On computing quadrature-based bounds for the $\mathbf{A}$-norm of the error in conjugate gradients

Petr Tichý

joint work with

Gerard Meurant and Zdeněk Strakoš

Institute of Computer Science AS CR

January 21–25, 2013, SNA 2013,
Rožnov pod Radhoštěm, Czech Republic

Problem formulation

Consider a system

$$\mathbf{A}x = b$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is **symmetric, positive definite**.

Without loss of generality, $\|b\| = 1$, $x_0 = 0$.

The conjugate gradient method

input $\mathbf{A}$, b

$r_0 = b$, $p_0 = r_0$

for $k = 1, 2, \ldots$ **do**

$$\begin{aligned}
\gamma_{k-1} &= \frac{r_{k-1}^T r_{k-1}}{p_{k-1}^T \mathbf{A}\, p_{k-1}} \\
x_k &= x_{k-1} + \gamma_{k-1} p_{k-1} \\
r_k &= r_{k-1} - \gamma_{k-1} \mathbf{A}\, p_{k-1} \\
\delta_k &= \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}} \\
p_k &= r_k + \delta_k p_{k-1}
\end{aligned}$$

test quality of x_k

end for

The Lanczos algorithm

Let $\mathbf{A}$ be symmetric, compute orthonormal basis of $\mathcal{K}_k(\mathbf{A}, b)$

input $\mathbf{A}$, b
$v_1 = b/\|b\|$, $\delta_1 = 0$
$\beta_0 = 0$, $v_0 = 0$
for $k = 1, 2, \ldots$ **do**
    $\alpha_k = v_k^T \mathbf{A} v_k$
    $w = \mathbf{A} v_k - \alpha_k v_k - \beta_{k-1} v_{k-1}$
    $\beta_k = \|w\|$
    $v_{k+1} = w/\beta_k$
end for

$$\mathbf{T}_k = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \ddots & & \\ & & \ddots & \beta_{k-1} \\ & & \beta_{k-1} & \alpha_k \end{bmatrix}$$

The Lanczos algorithm can be represented by

$$\mathbf{A}\mathbf{V}_k = \mathbf{V}_k \mathbf{T}_k + \beta_k v_{k+1} e_k^T, \qquad \mathbf{V}_k^* \mathbf{V}_k = \mathbf{I}\,.$$

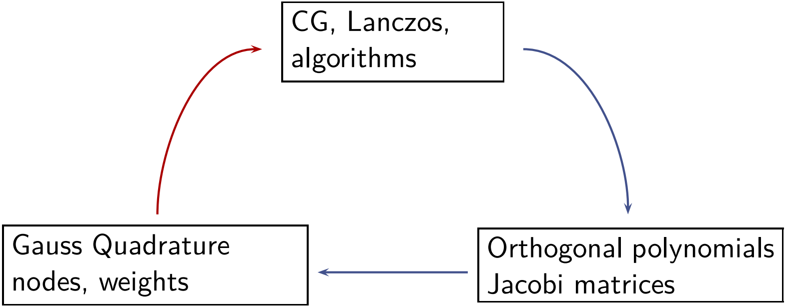
CG versus Lanczos

- Both algorithms generate an orthogonal basis of the Krylov subspace $\mathcal{K}_k(\mathbf{A}, b)$.
- Lanczos generates an orthonormal basis $v_1, \ldots, v_k$ using a three-term recurrence $\rightarrow \mathbf{T}_k$.
- CG generates an orthogonal basis $r_0, \ldots, r_{k-1}$ using a coupled two-term recurrence $\rightarrow \mathbf{T}_k = \mathbf{L}_k \mathbf{D}_k \mathbf{L}_k^T$.

$$\mathbf{L}_k \equiv \begin{bmatrix} 1 & & & \\ \sqrt{\delta_1} & \ddots & & \\ & \ddots & \ddots & \\ & & \sqrt{\delta_{k-1}} & 1 \end{bmatrix}, \quad \mathbf{D}_k \equiv \begin{bmatrix} \gamma_0^{-1} & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \gamma_{k-1}^{-1} \end{bmatrix}.$$

