On the Convergence Rate of Different Variants of the Conjugate Gradient Algorithm

Hexuan Liu, Anne Greenbaum, Tyler Chen

University of Washington, Department of Applied Mathematics

February 24, 2019

Open Questions

1. For convergence rate, we are interested in two quantities:

$$\epsilon_1 := \max_{j=1,\cdots,J} \|f_j\|/\|A\|,$$

$$\epsilon_2 := \max_{j=1,\cdots,J} |\langle \beta_j q_{j+1}, q_j \rangle|/\|A\|.$$

Which variants have small values of these quantities?

2. If an implementation does not have small ϵ_1, ϵ_2, does it necessarily perform poorly?

Convergence Rate

Recall: CG can be written in a form like Lanczos, where Lanczos vectors are normalized CG residuals. In exact arithmetic,

$$AQ_J = Q_J T_J + \beta_J q_{J+1} e_J^T$$

In finite precision arithmetic,

$$AQ_J = Q_J T_J + \beta_J q_{J+1} e_J^T + F_J$$

Define

$$\epsilon_1 := \max_{j=1,\cdots,J} \|f_j\| / \|A\|$$

$$\epsilon_2 := \max_{j=1,\cdots,J} |\langle \beta_j q_{j+1}, q_j \rangle| / \|A\|$$

Convergence Rate

[Greenbaum 89'] If $\epsilon_1,\ \epsilon_2$ are small, finite precision CG is like exact CG for a matrix with eigenvalues in small intervals about the eigenvalues of A. Generally, the smaller ϵ_1, ϵ_2, the smaller the interval size.

Algorithm 1 HSCG

$r_0 = b - Ax_0,\ p_0 = r_0$
for $k = 1$ to $nmax$ **do**
  $a_{k-1} = \frac{\langle r_{k-1}, r_{k-1} \rangle}{\langle p_{k-1}, Ap_{k-1} \rangle}$
  $x_k = x_{k-1} + a_{k-1} p_{k-1}$
  $r_k = r_{k-1} - a_{k-1} Ap_{k-1}$
  $b_k = \frac{\langle r_k, r_k \rangle}{\langle r_{k-1}, r_{k-1} \rangle}$
  $p_k = r_k + b_k p_{k-1}$
end for

$$r_k = r_{k-1} - a_{k-1} Ap_{k-1} + \delta_{r_k}, \quad \delta_{r_k} \le \xi(\|r_{k-1}\| + 2|a_{k-1}|c\|A\|\|p_{k-1}\|)$$

$$p_k = r_k + b_k p_{k-1} + \delta_{p_k}, \quad \delta_{p_k} \le \xi(\|r_k\| + 2|b_k|\|p_{k-1}\|)$$

$$f_k = \frac{(-1)^k}{\|r_{k-1}\|}\left(\frac{b_{k-1}}{a_{k-2}}\delta_{r_{k-1}} + A\delta_{p_{k-1}} - \frac{1}{a_{k-1}}\delta_{r_k}\right)$$

Chronopoulos and Gear CG

Algorithm 2 CGCG

$r_0 = b - Ax_0$, $p_0 = r_0$, $s_0 = Ap_0$, $\nu_0 = \langle r_0, r_0 \rangle$, $\eta_0 = \langle s_0, p_0 \rangle$, $a_0 = \nu_0 / \eta_0$

for $k = 1$ to *nmax* **do**

$x_k = x_{k-1} + a_{k-1} p_{k-1}$

$r_k = r_{k-1} - a_{k-1} s_{k-1}$

$w_k = Ar_k$, $\nu_k = \langle r_k, r_k \rangle$

$b_k = \nu_k / \nu_{k-1}$

$p_k = r_k + b_k p_{k-1}$, $s_k = w_k + b_k s_{k-1}$, $\eta_k = \langle w_k, r_k \rangle$

$a_k = \nu_k / (\eta_k - (b_k / a_{k-1}) \nu_k)$

end for

$$f_k = \frac{(-1)^k}{\|r_{k-1}\|} \left(\frac{b_{k-1}}{a_{k-2}} \delta_{r_{k-1}} + \delta_{s_{k-1}} - \frac{1}{a_{k-1}} \delta_{r_k} \right)$$

Ghysels and Vanroose CG (pipelined)

Algorithm 3 GVCG

$r_0 = b - Ax_0,\ p_0 = r_0,\ s_0 = Ap_0,\ w_0 = s_0,\ u_0 = Aw_0,\ \nu_0 = \langle r_0, r_0\rangle,\ a_0 = \nu_0/\langle s_0, p_0\rangle$

for $k = 1$ to $nmax$ **do**

$x_k = x_{k-1} + a_{k-1}p_{k-1}$

$r_k = r_{k-1} - a_{k-1}s_{k-1}$

$w_k = w_{k-1} - a_{k-1}u_{k-1}$

$t_k = Aw_k,\ \nu_k = \langle r_k, r_k\rangle$

$b_k = \nu_k/\nu_{k-1}$

$a_k = \nu_k/(\langle w_k, r_k\rangle - (b_k/a_{k-1})\nu_k)$

$p_k = r_k + b_k p_{k-1},\ s_k = w_k + b_k s_{k-1},\ u_k = t_k + b_k u_{k-1}$

end for

In GVCG, w_k is computed recursively:

