Parallel sparse linear solvers in the team SAGE

Jocelyne Erhel
Joint work with Désiré Nuentsa Wakam (GMRES)
and Baptiste Poirriez (PCG)

SAGE team, Inria Rennes, France

Workshop on linear solvers, organized by C2S@EXA, 23rd September 2013

Outline

Solver interface

- Interface to direct and iterative solvers: MUMPS, SuperLU_Dist, Hypre, Petsc, pArms,etc
- SLSI [Nuentsa Wakam et al 2010] available on demand
- System solver in H2OLab platform [Erhel et al 2009]
- Application to CFD problems

GMRES(m): a Krylov method

- combining Domain Decomposition and deflation

PCG: a Krylov method for SPD matrices

- combining Domain Decomposition and deflation

Preconditioned GMRES

$Ax = b, \quad A \in \mathbb{R}^{n \times n} \quad x, b \in \mathbb{R}^n \quad B = AM^{-1}$

GMRES(m): a Krylov subspace method

- [Saad and Schultz 1986, Meurant's book 1999, Saad's book 2003, Simoncini and Szyld 2007, Erhel 2011, ...]
- Fix x_0, then $r_0 = b - Ax_0$
- $\mathcal{K}_m(B, r_0) = span\{r_0, Br_0, \ldots B^{m-1} r_0\}$
- Find $x_m \in x_0 + \mathcal{K}_m(B, r_0)$ such that $\|r_m\|_2 = \|b - Bx_m\|_2 = \min_{u \in x_0 + \mathcal{K}_m(B, r_0)} \|b - Bu\|_2$

Building blocks of GMRES

- Initial step: choose x_0, compute r_0
- First step: generate an orthonormal basis $V_{m+1} = [v_0, \ldots, v_m]$ of $\mathcal{K}_{m+1}(B, r_0)$ such that

$$v_0 = r_0/\beta, \quad \beta = \|r_0\|_2, \quad BV_m = V_{m+1}\bar{H}_m$$

- Second step: approximate solution $x_m = x_0 + M^{-1} V_m y_m$

$$\Rightarrow r_m = r_0 - BV_m y_m = V_{m+1}(\beta e_1 - \bar{H}_m y_m)$$

$$\Rightarrow y_m = min_{y \in \mathbb{R}^m} \|\beta e_1 - \bar{H}_m y\|_2$$

GMRES ... practical issues

Arnoldi process

```
$v_0 = r_0/\|r_0\|_2$
for $k = 0, \ldots$ do
    $p = Bv_k$
    for $i = 1 : k$ do
        $h_{ik} = v_i^T p$
        $p = p - h_{ik} v_i$
    end for
    $h_{k+1,k} = ||p||_2$
    $v_{k+1} = p/h_{k+1,k}$
end for
```

$\Downarrow$

$$BV_m = V_{m+1}\bar{H}_m$$

Granularity issues in parallel algorithms

⇒ Communication-avoiding strategies

- Generate the basis vectors [Reichel 1990, Bai et al 1994]
- Orthogonalize the basis [De Sturler 1994, Erhel 1995, Sidje 1997]
- Improve the strategy [Hoemmen 2010, Demmel et al 2011]

Complexity issues with restarted GMRES(m)

⇒ Use deflation to recover possible loss of information

- Deflation by preconditioning [Erhel et al 1996, Burrage et al 1998, Baglama et al 1998, ...]
- Deflation by augmented basis [Morgan 1995, Morgan 2002,...]

Preconditioning issues

⇒ use multilevel methods to deal with large systems

- Schwarz preconditioning [Atenekeng Kahou et al 2007, Dufaud+Tromeur-Dervout 2010, Giraud+Haidar 2009, Smith et al's book 1996,...]
- Filtering and Schur complement [Li et al 2003, Grigori et al 2011]
- Multilevel parallelism [Nuentsa Wakam et al 2011, Giraud et al 2010, ...]

