WP1: Overview of recent solver developments

A general framework for deriving pipelined Krylov methods: application to BiCGStab for large and sparse unsymmetric linear systems

EXA2CT meeting, Bordeaux, 22-23 September 2016

University of Antwerp, Applied Mathematics
Siegfried Cools, Wim Vanroose

Universiteit Antwerpen
Bi-Conjugate Gradients Stabilized (BiCGStab)

Algorithm 4 Standard BiCGStab

1: function BICGSTAB($A, b, x_0$)
2:     $r_0 := b - Ax_0$; $p_0 := r_0$
3:     for $i = 0,...$ do
4:         $s_i := Ap_i$
5:         compute $(r_0, s_i)\text{; } (\alpha_i, q_i, y_i)$
6:         $\alpha_i := (r_0, r_i) / (r_0, s_i)$
7:         $q_i := r_i - \alpha_i s_i$
8:         $y_i := Aq_i$
9:         compute $(q_i, y_i)\text{; } (y_i, y_i)$
10:        $\omega_i := (q_i, y_i) / (y_i, y_i)$
11:        $x_{i+1} := x_i + \alpha_i p_i + \omega_i q_i$
12:        $r_{i+1} := q_i - \omega_i y_i$
13:        compute $(r_0, r_{i+1})\text{; } (\beta_i, \rho_{i+1}, p_{i+1})$
14:        $\beta_i := (\alpha_i/\omega_i) (r_0, r_{i+1}) / (r_0, r_i)$
15:        $p_{i+1} := r_{i+1} + \beta_i (p_i - \omega_i s_i)$
16:     end for
17: end function

Traditional BiCGStab:
(non-preconditioned)

Global communication
- 3 global reduction phases

Semi-local communication
- 2 non-overlapping SpMVs

Local communication
- 4 axpy(-like) operations

General two-step framework for deriving pipelined Krylov methods:

**Step 1.** Avoiding communication: merge global reductions

**Step 2.** Hiding communication: overlap SpMVs & global reductions
Step 1. Avoiding global communication

(a) **Identify** two global comm. phases for merger (lines 5-6 & 13-14)

(b) **Rewrite** SpMV as recurrence:

\[
s_i = Ap_i = w_i + \beta_{i-1} (s_{i-1} - \omega_{i-1} z_{i-1}),
\]

*define* \( w_i := Ar_i \), \( z_i := As_i \) and note that \( y_i := w_i - \alpha_i z_i \)

(c) **Rewrite** dot-product using (b):

\[
(r_0, s_i) = (r_0, w_i + \beta_{i-1} (s_{i-1} - \omega_{i-1} z_{i-1})),
\]

independent of interlying variables

(d) **Move** dot-product (lines 5-6) upward and merge with existing global comm. phase (lines 13-14)
Communication-avoiding BiCGStab (CA-BiCGStab)

Algorithm 5 Communication avoiding BiCGStab

1: function CA-BiCGSTAB(A, b, x₀)
2:     r₀ := b − Ax₀; w₀ := Ar₀; α₀ := (r₀, r₀) / (r₀, w₀); β₀ := 0
3:     for i = 0, . . . do
4:         pᵢ := rᵢ + βᵢ₋₁(pᵢ₋₁ − ωᵢ₋₁sᵢ₋₁)
5:         sᵢ := wᵢ + βᵢ₋₁(sᵢ₋₁ − ωᵢ₋₁zᵢ₋₁)
6:         zi := Asi
7:         qi := rᵢ − αᵢsi
8:         yi := wi − αᵢzi
9:         compute (qi, yi) ; (yi, yi)
10:        ωᵢ := (qi, yi) / (yi, yi)
11:        xi₊₁ := xi + αᵢpi + ωᵢqi
12:        rᵢ₊₁ := qi − ωᵢyi
13:        wi₊₁ := Arᵢ₊₁
14:        compute (r₀, rᵢ₊₁) ; (r₀, wi₊₁) ; (r₀, sᵢ) ; (r₀, zᵢ)
15:     βᵢ := (αᵢ/ωᵢ) (r₀, rᵢ₊₁) / (r₀, rᵢ)
16:     αᵢ₊₁ := (r₀, rᵢ₊₁) / ((r₀, wi₊₁) + βᵢ(r₀, sᵢ) − βᵢωᵢ(r₀, zᵢ))
17:     end for
18: end function

