Minimizing convex quadratics with variable precision Krylov methods

Philippe Toint (with Serge Gratton and Ehouarn Simon)

Namur Center for Complex Systems (naXys), University of Namur, Belgium
CIMI, INP, Toulouse, France

(philippe.toint@unamur.be)

XII BRAZOPT, Iguazu, Brazil, July 2018

Thanks

- CIMI, Institut National Polytechnique, Toulouse, France
  (ANR-11-IDEX-0002-02)
- University of Namur, Belgium
- Belgian National Fund for Scientific Research

The (simple?) problem

We consider the unconstrained quadratic optimization (QO) problem:

$$\text{minimize} \quad q(x) = \tfrac{1}{2}x^T A x - b^T x$$

for $x, b \in \mathbb{R}^n$ and A an $n \times n$ symmetric positive-definite matrix.

A truly "core" problem in optimization (and linear algebra)

- the simplest nonlinear optimization problem
- subproblem in many methods for general nonlinear unconstrained optimization
- central in linear algebra (including solving elliptic PDEs)

Working assumptions

For what follows, we assume that

- the problem size n is large enough and A is dense enough to make factorization of A unavailable
- a Krylov iterative method (Conjugate Gradients, FOM) is used
- the cost of running this iterative method is dominated by the products Av

Focus on an optimization point of view: look at decrease in q rather than at decrease in the associated system's residual

ex: ensuring sufficient decrease in trust-region methods

Our aim, for x_* solution of QO,

$$\boxed{\text{Find } x_k \text{ such that } |q(x_k) - q(x_*)| \leq \epsilon |q(x_0) - q(x_*)|.}$$

A first motivating example: weather forecasting (1)

The $\boxed{\text{weakly-constrained 4D-Var}}$ formulation:

$$\min_{x \in \mathbb{R}^n} \frac{1}{2}\|x_0 - x_b\|^2_{B^{-1}} + \frac{1}{2}\sum_{j=0}^{N} \|\mathcal{H}_j(x_j) - y_j\|^2_{R_j^{-1}} + \frac{1}{2}\sum_{j=1}^{N} \|\underbrace{x_j - \mathcal{M}_j(x_{j-1})}_{q_j}\|^2_{Q_j^{-1}}$$

- $x = (x_0, \ldots, x_N)^T$ is the state control variable (with $x_j = x(t_j)$)
- x_b is the background given at the initial time (t_0).
- $y_j \in \mathbb{R}^{m_j}$ is the observation vector over a given time interval
- $\mathcal{H}_j$ maps the state vector x_j from model space to observation space
- $\mathcal{M}_j$ is an integration of the numerical model from time t_{j-1} to t_j
- B, R_j and Q_j are the covariance matrices of background, observation and model error. B and Q_j impractical to "invert"

A first motivating example: weather forecasting (2)

Solve by a Gauss-Newton method whose subproblem (at iteration k) is

$$\min_{\delta x} \frac{1}{2}\|\delta x_0 - b^{(k)}\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\sum_{j=0}^{N}\left\|H_j^{(k)}\delta x_j - d_j^{(k)}\right\|^2_{\mathbf{R}_j^{-1}} + \frac{1}{2}\sum_{j=1}^{N}\|\underbrace{\delta x_j - M_j^{(k)}\delta x_{j-1}}_{\delta q_j} - c_j^{(k)}\|^2_{\mathbf{Q}_j^{-1}}$$

- δx is the increment in x.
- The vectors $b^{(k)}$, $c_j^{(k)}$ and $d_j^{(k)}$ are defined by

$$b^{(k)} = x_b - {x_0}^{(k)}, \quad c_j^{(k)} = q_j^{(k)}, \quad d_j^{(k)} = \mathcal{H}_j({x_j}^{(k)}) - y_j$$

and are calculated at the outer loop.

