Accelerating augmented and deflated Krylov space methods for convection-diffusion problems

Giuseppe Pitton, Luca Heltai

SISSA

July 23, 2018

Krylov space methods

Solve for $\mathbf{x}$:

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

- $\mathbf{A}$ may be too large for direct methods;
  $\hookrightarrow$ set up a sequence of approximations $\{\mathbf{x}_k\}$ of $\mathbf{x}$, so that $\mathbf{x}_k$ is a good approximation for k reasonably small ($\lesssim 100$).
- How to measure how good an approximation is?
  $\hookrightarrow$ 2-norm of the residual $\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$.
- (Not always a good indication.)

Krylov space methods

Construction of the tentative solution $\mathbf{x}_k$:

- search space: $\mathbf{x}_k \in \mathbf{x}_0 + \mathcal{S}_k$;
- constraint space: $\mathbf{r}_k \perp \mathcal{C}_k$.

Let $\{\mathbf{s}_i\}_{i=1}^k$ o.n. basis for $\mathcal{S}_k$ and $\{\mathbf{c}_j\}_{j=1}^k$ o.n. basis for $\mathcal{C}_k$.
Let $\mathbf{S}_k = [\mathbf{s}_1\, \mathbf{s}_2\, \ \ldots\ \mathbf{s}_k] \in \mathbb{C}^{n\times k}$
and $\mathbf{C}_k = [\mathbf{c}_1\, \mathbf{c}_2\, \ \ldots\ \mathbf{c}_k] \in \mathbb{C}^{n\times k}$
unitary matrices.
Then, write $\mathbf{x}_k = \mathbf{x}_0 + \mathbf{S}_k\mathbf{t}_k$ for some $\mathbf{t}_k \in \mathbb{C}^k$.
$\mathbf{r}_k \perp \mathcal{C}_k \qquad \Rightarrow \qquad \mathbf{C}_k^H\mathbf{r}_k = 0.$

see: [Liesen and Strakoš, 2012, Saad, 2003, Greenbaum, 1997]

Krylov space methods

$$\begin{cases} \mathbf{x}_k = \mathbf{x}_0 + \mathbf{S}_k \mathbf{t}_k \\ \mathbf{C}_k^H \mathbf{r}_k = 0 \end{cases} \quad \Rightarrow \quad \mathbf{C}_k^H \mathbf{A} \mathbf{S}_k \mathbf{t}_k = \mathbf{C}_k^H \mathbf{r}_0.$$

$\mathbf{C}_k^H \mathbf{A} \mathbf{S}_k \in \mathbb{C}^{k \times k}$: if $k \ll n$, much smaller operation count.
Approximate solution:

$$\mathbf{x}_k = \mathbf{x}_0 + \mathbf{S}_k (\mathbf{C}_k^H \mathbf{A} \mathbf{S}_k)^{-1} \mathbf{C}_k^H \mathbf{r}_0.$$

see: [Liesen and Strakoš, 2012, Saad and Schultz, 1986, Greenbaum, 1997]

GMRES

- How to construct $\mathbf{S}_k$ and $\mathbf{C}_k$?
  $\hookrightarrow$ Arnoldi algorithm:

$$\mathcal{S}_k = \operatorname{Im} \mathbf{S}_k = \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0) = \operatorname{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0\}$$

- taking $\mathcal{C}_k = \mathbf{A}\mathcal{S}_k$ ensures $\exists!$ solution $\mathbf{t}_k$;
- optimality property: $\mathbf{x}_k$ such that

$$\mathbf{r}_k = \arg \min_{\mathbf{y} \in \mathbf{x}_0 + \mathcal{S}_k} \|\mathbf{b} - \mathbf{A}\mathbf{y}\|_2$$

Preconditioning

- work and storage of Arnoldi algorithm increase linearly with k;
- if $k \gtrsim 50$, iterations may become too costly;
  $\hookrightarrow$ restart: take $\mathbf{x}_k$ as $\mathbf{x}_0$ and set:

$$\mathcal{S}_k = \mathcal{K}_k(\mathbf{A}, \mathbf{r}_m) = \mathcal{K}_k(\mathbf{A}, p_{m-1}(\mathbf{A})\mathbf{r}_0) \not\angle \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0);$$

