Krylov subspace methods

Short Recurrence Methods for Non-Hermitian Problems

Krylov subspace methods

GMRES optimal in iterations but expensive in time and memory if many iterations requires. Main cost is keeping all vectors and complete orthogonalization.

Can we devise an optimal method with a short recurrence?

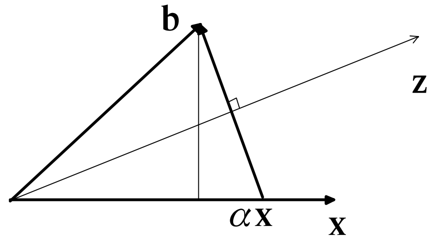
No, unless (non-Hermitian) matrix very special.

(Faber&Manteuffel result)

However, we can construct short recurrence methods that are very good in most cases.

Oblique Projection

Find approximation to b in $\mathrm{range}(x)$ such that $b - \alpha x \perp z$.



This gives $z^H(b - \alpha x) = 0 \Rightarrow \alpha = \frac{z^H b}{z^H x}$.

If $z^H x = 0$ no solution exists.

The quality of the approximation from an oblique projection depends on the angle between the search space, $\mathrm{range}(x)$, and the space that defines the projection, $\mathrm{range}(z)$.

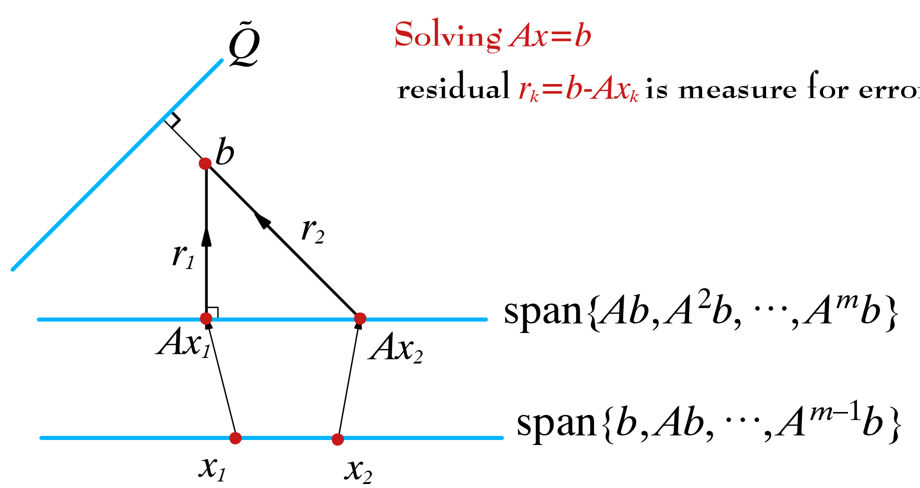
Krylov subspace methods

Two types of Krylov subspace methods (general matrices):

1. Orthogonal projection methods: optimal, but expensive (GMRES).
2. Oblique projection methods: cheap, but often converge poorly.
   a. economize on optimal methods (restart or truncate)
   b. create different space (BiCG): $\mathrm{span}\{\tilde{b}, A^*\tilde{b}, \cdots, (A^*)^{m-1}\tilde{b}\}$

Solving $Ax=b$

residual $r_k=b-Ax_k$ is measure for error



Oblique projection

In multiple dimensions the quality of the approximation from an oblique projection depends on *the canonical or principle angles* between search space and space that defines projection. The residual from an oblique projection may be much larger than the optimal residual.

Result may be very poor if one or more angles near $\pi/2$.

Linear system: $Ax = b$,
$x \in \text{range}(V_m)$ and $r = b - AV_m y \perp \text{range}(\tilde{V}_m)$

$$\tilde{V}_m^H(b - AV_m y) = 0 \quad \Leftrightarrow \quad \tilde{V}_m^H A V_m y = \tilde{V}_m^H b$$

Solve $m \times m$ system (small).