A first motivating example: weather forecasting (3)

Can be rewritten as

$$\min_{\delta x} \; q_{\rm st} = \frac{1}{2}\|L\delta x - b\|^2_{D^{-1}} + \frac{1}{2}\|H\delta x - d\|^2_{R^{-1}}$$

where

- $L = \begin{pmatrix} I & & & & \\ -M_1 & I & & & \\ & -M_2 & I & & \\ & & \ddots & \ddots & \\ & & & -M_N & I \end{pmatrix}$
- $d = (d_0, d_1, \dots, d_N)^T$ and $b = (b, c_1, \dots, c_N)^T$
- $H = \mathrm{diag}(H_0, H_1, \dots, H_N)$
- $D = \mathrm{diag}(B, Q_1, \dots, Q_N)$ and $R = \mathrm{diag}(R_0, R_1, \dots, R_N)$

A first motivating example: weather forecasting (3)

$$\min_{\delta x} \; q_{\rm st} = \frac{1}{2}\|L\delta x - b\|^2_{D^{-1}} + \frac{1}{2}\|H\delta x - d\|^2_{R^{-1}}$$

This is a standard QO, but **HUGE!** Note that

$$\nabla^2 q_{\rm st} = L^T D^{-1} L + H^T R^{-1} H$$

In addition $\boxed{D^{-1} = \mathrm{diag}(B^{-1}, Q_1^{-1}, \ldots, Q_N^{-1}) \text{ is unavailable!}}$

Thus $\nabla^2 q_{st} v$ (a Hessian times vector product) must be computed by

- $w = Lv$,
- solve $Dz = w$ using some (preconditioned) Krylov method
- $v = L^T z + H^T R^{-1} H v$

A second motivating example: variable precision arithmetic

Next barrier in hyper computing: energy dissipation!

Heat production is proportional to chip surface, hence

$$\text{energy output} \;\approx \left(\;\text{number of digits used}\;\right)^2$$

Architectural trend: use multiprecision arithmetic

- graphical processing units (GPUs)
- hierarchy of specialized CPUs (double, single, half, ...)

How to use this hierarchy optimally for fully accurate results?

Inaccuracy frameworks

Our proposal;

> Make the Krylov methods for QO more efficient by allowing error on the matrix-vector product (the dominant computation)

Two frameworks of interest:

- Continuous accuracy levels
  ex: WC-4D-VAR, where accuracy in the inversion $Dz = w$ can be continuously chosen
- Discrete accuracy levels
  ex: double-single-half precision arithmetic

Considered here:

> - Full orthonormalisation method (FOM)
> - Conjugate Gradients (CG)

with (wlog) $x_0 = 0$ and $q(x_0) = 0$.

A central equality

Define $r(x) \stackrel{\rm def}{=} Ax - b = \nabla q(x)$ and $Ax_* = b$.

$$\boxed{q(x) - q(x_*) = \tfrac{1}{2}\|r(x)\|^2_{A^{-1}}}$$

$$\begin{aligned}
\tfrac{1}{2}\|r(x)\|^2_{A^{-1}} &= \tfrac{1}{2}(Ax-b)^T A^{-1}(Ax-b) \\
&= \tfrac{1}{2}(x-x_*)^T A(x-x_*) \\
&= \tfrac{1}{2}(x^T Ax - 2x^T Ax_* + x_*^T Ax_*) \\
&= q(x) - q(x_*)
\end{aligned}$$

Hence

Decrease in q can be monitored by considering the A^{-1} norm of its gradient

The primal-dual norm

$\Rightarrow$ natural to consider the inaccuracy on the product Av by measuring the backward error

$$\|E\|_{A^{-1},A} \stackrel{\text{def}}{=} \sup_{x\neq 0} \frac{\|Ex\|_{A^{-1}}}{\|x\|_A} = \|A^{-1/2}EA^{-1/2}\|_2$$

(primal-dual norm)

Let A be a symmetric and positive definite matrix and E be any symmetric perturbation. Then, if $\|E\|_{A^{-1},A} < 1$, the matrix $A + E$ is symmetric positive definite.

The main idea

Krylov methods reduce the (internally recurred) residual r_k on successive nested Krylov spaces

$\Rightarrow$ can expect r_k to converge to zero

$\Rightarrow$ keep $r(x_k) - r_k$ small in the appropriate norm

For FOM and CG, if

$$\max\left[\|r_k - r(x_k)\|_{A^{-1}}, \|r_k\|_{A^{-1}}\right] \leq \frac{\sqrt{\epsilon}}{2}\|b\|_{A^{-1}}$$

then

$$|q(x_k) - q(x_*)| \leq \epsilon |q(x_*)|$$

The inexact FOM algorithm

Theoretical inexact FOM algorithm

1. Set $\beta = \|b\|_2$, and $v_1 = [b/\beta]$,
2. For k=1, 2, . . . , do
3. $\quad w_k = (A + E_k) v_k$
4. $\quad$ For $i = 1, \ldots, k$ do
5. $\quad\quad h_{i,k} = v_i^T w_k$
6. $\quad\quad w_k = w_k - h_{i,k} v_i$
7. $\quad$ EndFor
8. $\quad h_{k+1,k} = \|w_k\|_2$
9. $\quad y_k = H_k^{-1}(\beta e_1)$
10. $\quad$ if $|h_{k+1,k} e_k^T y_k|$ is small enough then go to 13
11. $\quad v_{k+1} = w_k / h_{k+1,k}$
12. EndFor
13. $x_k = V_k y_k$