Work in the team SAGE

Combine 'communication-avoiding' GMRES ... and Deflation ... and domain decomposition preconditioners

[Nuentsa Wakam et al 2013 and to appear]

GMRES … practical issues

Arnoldi process

```
$v_0 = r_0/\|r_0\|_2$
for $k = 0, \ldots$ do
    $p = Bv_k$
    for $i = 1 : k$ do
        $h_{ik} = v_i^T p$
        $p = p - h_{ik} v_i$
    end for
    $h_{k+1,k} = ||p||_2$
    $v_{k+1} = p/h_{k+1,k}$
end for
```

⇓

$$BV_m = V_{m+1}\bar{H}_m$$

Granularity issues in parallel algorithms

⇒ Communication-avoiding strategies

- Generate the basis vectors [Reichel 1990, Bai et al 1994]
- Orthogonalize the basis [De Sturler 1994, Erhel 1995, Sidje 1997]
- Improve the strategy [Hoemmen 2010, Demmel et al 2011]

Complexity issues with restarted GMRES(m)

⇒ Use deflation to recover possible loss of information

- Deflation by preconditioning [Erhel et al 1996, Burrage et al 1998, Baglama et al 1998, ...]
- Deflation by augmented basis [Morgan 1995, Morgan 2002,...]

Preconditioning issues

⇒ use multilevel methods to deal with large systems

- Schwarz preconditioning [Atenekeng Kahou et al 2007, Dufaud+Tromeur-Dervout 2010, Giraud+Haidar 2009, Smith et al's book 1996,...]
- Filtering and Schur complement [Li et al 2003, Grigori et al 2011]
- Multilevel parallelism [Nuentsa Wakam et al 2011, Giraud et al 2010, ...]

Work in the team SAGE

Combine 'communication-avoiding' GMRES ... and Deflation ... and domain decomposition preconditioners
[Nuentsa Wakam et al 2013 and to appear]

GMRES ... practical issues

Arnoldi process

```
$v_0 = r_0 / \|r_0\|_2$
for $k = 0, \ldots$ do
    $p = Bv_k$
    for $i = 1 : k$ do
        $h_{ik} = v_i^T p$
        $p = p - h_{ik} v_i$
    end for
    $h_{k+1,k} = ||p||_2$
    $v_{k+1} = p / h_{k+1,k}$
end for
```

$\Downarrow$

$$BV_m = V_{m+1} \bar{H}_m$$

Granularity issues in parallel algorithms

$\Rightarrow$ Communication-avoiding strategies

- Generate the basis vectors [Reichel 1990, Bai et al 1994]
- Orthogonalize the basis [De Sturler 1994, Erhel 1995, Sidje 1997]
- Improve the strategy [Hoemmen 2010, Demmel et al 2011]

Complexity issues with restarted GMRES(m)

$\Rightarrow$ Use deflation to recover possible loss of information

- Deflation by preconditioning [Erhel et al 1996, Burrage et al 1998, Baglama et al 1998, ...]
- Deflation by augmented basis [Morgan 1995, Morgan 2002,...]

Preconditioning issues

$\Rightarrow$ use multilevel methods to deal with large systems

- Schwarz preconditioning [Atenekeng Kahou et al 2007, Dufaud+Tromeur-Dervout 2010, Giraud+Haidar 2009, Smith et al's book 1996,...]
- Filtering and Schur complement [Li et al 2003, Grigori et al 2011]
- Multilevel parallelism [Nuentsa Wakam et al 2011, Giraud et al 2010, ...]

Work in the team SAGE

Combine 'communication-avoiding' GMRES ... and Deflation ... and domain decomposition preconditioners
[Nuentsa Wakam et al 2013 and to appear]

GMRES ... practical issues

Arnoldi process

```
$v_0 = r_0/\|r_0\|_2$
for $k = 0, \ldots$ do
    $p = Bv_k$
    for $i = 1 : k$ do
        $h_{ik} = v_i^T p$
        $p = p - h_{ik} v_i$
    end for
    $h_{k+1,k} = ||p||_2$
    $v_{k+1} = p/h_{k+1,k}$
end for
```

$\Downarrow$

$$BV_m = V_{m+1}\bar{H}_m$$

Granularity issues in parallel algorithms

⇒ Communication-avoiding strategies

- Generate the basis vectors [Reichel 1990, Bai et al 1994]
- Orthogonalize the basis [De Sturler 1994, Erhel 1995, Sidje 1997]
- Improve the strategy [Hoemmen 2010, Demmel et al 2011]

Complexity issues with restarted GMRES(m)

⇒ Use deflation to recover possible loss of information

- Deflation by preconditioning [Erhel et al 1996, Burrage et al 1998, Baglama et al 1998, ...]
- Deflation by augmented basis [Morgan 1995, Morgan 2002,...]

