Pipelined, Flexible Krylov Subspace Methods

Patrick Sanan[†], Sascha M. Schnepp[×], Dave A. May[×]
[†] USI Lugano [×] ETH Zürich

June 17, 2015

PETSc-20

- Hard problem $Ax = b$
  $\rightarrow$ nonlinear preconditioner
  $\rightarrow$ flexible Krylov method

```
-ksp_type fgmres
-pc_type fieldsplit
-pc_fieldsplit_type schur
-pc_fieldsplit_schur_fact_type upper
-fieldsplit_u_ksp_type preonly
-fieldsplit_u_pc_type mg
-fieldsplit_u_pc_mg_galerkin
-fieldsplit_u_pc_mg_levels 3
-fieldsplit_p_ksp_type preonly
-fieldsplit_u_mg_coarse_pc_type gamg
-fieldsplit_u_mg_levels_2_ksp_type chebyshev
-fieldsplit_u_mg_levels_2_pc_type jacobi
-fieldsplit_u_mg_levels_2_ksp_max_it 10
-fieldsplit_u_mg_levels_1_pc_type asm
-fieldsplit_u_mg_levels_1_pc_asm_type restrict
-fieldsplit_p_pc_type bjacobi
-fieldsplit_u_mg_levels_1_sub_pc_type  ilu
-fieldsplit_u_mg_levels_1_pc_asm_dm_subdomains 1
-fieldsplit_u_mg_levels_1_pc_asm_overlap 4
-fieldsplit_u_mg_levels_1_pc_asm_type basic
-fieldsplit_u_mg_levels_1_ksp_type fgmres
-fieldsplit_u_mg_levels_1_ksp_max_it 2
```

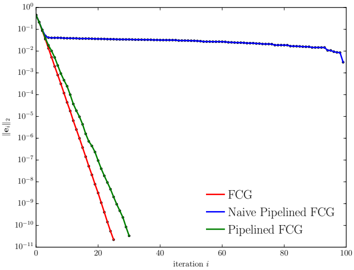
- Large process count
  $\rightarrow$ pipelined Krylov method
  (Ghysels, Vanroose, Ashby, Meerbergen, Reps)

function $\mathrm{PIPECG}(A, M^{-1}, b, x_0)$
  [.. Initialize, fill pipeline ..]
  for $i = 1, 2, \ldots$ **do**
    $x_i \leftarrow x_{i-1} + \alpha_{i-1} p_{i-1}$
    $r_i \leftarrow r_{i-1} - \alpha_{i-1} s_{i-1}$
    $u_i \leftarrow u_{i-1} - \alpha_{i-1} q_{i-1}$
    $w_i \leftarrow w_{i-1} - \alpha_{i-1} z_{i-1}$
    $m_i \leftarrow M^{-1} w_i$
    $n_i \leftarrow A m_i$
    $\gamma_i \leftarrow \langle u_i, r_i \rangle$
    $\delta_i \leftarrow \langle u_i, w_i \rangle$
    $\beta_i \leftarrow \gamma_i / \gamma_{i-1}$
    $\eta_i \leftarrow \delta_i - \beta_i^2 \eta_{i-1}$
    $\alpha_i \leftarrow \gamma_i / \eta_i$
    $p_i \leftarrow u_i + \beta_i p_{i-1}$
    $s_i \leftarrow w_i + \beta_i s_{i-1}$
    $q_i \leftarrow m_i + \beta_i q_{i-1}$
    $z_i \leftarrow n_i + \beta_i z_{i-1}$

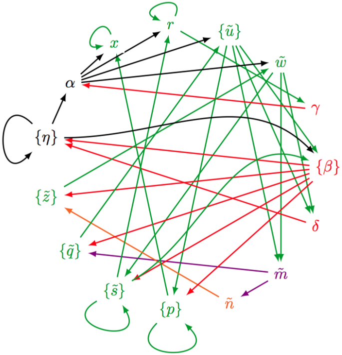
Pipelining loop transformations:

- Linear 'unrolling'
  $Ax_i = Ax_{i-1} + \alpha_{i-1} A u_{i-1}$
- The Pythagorean Theorem
  $||p_i||_A^2 = ||u_i||_A^2 - \beta_i^2 ||p_{i-1}||_A^2$
- Nonlinear preconditioners B
  $u_i = B(r_i) \overset{?}{\approx} B(r_{i-1}) - \alpha_{i-1} B(Au_{i-1})$
- We examine an inexpensive yet effective trick to retain the effectiveness of the preconditioner, based on the assumption of a strong-enough preconditioner.
  $u_i = B(r_i) \approx (1-\alpha_{i-1})B(r_{i-1}) - \alpha_{i-1} B(Au_{i-1} - r_{i-1})$

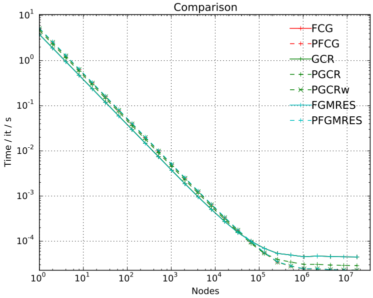
function PIPEFCG(A, B, b, x_0)
  [.. Initialize, fill pipeline ..]
  for $i = 1, 2, \ldots$ **do**
    $x_i \leftarrow x_{i-1} + \alpha_{i-1} p_{i-1}$
    $r_i \leftarrow r_{i-1} - \alpha_{i-1} \tilde{s}_{i-1}$
    $\tilde{u}_i \leftarrow \tilde{u}_{i-1} - \alpha_{i-1} \tilde{q}_{i-1}$
    $\tilde{w}_i \leftarrow \tilde{w}_{i-1} - \alpha_{i-1} \tilde{z}_{i-1}$
    $\mathbf{\tilde{m}_i = B(\tilde{w}_i - r_i) + u_i}$
    $\tilde{n}_i = A\tilde{m}_i$
    $\gamma_i \leftarrow \langle \tilde{u}_i, r_i \rangle$
    for $k = i - \nu_i, \ldots, i-1$ **do**
      $\beta_{i,k} \leftarrow \frac{-1}{\eta_k} \langle \tilde{u}_i, \tilde{s}_k \rangle$
    $\delta_i \leftarrow \langle \tilde{u}_i, \tilde{w}_i \rangle$
    $p_i \leftarrow \tilde{u}_i + \sum_{k=i-\nu_i}^{i-1} \beta_{i,k} p_k$
    $\tilde{s}_i \leftarrow \tilde{w}_i + \sum_{k=i-\nu_i}^{i-1} \beta_{i,k} \tilde{s}_k$
    $\tilde{q}_i \leftarrow \tilde{m}_i + \sum_{k=i-\nu_i}^{i-1} \beta_{i,k} \tilde{q}_k$
    $\tilde{z}_i \leftarrow \tilde{n}_i + \sum_{k=i-\nu_i}^{i-1} \beta_{i,k} \tilde{z}_k$
    $\eta_i \leftarrow \delta_i - \sum_{k=i-\nu_i}^{i-1} \beta_{i,k}^2 \eta_k$
    $\alpha_i \leftarrow \gamma_i / \eta_i$



Complex ..



But promising for some use cases.



`KSPPIPEFCG`, `KSPPIPEGCR`, `KSPPIPEFGMRES` written!