Results for the inexact FOM

Let $\epsilon_\pi > 0$ and let $\phi \in \mathbb{R}^k_+$ such that $\sum_{j=1}^k \phi_j^{-1} \leq 1$. Suppose also that, for all $j \in \{1, \ldots, k\}$,

$$\|E_j\|_{A^{-1},A} \leq \omega_j^{\rm FOM} \stackrel{\rm def}{=} \min\left[1, \frac{\epsilon_\pi \, \|b\|_{A^{-1}}}{\phi_j \|v_j\|_A \|H_k^{-1}\|_2 \|r_{j-1}\|_2}\right] \quad (2.1)$$

Then

$$\|r(x_k) - r_k\|_{A^{-1}} \leq \epsilon_\pi \, \|b\|_{A^{-1}}.$$

Let $\epsilon > 0$ and suppose that, at iteration $k > 0$ of the FOM algorithm,

$$\|r_k\|_{A^{-1}} \leq \tfrac{1}{2}\sqrt{\epsilon}\, \|b\|_{A^{-1}}$$

and the product error matrices E_j satisfy (2.1) with $\epsilon_\pi = \frac{1}{2}\sqrt{\epsilon}$ for some $\phi \in \mathbb{R}^k$ (as above). Then

$$|q(x_k) - q(x_*)| \leq \epsilon |q(x_*)|$$

The inexact Conjugate Gradients algorithm

Theoretical inexact CG algorithm

1. Set $x_0 = 0$, $\beta_0 = \|b\|_2^2$, $r_0 = -b$ and $p_0 = r_0$
2. For k=0, 1, . . . , do
3. $\quad c_k = (A + E_k)p_k$
4. $\quad \alpha_k = \beta_k / p_k^T c_k$
5. $\quad x_{k+1} = x_k + \alpha_k p_k$
6. $\quad r_{k+1} = r_k + \alpha_k c_k$
7. $\quad$ if r_{k+1} is small enough then stop
8. $\quad \beta_{k+1} = r_{k+1}^T r_{k+1}$
9. $\quad p_{k+1} = -r_{k+1} + (\beta_{k+1}/\beta_k) p_k$
10. EndFor

Results for the inexact CG

Let $\epsilon_\pi > 0$ and let $\phi \in \mathbb{R}^k_+$ such that $\sum_{j=1}^k \phi_j^{-1} \leq 1$. Suppose also that, for all $j \in \{0, \dots, k-1\}$,

$$\|E_j\|_{A^{-1},A} \leq \omega_j^{\text{CG}} \stackrel{\text{def}}{=} \frac{\epsilon_\pi \, \|b\|_{A^{-1}} \|p_j\|_A}{\phi_{j+1}\|r_j\|_2^2 + \epsilon_\pi \, \|b\|_{A^{-1}} \|p_j\|_A} \tag{2.2}$$

Then

$$\|r(x_k) - r_k\|_{A^{-1}} \leq \epsilon_\pi \, \|b\|_{A^{-1}}.$$

Let $\epsilon > 0$ and suppose that, at iteration $k > 0$ of the CG algorithm,

$$\|r_k\|_{A^{-1}} \leq \tfrac{1}{2}\sqrt{\epsilon} \, \|b\|_{A^{-1}}$$

and the product error matrices E_j satisfy (2.2) with $\epsilon_\pi = \frac{1}{2}\sqrt{\epsilon}$ for some $\phi \in \mathbb{R}^k$ (as above). Then

$$|q(x_k) - q(x_*)| \leq \epsilon |q(x_*)|$$

Achieved vs optimal decrease

Let q be the value of the quadratic recurred internally by FOM or CG.

Let x be the result of applying the FOM or CG algorithm with inexact products and suppose that the above error bounds hold with $\epsilon_\pi = \frac{1}{2}\sqrt{\epsilon}$. Then

$$\frac{|q(x) - q|}{|q(x_*)|} \leq \sqrt{\epsilon}(1 + \sqrt{\epsilon}).$$

Can one trust the internally computed decrease? Rather pessismistic!