CG, Lanczos and Gauss quadrature

Overview



At any iteration step k, CG (implicitly) determines weights and nodes of the k-point Gauss quadrature

$$\int_{\zeta}^{\xi} f(\lambda)\, d\omega(\lambda) \;=\; \sum_{i=1}^{k} \omega_i^{(k)} f(\theta_i^{(k)}) \;+\; \mathcal{R}_k[f]\,.$$

Let $f(\lambda) \equiv \lambda^{-1}$.

Lanczos-related quantities:

$$\left(\mathbf{T}_n^{-1}\right)_{1,1} \;=\; \left(\mathbf{T}_k^{-1}\right)_{1,1} + \mathcal{R}_k[\lambda^{-1}].$$

CG-related quantities

$$\|x\|_{\mathbf{A}}^2 \;=\; \sum_{j=0}^{k-1} \gamma_j \|r_j\|^2 + \|x - x_k\|_{\mathbf{A}}^2\,.$$

Gauss-Radau quadrature

$$\int_{\zeta}^{\xi} f \, d\omega(\lambda) = \sum_{i=1}^{k} w_i f(\nu_i) + w_{k+1} f(\mu) + \mathcal{R}_k[f],$$

μ is prescribed outside the open integration interval,
$[w_i]_{i=1}^{k+1}$ and $[\nu_i]_{i=1}^{k}$ are unknowns.

Algebraically: Given μ, find $\widetilde{\alpha}_{k+1}$ so that μ is an eigenvalue of

$$\widetilde{\mathbf{T}}_{k+1} = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \ddots & \ddots & & \\ & \ddots & \ddots & \beta_{k-1} & \\ & & \beta_{k-1} & \alpha_k & \beta_k \\ & & & \beta_k & \widetilde{\alpha}_{k+1} \end{bmatrix}.$$

Quadrature for $f(\lambda) = \lambda^{-1}$ is given by $\left(\widetilde{\mathbf{T}}_{k+1}^{-1}\right)_{1,1}$.

Quadrature formulas

Golub - Meurant - Strakoš approach

Quadrature formulas for $f(\lambda) = \lambda^{-1}$ take the form

$$\begin{aligned} \left(\mathbf{T}_n^{-1}\right)_{1,1} &= \left(\mathbf{T}_k^{-1}\right)_{1,1} + \mathcal{R}_k^{(G)}\,, \\ \left(\mathbf{T}_n^{-1}\right)_{1,1} &= \left(\widetilde{\mathbf{T}}_k^{-1}\right)_{1,1} + \mathcal{R}_k^{(R)}\,, \end{aligned}$$

and $\mathcal{R}_k^{(G)} > 0$ while $\mathcal{R}_k^{(R)} < 0$ if $\mu \leq \lambda_{\min}$. Equivalently

$$\begin{aligned} \|x\|_{\mathbf{A}}^2 &= \tau_k + \|x - x_k\|_{\mathbf{A}}^2\,, \\ \|x\|_{\mathbf{A}}^2 &= \widetilde{\tau}_k + \mathcal{R}_k^{(R)}\,. \end{aligned}$$

where $\tau_k \equiv \left(\mathbf{T}_k^{-1}\right)_{1,1}$, $\widetilde{\tau}_k \equiv \left(\widetilde{\mathbf{T}}_k^{-1}\right)_{1,1}$.