CA-BiCGStab:
(non-preconditioned)

Global communication
▶ 2 global red. phases (vs. 3)

Semi-local communication
▶ 2 non-overlapping SpMVs

Local communication
▶ 6 axpy(-like) operations (vs. 4)

Status after Step 1:
no. global comm. phases reduced from 3 to 2, at the cost of 2 additional axpys

Note: further reduction from 2 to 1 global comm. phase possible, but not recommended (see later).
Step 2. Hiding global communication

(a) **Identify** SpMV / global reduction pairs (lines 6 & 9 and 13 & 14)

(b) **Rewrite** SpMVs as recurrences:

\[ z_i := A s_i = t_i + \beta_{i-1} (z_{i-1} - \omega_{i-1} v_{i-1}) \],
\[ w_{i+1} := A r_{i+1} = y_i - \omega_i (t_i - \alpha_i v_i), \]

**define** \( t_i := A w_i, v_i := A z_i \)

(c) **Check** SpMV / global reduction pairwise dependencies:

- line 9 independent of \( v_i \)? **yes**
- line 14 indep. of \( t_{i+1} \)? **yes**

(d) **Insert** new SpMVs **below**
corresponding global comm. phases

---

**Algorithm 5** Communication avoiding BiCGStab

1: function CA-BICGSTAB(A, b, x₀)
2: \( r_0 := b - A x_0; w_0 := A r_0; \alpha_0 := (r_0, r_0) / (r_0, w_0); \beta_{-1} := 0 \)
3: for \( i = 0, \ldots \) do
4: \( p_i := r_i + \beta_{i-1} (p_{i-1} - \omega_{i-1} s_{i-1}) \)
5: \( s_i := w_i + \beta_{i-1} (s_{i-1} - \omega_{i-1} z_{i-1}) \)
6: \( z_i := A s_i \)
7: \( q_i := r_i - \alpha_i s_i \)
8: \( y_i := w_i - \alpha_i z_i \)
9: **compute** \((q_i, y_i) ; (y_i, y_i)\) \hspace{1cm} **dot-prod**
10: \( \omega_i := (q_i, y_i) / (y_i, y_i) \)
11: \( x_{i+1} := x_i + \alpha_i p_i + \omega_i q_i \)
12: \( r_{i+1} := q_i - \omega_i y_i \)
13: \( w_{i+1} := A r_{i+1} \)
14: **compute** \((r_0, r_{i+1}) ; (r_0, w_{i+1}) ; (r_0, s_i) ; (r_0, z_i)\)
15: \( \beta_i := (\alpha_i / \omega_i) (r_0, r_{i+1}) / (r_0, r_i) \)
16: \( \alpha_{i+1} := (r_0, r_{i+1}) / ((r_0, w_{i+1}) + \beta_i (r_0, s_i) - \beta_i \omega_i (r_0, z_i)) \)
17: end for
18: end function
Pipelined BiCGStab (p-BiCGStab)

Algorithm 6 Pipelined BiCGStab

```
function PIPE-BICGSTAB(A, b, x₀)
  r₀ := b - Ax₀; w₀ := Ar₀; t₀ := Aw₀;
  for i = 0,... do
    pᵢ := rᵢ + βᵢ₋₁ (pᵢ₋₁ - ωᵢ₋₁ sᵢ₋₁)
    sᵢ := wᵢ + βᵢ₋₁ (sᵢ₋₁ - ωᵢ₋₁ zᵢ₋₁)
    zᵢ := tᵢ + βᵢ₋₁ (zᵢ₋₁ - ωᵢ₋₁ vᵢ₋₁)
    qᵢ := rᵢ - αᵢ sᵢ
    yᵢ := wᵢ - αᵢ zᵢ
    compute (qᵢ, yᵢ); (yᵢ, yᵢ)
    ωᵢ := (qᵢ, yᵢ) / (yᵢ, yᵢ)
    overlap vᵢ := Azᵢ
    xᵢ₊₁ := xᵢ + αᵢ pᵢ + ωᵢ qᵢ
    rᵢ₊₁ := qᵢ - ωᵢ yᵢ
    wᵢ₊₁ := yᵢ - ωᵢ (tᵢ - αᵢ vᵢ)
    compute (r₀, rᵢ₊₁); (r₀, wᵢ₊₁); (r₀, sᵢ); (r₀, zᵢ)
    βᵢ := (αᵢ/ωᵢ) (r₀, rᵢ₊₁) / (r₀, rᵢ)
    αᵢ₊₁ := (r₀, rᵢ₊₁) / ((r₀, wᵢ₊₁) + βᵢ (r₀, sᵢ) - βᵢ ωᵢ (r₀, zᵢ))
    overlap tᵢ₊₁ := Awᵢ₊₁
  end for
end function
```

