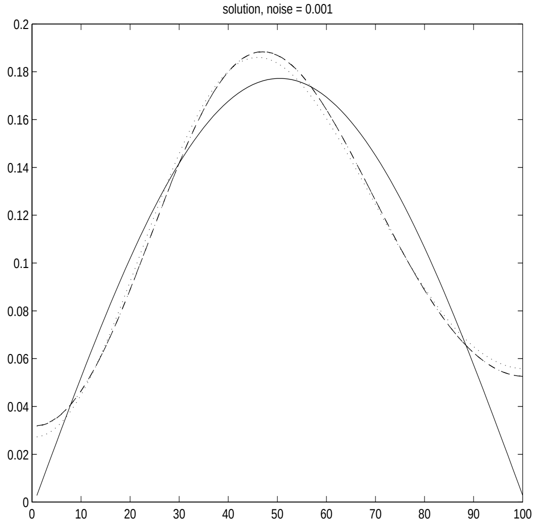
## GCV algorithms, `Phillips` problem

| | noise | $\mu$ | $\|c - Ax\|$ | $\|x - x_0\|$ | t (s) |
|---|---|---|---|---|---|
| vm | $10^{-7}$ | $8.7929\ 10^{-11}$ | $9.0162\ 10^{-8}$ | $2.2391\ 10^{-4}$ | 29.50 |
| | $10^{-5}$ | $4.5432\ 10^{-9}$ | $9.0825\ 10^{-6}$ | $2.2620\ 10^{-3}$ | 6.09 |
| | $10^{-3}$ | $4.3674\ 10^{-7}$ | $9.7826\ 10^{-4}$ | $1.0057\ 10^{-2}$ | 1.14 |
| | $10^{-1}$ | $3.8320\ 10^{-5}$ | $9.8962\ 10^{-2}$ | $9.3139\ 10^{-2}$ | 0.16 |
| gm-opt | $10^{-7}$ | $1.6343\ 10^{-10}$ | $1.1260\ 10^{-7}$ | $2.2163\ 10^{-4}$ | 15.30 |
| | $10^{-5}$ | $5.3835\ 10^{-9}$ | $9.1722\ 10^{-6}$ | $2.1174\ 10^{-3}$ | 6.09 |
| | $10^{-3}$ | $4.1814\ 10^{-7}$ | $9.7737\ 10^{-4}$ | $1.0375\ 10^{-2}$ | 0.66 |
| | $10^{-1}$ | $4.1875\ 10^{-5}$ | $9.9016\ 10^{-2}$ | $9.0659\ 10^{-2}$ | 0.22 |

# Comparisons of methods

Baart problem, $n = 100$

| noise | meth | $\mu$ | $\|c - Ax\|$ | $\|x - x_0\|$ |
|---|---|---|---|---|
| $10^{-3}$ | $\mu$ opt | $2.7826\ 10^{-8}$ | $2.3501\ 10^{-3}$ | $1.5084\ 10^{-1}$ |
| | vm | $1.2018\ 10^{-8}$ | $9.8573\ 10^{-4}$ | $1.5239\ 10^{-1}$ |
| | gm-opt | $1.3077\ 10^{-8}$ | $9.8582\ 10^{-4}$ | $1.5205\ 10^{-1}$ |
| | gcv | $9.4870\ 10^{-9}$ | $9.8554\ 10^{-4}$ | $1.5362\ 10^{-1}$ |
| | disc | $8.4260\ 10^{-8}$ | $1.0000\ 10^{-3}$ | $1.5556\ 10^{-1}$ |
| | gr | $1.7047\ 10^{-7}$ | $1.0235\ 10^{-3}$ | $1.6373\ 10^{-1}$ |
| | lc | $4.5414\ 10^{-9}$ | $9.8524\ 10^{-4}$ | $1.6028\ 10^{-1}$ |
| | qo | $1.2586\ 10^{-8}$ | $9.8450\ 10^{-4}$ | $6.6072\ 10^{-1}$ |
| | L-rib | $6.3232\ 10^{-9}$ | $9.8534\ 10^{-4}$ | $1.5669\ 10^{-1}$ |
| | L-cur | $5.8220\ 10^{-9}$ | $9.8531\ 10^{-4}$ | $1.5749\ 10^{-1}$ |




Solutions for the Baart problem, $m = n = 100$, $noise = 10^{-3}$,
solid=unperturbed solution, dashed=vm, dot–dashed=gm-opt