[Golub & Meurant 1994, 1997, 2010, Golub & Strakoš 1994]

Idea of estimating the $\mathbf{A}$-norm of the error
[Golub & Strakoš 1994]

Consider two quadrature rules at steps k and $k+d$, $d>0$,

$$\begin{aligned} \|x\|_{\mathbf{A}}^2 &= \tau_k + \|x-x_k\|_A^2 \,, \\ \|x\|_{\mathbf{A}}^2 &= \widehat{\tau}_{k+d} + \widehat{\mathcal{R}}_{k+d} \,. \end{aligned} \tag{1}$$

Then

$$\|x-x_k\|_{\mathbf{A}}^2 = \widehat{\tau}_{k+d} - \tau_k + \hat{\mathcal{R}}_{k+d} \,.$$

Gauss quadrature: $\hat{\mathcal{R}}_{k+d} = \mathcal{R}_{k+d}^{(G)} > 0 \rightarrow$ lower bound,
Radau quadrature: $\hat{\mathcal{R}}_{k+d} = \mathcal{R}_{k+d}^{(R)} < 0 \rightarrow$ upper bound.

How to compute efficiently

$$\widehat{\tau}_{k+d} - \tau_k \,?$$

How to compute $\widehat{\tau}_{k+d} - \tau_k$?

For numerical reasons, it is not good to compute explicitly τ_k, $\widehat{\tau}_{k+d}$, and subtract .

Instead, we use the formula,

$$\begin{aligned} \widehat{\tau}_{k+d} - \tau_k &= \sum_{j=k}^{k+d-2} (\tau_{j+1} - \tau_j) + (\widehat{\tau}_{j+d} - \tau_{j+d-1}) \\ &\equiv \sum_{j=k}^{k+d-2} \Delta_j + \widehat{\Delta}_{k+d-1}\,, \end{aligned}$$

and update the Δ's without subtraction. Recall that

$$\begin{aligned} \Delta_j &= \left(\mathbf{T}_{j+1}^{-1}\right)_{1,1} - \left(\mathbf{T}_{j}^{-1}\right)_{1,1}\,, \\ \widehat{\Delta}_{k+d-1} &= \left(\widehat{\mathbf{T}}_{k+d}^{-1}\right)_{1,1} - \left(\mathbf{T}_{k+d-1}^{-1}\right)_{1,1}\,. \end{aligned}$$

Golub and Meurant approach

[Golub & Meurant 1994, 1997]: Use tridiagonal matrices,

$$\boxed{\text{CG}} \;\rightarrow\; \boxed{\mathbf{T}_k} \;\rightarrow\; \boxed{\mathbf{T}_k - \mu\mathbf{I}} \;\rightarrow\; \boxed{\widetilde{\mathbf{T}}_k}$$

Compute the Δ's,

$$\begin{aligned} \Delta_{k-1} &\equiv \left(\mathbf{T}_k^{-1}\right)_{1,1} - \left(\mathbf{T}_{k-1}^{-1}\right)_{1,1}, \\ \Delta_k^{(\mu)} &\equiv \left(\widetilde{\mathbf{T}}_{k+1}^{-1}\right)_{1,1} - \left(\mathbf{T}_k^{-1}\right)_{1,1}. \end{aligned}$$

Use the formulas

$$\begin{aligned} \|x - x_k\|_{\mathbf{A}}^2 &= \sum_{j=k}^{k+d-1} \Delta_j + \|x - x_{k+d}\|_{\mathbf{A}}^2, \\ \|x - x_k\|_{\mathbf{A}}^2 &= \sum_{j=k}^{k+d-2} \Delta_j + \Delta_{k+d-1}^{(\mu)} + \mathcal{R}_{k+d}^{(R)}. \end{aligned}$$