**p-BiCGStab:**
(non-preconditioned)

**Global communication**
- 2 global red. phases (vs. 3)

**Semi-local communication**
- 2 overlapping SpMVs

**Local communication**
- 8 axpy(-like) operations (vs. 4)

**Status after Step 2:**
both global comm. phases are overlapped with SpMV computations ('hidden'),
at the cost of 4 additional axyps compared to standard BiCGStab
Like for any pipelined method, including a preconditioner is easy.

p-BiCGStab:
(preconditioned)

Global communication
- 2 global red. phases (vs. 3)

Semi-local communication
- 2 overlapping Prec + SpMVs

Local communication
- 11 axpy(-like) operations (vs. 4)

<table>
<thead>
<tr>
<th></th>
<th>GLRED</th>
<th>SPMV</th>
<th>Flops (AXPY + DOT-PROD)</th>
<th>Time (GLRED + SPMV)</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiCGStab</td>
<td>3</td>
<td>2</td>
<td>20</td>
<td>3 GLRED + 2 SPMV</td>
<td>7</td>
</tr>
<tr>
<td>IBiCGStab</td>
<td>1</td>
<td>2</td>
<td>30</td>
<td>1 GLRED + 2 SPMV</td>
<td>10</td>
</tr>
<tr>
<td>p-BiCGstab</td>
<td>2</td>
<td>2*</td>
<td>38</td>
<td>2 max(GLRED, SPMV)</td>
<td>11</td>
</tr>
</tbody>
</table>

If $\text{time(}GLRED\text{)} \approx \text{time(}SPMV\text{)}$:
- speed-up factor $p\text{-BiCGStab/BiCGStab} = 2.5$
- speed-up factor $IBiCGStab/BiCGStab = 1.66$

If $\text{time(}GLRED\text{)} \gg \text{time(}SPMV\text{)}$:
- speed-up factor $p\text{-BiCGStab/BiCGStab} = 2.5$
- speed-up factor $IBiCGStab/BiCGStab = 3.0$

<table>
<thead>
<tr>
<th>Method</th>
<th>GLRED</th>
<th>SPMV</th>
<th>Flops (AXPY + DOT-PROD)</th>
<th>Time (GLRED + SPMV)</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiCGStab</td>
<td>3</td>
<td>2</td>
<td>20</td>
<td>3 GLRED + 2 SPMV</td>
<td>7</td>
</tr>
<tr>
<td>IBiCGStab</td>
<td>1</td>
<td>2</td>
<td>30</td>
<td>1 GLRED + 2 SPMV</td>
<td>10</td>
</tr>
<tr>
<td>p-BiCGstab</td>
<td>2</td>
<td>2*</td>
<td>38</td>
<td>2 max(GLRED, SPMV)</td>
<td>11</td>
</tr>
</tbody>
</table>

Is combination of both methods (1 GLRED overlapped with all SPMVs) possible?

Theoretically, yes, however:
- no. axpys is much larger → algorithm robustness decreases (rounding errors)
- one extra SpMV required → increase in computational cost → further loss of robustness/flop equivalence
Convergence results: p-BiCGStab

MatrixMarket collection unsymmetric test problems \[ \text{tol} = 1e-6 \]

