Numerical Linear Algebra

Maya Neytcheva, TDB, February-March 2021

Generalized Conjugate Gradient Methods

GCG

GCG-type of methods:

Reasons to be widely used:

- parameter-free iterative methods
- finite termination property
- optimality approximation property
- favourable memory requirements and computational complexity per iteration
- the use of a *good* preconditioner can significantly improve the performance
- the influence of roundoff error is usually acceptable

Derivation of the GCG method

GCG can be derived within the framework of the (generalized) Least Squares methods, where at each step the square of the residual norm is minimized.

We want to solve

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

The matrix A can be even rectangular of size $n \times m$. One way to go is to consider some auxiliary matrix Q and solve either

$$QA\mathbf{x} = Q\mathbf{a}$$

or

$$AQ\mathbf{y} = \mathbf{a} \quad \text{with} \quad \mathbf{x} = Q\mathbf{y}.$$

Derivation of the GCG method (cont)

For the special choice $Q = A^T$ we obtain the normal equation to solve:

$$A^T A\mathbf{x} = A^T\mathbf{a} \text{ – Least Squares residuals}$$
or
$$AA^T\mathbf{y} = \mathbf{a} \text{ – Least Squares error.}$$

If A is square, Q can be seen as a preconditioner to A (left or right, correspondingly).
Further we will work only with a square matrix B, where

$$B = QA$$
$$B = AQ$$
$$B = C^{-1}A$$

and we are going to solve the system

$$B\mathbf{x} = \mathbf{b}.$$

Derivation of the GCG method (cont)

Consider now the following quadratic form:

$$f(\mathbf{x}) = \frac{1}{2}(\mathbf{r}, \mathbf{r})_0 = \frac{1}{2}(B\mathbf{x} - \mathbf{b}, B\mathbf{x} - \mathbf{b}) \quad (1)$$

where $(\cdot, \cdot)_0$ is defined as $(\mathbf{u}, \mathbf{v})_0 = (\mathbf{u}^T M_0 \mathbf{v})$ for some given positive definite matrix M_0.

- If $\mathbf{b} \in R(B)$, then (1) has a minimizer, $\widetilde{\mathbf{x}}$, for which $f(\mathbf{x}) = 0$
- If $\mathbf{b} \notin R(B)$, then (1) is solved so that at each step $\|\mathbf{r}^{(k)}\|_0^2$ is minimized, which gives the name of the method.

Derivation of the GCG method (cont)

The minimization takes place on a subspace, V, spanned by a number (s) of search directions $\{\mathbf{v}^{(j)}\}$, such that $B\mathbf{v}^{(j)}$ are linearly independent, i.e.,

$$(B\mathbf{v}^{(i)}, B\mathbf{v}^{(j)}) = 0$$

The parameter s is the max number of search directions to be used when updating the current solution $\mathbf{x}^{(k)}$

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \sum_{k-s_k}^{k-1} \alpha_j^{(k)} \mathbf{v}^{(j)} \quad (2)$$

$s_k = \min(s_{k-1} + 1, s), 1 \le s_k \le k$

Derivation of the GCG method (cont)

Repeat: $\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \sum_{k-s_k}^{k-1} \alpha_j^{(k)} \mathbf{v}^{(j)}$

Then, the corresponding residual can be expressed as

$\mathbf{r}^{(k)} = B\mathbf{x}^{(k)} - \mathbf{b} = B\mathbf{x}^{(k-1)} - \mathbf{b} + \sum_{k-s_k}^{k-1} \alpha_j^{(k)} B\mathbf{v}^{(j)}$

$$\mathbf{r}^{(k)} = \mathbf{r}^{(k-1)} + \sum_{j=k-s_k}^{k-1} \alpha_j^{(k)} B\mathbf{v}^{(j)}.$$

Now we are in a position to choose the coefficients $\alpha_j^{(k)}$ ($s_k + 1$ of them) such that $f(\mathbf{x})$ is minimized.
The necessary condition for that is to impose

$$\frac{\partial f}{\partial \alpha_j^{(k)}} = 0.$$

Derivation of the GCG method (cont)