Preconditioning issues

⇒ use multilevel methods to deal with large systems

- Schwarz preconditioning [Atenekeng Kahou et al 2007, Dufaud+Tromeur-Dervout 2010, Giraud+Haidar 2009, Smith et al's book 1996,...]
- Filtering and Schur complement [Li et al 2003, Grigori et al 2011]
- Multilevel parallelism [Nuentsa Wakam et al 2011, Giraud et al 2010, ...]

Work in the team SAGE

Combine 'communication-avoiding' GMRES ... and Deflation ... and domain decomposition preconditioners

[Nuentsa Wakam et al 2013 and to appear]

GMRES … practical issues

Arnoldi process

```
$v_0 = r_0/\|r_0\|_2$
for $k = 0, \ldots$ do
    $p = Bv_k$
    for $i = 1 : k$ do
        $h_{ik} = v_i^T p$
        $p = p - h_{ik} v_i$
    end for
    $h_{k+1,k} = ||p||_2$
    $v_{k+1} = p/h_{k+1,k}$
end for
```

$\Downarrow$

$BV_m = V_{m+1}\bar{H}_m$

Granularity issues in parallel algorithms

⇒ Communication-avoiding strategies

- Generate the basis vectors [Reichel 1990, Bai et al 1994]
- Orthogonalize the basis [De Sturler 1994, Erhel 1995, Sidje 1997]
- Improve the strategy [Hoemmen 2010, Demmel et al 2011]

Complexity issues with restarted GMRES(m)

⇒ Use deflation to recover possible loss of information

- Deflation by preconditioning [Erhel et al 1996, Burrage et al 1998, Baglama et al 1998, ...]
- Deflation by augmented basis [Morgan 1995, Morgan 2002,...]

Preconditioning issues

⇒ use multilevel methods to deal with large systems

- Schwarz preconditioning [Atenekeng Kahou et al 2007, Dufaud+Tromeur-Dervout 2010, Giraud+Haidar 2009, Smith et al's book 1996,...]
- Filtering and Schur complement [Li et al 2003, Grigori et al 2011]
- Multilevel parallelism [Nuentsa Wakam et al 2011, Giraud et al 2010, ...]

Work in the team SAGE

Combine 'communication-avoiding' GMRES ... and Deflation ... and domain decomposition preconditioners
[Nuentsa Wakam et al 2013 and to appear]

Communication-avoiding strategy: Newton basis

building blocks

- Initial step: run one cycle of GMRES(m) and compute shifts for the Newton basis
- First step: build a basis $K_{m+1} = [k_0, k_1, \ldots, k_m]$ of the Krylov subspace $\mathcal{K}_{m+1}(B, r_0)$ such that

$$BK_m = K_{m+1}\bar{T}_m$$

- Second step: compute an orthonormal basis of $\mathcal{K}_{m+1}(B, r_0)$
  Compute the QR factorization $K_{m+1} = V_{m+1}R_{m+1}$
  RODDEC [Sidje 1997, Erhel 1995] or TSQR [Demmel et al 2011]

$$\Rightarrow BK_m = V_{m+1}R_{m+1}\bar{T}_m \Rightarrow BV_m = V_{m+1}\underbrace{R_{m+1}\bar{T}_mR_m^{-1}}_{\bar{H}_m}$$

- Third step: approximate solution $x_m = x_0 + M^{-1}V_my_m$

$$\Rightarrow r_m = r_0 - BK_my_m = V_{m+1}(\beta e_1 - \bar{H}_my_m) \quad \text{with } \beta = \|r_0\|_2$$

$$\Rightarrow y_m = min_{y\in\mathbb{R}^m}\|\beta e_1 - \bar{H}_my\|_2$$

Communication-avoiding strategy: Newton basis

building blocks

- Initial step: run one cycle of GMRES(m) and compute shifts for the Newton basis
- First step: build a basis $K_{m+1} = [k_0, k_1, \ldots, k_m]$ of the Krylov subspace $\mathcal{K}_{m+1}(B, r_0)$ such that

$$BK_m = K_{m+1}\bar{T}_m$$

- Second step: compute an orthonormal basis of $\mathcal{K}_{m+1}(B, r_0)$
  Compute the QR factorization $K_{m+1} = V_{m+1}R_{m+1}$
  RODDEC [Sidje 1997, Erhel 1995] or TSQR [Demmel et al 2011]

$$\Rightarrow BK_m = V_{m+1}R_{m+1}\bar{T}_m \Rightarrow BV_m = V_{m+1}\underbrace{R_{m+1}\bar{T}_m R_m^{-1}}_{\bar{H}_m}$$

