Aspects of deflated restarts in mixed precision computations for lattice QCD

Andreas Frommer

Bergische Universität Wuppertal

September 8, 2010
Joint work with

- Matthias Bolten, Karsten Kahl, Paul Zingler (Wuppertal U)
- Yoshifumi Nakamura, Andrea Nobile, Tilo Wettig (Regensburg U)

within the DFG collaborative research center

“SFB-TR 55 Hadron Physics from Lattice QCD”
Outline

1. The QPACE project
2. SAP preconditioning
3. Variants of GMRES
4. Deflation
5. Mixed precision
QPACE: Hardware project within SFB-TR 55

Joint development academia–industry:

- Universities of Regensburg, Wuppertal, Ferrara, Milano, DESY Zeuthen, Research Lab Jülich
- IBM Böblingen, Rochester, La Gaude
- Support from Eurotech (I), Knürr (D) and Xilinx (US)
QPACE: QCD Parallel Computing on the Cell

- **Goal:**
  - Scalable architecture optimized for lattice QCD calculations

- **Concept:**
  - Fast commodity processor = IBM PowerXCell 8i
  - Custom network → custom network processor (FPGA based)
  - Custom system design

- **Challenges:**
  - High single node sustained performance
  - Scalability = high bandwidth, low latency network
  - Cost efficient system integration

- **Deployment:**
  - University of Wuppertal, Research Lab Jülich
  - 4 QPACE racks at each site
QPACE rack:
- 256 nodes (1 PowerXCell 8i, 1 network processor, 4 GB DDR2)
- Performance density: 26/52 TFlops/rack DP/SP, O(29) kW

PowerXCell 8i Processor:
- 8 Synergistic Processing Elements: double precision, IEEE rounding, 12.8/25.6 GFlops DP/SP, 256 kBytes local store
- DDR2 memory, 25 Gbyte/s

Custom 3D torus network:
- High-bandwidth, low-latency network for inter-node comm.
- Goal: O(1) $\mu$s latency, 1 GByte/s per link and direction
- Nearest neighbour communication
QPACE: LQCD kernel optimisation

- Optimisation challenge:
  - Optimise data re-use in on-chip memory
  - On-chip parallelism, SIMDization
  - Hide network latencies $O(3 \, \mu s)$
- Process small lattice blocks $\rightarrow$ Suitable algorithms implemented
  - Max block size per node: 8x8x4x4
  - Block solver performance: $>45\%$ peak single prec.
- Measured performance of $O(20\%)$
  - Room for further improvement up to 30%
Simulation on QPACE: 4 nodes

- $16^3 \times 32$ configuration
- $\beta = 5.29$, $\kappa_{\text{sea}} = 0.1350$, $\kappa = 0.13768$, $\kappa_c \approx 0.1377$
Simulation on QPACE: 2 x 128 nodes

- $48^3 \times 64$ configuration
- $\beta = 5.40, c_{sw} = 1.8228, \kappa_{sea} = 0.1366, \kappa = 0.1373$

![Graph showing iterations and norms for different methods]
Preconditioned GMRES

SAP: Schwarz alternating procedure (Lüscher)

- $M$: (clover improved) Wilson fermion matrix
- nearest neighbor coupling on 4d torus
- solve $Mx = b$ using restarted preconditioned GMRES
- preconditioner: SAP

Important aspects:
- restarts
- non-stationary preconditioner
SAP preconditioner

- use red-black ordering of blocks
- on each block \( i \) solve \( M_i \Delta x_i = b - Mx \) (smaller system)
- for each color: correct \( x \rightarrow x + \sum_{blocks} \Delta x_i \)
- do several sweeps, each color in parallel (we take 6 sweeps)
SAP preconditioner

- use red-black ordering of blocks
- on each block $i$ solve $M_i \Delta x_i = b - \hat{M} x$ (smaller system)
- for each color: correct $x \rightarrow x + \sum_{\text{blocks}} \Delta x_i$
- do several sweeps, each color in parallel (we take 6 sweeps)
- aka: red-black block Gauss-Seidel
$K_k(M, b) = \text{span}\{b, Mb, \ldots, M^{k-1}b\}$ k-th Krylov subspace

- **Minimal residual property:**
  
  $x^k \in K_{k-1}$ s.t. $\|b - Mx\|_2$ is minimal over all $x \in K_k$

  **GMRES computes such $x^k$:**
  
  - Arnoldi process (→ long recurrence, restart necessary)
  - $(k + 1) \times k$ least squares problem

- **Conjugate residuals property:**
  
  $\langle r^k, Mr_j \rangle = 0$ for $j = 0, \ldots, k - 1$.

  **GCR computes such $x^k$:**
  
  - generate $M^\dagger M$-conjugate search directions $p_k$
    (→ long recurrence, restart necessary)
**Theorem:** The following are equivalent:

- minimal residual property
- conjugate residuals property
- $r^k \perp M \cdot K_k$
Theorem: The following are equivalent:

- minimal residual property
- conjugate residuals property
- $r^k \perp M \cdot K_k$

Consequences: GMRES and GCR . . .