Managing the inaccuracy budget

Assume $k_{\max}$, an estimate of the maximum number of iterations, is known.

At iteration j of FOM/CG:

$$\left.\begin{matrix} v_j \\ r_{j-1} \\ \phi_j \end{matrix}\right\} \rightarrow \left.\begin{matrix} v_j \\ \omega_j \end{matrix}\right\} \rightarrow \boxed{\begin{matrix}\text{product} \\ \text{routine}\end{matrix}} \rightarrow \begin{matrix} \boxed{(A+E_j)v_j} \\ \|E_j\|_{A^{-1},A} \end{matrix} \rightarrow \hat{\phi}_j \rightarrow \phi_{j+1}$$

where

$$\hat{\phi}_j^{\rm FOM} = \frac{\epsilon_\pi \, \|b\|_{A^{-1}}}{\|E_j\|_{A^{-1},A}\|v_j\|_A\|H_k^{-1}\|_2\|r_{j-1}\|_2}$$

$$\hat{\phi}_j^{\rm CG} = \frac{(1-\|E_j\|_{A^{-1},A})\,\epsilon_\pi \, \|b\|_{A^{-1}}\|p_j\|_A}{\|E_j\|_{A^{-1},A}\|r_j\|_2^2}$$

and $\quad \phi_{j+1} = \dfrac{k_{\max} - j}{1-\sum_{p=1}^{j} \hat{\phi}_p^{-1}}$

So what?

- The primal-dual norm $\|E_j\|_{A^{-1},A}$ is sometimes difficult to evaluate
- The error bounds remain unfortunately impractical (they involve $\|b\|_{A^{-1}}$, $\|v_j\|_A$ or $\|p_j\|_A$, which cannot be computed readily in the course of the FOM or CG algorithm).
- The termination test $\|r_k\|_{A^{-1}} \leq \frac{1}{2}\sqrt{\epsilon}\,\|b\|_{A^{-1}}$ also involves the unavailable $\|r_k\|_{A^{-1}}$

Give up? Not quite...

- the FOM error bound allows a growth of the error in $\|r_j\|^{-1}$ while (2.2) allows a growth of the order of $\|r_j\|^{-2}\|p_j\|_A$ instead.
- The ϕ_j may be viewed as an error management strategy. A simple choice is to define $\phi_j = n$ for all j but there may be better options (discussed later).

Using the true (unavailable) quantities (1)

Would this work at all if using the true $\|b\|_{A^{-1}}$, $\|v_j\|_A$ and $\|p_j\|_A$?

Consider 6 algorithms:

- FOM: the standard full-accuracy FOM
- iFOM: the inexact FOM (with exact bounds, for now)
- CG: the standard full-accuracy CG
- CGR: the full-accuracy CG *with reorthogonalization*
- iCG: the inexact CG (with exact bounds, for now)
- iCGR: the inexact CGR (with exact bounds, for now)

Continuous accuracy levels (1)

Comparing equivalent numbers of full accuracy products:

- Assume obtaining full accuracy is a linearly convergent process of rate ρ
  (realistic for our weather prediction data assimilation example)
- Cost of an ϵ-accurate solution:

$$\frac{\log(\epsilon)}{\log(\rho)}$$

- Cost of an ω-accurate solution

$$\frac{\log(\omega)}{\log(\rho)}$$

$\Rightarrow$ sum these values during computing and compare them.

Continuous accuracy levels (2)

Compare on:

- synthetic matrices of size 1000×1000 with varying conditioning (from 10^1 to 10^8) and log-linearly spaced eigenvalues
- "real" matrices from the NIST Matrix Market (paper only)
- use different levels of final accuracy ($\epsilon = 10^{-3},\ 10^{-5},\ 10^{-7}$)

Note that

> Continuous accuracy levels $\Rightarrow$ no room for inaccuracy budget management!