<table>
<thead>
<tr>
<th>Matrix</th>
<th>Prec</th>
<th>(\kappa(A))</th>
<th>(N)</th>
<th>#nnz</th>
<th>(|r_0|_2)</th>
<th>BiCGStab (\text{iter})</th>
<th>(|r_i|_2)</th>
<th>p-BiCGStab (\text{iter})</th>
<th>(|r_i|_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1138_bus</td>
<td>ILU</td>
<td>8.6e+06</td>
<td>1138</td>
<td>4054</td>
<td>4.3e+01</td>
<td>89</td>
<td>1.4e-05</td>
<td>95</td>
<td>1.8e-05</td>
</tr>
<tr>
<td>add32</td>
<td>ILU</td>
<td>1.4e+02</td>
<td>4960</td>
<td>19,848</td>
<td>8.0e+03</td>
<td>19</td>
<td>5.9e-09</td>
<td>19</td>
<td>5.9e-09</td>
</tr>
<tr>
<td>bcsstk14</td>
<td>ILU</td>
<td>1.3e+10</td>
<td>1806</td>
<td>63,454</td>
<td>2.1e+09</td>
<td>315</td>
<td>1.6e+03</td>
<td>322</td>
<td>1.2e+03</td>
</tr>
<tr>
<td>bcsstk18</td>
<td>ILU</td>
<td>6.5e+01</td>
<td>11,948</td>
<td>149,090</td>
<td>2.6e+09</td>
<td>84</td>
<td>2.2e+03</td>
<td>102</td>
<td>2.0e+03</td>
</tr>
<tr>
<td>bcsstk26</td>
<td>ILU</td>
<td>1.7e+08</td>
<td>1922</td>
<td>30,336</td>
<td>3.5e+09</td>
<td>113</td>
<td>2.8e+03</td>
<td>107</td>
<td>2.9e+03</td>
</tr>
<tr>
<td>bcsstm25</td>
<td>-</td>
<td>6.1e+09</td>
<td>15,439</td>
<td>15,439</td>
<td>6.9e+07</td>
<td>928</td>
<td>6.8e+01</td>
<td>825</td>
<td>6.5e+01</td>
</tr>
<tr>
<td>bfw782a</td>
<td>ILU</td>
<td>1.7e+03</td>
<td>782</td>
<td>7514</td>
<td>3.2e-01</td>
<td>72</td>
<td>1.1e-07</td>
<td>65</td>
<td>6.2e-08</td>
</tr>
<tr>
<td>bw2000</td>
<td>-</td>
<td>2.4e+05</td>
<td>2000</td>
<td>7996</td>
<td>1.1e+03</td>
<td>1156</td>
<td>6.6e-04</td>
<td>1162</td>
<td>9.1e-04</td>
</tr>
<tr>
<td>cdce6</td>
<td>ILU</td>
<td>1.8e+02</td>
<td>961</td>
<td>4681</td>
<td>5.8e-01</td>
<td>9</td>
<td>2.2e-07</td>
<td>9</td>
<td>2.2e-07</td>
</tr>
<tr>
<td>fidap014</td>
<td>-</td>
<td>3.5e+16</td>
<td>3251</td>
<td>65,747</td>
<td>2.7e+06</td>
<td>121</td>
<td>2.6e+00</td>
<td>123</td>
<td>2.6e+00</td>
</tr>
<tr>
<td>fs_760.3</td>
<td>ILU</td>
<td>1.0e+20</td>
<td>760</td>
<td>5816</td>
<td>1.6e+07</td>
<td>930</td>
<td>1.4e+01</td>
<td>709</td>
<td>1.1e+01</td>
</tr>
<tr>
<td>jagmesh9</td>
<td>-</td>
<td>6.0e+03</td>
<td>1349</td>
<td>9101</td>
<td>6.8e+00</td>
<td>1022</td>
<td>6.4e-06</td>
<td>996</td>
<td>6.6e-06</td>
</tr>
<tr>
<td>jpwh_991</td>
<td>ILU</td>
<td>1.4e+02</td>
<td>991</td>
<td>6027</td>
<td>3.8e-01</td>
<td>9</td>
<td>2.9e-07</td>
<td>9</td>
<td>2.9e-07</td>
</tr>
<tr>
<td>orsreg_1</td>
<td>ILU</td>
<td>6.7e+03</td>
<td>2205</td>
<td>14,133</td>
<td>4.8e+00</td>
<td>25</td>
<td>2.7e-06</td>
<td>25</td>
<td>2.7e-06</td>
</tr>
<tr>
<td>pde2961</td>
<td>ILU</td>
<td>6.4e+02</td>
<td>2961</td>
<td>14,585</td>
<td>2.9e-01</td>
<td>31</td>
<td>1.3e-07</td>
<td>31</td>
<td>4.5e-08</td>
</tr>
<tr>
<td>rdb3200l</td>
<td>-</td>
<td>1.1e+03</td>
<td>3200</td>
<td>18,880</td>
<td>1.0e+01</td>
<td>149</td>
<td>5.1e-06</td>
<td>145</td>
<td>9.1e-06</td>
</tr>
<tr>
<td>s3dkq4m2</td>
<td>-</td>
<td>1.9e+11</td>
<td>90,449</td>
<td>2,455,670</td>
<td>6.8e+01</td>
<td>3736</td>
<td>5.8e-05</td>
<td>3500</td>
<td>6.2e-05</td>
</tr>
<tr>
<td>saylr4</td>
<td>ILU</td>
<td>6.9e+06</td>
<td>3564</td>
<td>22,316</td>
<td>3.1e-03</td>
<td>40</td>
<td>3.0e-09</td>
<td>39</td>
<td>1.5e-09</td>
</tr>
<tr>
<td>sherman3</td>
<td>ILU</td>
<td>5.5e+18</td>
<td>5005</td>
<td>20,033</td>
<td>1.8e+01</td>
<td>98</td>
<td>3.7e-06</td>
<td>83</td>
<td>9.1e-06</td>
</tr>
<tr>
<td>sstmodel</td>
<td>-</td>
<td>2.7e+18</td>
<td>3345</td>
<td>22,749</td>
<td>7.9e+00</td>
<td>6968</td>
<td>7.7e-06</td>
<td>4399</td>
<td>7.5e-06</td>
</tr>
<tr>
<td>utm5940</td>
<td>ILU</td>
<td>4.3e+08</td>
<td>5940</td>
<td>83,842</td>
<td>3.6e-01</td>
<td>223</td>
<td>4.0e-08</td>
<td>244</td>
<td>3.5e-07</td>
</tr>
</tbody>
</table>