Computation of $\tilde{V}_m^H A V_m y$, $\tilde{V}_m$, and V_m using short recurrences requires special choices for $\tilde{V}_m$.

Short recurrence methods: BiCG

In order to use a short recurrence we need to make special choices.
Given arbitrary v_1 and $\tilde{v}_1$ we generate two Krylov sequences:

$$v_i = \left(Av_{i-1} - \textstyle\sum_{j<i} \alpha_j v_j\right)/\|.\| \in K^i(A, v_1)$$
$$\tilde{v}_i = \left(A^H\tilde{v}_{i-1} - \textstyle\sum_{j<i} \alpha_j \tilde{v}_j\right)/\|.\| \in K^i(A^H, \tilde{v}_1)$$

Neither the vectors v_i nor the vectors $\tilde{v}_i$ are orthogonal.
However, $v_i^H\tilde{v}_j = 0$ for $i \neq j$ and $v_i^H\tilde{v}_i \neq 0$ by choice of α_i and $\tilde{\alpha}_i$.

We now have the following orthogonality results (analogous to CG):

$Av_i \perp \tilde{v}_j$ for $j < i - 1$ since $(Av_i)^H\tilde{v}_j = v_i^H A^H \tilde{v}_j = v_i^H\left(\sum_{k=1}^{j+1} \gamma_k \tilde{v}_k\right)$, and
$A^H\tilde{v}_i \perp v_j$ for $j < i - 1$.

So we can generate two mutually orthogonal (≡ biorthogonal) sequences of vectors using short recurrences.

Note that in this case $\tilde{V}_i^H A V_i$ is tridiagonal.

BiConjugate Gradient Method (3-term rec.)

Solve $Ax = b$; choose $x_0 \to r_0 = b - Ax_0$; choose $\tilde{r}_0$;
Iterate:

$$\delta_m = \tilde{r}_m^* r_m, \quad \alpha_m = \tilde{r}_m^* A r_m / \delta_m, \quad \beta_{m-1} = \gamma_{m-1} \frac{\delta_m}{\delta_{m-1}}, \quad \gamma_m = -\alpha_m - \beta_{m-1}.$$

$$r_{m+1} = \gamma_m^{-1}(Ar_m - \alpha_m r_m - \beta_{m-1} r_{m-1}); \qquad AR_m = R_{m+1} \underline{T_m};$$
$$\tilde{r}_{m+1} = \bar{\gamma}_m^{-1}(A^* \tilde{r}_m - \bar{\alpha}_m \tilde{r}_m - \bar{\beta}_{m-1} \tilde{r}_{m-1}); \qquad A^* \tilde{R}_m = \tilde{R}_{m+1} \underline{\bar{T}_m};$$
$$x_{m+1} = -\frac{\alpha_m}{\gamma_m} x_m - \frac{\beta_{m-1}}{\gamma_m} x_{m-1} - \gamma_m^{-1} r_m;$$

$$\tilde{R}_m^* R_m = \Delta_m = diag(\delta_0, \delta_1, \ldots, \delta_{m-1});$$
$$x_{m+1} = x_0 + R_m y \;\to\; r_{m+1} = r_0 - R_{m+1} \underline{T_m} y; \; r_{m+1} \perp \tilde{R}_m;$$
$$\tilde{R}_m^*(r_0 - AR_m y) = 0 \;\Rightarrow\; \delta_0 e_1 - \Delta_n T_n y = 0 \;\Rightarrow\; y = T_n^{-1} e_1;$$
$$r_{m+1} = R_{m+1}(e_1 - \underline{T}_m T_m^{-1} e_1);$$

BiConjugate Gradient Method

Using again an implicit decomposition of the tridiagonal matrix $T_k = L_k D_k U_k$ we get a coupled two term recurrence that allows us to discard old vectors.

$x_0 \to r_0 = b - Ax_0; p_0 = r_0;$ choose $\tilde{r}_0 : \tilde{r}_0^H r_0 \neq 0$ and $\tilde{p}_0 = \tilde{r}_0$.