Continuous accuracy levels (3)

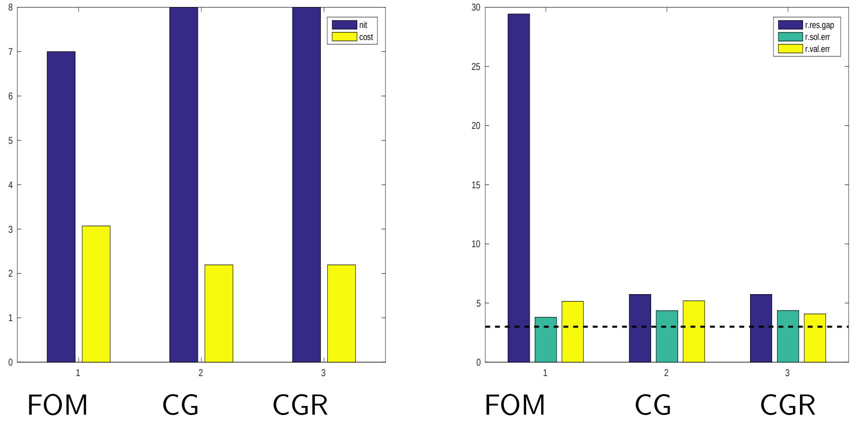


Figure: Exact bounds, $\kappa(A) = 10^1$, $\epsilon = 10^{-3}$ (continuous case)

Continuous accuracy levels (4)

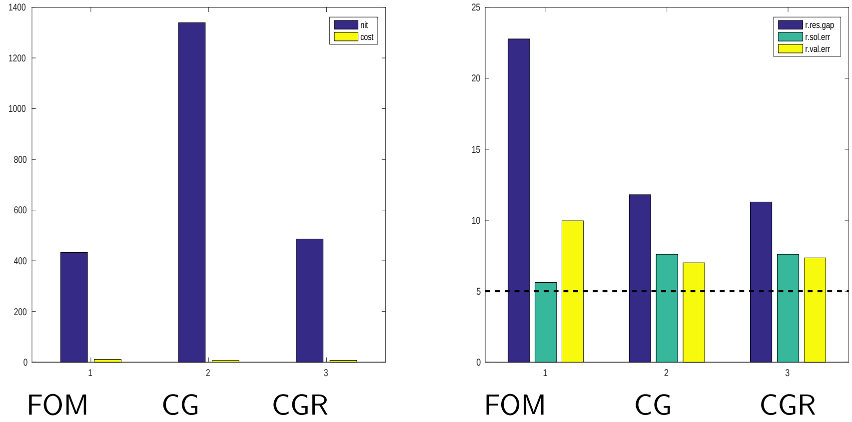


Figure: Exact bounds, $\kappa(A) = 10^5$, $\epsilon = 10^{-5}$ (continuous case)

Continuous accuracy levels (5)

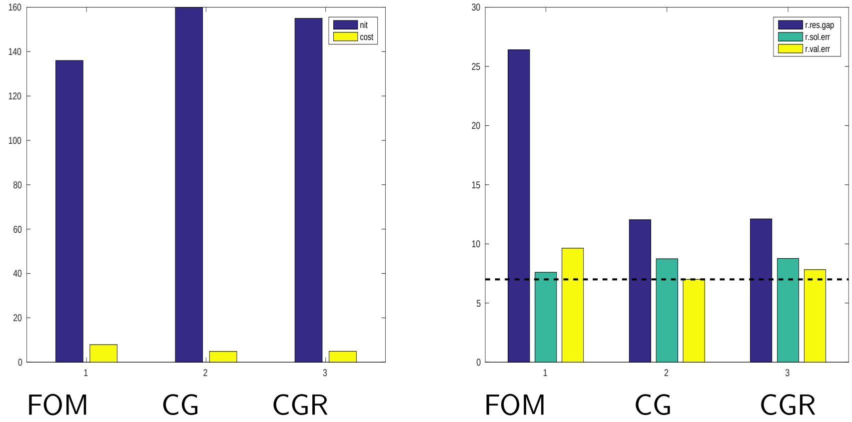


Figure: Exact bounds, $\kappa(A) = 10^3$, $\epsilon = 10^{-7}$ (continuous case)

Discontinuous accuracy levels (1)

Focus on $\boxed{\text{multiprecision arithmetic}}$. Assume

- 3 levels of accuracy (double, single, half)
- a ratio of 4 in efficiency when moving from one level to the next

Use the sames matrices and final accuracies as above.

Apply the inaccuracy budget management!