- Third step: approximate solution $x_m = x_0 + M^{-1}V_m y_m$

$$\Rightarrow r_m = r_0 - BK_m y_m = V_{m+1}(\beta e_1 - \bar{H}_m y_m) \quad \text{with } \beta = \|r_0\|_2$$

$$\Rightarrow y_m = min_{y\in\mathbb{R}^m}\|\beta e_1 - \bar{H}_m y\|_2$$

Communication-avoiding strategy: Newton basis

building blocks

- Initial step: run one cycle of GMRES(m) and compute shifts for the Newton basis
- First step: build a basis $K_{m+1} = [k_0, k_1, \ldots, k_m]$ of the Krylov subspace $\mathcal{K}_{m+1}(B, r_0)$ such that

$$BK_m = K_{m+1}\bar{T}_m$$

- Second step: compute an orthonormal basis of $\mathcal{K}_{m+1}(B, r_0)$
  Compute the QR factorization $K_{m+1} = V_{m+1}R_{m+1}$
  RODDEC [Sidje 1997, Erhel 1995] or TSQR [Demmel et al 2011]

$$\Rightarrow BK_m = V_{m+1}R_{m+1}\bar{T}_m \Rightarrow BV_m = V_{m+1}\underbrace{R_{m+1}\bar{T}_m R_m^{-1}}_{\bar{H}_m}$$

- Third step: approximate solution $x_m = x_0 + M^{-1}V_m y_m$

$$\Rightarrow r_m = r_0 - BK_m y_m = V_{m+1}(\beta e_1 - \bar{H}_m y_m) \quad \text{with } \beta = \|r_0\|_2$$

$$\Rightarrow y_m = min_{y \in \mathbb{R}^m} \|\beta e_1 - \bar{H}_m y\|_2$$

Communication-avoiding strategy: Newton basis

building blocks

- Initial step: run one cycle of GMRES(m) and compute shifts for the Newton basis
- First step: build a basis $K_{m+1} = [k_0, k_1, \ldots, k_m]$ of the Krylov subspace $\mathcal{K}_{m+1}(B, r_0)$ such that

$$BK_m = K_{m+1}\bar{T}_m$$

- Second step: compute an orthonormal basis of $\mathcal{K}_{m+1}(B, r_0)$
  Compute the QR factorization $K_{m+1} = V_{m+1}R_{m+1}$
  RODDEC [Sidje 1997, Erhel 1995] or TSQR [Demmel et al 2011]

$$\Rightarrow BK_m = V_{m+1}R_{m+1}\bar{T}_m \Rightarrow BV_m = V_{m+1}\underbrace{R_{m+1}\bar{T}_mR_m^{-1}}_{\bar{H}_m}$$

- Third step: approximate solution $x_m = x_0 + M^{-1}V_my_m$

$$\Rightarrow r_m = r_0 - BK_my_m = V_{m+1}(\beta e_1 - \bar{H}_my_m) \quad \text{with } \beta = \|r_0\|_2$$

$$\Rightarrow y_m = min_{y\in\mathbb{R}^m}\|\beta e_1 - \bar{H}_my\|_2$$

Deflation strategies

Restarted GMRES(m)

- $x_m = x_0 + M^{-1} V_m y_m$ where y_m minimizes $\|r_m\|_2$
- The convergence rate depends on the spectral distribution in B
- Smallest eigenvalues slow down the convergence
- Deflation occurs when the Krylov subspace is large enough
- With restarting : loss of spectral information, risk of stalling

Accelerating the restarted GMRES[Simoncini and Szyld, 2007]

- Approximate the smallest eigenvalues and the associated invariant subspace U_r
- Explicit deflation technique [Erhel et al 1996; Burrage et al 1998; Moriya et al 2000]:

$$B\bar{M}^{-1}\tilde{x} = b$$

  with $\bar{M}^{-1} = (I_n + U_r(|\lambda_n|T^{-1} - I_r)U_r^T$ and $T = U_r^T B U_r$
- Augmented techniques [Morgan 2000, 2002, Giraud et al, 2010]:

$$x_m \in x_0 + span\{U_r\} + \mathcal{K}_m(B, r_0)$$

Deflation strategies

Restarted GMRES(m)