Recall, $f(\mathbf{x}) = \|\mathbf{r}^{(k)}\|_0^2 = \|\mathbf{r}^{(k-1)} + \sum_{k-s_k}^{k-1} \alpha_j^{(k)} B\mathbf{v}^{(j)}\|_0^2$

$$\frac{\partial f}{\partial \alpha_j^{(k)}} = \frac{\partial}{\partial \alpha_j^{(k)}} \left(\mathbf{r}^{(k-1)} + \sum_{k-s_k}^{k-1} \alpha_j^{(k)} B\mathbf{v}^{(j)}, \mathbf{r}^{(k-1)} + \sum_{k-s_k}^{k-1} \alpha_j^{(k)} B\mathbf{v}^{(j)} \right) = 0$$

which latter is equivalent to the following orthogonality condition

$$(\mathbf{r}^{(k)}, B\mathbf{v}^{(j)}) = 0, \ \forall j = k - s_k, \cdots, k-1$$

In other words,

$$\sum_{k-s_k}^{k-1} \alpha_j^{(k)} (B\mathbf{v}^{(j)}, B\mathbf{v}^{i})_0 = -(\mathbf{r}^{(k-1)}, B\mathbf{v}^{i})_0, \ i = 1, \cdots, s_k \qquad (3)$$

Derivation of the GCG method (cont)

Repeat: $\sum_{k-s}^{k-1} \alpha_j^{(k)} (B\mathbf{v}^{(j)}, B\mathbf{v}^{i})_0 = -(\mathbf{r}^{(k-1)}, B\mathbf{v}^{i})_0, \ i = 1, \cdots, s_k$

Thus, $\alpha_j^{(k)}$ are solutions of the system of equations

$$\Lambda^{(k)} \underline{\alpha}^{(k)} = \underline{\gamma}^{(k)} \qquad (4)$$

Derivation of the GCG method (cont)

Observations regarding the above system
$\Lambda^{(k)} = [(B\mathbf{v}^{k+1-j}, B\mathbf{v}^{k+1-i})], 1 \leq i, j \leq s_k + 1$ and $(\alpha^{(k)})_j = \alpha_{k+1-j}^{(k)}$:

- $\Lambda^{(j)}$ is symmetric and positive definite
- If the vectors $B\mathbf{v}^{(j)}$ are linearly independent, then Λ is nonsingular.
- The vector $\underline{\gamma}^{(k)}$ is of the form: $[0, \cdots, 0, -(\mathbf{r}^{(k-1)}, B\mathbf{v}^{k-1})_0]^T$
- The transition from Λ_{k-1} to $\Lambda^{(k)}$ means to augment Λ_{k-1} with one row and one column.

At stage k we have $k - s_k$ search directions and after solving (4) we can update $\mathbf{x}^{(k+1)}$ and eventually enlarge the search space with a new vector $\mathbf{v}^{k+1}$.

Derivation of the GCG method (cont)

The search directions

- can be chosen quite freely;
- special choice no.1: $\mathbf{v}^k = -\mathbf{r}^{(k)} + \sum_{k-s_k}^{k-1} \beta_j^{(k)} \mathbf{v}^{(j)}$
- special choice no.2: $\mathbf{v}^k = B\mathbf{v}^{k-1} + \sum_{k-s_k}^{k-1} \beta_j^{(k)} \mathbf{v}^{(j)}$

The coefficients β are frequently determined by a conjugate orthogonality condition

$$(B\mathbf{v}^{i}, B\mathbf{v}^{j})_1 = 0, k - s_k \leq i, j \leq k-1$$

OBS! $(\cdot, \cdot)_1$ can be another inner product $(\mathbf{u}, \mathbf{v})_1 = (\mathbf{u}, M_1 \mathbf{v})$ for some other symmetric positive definite matrix M_1.
The relation to determine $\beta_j^{(k)}$ becomes <back>

$$\beta_j^{(k)} = \frac{(B\mathbf{r}^{(k)}, B\mathbf{v}^{i})_1}{(B\mathbf{v}^{j}, B\mathbf{v}^{i})_1}$$

Possible breakdown of GCG:

ONLY if $\Lambda^{(k)}$ becomes singular!

For a nonsingular matrix B, $\Lambda^{(k)}$ becomes singular only if the vectors $\mathbf{v}^i$ become linearly dependent.

Since $\mathbf{v}^k = -\mathbf{r}^{(k)} + \sum_{k-s_k}^{k-1} \beta_j^{(k)} \mathbf{v}^{(j)}$, for the vectors to be linearly dependent means $\mathbf{r}^{(k-1)} = \mathbf{0}$, i.e., the solution has already been found.

No breakdowns!

Stagnation:

If after solving the system $\Lambda^{(k)}\alpha^{(k)} = \gamma^{(k)}$ the computed coefficients $\alpha^{(k)} = 0$, which latter is possible if $(\mathbf{r}^{(k-1)}, B\mathbf{v}^{k-1})_0 = 0$, then $\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)}$, i.e., no update occurs, the situation is referred to as *stagnation*.