- restart may severely hinder convergence rate (*GMRES stagnation*);
  $\hookrightarrow$ preconditioning: solve instead

$$\mathbf{MAx} = \mathbf{Mb}$$

  induces a new Krylov space

$$\mathcal{S}_k = \mathcal{K}_k(\mathbf{MA}, \mathbf{Mr}_0).$$

Augmented and deflated methods

Let $\mathcal{V}_s \subset \mathbb{C}^n$ containing some “relevant information”.

(i) $\|\mathbf{\Pi}_{\mathcal{V}_s}\mathbf{x}\|$ “large”;

(ii) $\mathcal{V}_s$ contains some eigenvectors of $\mathbf{A}$ with small eigenvalues;

(iii) ... (usually heuristics)

- Augmentation (strategy (i), (ii), (iii))

$$\mathcal{S}_k = \mathcal{K}_k(\mathbf{A}, \mathbf{r}_0) + \mathcal{V}_s$$

$$\mathcal{C}_k = \mathcal{K}_k(\mathbf{A}, \mathbf{A}\mathbf{r}_0) + \mathbf{A}\mathcal{V}_s$$

- Deflation preconditioners (strategy (ii))

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

$$\mathbf{M}_D = \mathbf{I} - \mathbf{A}\mathbf{V}_s\mathbf{E}_s^{-1}\mathbf{V}_s^H + \lambda^*\mathbf{V}_s\mathbf{E}_s^{-1}\mathbf{V}_s^H.$$

with $\mathbf{E}_s = \mathbf{V}_s^H\mathbf{A}\mathbf{V}_s$.

see: [Erlangga and Nabben, 2008, Tang et al., 2010, Nabben and Vuik, 2006]

Augmented and deflated methods

Common strategy: $\mathbf{x} = \mathbf{x}_0 + \mathbf{x}_c$

1. initial guess on subspace:

$$\mathbf{x}_0 = \mathbf{V}\mathbf{E}^{-1}\mathbf{V}^H\mathbf{b};$$

2. correction equation:

$$\mathbf{A}\mathbf{x}_c = \mathbf{r}_0 = (\mathbf{I} - \mathbf{\Pi}_{\mathcal{V},\mathbf{A}\mathcal{V}})\mathbf{b}$$

   with the Krylov space

$$\begin{aligned}\mathcal{S}_k &= \mathcal{K}_k(\mathbf{\Pi}_i\mathbf{A}, \mathbf{\Pi}_j(\mathbf{I} - \mathbf{\Pi}_{\mathcal{V},\mathbf{A}\mathcal{V}})\mathbf{b}) \\ &= \mathcal{K}_k(\mathbf{\Pi}_i\mathbf{A}, \mathbf{\Pi}_j\mathbf{r}_0)\end{aligned}$$

Augmented and deflated methods

Some projectors:

$$\mathbf{\Pi}_1 = (\mathbf{I} - \mathbf{V}\mathbf{V}^H)$$
$$\mathbf{\Pi}_2 = (\mathbf{I} - \mathbf{A}\mathbf{V}\mathbf{E}^{-1}\mathbf{V}^H)$$
$$\mathbf{\Pi}_3 = (\mathbf{I} - \mathbf{A}\mathbf{V}(\mathbf{V}^H\mathbf{A}^H\mathbf{A}\mathbf{V})^{-1}\mathbf{V}^H\mathbf{A}^H)$$

Methods:

method name	Krylov operator	Krylov initial vector
augmented orthogonal	$\mathbf{\Pi}_1\mathbf{A}$	$\mathbf{r}_0$
augmented oblique	$\mathbf{\Pi}_2\mathbf{A}$	$\mathbf{r}_0$
augmented LS	$\mathbf{\Pi}_3\mathbf{A}$	$\mathbf{r}_0$
deflated orthogonal	$\mathbf{\Pi}_1\mathbf{A}$	$\mathbf{\Pi}_1\mathbf{r}_0$
deflated oblique	$\mathbf{\Pi}_2\mathbf{A}$	$\mathbf{\Pi}_2\mathbf{r}_0$
deflated LS	$\mathbf{\Pi}_3\mathbf{A}$	$\mathbf{\Pi}_3\mathbf{r}_0$