- $x_m = x_0 + M^{-1} V_m y_m$ where y_m minimizes $\|r_m\|_2$
- The convergence rate depends on the spectral distribution in B
- Smallest eigenvalues slow down the convergence
- Deflation occurs when the Krylov subspace is large enough
- With restarting : loss of spectral information, risk of stalling

Accelerating the restarted GMRES[Simoncini and Szyld, 2007]

- Approximate the smallest eigenvalues and the associated invariant subspace U_r
- Explicit deflation technique [Erhel et al 1996; Burrage et al 1998; Moriya et al 2000]:

$$B\bar{M}^{-1}\tilde{x} = b$$

  with $\bar{M}^{-1} = (I_n + U_r(|\lambda_n| T^{-1} - I_r) U_r^T$ and $T = U_r^T B U_r$
- Augmented techniques [Morgan 2000, 2002, Giraud et al, 2010]:

$$x_m \in x_0 + span\{U_r\} + \mathcal{K}_m(B, r_0)$$

DGMRES: GMRES with adaptive preconditioning deflation

DGMRES(m, r)

- Perform one cycle of restarted GMRES(m) and compute a coarse subspace of basis U_r
- Build $\bar{M}^{-1} \equiv I_n + U_r(|\lambda_n| T^{-1} - I_r)U_r^T, \quad T = U_r^T B U_r$
- At each restart, update r and the basis U_r

Adaptive DGMRES(m,r)

- Switch to DGMRES(m,r) only if necessary [Nuentsa Wakam Erhel 2013]
- Detect a potential slow convergence [Sosonkina et al 1998]

Module DGMRES

- KSP module in Petsc library
- Distributed with Petsc
- Application to CFD problems [Nuentsa Wakam Pacull 2013]

DGMRES: GMRES with adaptive preconditioning deflation

DGMRES(m, r)

- Perform one cycle of restarted GMRES(m) and compute a coarse subspace of basis U_r
- Build $\bar{M}^{-1} \equiv I_n + U_r(|\lambda_n| T^{-1} - I_r)U_r^T, \quad T = U_r^T B U_r$
- At each restart, update r and the basis U_r

Adaptive DGMRES(m,r)

- Switch to DGMRES(m,r) only if necessary [Nuentsa Wakam Erhel 2013]
- Detect a potential slow convergence [Sosonkina et al 1998]

Module DGMRES

- KSP module in Petsc library
- Distributed with Petsc
- Application to CFD problems [Nuentsa Wakam Pacull 2013]

DGMRES: GMRES with adaptive preconditioning deflation

DGMRES(m, r)

- Perform one cycle of restarted GMRES(m) and compute a coarse subspace of basis U_r
- Build $\bar{M}^{-1} \equiv I_n + U_r(|\lambda_n| T^{-1} - I_r) U_r^T, \quad T = U_r^T B U_r$
- At each restart, update r and the basis U_r

Adaptive DGMRES(m,r)

- Switch to DGMRES(m,r) only if necessary [Nuentsa Wakam Erhel 2013]
- Detect a potential slow convergence [Sosonkina et al 1998]

Module DGMRES

- KSP module in Petsc library
- Distributed with Petsc
- Application to CFD problems [Nuentsa Wakam Pacull 2013]

DGMRES combined with Domain Decomposition

Implementation in PETSc

Options for DGMRES accelerator

-ksp_type <dgmres>, -ksp_dgmres_eigen <1>, -ksp_dgmres_smv <0.5>, -ksp_gmres_restart <48, 64>, -ksp_maxit <1000>

ASM Preconditioner

ParMETIS partitioning, RAS preconditioner ($D = 16, 32, 64, 128$), overlap $= 1$, Sequential MUMPS in subdomains.