$$w_k = w_{k-1} - a_{k-1}u_{k-1} + \delta_{w_k}$$

$$s_k = w_k + b_k s_{k-1} + \delta_{s_k}$$

$$u_k = t_k + b_k u_{k-1} + \delta_{u_k}$$

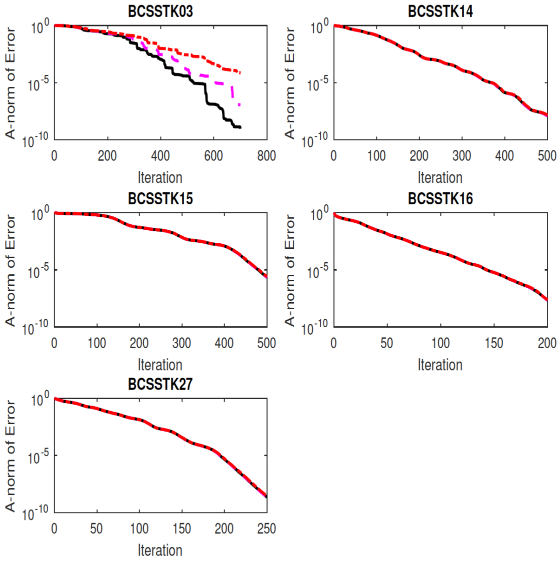
f_k depends not only on local rounding errors, but also on the amount by which w_k differs from Ar_k. This involves rounding errors made at all previous steps. Therefore, we expect ϵ_1 to be large for GVCG.

Convergence Rate: Numerical Examples

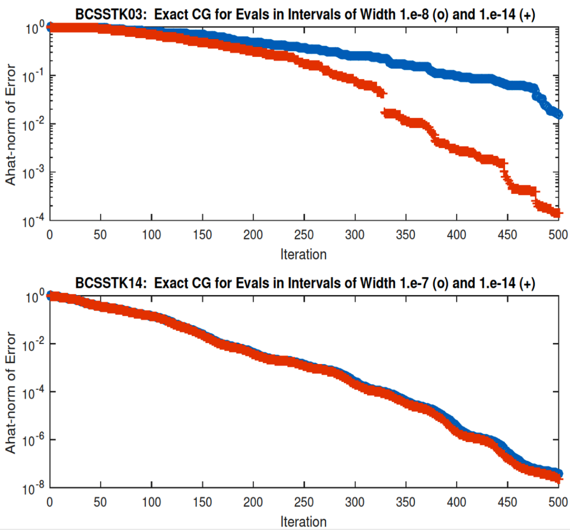
	HSCG	CGCG	GVCG
bcsstk03	$\epsilon_1 = 9.5e-16$	$\epsilon_1 = 4.9e-16$	$\epsilon_1 = 4.5e-8$
	$\epsilon_2 = 1.9e-12$	$\epsilon_2 = 1.5e-13$	$\epsilon_2 = 1.7e-13$
bcsstk14	$\epsilon_1 = 1.4e-15$	$\epsilon_1 = 1.8e-15$	$\epsilon_1 = 3.2e-6$
	$\epsilon_2 = 1.2e-15$	$\epsilon_2 = 3.8e-15$	$\epsilon_2 = 3.9e-15$
bcsstk15	$\epsilon_1 = 1.7e-15$	$\epsilon_1 = 1.8e-15$	$\epsilon_1 = 3.1e-8$
	$\epsilon_2 = 8.6e-16$	$\epsilon_2 = 4.0e-15$	$\epsilon_2 = 2.7e-15$
bcsstk16	$\epsilon_1 = 1.7e-15$	$\epsilon_1 = 2.2e-15$	$\epsilon_1 = 1.1e-7$
	$\epsilon_2 = 4.1e-16$	$\epsilon_2 = 2.9e-15$	$\epsilon_2 = 1.5e-15$
bcsstk27	$\epsilon_1 = 9.3e-16$	$\epsilon_1 = 8.5e-16$	$\epsilon_1 = 2.8e-6$
	$\epsilon_2 = 3.5e-16$	$\epsilon_2 = 1.7e-15$	$\epsilon_2 = 2.3e-15$

- Finite precision CG implementations can be shown to behave like exact CG applied to an extended matrix with eigenvalues in $[\lambda_i - \delta, \lambda_i + \delta]$ if ϵ_1 and ϵ_2 are tiny.
- ϵ_1 and ϵ_2 only give an upper bound on δ. If an implementation does not have small ϵ_1, ϵ_2, does it necessarily perform poorly?