Average iter deviation wrt BiCGStab: \(-3.5\%\)
Convergence results: p-BiCGStab

1138_bus with ILU preconditioner

bcsstk18 with ILU preconditioner

bwm2000 without preconditioner

sherman3 with ILU preconditioner
Robustness and attainable accuracy: p-BiCGStab-rr
Robustness and attainable accuracy: p-BiCGStab-rr

Residual replacement every \( rr \)-th iteration
(non-automated, i.e. \( rr \) is a parameter of the method, but chosen large s.t. no. res. repl. is small)

\[
\begin{align*}
  r_i &:= b - Ax_i, & \hat{r}_i &:= M^{-1}r_i, & w_i &:= A\hat{r}_i, \\
  s_i &:= A\hat{p}_i, & \hat{s}_i &:= M^{-1}s_i, & z_i &:= A\hat{s}_i.
\end{align*}
\]

- increased **maximal attainable accuracy**: comparable to BiCGStab level
- increased **robustness**: negates instable true residual behaviour
- increased **number of iterations** possible
Robustness and attainable accuracy: p-BiCGStab-rr

MatrixMarket collection unsymmetric test problems $\text{tol} = 1e^{-20}$

<table>
<thead>
<tr>
<th>Matrix</th>
<th>$|r_0|_2$</th>
<th>BiCGStab</th>
<th>p-BiCGStab</th>
<th>p-BiCGStab-rr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>iter</td>
<td>$|r_i|_2$</td>
<td>iter</td>
</tr>
<tr>
<td>1138_bus</td>
<td>4.3e+01</td>
<td>124</td>
<td>1.8e-11</td>
<td>130</td>
</tr>
<tr>
<td>add32</td>
<td>8.0e-03</td>
<td>46</td>
<td>7.8e-18</td>
<td>42</td>
</tr>
<tr>
<td>bcsstk14</td>
<td>2.1e+09</td>
<td>559</td>
<td>7.3e-06</td>
<td>444</td>
</tr>
<tr>
<td>bcsstk18</td>
<td>2.6e+09</td>
<td>523</td>
<td>4.8e-06</td>
<td>450</td>
</tr>
<tr>
<td>bcsstk26</td>
<td>3.5e+09</td>
<td>414</td>
<td>1.1e-05</td>
<td>216</td>
</tr>
<tr>
<td>bcsstm25</td>
<td>6.9e+07</td>
<td>-</td>
<td>3.2e+00</td>
<td>-</td>
</tr>
<tr>
<td>bfw782a</td>
<td>3.2e-01</td>
<td>117</td>
<td>9.5e-14</td>
<td>106</td>
</tr>
<tr>
<td>bwm2000</td>
<td>1.1e+03</td>
<td>1733</td>
<td>2.5e-09</td>
<td>1621</td>
</tr>
<tr>
<td>cdde6</td>
<td>5.8e-01</td>
<td>151</td>
<td>8.1e-14</td>
<td>147</td>
</tr>
<tr>
<td>fidap014</td>
<td>2.7e+06</td>
<td>-</td>
<td>4.3e-03</td>
<td>-</td>
</tr>
<tr>
<td>fs_760_3</td>
<td>1.6e+07</td>
<td>1979</td>
<td>1.2e-05</td>
<td>1039</td>
</tr>
<tr>
<td>jagmesh9</td>
<td>6.8e+00</td>
<td>6230</td>
<td>2.4e-14</td>
<td>3582</td>
</tr>
<tr>
<td>jpw_991</td>
<td>3.8e-01</td>
<td>53</td>
<td>1.3e-14</td>
<td>54</td>
</tr>
<tr>
<td>orsreg_1</td>