Cluster Parapide @ GRID'5000; 25 nodes (2 CPUs Intel@2.93GHz, 4 cores/CPU, 24GB RAM), Infiniband network

RM07R : $n = 381{,}689$; $nz = 37{,}464{,}962$

D	GMRES(48)		DGMRES(47,1)		GMRES(64)		DGMRES(63,1)		Memory (MB)
	Matvecs	Time	Matvecs	Time	Matvecs	Time	Matvecs	Time	
16	551	230	212	173.4	355	193.8	208	168.9	1,070
32	-	-	533	109.2	2217	244.6	455	94.6	513
64	-	-	410	56.8	-	-	453	50.8	299
128	-	-	791	51.5	-	-	638	44.3	225

DGMRES combined with Domain Decomposition

Implementation in PETSc

Options for DGMRES accelerator

-ksp_type <dgmres>, -ksp_dgmres_eigen <1>, -ksp_dgmres_smv <0.5>, -ksp_gmres_restart <48, 64>, -ksp_maxit <1000>

ASM Preconditioner

ParMETIS partitioning, RAS preconditioner ($D = 16, 32, 64, 128$), overlap $= 1$, Sequential MUMPS in subdomains.

Cluster Parapide @ GRID'5000; 25 nodes (2 CPUs Intel@2.93GHz, 4 cores/CPU, 24GB RAM), Infiniband network

RM07R : n = 381,689; nz = 37,464,962

D	GMRES(48)		DGMRES(47,1)		GMRES(64)		DGMRES(63,1)		Memory (MB)
	Matvecs	Time	Matvecs	Time	Matvecs	Time	Matvecs	Time	
16	551	230	212	173.4	355	193.8	208	168.9	1,070
32	-	-	533	109.2	2217	244.6	455	94.6	513
64	-	-	410	56.8	-	-	453	50.8	299
128	-	-	791	51.5	-	-	638	44.3	225

AGMRES: deflation with an adaptive augmented basis

Building blocks

- Initial step: run one cycle of GMRES(m) and compute shifts for the Newton basis
  Compute $U_r = [u_0, u_1, \ldots, u_{r-1}]$ a basis of a coarse subspace
- First step: build a basis $K_{m+1} = [k_0, k_1, \ldots, k_m]$ of the Krylov subspace $\mathcal{K}_{m+1}(B, r_0)$ such that

  $$BK_m = K_{m+1} \bar{T}_m$$

  Define the augmented subspace $\mathcal{C}_s = \mathcal{K}_m(B, r_0) + span\{U_r\}$ with $s = m + r$ with the basis

  $$\left[\begin{array}{cc} K_m & U_r \end{array} \right]$$

- compute

  $$BU_r = \hat{K}_r D_r$$

  Define the augmented subspace $\hat{\mathcal{C}}_{s+1} = \mathcal{K}_{m+1}(B, r_0) + span\{BU_r\}$ with the basis

  $$\left[\begin{array}{cc} K_{m+1} & \hat{K}_r \end{array} \right]$$

AGMRES: deflation with an adaptive augmented basis

Building blocks

- Initial step: run one cycle of GMRES(m) and compute shifts for the Newton basis
  Compute $U_r = [u_0, u_1, \ldots, u_{r-1}]$ a basis of a coarse subspace
- First step: build a basis $K_{m+1} = [k_0, k_1, \ldots, k_m]$ of the Krylov subspace $\mathcal{K}_{m+1}(B, r_0)$ such that

$$BK_m = K_{m+1}\bar{T}_m$$

Define the augmented subspace $\mathcal{C}_s = \mathcal{K}_m(B, r_0) + span\{U_r\}$ with $s = m + r$ with the basis

$$\left[\begin{array}{cc} K_m & U_r \end{array}\right]$$

- compute

$$BU_r = \hat{K}_r D_r$$

Define the augmented subspace $\hat{\mathcal{C}}_{s+1} = \mathcal{K}_{m+1}(B, r_0) + span\{BU_r\}$ with the basis

$$\left[\begin{array}{cc} K_{m+1} & \hat{K}_r \end{array}\right]$$

AGMRES: deflation with an adaptive augmented basis

Building blocks

- Initial step: run one cycle of GMRES(m) and compute shifts for the Newton basis
  Compute $U_r = [u_0, u_1, \ldots, u_{r-1}]$ a basis of a coarse subspace
- First step: build a basis $K_{m+1} = [k_0, k_1, \ldots, k_m]$ of the Krylov subspace $\mathcal{K}_{m+1}(B, r_0)$ such that

  $$BK_m = K_{m+1}\bar{T}_m$$

  Define the augmented subspace $\mathcal{C}_s = \mathcal{K}_m(B, r_0) + span\{U_r\}$ with $s = m + r$ with the basis