Convergence Rate: Numerical Examples



Convergence Rate: Numerical Examples



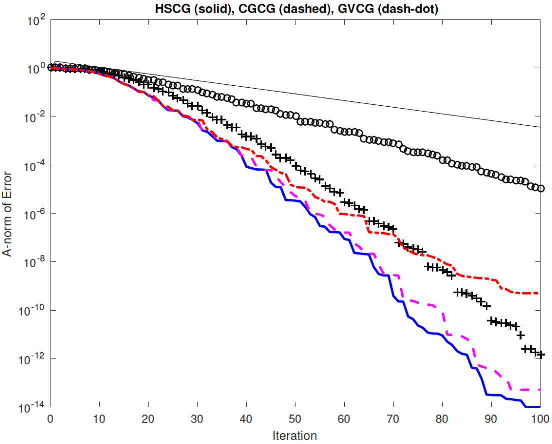
Convergence Rate

- For problems where the interval size has a big effect on convergence of exact CG, expect significant differences in convergence rates for different finite precision CG implementations.
- For problems that do not depend on δ, it seems that all three variants converge at similar rate. (This is the case for most problems in practice)

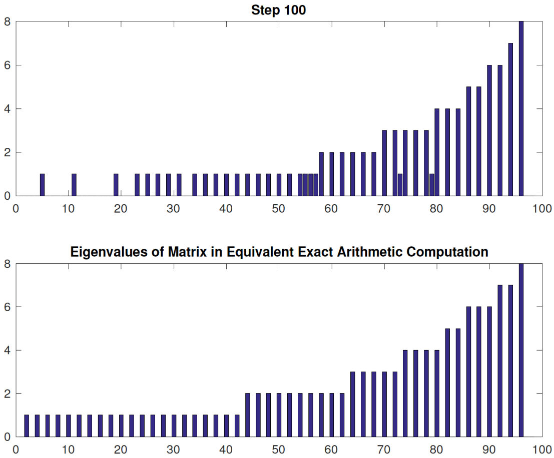
A Model Problem

$$\lambda_1 = 0.001,\ \lambda_n = 1,\ \lambda_i = \lambda_1 + \frac{i-1}{n-1}(\lambda_n - \lambda_1)\rho^{n-i},\ i = 2, \cdots, n-1$$

For $n = 48$, $\rho = 0.8$:

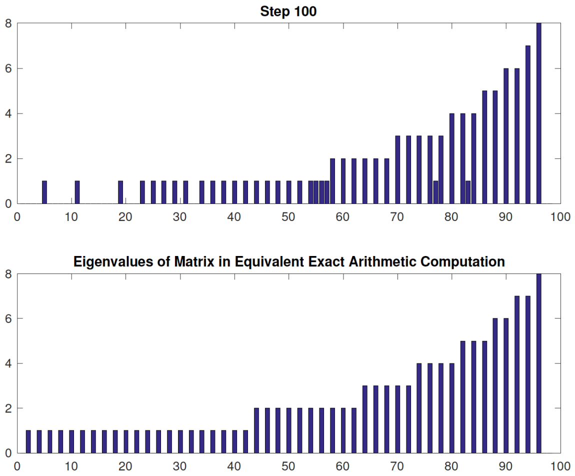


A Model Problem: HSCG



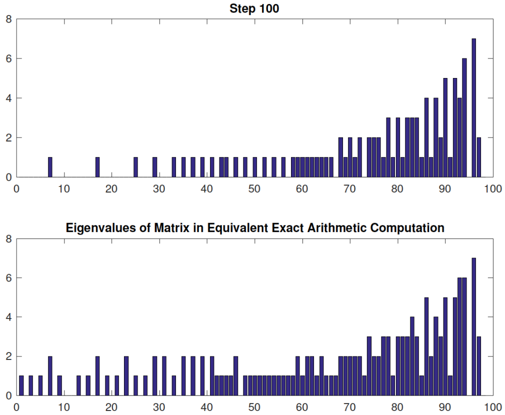
Eigenvalues of $\hat{A}$ were all within $1.7e-15$ of an eigenvalue of A.

A Model Problem: CGCG



Eigenvalues of $\hat{A}$ were all within $\color{red}{4.2e-15}$ of an eigenvalue of A.

A Model Problem: GVCG



Eigenvalues of $\hat{A}$ were as far as $4.1e-2$ from an eigenvalue of A. Also, by Cauchy interlacing theorem, the interval width cannot be less than $1.5e-6$.

A Model Problem

- For GVCG, we can still find an extended matrix $\hat{A}$ such that the A-norm of the error in the finite precision computation matches the $\hat{A}$-norm of the error in the equivalent exact CG computation.
- However, the eigenvalues of $\hat{A}$ lie in significant larger intervals about the eigenvalues of A, and (for this model problem) the best exact arithmetic bounds are significantly weakened.

Possible Remedies

To improve convergence, do things that will reduce ϵ_1 and ϵ_2.

1. The gap between w_k and Ar_k plays an important role in the three-term recurrence of the residual, as well as in the gap between the true and updated residual.
   So we propose to occasionally replace w_k with Ar_k.
   (If $w_k = Ar_k$ for every step, GVCG becomes CGCG)
2. Preconditioning/ prescaling to make the eigenvalues distribute more uniformly.