If this happens, a a new search direction $\mathbf{v}^{k-1}$ has to be found.

Derivation of the GCG method (cont)

Theorem: If $\Lambda^{(j)}, j = 0, 1, \cdots, k$ is nonsingular, then there holds

(1) $(\mathbf{r}^{(k+1)}, B\mathbf{v}^i)_0 = 0$ for $k - s_k \le i \le k$

(2) $(\mathbf{r}^{(k+1)}, B\mathbf{r}^i)_0 = 0$ for $s_{i-1} + k - s_k + 1 \le i \le k$

(3) $(\mathbf{r}^{(k+1)}, B\mathbf{r}^i)_0 = 0$ for $0 \le i \le k-1$, (for the full recursion, $s_j = j, j = 0, 2, \cdots$)

(4) If $\mathbf{v}^k$ is computed from special recursions 1 or 2, then

$$(\mathbf{r}^{(k)}, B\mathbf{v}^k)_0 = -(\mathbf{r}^{(k)}, B\mathbf{r}^{(k)})_0$$

(5) If $M_0B + B^TM_0$ is positive definite, then $\Lambda^{(k)}$ is nonsingular, and thus $\mathbf{r}^{(k)} \ne \mathbf{0}$.

The GCG method (cont)

The method defined by

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \sum_{k-s_k}^{k-1} \alpha_j^{(k)} \mathbf{v}^{(j)}$$

$$\alpha_j^{(k)} \text{ from } \sum_{k-s_k}^{k-1} \alpha_j^{(k)} (B\mathbf{v}^{(j)}, B\mathbf{v}^i)_0 = -(\mathbf{r}^{(k-1)}, B\mathbf{v}^i)_0$$

is referred to as GCG-MR(s) (minimal residuals)

Convergence of the GCG method:

Theorem: Consider GCG-MR(s).
- Denote $W_{k,t} = \{B\mathbf{v}^{k-t}, \cdots, B\mathbf{v}^{k-s_k}\}$, $1 \le t \le s_k$
- Let B, $\mathbf{b}$ and $W_{k,t}$ be real.

It there is no breakdown, i.e., $\Lambda^{(k)}$ is nonsingular, then the following holds:

(a) $\alpha_{k-1}^{(k)} = \frac{det(\Lambda_0^{(k)})}{det(\Lambda^{(k)})}(\mathbf{r}^{(k-1)}, B\mathbf{r}^{k-1})_0$,
where $\Lambda_0^{(k)}$ is the first principal minor of $\Lambda^{(k)}$.
If $M_0B + B^TM_0$ is p.d., and $\mathbf{r}^{(k-1)} \neq \mathbf{0}$, then $\alpha_{k-1}^{(k)} > 0$.

(b) The method converges monotonically, i.e., $f(\mathbf{x}^{(k+1)} < f(\mathbf{x}^{(k)})$ as long as $(\mathbf{r}^{(k)}, B\mathbf{r}^k)_0 \neq 0$.

(c) The rate of convergence is defined by

$$(\mathbf{r}^{(k+1)}, \mathbf{r}^{(k+1)})_0 = (\mathbf{r}^{(k)}, \mathbf{r}^{(k)})_0 - \frac{det(\Lambda_0^{(k)})}{det(\Lambda^{(k)})}(\mathbf{r}^{(k)}, B\mathbf{r}^{(k)})_0^2.$$

Convergence of the GCG method (cont):

Theorem (cont):

(c) If $s_k \ge 1$, then

$$\begin{aligned}(\mathbf{r}^{(k+1)}, \mathbf{r}^{(k+1)})_0 &= (\mathbf{r}^{(k)}, \mathbf{r}^{(k)})_0 - \frac{(\mathbf{r}^{(k)}, B\mathbf{r}^{(k)})_0^2}{\min\limits_{\mathbf{g}\in W_{k-1}} \|B\mathbf{r}^{(k)} - \mathbf{g}\|_0^2} \\ &\le (1-\xi)(\mathbf{r}^{(k)}, \mathbf{r}^{(k)})_0\end{aligned}$$

where $\xi = \lambda_{max}(\widetilde{B} + \widetilde{B}^T)\lambda_{min}(\widetilde{B} + \widetilde{B}^T)^{-1}$ and $\widetilde{B} = M_0^{1/2}BM_0^{-1/2}$.