  $$\begin{bmatrix} K_m & U_r \end{bmatrix}$$

- compute

  $$BU_r = \hat{K}_r D_r$$

  Define the augmented subspace $\hat{\mathcal{C}}_{s+1} = \mathcal{K}_{m+1}(B, r_0) + span\{BU_r\}$ with the basis

  $$\begin{bmatrix} K_{m+1} & \hat{K}_r \end{bmatrix}$$

AGMRES: deflation with an adaptive augmented basis

Building blocks

- Second step: Compute an orthonormal basis of $\hat{\mathcal{C}}_{s+1}$
  QR factorize the augmented basis $\begin{bmatrix} K_{m+1} & \hat{K}_r \end{bmatrix} = V_{s+1} R_{s+1}$

$$\Rightarrow BK_m = V_{m+1} R_{m+1} \bar{T}_m \Rightarrow BV_m = V_{m+1} R_{m+1} \bar{T}_m R_m^{-1}$$

$$\Rightarrow BU_r = (V_{m+1} R_{m+1,r} + V_r R_r) D_r$$

  Define the basis $W_s = \begin{bmatrix} V_m & U_r \end{bmatrix}$

$$\Rightarrow BW_s = V_{s+1} \bar{H}_s$$

- Third step: $x_s = x_0 + M^{-1} W_s y_s$

$$\Rightarrow r_s = r_0 - BW_s y_s = V_{s+1}(\beta e_1 - \bar{H}_s y_s) \quad \text{and } \beta = \|r_0\|_2$$

$$y_s = min_{y \in \mathbb{R}^s} \|\beta e_1 - \bar{H}_s y\|_2$$

- Final step: Adaptively update r and the coarse basis U_r

AGMRES: deflation with an adaptive augmented basis

Building blocks

- Second step: Compute an orthonormal basis of $\hat{\mathcal{C}}_{s+1}$
  QR factorize the augmented basis $\begin{bmatrix} K_{m+1} & \hat{K}_r \end{bmatrix} = V_{s+1} R_{s+1}$

  $$\Rightarrow BK_m = V_{m+1} R_{m+1} \bar{T}_m \Rightarrow BV_m = V_{m+1} R_{m+1} \bar{T}_m R_m^{-1}$$

  $$\Rightarrow BU_r = (V_{m+1} R_{m+1,r} + V_r R_r) D_r$$

  Define the basis $W_s = \begin{bmatrix} V_m & U_r \end{bmatrix}$

  $$\Rightarrow BW_s = V_{s+1} \bar{H}_s$$

- Third step: $x_s = x_0 + M^{-1} W_s y_s$

  $$\Rightarrow r_s = r_0 - BW_s y_s = V_{s+1}(\beta e_1 - \bar{H}_s y_s) \quad \text{and } \beta = \|r_0\|_2$$

  $$y_s = min_{y \in \mathbb{R}^s} \|\beta e_1 - \bar{H}_s y\|_2$$

- Final step: Adaptively update r and the coarse basis U_r

AGMRES: deflation with an adaptive augmented basis

Building blocks

- Second step: Compute an orthonormal basis of $\hat{\mathcal{C}}_{s+1}$
  QR factorize the augmented basis $\begin{bmatrix} K_{m+1} & \hat{K}_r \end{bmatrix} = V_{s+1} R_{s+1}$

$$\Rightarrow BK_m = V_{m+1} R_{m+1} \bar{T}_m \Rightarrow BV_m = V_{m+1} R_{m+1} \bar{T}_m R_m^{-1}$$

$$\Rightarrow BU_r = (V_{m+1} R_{m+1,r} + V_r R_r) D_r$$

  Define the basis $W_s = \begin{bmatrix} V_m & U_r \end{bmatrix}$

$$\Rightarrow BW_s = V_{s+1} \bar{H}_s$$

- Third step: $x_s = x_0 + M^{-1} W_s y_s$

$$\Rightarrow r_s = r_0 - BW_s y_s = V_{s+1}(\beta e_1 - \bar{H}_s y_s) \quad \text{and } \beta = \|r_0\|_2$$