Augmentation/deflation space

Usual choices:

(i) Ritz vectors: $(\mathbf{A} - \theta_i \mathbf{I})\mathbf{y}_i \in \mathcal{V}^\perp$

$$\mathbf{V}^H \mathbf{A} \mathbf{V} \mathbf{z}_i - \theta_i \mathbf{V}^H \mathbf{V} \mathbf{z}_i = \mathbf{0} \qquad \mathbf{y}_i = \mathbf{V} \mathbf{z}_i;$$

(ii) harmonic Ritz vectors (Ritz vectors for $\mathbf{A}^{-1}$)

$$\mathbf{V}^H \mathbf{A}^H \mathbf{A} \mathbf{V} \mathbf{z}_i - \theta_i \mathbf{V}^H \mathbf{A}^H \mathbf{V} \mathbf{z}_i = \mathbf{0};$$

(iii) angles between subspaces (GCROT [De Sturler, 1999]);

Less usual choices:

(iv) previous solutions for time-dependent problems:

$$\mathbf{A}_{n-1} \mathbf{x}_{k_{n-1}} = \mathbf{b}_{n-1} \quad \mathbf{A}_{n-2} \mathbf{x}_{k_{n-2}} = \mathbf{b}_{n-2} \quad \ldots$$

(v) as (iv), with SVD selection

Recycling algorithm

Given:

- m: number of retained previous solutions;
- $s \leqslant m$: dimension of the recycling space $\mathcal{V}_s$;
- $\ell \geqslant 1$: timesteps between SVD

Do:

1. solve $\mathbf{A}_i \mathbf{x}_i = \mathbf{b}_i$;
2. update $\mathbf{X}_m = [\mathbf{x}_{i-m}, \dots, \mathbf{x}_i]$;
3. if $(i \mod \ell) = 0$, compute $\Psi_m \Sigma_m \Phi_m^H = \mathbf{X}_m$ and update $\mathbf{V}_s = \Psi_m[1:s]$.

Code details

```
void DifferentialProblem<dim>::update_V()
{
  FullMatrix<double>  V(n_dofs, how_many);
  EM eigV(&V(0,0), n_dofs, how_many);

  EV eigsol(&solution(0), n_dofs);
  Vector<double> tmp1 (solution), tmp2(
     n_dofs);
  EV eigtmp1(&tmp(0), n_dofs);
  EV eigtmp2(&tmp2(0), n_dofs);

  for (int i=0;i<how_many-1;++i)
  {
      eigtmp2 = eigV.col(i);
      double gs_coeff =  tmp1*tmp2;
      eigtmp1 = eigtmp1 - gs_coeff*eigtmp2
         ;
  }
```

Code details

```
    double tmp1_norm = tmp1.l2_norm();
    eigtmp1 /= tmp1_norm;
    eigV.col(column) = eigtmp1;
    column = (column + 1) % RECYCLE_BASIS;

  }
```

Code details

```
void DifferentialProblem<dim>::
   setup_system()
{
  // LinearOperator<> Aop, Vop, VTop, Iop,
       system_matrix_deflated;
  preconditioner.initialize(system_matrix)
     ;
  Aop = linear_operator(system_matrix);
  Vop = linear_operator(V);
  VTop = transpose_operator(Vop);
  Iop  = identity_operator(Aop.
     reinit_range_vector);
  system_matrix_deflated = (Iop-Vop*VTop)*
     Aop;
}
```

Code details

```
int DifferentialProblem<dim>::
   solve_remove_V_space()
{
  system_matrix.vmult(residual, solution);
  residual -= system_rhs;

  solver.solve(system_matrix_deflated, y,
     residual,
           preconditioner);
  solution += y;

  constraints.distribute(solution);
  std::cout << solver_control.last_step()
     << std::endl;
  return solver_control.last_step();
}
```