<td>4.8e+00</td>
<td>51</td>
<td>4.0e-11</td>
<td>52</td>
</tr>
<tr>
<td>pde2961</td>
<td>2.9e-01</td>
<td>50</td>
<td>4.5e-15</td>
<td>48</td>
</tr>
<tr>
<td>rdb3200l</td>
<td>1.0e+01</td>
<td>178</td>
<td>3.7e-08</td>
<td>167</td>
</tr>
<tr>
<td>s3dkq4m2</td>
<td>6.8e+01</td>
<td>-</td>
<td>1.0e-05</td>
<td>-</td>
</tr>
<tr>
<td>saylr4</td>
<td>3.1e-03</td>
<td>52</td>
<td>4.0e-12</td>
<td>43</td>
</tr>
<tr>
<td>sherman3</td>
<td>1.8e+01</td>
<td>111</td>
<td>2.5e-11</td>
<td>100</td>
</tr>
<tr>
<td>sstmodel</td>
<td>7.9e+00</td>
<td>-</td>
<td>5.1e-06</td>
<td>-</td>
</tr>
<tr>
<td>utm5940</td>
<td>3.6e-01</td>
<td>256</td>
<td>3.0e-12</td>
<td>248</td>
</tr>
</tbody>
</table>

Average iter deviation wrt BiCGStab: $-11.0\%$
Average #nrr wrt p-BiCGStab-rr iter: $22.1\%$
Performance benchmark: strong scaling results

- PETSc implementation using MPICH-3.1.3 communication
- Hardware specs: 1-20 LYNX nodes (12-240 cores)
- Benchmark problem: 1.000.000 unknowns 2D model with asymmetric stencil

\[ A_{\text{stencil}} = \begin{pmatrix} 1 & \frac{1}{\varepsilon} \\ \varepsilon & \varepsilon \end{pmatrix}, \quad \text{with } \varepsilon = 0.999 \]

CPU time i.f.o. number of nodes | Speedup over 1-node BiCGStab

![Graph showing performance with number of nodes](image1)

![Graph showing speedup with number of nodes](image2)
Performance benchmark: accuracy results

- PETSc implementation using MPICH-3.1.3 communication
- Hardware specs: 20 LYNX nodes (240 cores)
- Benchmark problem: 1,000,000 unknowns 2D model with asymmetric stencil

\[
A^{stencil} = \begin{pmatrix}
1 & 1 \\
-4 & \varepsilon \\
\end{pmatrix}, \quad \text{with } \varepsilon = 0.999
\]

Residuals i.f.o. number of iterations

Residuals i.f.o. total time spent
Conclusions and outlook

Overall conclusions

▶ General framework for deriving pipelined variants of existing Krylov methods
▶ Prove-of-concept: successfully applied to BiCGStab
▶ p-BiCGStab displays good performance, but slightly decreased robustness
▶ Residual replacement strategy improves robustness and max. att. accuracy

Status summary

▶ p-BiCGStab prototype Matlab code available
▶ p-BiCGStab PETSc implementation finished
  → to be made publicly available in open-source PETSc distribution asap
▶ Technical report nearly (80%) completed
  → to be submitted as full paper in ~ 1-2 months
References