Proof: (b): From $(\mathbf{r}^{(k)}, B\mathbf{v}^{(j)})_0 = 0$, $k - s_k \le j \le k$ we get

$$\begin{aligned}(\mathbf{r}^{(k+1)}, \mathbf{r}^{(k+1)})_0 &= (\mathbf{r}^{(k+1)}, \mathbf{r}^{(k)} + \sum \alpha_j^{(k)} B\mathbf{v}^{(j)})_0 \\ &= (\mathbf{r}^{(k+1)}, \mathbf{r}^{(k)})_0 + \alpha_k^{(k)}(B\mathbf{v}^k, \mathbf{r}^{(k)})_0 \\ &= (\mathbf{r}^{(k+1)}, \mathbf{r}^{(k)})_0 + \alpha_k^{(k)}(B\mathbf{r}^{(k)}, \mathbf{r}^{(k)})_0\end{aligned}$$

GCG - final termination property:

Consider now the full (untruncated) version of GCG: $s_k = k$.

Let $\mathbf{v}^0 = -\mathbf{r}^{(0)}$. Then, since $\mathbf{v}^k = -\mathbf{r}^{(k)} + \sum\limits_{k-s_k}^{k-1} \beta_j^{(k)}\mathbf{v}^{(j)}$, then

$\mathbf{v}^k \in V^k(\mathbf{v}^0, B) = span\{\mathbf{r}^{(0)}, B\mathbf{r}^{(0)}, \cdots, B^{k-1}\mathbf{r}^{(0)}\}$

$\Longrightarrow \quad \mathbf{v}^k = (I + P_{k-1}(B))\mathbf{r}^{(0)}$ for some polynomial of degree $k-1$.

$\Longrightarrow \quad f(\mathbf{x}^k) = \frac{1}{2}(\mathbf{r}^{(k)}, \mathbf{r}^{(k)}) = \frac{1}{2}\|(I + P_{k-1}(B))\mathbf{r}^{(0)}\|_0^2$

Note: $P_k(B)\mathbf{r}^{(0)}$ can be considered as an approximation of $\mathbf{r}^{(k)}$

$\Longrightarrow \quad (\mathbf{r}^{(k)}, \mathbf{r}^{(k)}) = \min\limits_{P_k \in \Pi_k^0} \|(I + P_{k-1}(B))\mathbf{r}^{(0)}\|_0^2.$

GCG - final termination property (cont):

Theorem: (Use Hamilton-Kayley's theorem)
Unless stagnation, there exists a minimal degree polynomial of B, $\widetilde{P}_m(B)$ of degree m such that $m \le n$, where n is the size of the matrix B
and the method will automatically stop after at most n iterations.

In case of ν distinct eigenvalues of B, then $m \le \nu$.

Special forms of the GCG method:

We have in hand two parameters to choose:

the two scalar products $(\cdot,\cdot)_0$ and $(\cdot,\cdot)_1$, or, respectively, the two matrices $\boxed{M_0, M_1}$.

Case 1: $(\cdot,\cdot)_0 = (\cdot,\cdot)_1$ and $M_0 = M_1 = I_n$

Let $\mathbf{v}^k = -\mathbf{r}^{(k)} + \sum_{k-s_k}^{k-1} \beta_j^{(k)} \mathbf{v}^{(j)}$.

The vectors $\mathbf{v}^k$ are mutually orthogonal and since $(B\mathbf{v}^k, B\mathbf{v}^j)_0 = (B\mathbf{v}^k, B\mathbf{v}^j)_1 = 0$, then $\Lambda^{(k)}$ becomes diagonal.

Special forms of the GCG method: $M_0 = M_1 = I_n$

Algorithm:[GCG–LS(1)]

Given: $\mathbf{x}^{(0)}, \mathbf{r}^{(0)} = B\mathbf{x}^{(0)} - \mathbf{b}, \mathbf{v}^0 = -\mathbf{r}^{(0)}$

Compute $B\mathbf{r}^{(0)} = -B\mathbf{v}^k$ and set $k = 1$

Loop over k

$\alpha_k = (B\mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)})/(B\mathbf{v}^{k-1}, B\mathbf{v}^{k-1})$

$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \alpha_k \mathbf{v}^{k-1}$

$\mathbf{r}^{(k)} = \mathbf{r}^{(k-1)} + \alpha_k B\mathbf{v}^{k-1}$

Check if $(\mathbf{r}^{(k)}, \mathbf{r}^{(k)}) < \varepsilon$, stop if 'yes'