## Phillips problem, $n = 200$

| noise | meth | $\mu$ | $\|c - Ax\|$ | $\|x - x_0\|$ |
|---|---|---|---|---|
| $10^{-5}$ | $\mu$ opt | $1.3725\ 10^{-7}$ | $2.9505\ 10^{-14}$ | $1.6641\ 10^{-3}$ |
| | vm | $4.5432\ 10^{-9}$ | $9.0825\ 10^{-6}$ | $2.2620\ 10^{-3}$ |
| | gm-opt | $5.3835\ 10^{-9}$ | $9.1722\ 10^{-6}$ | $2.1174\ 10^{-3}$ |
| | gcv | $3.1203\ 10^{-9}$ | $8.9283\ 10^{-6}$ | $2.6499\ 10^{-3}$ |
| | disc | $1.2107\ 10^{-8}$ | $1.0000\ 10^{-5}$ | $1.6873\ 10^{-3}$ |
| | gr | $4.1876\ 10^{-8}$ | $1.5784\ 10^{-5}$ | $1.9344\ 10^{-3}$ |
| | lc | $3.6731\ 10^{-14}$ | $2.4301\ 10^{-6}$ | $7.9811\ 10^{-1}$ |
| | qo | $1.5710\ 10^{-8}$ | $1.0542\ 10^{-5}$ | $1.6463\ 10^{-3}$ |
| | L-rib | $2.6269\ 10^{-14}$ | $2.2118\ 10^{-6}$ | $8.9457\ 10^{-1}$ |
| | L-cur | $4.7952\ 10^{-14}$ | $2.6093\ 10^{-6}$ | $7.2750\ 10^{-1}$ |

D. Calvetti, G.H. Golub and L. Reichel, *Estimation of the L–curve via Lanczos bidiagonalization*, BIT, v 39 n 4, (1999), pp 603–619

D. Calvetti, P.C. Hansen and L. Reichel, *L–curve curvature bounds via Lanczos bidiagonalization*, Elec. Trans. Numer. Anal., v 14, (2002), pp 20–35

H. Gfrerer, *An a posteriori parameter choice for ordinary and iterated Tikhonov regularization of ill–posed problems leading to optimal convergence rates*, Math. Comp., v 49, (1987), pp 507–522

G.H. Golub, M. Heath and G. Wahba, *Generalized cross–validation as a method to choosing a good ridge parameter*, Technometrics, v 21 n 2, (1979), pp 215–223

G. H. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, SIAM J. Numer. Anal., v 2 (1965), pp 205–224

- G.H. Golub and U. von Matt, *Tikhonov regularization for large scale problems*, in Scientific Computing, G.H. Golub, S.H. Lui, F. Luk and R. Plemmons Eds., Springer, (1997), pp 3–26
- G.H. Golub and U. von Matt, *Generalized cross–validation for large scale problems*, in Recent advances in total least squares techniques and errors in variable modeling, S. van Huffel ed., SIAM, (1997), pp 139–148
- M. Hanke and T. Raus, *A general heuristic for choosing the regularization parameter in ill–posed problems*, SIAM J. Sci. Comput., v 17, (1996), pp 956–972
- P.C. Hansen, *Regularization tools: a Matlab package for analysis and solution of discrete ill–posed problems*, Numer. Algo., v 6, (1994), pp 1–35
- P.C. Hansen and D.P. O'Leary, *The use of the L–curve in the regularization of discrete ill–posed problems*, SIAM J. Sci. Comput., v 14, (1993), pp 1487–1503

P.C. Hansen, T.K. Jensen and G. Rodriguez, *An adaptive pruning algorithm for the discrete L-curve criterion*, J. Comp. Appl. Math., v 198 n 2, (2007), pp 483–492

C.L. Lawson and R.J. Hanson, *Solving least squares problems*, SIAM, (1995)

A.S. Leonov, *On the choice of regularization parameters by means of the quasi–optimality and ratio criteria*, Soviet Math. Dokl., v 19, (1978), pp 537–540

V.A. Morozov, *Methods for solving incorrectly posed problems*, Springer, (1984)

A.N. Tikhonov, *Ill–posed problems in natural sciences*, Proceedings of the international conference Moscow August 1991, (1992), TVP Science publishers.

A.N. Tikhonov and V.Y. Arsenin, *Solutions of ill–posed problems*, (1977), Wyley