Discontinuous accuracy levels (2)

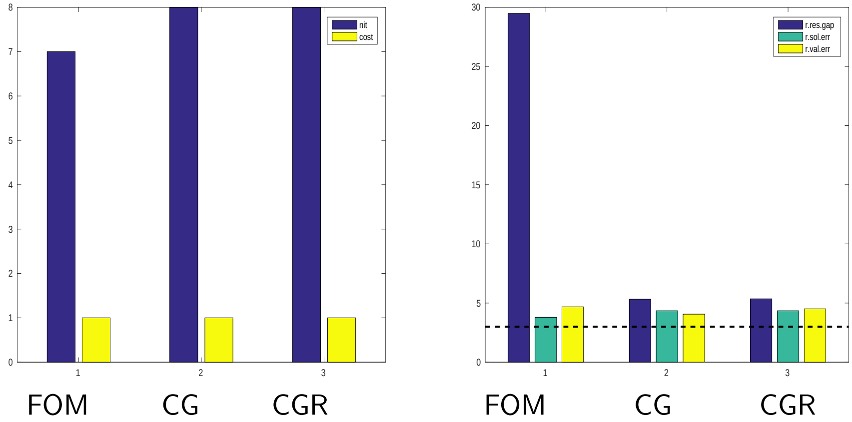
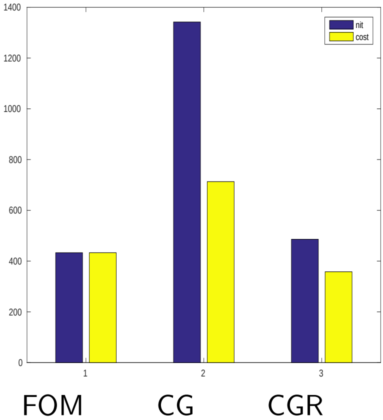


Figure: Exact bounds, $\kappa(A) = 10^1$, $\epsilon = 10^{-3}$ (continuous case)

Discontinuous accuracy levels (3)



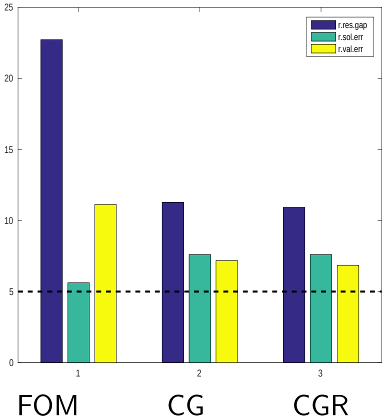
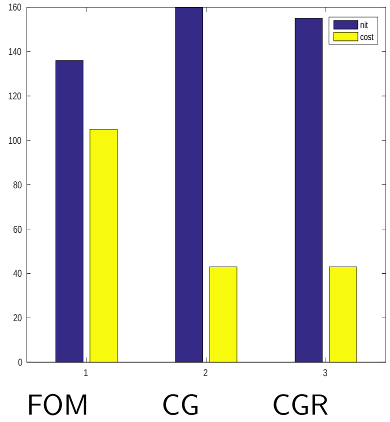


Figure: Exact bounds, $\kappa(A) = 10^5$, $\epsilon = 10^{-5}$ (continuous case)

Discontinuous accuracy levels (4)



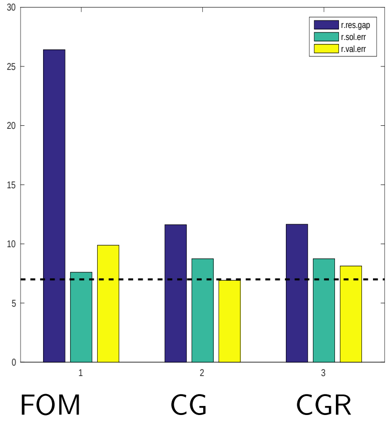


Figure: Exact bounds, $\kappa(A) = 10^3$, $\epsilon = 10^{-7}$ (continuous case)

Adhoc approximations

Abandon theoretical but unavailable quantities $\rightarrow$ approximate them:

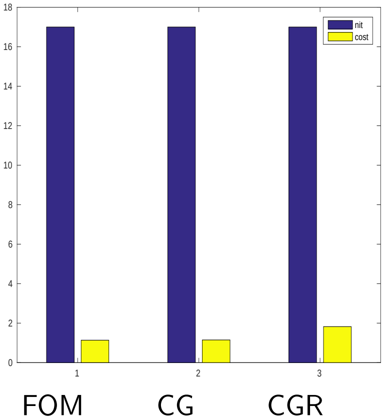
- $\|E\|_{A^{-1},A} \geq \lambda_{\min}(A)^{-1}\|E\|_2$
- $\|p\|_A \approx \frac{1}{n}\mathrm{Tr}(A)\|p\|_2$
  (ok for p with random independent components)
- $\|b\|_{A^{-1}} = |q(x_*)| \approx q_k \approx \frac{1}{2}|b^T x_k|$
- $\|H_k^{-1}\| = \frac{1}{\lambda_{\min}(H_k)} \leq \frac{1}{\lambda_{\min}(A)}$ (FOM only)
- $k_{\max} \approx \frac{\log(\epsilon)}{\log(\rho)}$ with $\rho \stackrel{\text{def}}{=} \frac{\sqrt{\lambda_{\max}/\lambda_{\min}}-1}{\sqrt{\lambda_{\max}/\lambda_{\min}}+1}$

Termination test (Arioli & Gratton):

$$q_{k-d} - q_k \leq \tfrac{1}{4}\epsilon|q_k|$$

for some stabilization delay d (e.g. 10)

Does it still work (continuous accuracy levels, 1)?



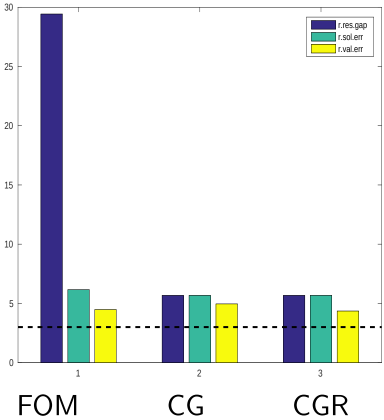


Figure: Approximate bounds, $\kappa(A) = 10^1$, $\epsilon = 10^{-3}$ (continuous case)

Does it still work (continuous accuracy levels, 2)?

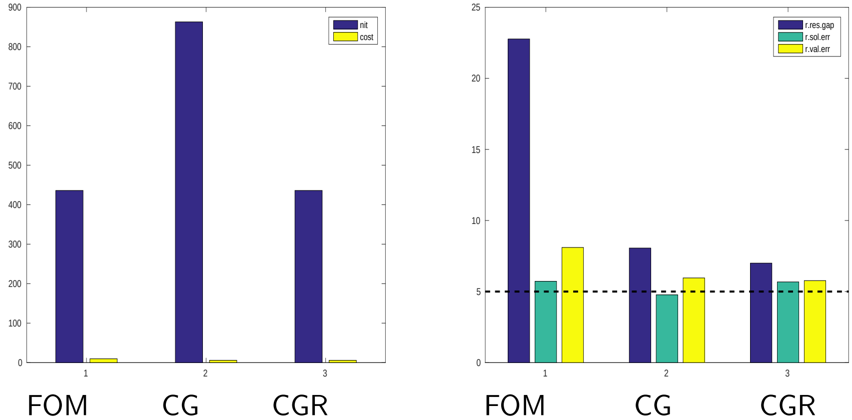


Figure: Approximate bounds, $\kappa(A) = 10^5$, $\epsilon = 10^{-5}$ (continuous case)

Does it still work (continuous accuracy levels, 3)?

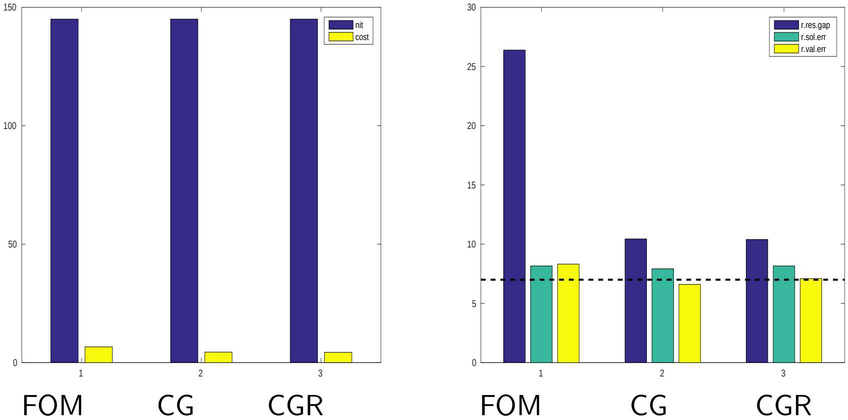


Figure: Approximate bounds, $\kappa(A) = 10^3$, $\epsilon = 10^{-7}$ (continuous case)

Does it still work (multiprecision, 1)?