Compute $B\mathbf{r}^{(k)}$

$\beta_j^k = (B\mathbf{r}^{(k)}, B\mathbf{v}^j)/(B\mathbf{v}^j, B\mathbf{v}^j),\ j = k - s_k, \cdots, k-1$

$\mathbf{v}^k = -\mathbf{r}^{(k)} + \sum_{k-s_k}^{k-1} \beta_j^k \mathbf{v}^j)$

$B\mathbf{v}^k = -B\mathbf{r}^{(k)} + \sum_{k-s_k}^{k-1} \beta_j^k B\mathbf{v}^j$

End

(ORTHOMIN, Vinsome, 1976)

GCG - version $M_0 = M_1 = I_n$:

Memory requirements

$\mathbf{v}^j$	$B\mathbf{v}^j$	$\mathbf{r}^{(k)}$	$\mathbf{x}^{(k)}$	$B\mathbf{r}^{(k)}$
k	k	1	1	1

$2k + 3$ vectors

Computational complexityů

$s_k + 1$	inner products
$2s_k$	linked triads
1	solve with C (remember: $B = C^{-1}A$)
1	multiplication with A

This version computes the minimal pseudoresidual solution, i.e., computes $\mathbf{x}^{(k)}$, such that $(\mathbf{r}^{(k)}, \mathbf{r}^{(k)})_0 \to min$, where $\mathbf{r}^{(k)} = B\mathbf{x}^{(k)}$

GCG - other versions:

Case 2: $M_0 = M_1 = C^TC$ – GCG-LS(s) minimizes the true residual on the cost of multiplications with A and A^T

Case 3: $M_0 = C^TC,\ M_1 = (BB^T)^{-1}$ – minimizes the true residual as well but does not need an extra multiplication with A^T, however we solve a small system of equations at each iteration.

Case 4: $\cdots$

Automatic truncation of the GCG method

Automatic truncation means that in the full version of the method, in the recursion

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \sum_{k-s_k}^{k-1} \alpha_j^{(k)} \mathbf{v}^{(j)}$$

some coefficients will become zero and we will work with less search directions. This holds for certain classes of matrices.

Theorem: Let

(1) $M_0 = M_1$ be Hermitian positive definite,

(2) B be M_0-normal with respect to $\mathbf{r}^{(0)}$ of M_0-normal degree $m = m(B, M_0, \mathbf{r}^{(0)})$

(3) $M_0 B + B^* M_0$ be positive definite.

Then GCG-LS(s) is identical to the full version if anf only if $s = m$.

GCG-MR: Matlab implementation

```
function [it,x]=gcgmr(A,rhs,x,max_vec,max_iter,eps,..
           absrel,nonzero_guess)
Hsub=[]; rnorm=1e13; it=0;
r = A*x-rhs;
h = my_favourite_prec(r); h = -h;  InitRes=sqrt(r'*r);
if absrel=='rel',  eps = eps_gcgmr*InitRes; end
    while (rnorm>eps)&(it<max_iter),
        it = it + 1;     ThisPos = mod(it-1, max_vec) + 1;
        d(:,ThisPos)=h;  Ad(:,ThisPos)=A*h;
        j0 = it - max_vec + 1; if it<=max_vec, j0 = 1; end
        [tau,Hsub,flagH]=solveH(r,Ad,Hsub,Ad(:,ThisPos),j0);
        for j=j0:it,
            ThisPos = mod(j-1,max_vec) + 1;
            x = x + tau(j-j0+1)*d(:,ThisPos);
            r = r + tau(j-j0+1)*Ad(:,ThisPos);
        end
        rnorm=sqrt(r'*r);
        h = my_favourite_prec(r);  h = -h;
    end
```


GCG-LS: MATLAB implementation

```
function [it,x]=gcgls(A,rhs,x,max_vec,eps)
rnorm=1e13; it=0; r=A*x-rhs;
h=my_favourite_prec(r); h=-h;
InitRes=sqrt(r'*r);
eps = eps*InitRes;
    while (rnorm>ceps),
        it = it + 1;
        ThisPos = mod(it-1, max_vec) + 1;
        NextPos = mod(it, max_vec) + 1;
        d=h; g=A*h; Gamma = d'*g; tau = -r'*d/Gamma;
        x = x + tau*d; r = r + tau*g;
        rnorm=sqrt(r'*r);
        h=my_favourite_prec(r); h=-h;
        for j=j0:it
            ThisPos = mod(j-1,max_vec) + 1;
            beta = h'*g; h    = h - beta*d;
        end
    end
```