Comparison

method	iterations (best case)	iterations (worst case)
$\mathcal{K}(\mathbf{A}, \mathbf{r}_0)$	37.5	42.0
$\mathcal{K}(\mathbf{\Pi}_1\mathbf{A}, \mathbf{r}_0)$	35.0	46.0
$\mathcal{K}(\mathbf{\Pi}_2\mathbf{A}, \mathbf{r}_0)$	26.0	27.8
$\mathcal{K}(\mathbf{\Pi}_3\mathbf{A}, \mathbf{r}_0)$	n.c.	n.c.
$\mathcal{K}(\mathbf{\Pi}_1\mathbf{A}, \mathbf{\Pi}_1\mathbf{r}_0)$	40.8	58.0
$\mathcal{K}(\mathbf{\Pi}_2\mathbf{A}, \mathbf{\Pi}_2\mathbf{r}_0)$	42.0	60.0
$\mathcal{K}(\mathbf{\Pi}_3\mathbf{A}, \mathbf{\Pi}_3\mathbf{r}_0)$	46.0	54.0

$$\begin{aligned}
\mathbf{\Pi}_1 &= (\mathbf{I} - \mathbf{V}\mathbf{V}^H) \\
\mathbf{\Pi}_2 &= (\mathbf{I} - \mathbf{A}\mathbf{V}\mathbf{E}^{-1}\mathbf{V}^H) \\
\mathbf{\Pi}_3 &= (\mathbf{I} - \mathbf{A}\mathbf{V}(\mathbf{V}^H\mathbf{A}^H\mathbf{A}\mathbf{V})^{-1}\mathbf{V}^H\mathbf{A}^H)
\end{aligned}$$

Cost comparison

Q_1 FEM:

method	average iteration count	solver wall time (s)
no aug	125	222
aug 20	78.4	249
aug 40	78.0	243
aug 60	77.5	268
aug 80	78.2	271

Q_{23} Legendre SEM-NI:

method	average iteration count	solver wall time (s)
no aug	53.7	191
aug 20	48.1	180
aug 40	44.8	170
aug 60	45.9	177
aug 80	46.4	175

Conclusions

- standard way: select some Arnoldi vectors at each restart (GCROT [De Sturler, 1999]);
- or: compute a few eigenvectors (even more costly);
- past solutions are already there;
- SVD selection is costly: use with a grain of salt;
- for high-order methods, surely worthwile.

Never forget: augmentation/deflation methods can not replace a preconditioner, but only complement it.

References I

De Sturler, E. (1999).
Truncation strategies for optimal krylov subspace methods.
SIAM Journal on Numerical Analysis, 36(3):864–889.

Erlangga, Y. A. and Nabben, R. (2008).
Deflation and balancing preconditioners for krylov subspace methods applied to nonsymmetric matrices.
SIAM Journal on Matrix Analysis and Applications, 30(2):684–699.

Greenbaum, A. (1997).
Iterative Methods for Solving Linear Systems.
Society for Industrial and Applied Mathematics.

Liesen, J. and Strakoš, Z. (2012).
Krylov Subspace Methods.
Oxford University Press.

References II

Nabben, R. and Vuik, C. (2006).
A comparison of deflation and the balancing preconditioner.
SIAM Journal on Scientific Computing, 27(5):1742–1759.

Saad, Y. (2003).
Iterative methods for sparse linear systems.
SIAM.

Saad, Y. and Schultz, M. H. (1986).
Gmres: a generalized minimal residual algorithm for solving nonsymmetric linear systems.
SIAM Journal on Scientific and Statistical Computing, 7(3):856–869.

References III

Tang, J. M., Maclachlan, S. P., Nabben, R., and Vuik, C. (2010).
A comparison of two level preconditioners based on multigrid and deflation.
SIAM Journal on Matrix Analysis and Applications, 31(4):1715–1739.