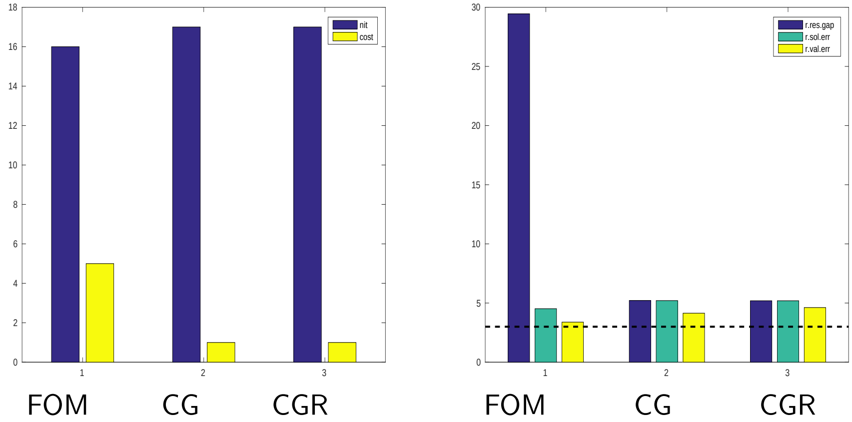
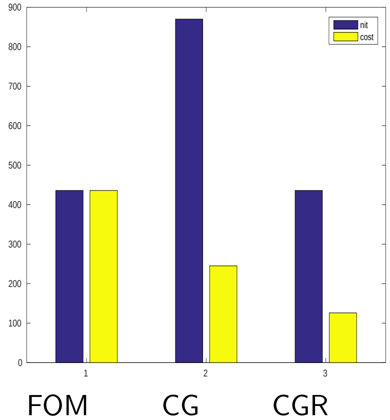


Figure: Approximate bounds, $\kappa(A) = 10^1$, $\epsilon = 10^{-3}$ (continuous case)

Does it still work (multiprecision, 2)?



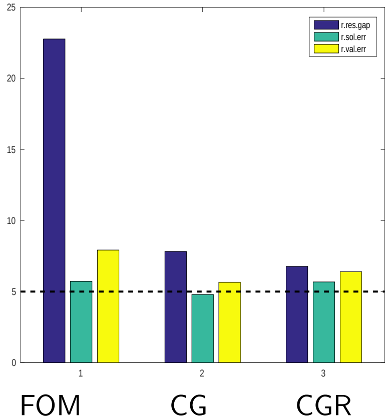


Figure: Approximate bounds, $\kappa(A) = 10^5$, $\epsilon = 10^{-5}$ (continuous case)

Does it still work (multiprecision, 3)?

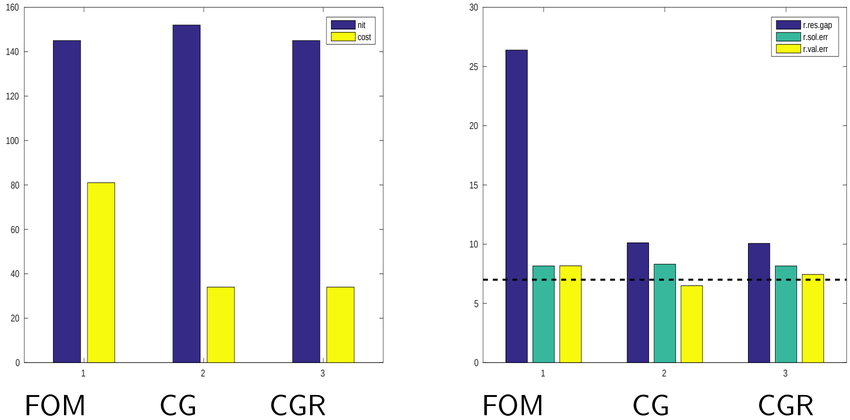


Figure: Approximate bounds, $\kappa(A) = 10^3$, $\epsilon = 10^{-7}$ (continuous case)

Conclusions and perspectives

Summary:

- Optimization-focused theory for iterative QO with inexact products
- Theoretical gains substantial
- Translates well to practice after approximations

Perspectives:

- More general (controlable) inexactness in optimization (inexactly weighted least-squares, . . .)

Thank your for your attention!

Reference

- S. Gratton, E. Simon, Ph. L. Toint,
  Minimizing convex quadratics with variable precision Krylov methods,
  arXiv:1807.07476