For $k = 1, 2, \ldots$

$$\alpha_{k-1} = \frac{\tilde{r}_{k-1}^H r_{k-1}}{\tilde{p}_{k-1}^H A p_{k-1}}; \qquad x_k = x_{k-1} + \alpha_{k-1} p_{k-1};$$

$$r_k = r_{k-1} - \alpha_{k-1} A p_{k-1}; \qquad \tilde{r}_k = \tilde{r}_{k-1} - \bar{\alpha}_{k-1} A^H \tilde{p}_{k-1};$$

$$\beta_{k-1} = \frac{\tilde{r}_k^H r_k}{\tilde{r}_{k-1}^H r_{k-1}};$$

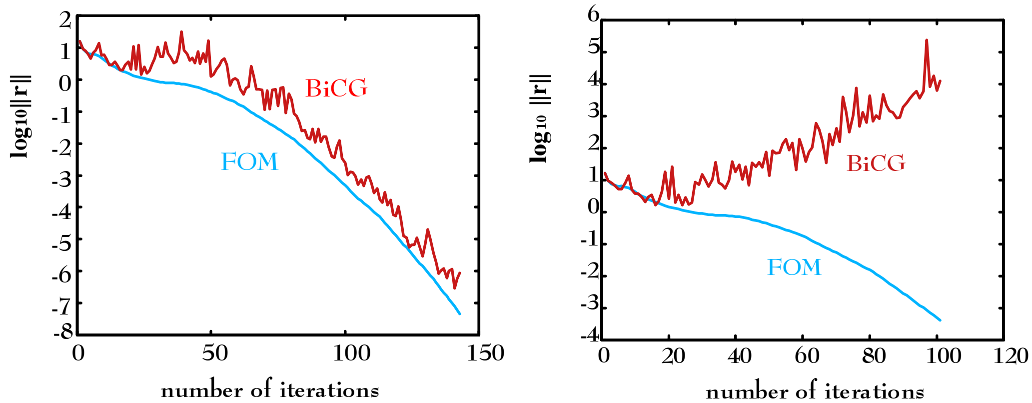
$$p_k = r_k + \beta_{k-1} p_k; \qquad \tilde{p}_k = \tilde{r}_k + \bar{\beta}_{k-1} \tilde{p}_k;$$

End

Drawbacks of BiCG?

Convergence

Some of the most popular methods today are derived from BiCG: CGS, BiCGstab, TFQMR, QMR, ...



Short recurrence methods: BiCG

1. In general, method does not satisfy any (strict) minimization property and hence may converge (very) erratically. (In practice the method often converges surprisingly well -- this is not a drawback--).

2. Two matvec.s per iteration, only one extends search space.

3. Need A^H which may be more expensive to work with or may not even be available (matrix-free implementations)

4. Breaks down without finding solution:

   a. $\tilde{r}_k = 0$, which means $K^k(A^H, \tilde{r}_k)$ is invariant

   b. $\tilde{r}_k^H r_k = 0$, where $\tilde{r}_k \neq 0$ and $r_k \neq 0$.

   c. T_k singular (only for coupled 2-term recurrence)

Short recurrence methods: QMR

If BiCG fails because T_k is singular, we may solve the system

$\|r_0\| e_1 - \underline{T}_k^{(k+1)\times k}$ in least squares sense:

$$r_k = r_0 - AV_k y_k = V_{k+1}\left(\|r_0\| e_1 - \underline{T}_k y_k\right)$$

Minimize $\left\| \|r_0\| e_1 - \underline{T}_k y_k \right\|_2$ as in GMRES (or MINRES).

Major difference: V_{k+1} is not (at all) orthogonal, hence not optimal.

This least squares system always has a solution (removes one breakdown condition).

Compared with GMRES we have $\|r_k^Q\|_2 \le \kappa(V_{k+1}) \|r_k^G\|_2$.

Unfortunately there is no bound (in general) for $\kappa(V_{k+1})$.

Other drawbacks remain. How to get rid of the A^H part.