CGQL (Conjugate Gradients and Quadrature via Lanczos)

input $\mathbf{A}$, b, x_0, μ
$r_0 = b - \mathbf{A}x_0$, $p_0 = r_0$
$\delta_0 = 0$, $\gamma_{-1} = 1$, $c_1 = 1$, $\beta_0 = 0$, $d_0 = 1$, $\tilde{\alpha}_1^{(\mu)} = \mu$,
for $k = 1, \dots,$ until convergence **do**
CG-iteration (k)

$$\begin{aligned}
\alpha_k &= \frac{1}{\gamma_{k-1}} + \frac{\delta_{k-1}}{\gamma_{k-2}}, \ \beta_k^2 = \frac{\delta_k}{\gamma_{k-1}^2} \\
d_k &= \alpha_k - \frac{\beta_{k-1}^2}{d_{k-1}}, \ \Delta_{k-1} = \|r_0\|^2 \frac{c_k^2}{d_k}, \\
\tilde{\alpha}_{k+1}^{(\mu)} &= \mu + \frac{\beta_k^2}{\alpha_k - \tilde{\alpha}_k^{(\mu)}}, \\
\Delta_k^{(\mu)} &= \|r_0\|^2 \frac{\beta_k^2 c_k^2}{d_k \left(\tilde{\alpha}_{k+1}^{(\mu)} d_k - \beta_k^2\right)}, \quad c_{k+1}^2 = \frac{\beta_k^2 c_k^2}{d_k^2}
\end{aligned}$$

Estimates(k,d)
end for

Our approach

[Meurant & T. 2012]

- We use tridiagonal matrices only implicitly.
- CG generates LDL^T factorization of $\mathbf{T}_k$.
- Update LDL^T factorizations of the tridiagonal matrices

$$\boxed{\widetilde{\mathbf{T}}_k}$$

- Quite complicated algebraic manipulations, but, in the end,
- we get **very simple formulas** for updating Δ_{k-1} and $\Delta_k^{(\mu)}$.
- This idea can be used also for other types of quadratures (Gauss-Lobatto, Anti-Gauss).

CGQ (Conjugate Gradients and Quadrature)

[Meurant & T. 2012]

input $\mathbf{A}$, b, x_0, μ,
$r_0 = b - \mathbf{A}x_0$, $p_0 = r_0$
$\Delta_0^{(\mu)} = \frac{\|r_0\|^2}{\mu}$,
for $k = 1, \ldots,$ until convergence **do**
CG-iteration(k)

$$\begin{aligned} \Delta_{k-1} &= \gamma_{k-1}\|r_{k-1}\|^2, \\ \Delta_k^{(\mu)} &= \frac{\|r_k\|^2 \left(\Delta_{k-1}^{(\mu)} - \Delta_{k-1}\right)}{\mu\left(\Delta_{k-1}^{(\mu)} - \Delta_{k-1}\right) + \|r_k\|^2} \end{aligned}$$

Estimates(k,d)
end for

Conclusions and questions

- The upper bound as well as the lower bound on the $\mathbf{A}$-norm of the error can be **computed in a simple way**.
- Unfortunately, the computation of the upper bound is **not always numerically stable**.
  - μ is far from λ_1 $\rightarrow$ overestimation,
  - μ is close to λ_1 $\rightarrow$ numerical troubles.
- The estimation of the $\mathbf{A}$-norm of the error **should be based** on the numerical stable **lower bound**.
- **How to detect** a reasonable decrease of the $\mathbf{A}$-norm of the error? (How to choose d adaptively?).
- Is there any way how to **involve** the upper bound? Understanding of numerical behaviour of the upper bound?

Related papers

- G. Meurant and P. Tichý, [On computing quadrature-based bounds for the $\mathbf{A}$-norm of the error in conjugate gradients, Numer. Algorithms, (2012)]
- G. H. Golub and G. Meurant, [Matrices, moments and quadrature with applications, Princeton University Press, USA, 2010.]
- Z. Strakoš and P. Tichý, [On error estimation in the conjugate gradient method and why it works in finite precision computations, Electron. Trans. Numer. Anal., 13 (2002), pp. 56–80.]
- G. H. Golub and G. Meurant, [Matrices, moments and quadrature. II. BIT, 37 (1997), pp. 687–705.]
- G. H. Golub and Z. Strakoš, [Estimates in quadratic formulas, Numer. Algorithms, 8 (1994), pp. 241–268.]

Thank you for your attention!