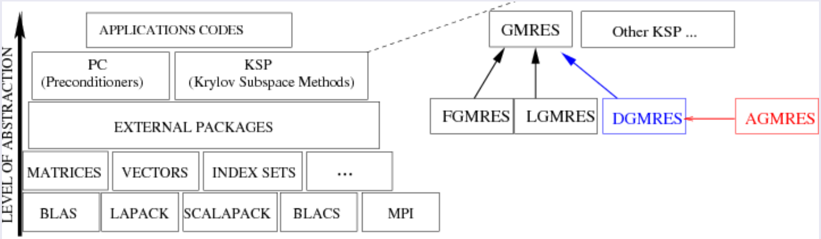
$$y_s = min_{y \in \mathbb{R}^s} \|\beta e_1 - \bar{H}_s y\|_2$$

- Final step: Adaptively update r and the coarse basis U_r

AGMRES in PETSc

New KSP type : AGMRES

figure



LEVEL OF ABSTRACTION

APPLICATIONS CODES

PC (Preconditioners)

KSP (Krylov Subspace Methods)

EXTERNAL PACKAGES

MATRICES | VECTORS | INDEX SETS | ...

BLAS | LAPACK | SCALAPACK | BLACS | MPI

GMRES | Other KSP ...

FGMRES | LGMRES | DGMRES | AGMRES

Usage in Petsc

- Use AGMRES just as GMRES
- $\Rightarrow$ KSPSetType(ksp, KSPAGMRES) or -ksp_type agmres, -pc_type asm, ...
- Options : -ksp_gmres_restart m, -ksp_agmres_eig r,
- -ksp_max_its maxits, -ksp_agmres_smv smv -ksp_agmres_bgv bgv, ...

AGMRES combined with Domain Decomposition

Main steps when using AGMRES

- Partition the weighted graph of the matrix in parallel with PARMETIS.
- Redistribute the matrix and right-hand-side according to the PARMETIS partitioning.
- Perform a parallel iterative row and column scaling on the matrix and the right-hand side vector [Amestoy et al, 2008].
- Define the overlap between the submatrices for the additive Schwarz preconditioner.

$$M_{RAS}^{-1} = \sum_{k=1}^{D} (R_k^0)^T (A_k^\delta)^{-1} R_k^\delta$$

- Setup the submatrices (ILU or LU factorization).
- Solve iteratively the preconditioned system using either AGMRES or GMRES.

Augmented Conjugate Gradient

PCG

- A Symmetric Positive Definite (SPD) matrix
- Krylov method
- short recurrences and minimization properties
- preconditioning M^{-1}

Coarse grid and balancing

[Nicolaides 1987, Mandel 1993, DD proceedings, Giraud et al.]

- Z basis of a coarse subspace
- $A_c = Z^T AZ$ restriction of A nonsingular small matrix
- $P = I - AZA_c^{-1}Z^T$ and $P^T = I - ZA_c^{-1}(AZ)^T$
- $C_b = P^T M^{-1} P + ZA_c^{-1} Z^T$

Coarse grid and augmented CG

[Erhel et al 2000, Saad et al. 2000, Tang et al. 2009, Poirriez 2011, Nataf et al]

- $x_0 = ZA_c^{-1}Z^T b$
- $C_a = P^T M^{-1}$
- C_a is equivalent to C_b

SIDNUR: AugCG and Domain Decomposition

Balancing Neumann Neumann

- PCG applied to a Schur complement
- Neumann-Neumann preconditioning M^{-1}
- Balancing with a coarse grid Z

SIDNUR

[Poirriez 2011, Pichot et al. DD21 proceedings to appear]

- domain decomposition provided by the user
- coarse grid : signature of subdomains [Frank and Vuik 2001]
- C++ library soon available
- mutual factorization of local Schur complements and local matrices
- management of floating subdomains
- numerical experiments with 3D fracture networks

Conclusion

GMRES

- DGMRES KSP module: deflation in GMRES(m) with or without Newton basis
- AGMRES KSP module: augmented Newton basis in GMRES(m)
- Deflation combined with Schwarz domain decomposition preconditioning
- Robustness: reduce the restarting effects and the domain decomposition effects
- Efficiency: increase granularity and scalability
- Numerical experiments with CFD problems: DGMRES and AGMRES faster than GMRES

PCG

- Deflation combined with Schur domain decomposition
- SIDNUR: Balancing Domain Decomposition
- Robustness: reduce the domain decomposition effects
- Efficiency: parallel Schur and Neumann Neumann computations
- Numerical experiments with 3D fracture networks: faster than multigrid and PCG
- Library soon available as free software