Short recurrence methods: CGS

Cunning Plan (P. Sonneveld):

If we only need $K(A^H, \tilde{r}_0)$ for projection, then all we need are the inner products with the vectors $\tilde{r}_i$ and $\tilde{p}_i$.

Since $\tilde{r}_i = \bar{R}_i(A^H)\tilde{r}_0$, we have

$$\tilde{r}_i^H r_i = [\bar{R}_i(A^H)\tilde{r}_0]^H [R_i(A) r_0] = \tilde{r}_0^H R_i^2(A) r_0.$$

Analogously, the other inner products are prod. polynomials in A.
If we can find (easy) recurrences to represent the products of polynomials times a vector (like $R_i^2(A)r_0$), then we only need $\tilde{r}_0$ and we can discard the Krylov space generated with A^H.

Moreover we now compute approximations from Krylov space $K^{2i}(A, r_0)$ using $2 \times i$ matvec.s. So , we no longer waste matvec.s.

Finally, if $R_i(A)r_0$ small, then *typically* $R_i^2(A)r_0$ much smaller.
Unfortunately, when $R_i(A)r_0$ large ...

Lanczos product methods

Since CGS 'squares the residual' the residual may be large when $R_i(A)r_0$ not small. In fact, even when $R_i(A)r_0$ small, $R_i^2(A)r_0$ may not be.

This may lead to very irregular (nervous) convergence behavior with large peaks in the residual norm. This may ruin accuracy and sometimes convergence.

Nevertheless, for a long time CGS was the method of choice for a large class of problems.

Instead of squaring the polynomial, to avoid large peaks in CGS convergence, we may multiply by another polynomial:

$$r_k = M_k(A)R_k(A)r_0,$$

where, M_k is used to improve convergence and avoid peaks.
For example, one-step minimum residual polynomial. This yields the BiCGStab (Stabilized) method. Currently among the most popular Krylov methods with GMRES and TFQMR (QMR squared).

Truncation for optimal methods

The Generalized Minimum Residual Method: GMRES

GMRES:
select x_0; $r_0 = b - Ax_0$; $k = 0$;
$\$ = 2r_0 2_2$; $v_1 = \$^{-1} r_0$;
do
  $k = k+1$;
  $v_{k+1} = Av_k$;
  do $i = 1,k$
    $h_{i,k} = v_i^* v_{k+1}$;
    $v_{k+1} = v_{k+1} - h_{i,k} v_i$;
  enddo
  $h_{k+1,k} = 2v_{k+1}2_2$; $v_{k+1} = v_{k+1}/h_{k+1,k}$;

  update QR-decomp. $\underline{H}_k = \underline{Q}_k R_k$
  if $(2(I-\underline{QQ}_k^*)\$e_1 2_2$ # *eps*) exit
enddo
$y_k = R_k^{-1}\underline{Q}_k^* \$e_1$; $x_k = x_0 + V_k y_k$;
$r_k = r_0 - V_{k+1}\underline{H}_k y_k$;

Arnoldi:
$AV_m = V_{m+1}\underline{H}_m$;
$V_{m+1}^* V_{m+1} = I_{m+1}$; $v_1 = r_0/\$$;
$x_m = x_0 + V_m y_m$; $r_m = r_0 - V_{m+1}\underline{H}_m y_m$

range$(V_m) = K^m(A,r_0)$;
range$(AV_m) = K^m(A,Ar_0)$;

QR-decomp. $\underline{H}_m = \underline{Q}_m R_m$ Y
QR-decomp. $AV_m = V_m \underline{Q}_m$ ŒR_m;

$r_m = V_{m+1}(\$e_1 - \underline{Q}_m R_m y_m)$;
min $2r_m 2$6 $y_m = R_m^{-1}\underline{Q}_m^* \$e_1$;
(Least Squares)

Drawbacks:
Orthogonalization cost: $O(Nm^2)$; Memory requirements: $O(Nm)$
Remedy: Restart