- . . . are mathematically equivalent
- . . . have different numerical quality:
  GCR is significantly less stable and accurate than GMRES when convergence is slow
  [Roloźnik, Tichý, Gutknecht et al, 2008-2010]
Inexact and flexible methods

**Motivation:** In SAP, small blocks cannot be solved *exactly*  
→ use *inner iteration*: GMRES, e.g. (we take 6 its.)

**Consequences:**

- SAP is *not* a stationary preconditioner
- We do *not* build up preconditioned Krylov subspaces $K_k(\text{MP}, b)$
- $K_k \rightarrow \text{span}\{b, MP_1 b, (MP_2)(MP_1)b, \ldots\} =: \hat{K}_k$
Inexact and flexible methods

**Motivation:** In SAP, small blocks cannot be solved exactly
→ use *inner iteration:* GMRES, e.g. (we take 6 its.)

**Consequences:**

- SAP is *not* a stationary preconditioner
- We do *not* build up preconditioned Krylov subspaces $K_k(MP, b)$
  - $K_k \rightarrow \text{span}\{b, MP_1 b, (MP_2)(MP_1)b, \ldots\} =: \hat{K}_k$

**But:**

- we can still enforce a conjugate residual property: FGCR
- we can still enforce the minimal residual property: FGMRES (flexible GMRES)
- again: both methods are mathematically equivalent
We saw this earlier . . .

- $16^3 \times 32$ configuration
- $\beta = 5.29$, $\kappa_{\text{sea}} = 0.1350$, $\kappa = 0.13768$, $\kappa_c \approx 0.1377$
A problem with restarts

**Restarted GMRES, GCR, FGMRES:** After every $m$ iterations ($= 1$ cycle):

- stop recurrence, get current iterate $x_c$
- compute current residual $r_c = b - Mx_c$
- next cycle: solve $Mx = b$ with initial guess $x_c$

( $\Leftrightarrow$ solve error equation $Me = r$ with $e_0 = 0$)
A problem with restarts

**Restarted GMRES, GCR, FGMRES:** After every $m$ iterations ($= 1$ cycle):

- stop recurrence, get current iterate $x_c$
- compute current residual $r_c = b - Mx_c$
- next cycle: solve $Mx = b$ with initial guess $x_c$

($\iff$ solve error equation $Me = r$ with $e_0 = 0$)

**Problem:** After restart, all possible information from previous $K_m$ is lost, e.g.

$$r^{m+k} \perp M \cdot K_m$$ in 2nd cycle
A problem with restarts

**Restarted GMRES, GCR, FGMRES:** After every $m$ iterations ($= 1$ cycle):

- stop recurrence, get current iterate $x_c$
- compute current residual $r_c = b - Mx_c$
- next cycle: solve $Mx = b$ with initial guess $x_c$
  
  $(\Leftrightarrow$ solve error equation $Me = r$ with $e_0 = 0)$

**Problem:** After restart, all possible information from previous $K_m$ is lost, e.g.

$$r^{m+k} \not\perp M \cdot K_m \text{ in 2nd cycle}$$

**Augmenting approach:** Keep subspace $U$ of previous $K_m$ and enforce

$$r^{m+k} \perp M \cdot (U + K_k(M, r_m))$$
Which subspace $\mathcal{U}$ for deflation?

**Motivation:** Eigendecomposition of $r$:

$$
    r = \sum \alpha_i y_i \Leftrightarrow e = M^{-1}r = \sum \frac{\alpha_i}{\lambda_i} y_i.
$$

‘Eliminate’ $\alpha_i y_i$ for which $|\lambda_i|$ is small.

**GMRES-DR approach** [Morgan, 2002]:

- ‘eliminate’ by augmenting Krylov space with approximations to small eigenvectors
- after cycle $s$: compute *harmonic Ritz values/vectors*:
  
  $$
  My - \lambda y \perp M(K_m + \mathcal{U}_s), \ y \in K_m + \mathcal{U}_s
  $$

- $\mathcal{U}_{s+1}$ for next cycle: small harmonic Ritz vectors
GMRES-DR, FGMRES-DR

**Important property:**

- $My - \lambda y \parallel r^m$ for harmonic Ritz pairs $(y, \lambda)$
- Orthogonalization process for $K_k + U_s$ can be cast as in implicitly restarted Arnoldi process $\rightarrow$ no mv-multiplications for $U$ part.

Resulting algorithm GMRES-DR can be generalized to FGMRES-DR

[Giraud, Gratton, Pinel, Vasseur, 2010]
Important property:

- \( M y - \lambda y \parallel r^m \) for harmonic Ritz pairs \((y, \lambda)\)
- Orthogonalization process for \( K_k + U_s \) can be cast as in implicitly restarted Arnoldi process \( \rightarrow \) no mv-multiplications for \( U \) part.

Resulting algorithm GMRES-DR can be generalized to FGMRES-DR
[Giraud, Gratton, Pinel, Vasseur, 2010]

By the way: Are there (efficient or inefficient) deflated GCR variants?
We saw this already . . .
...and this, too
Why mixed precision?

The **GPU curse**: Many GPUs (and also the CELL) are much faster in single precision.

**Consequence**: Many lattice QCD codes use iterative refinement:
1. run in single precision until convergence
2. recompute residual $r_c = b - Mx_c$ in *double precision*
3. solve for error $Me = r_c$, again in single precision
4. repeat if necessary
A new challenge for deflation à la FGMRES-DR:

- upon single precision convergence

\[ r_c \parallel r_{hr} \quad (= My_{hr} - \lambda_{hr} y_{hr}, \quad (\lambda_{hr}, y_{hr}) \text{ harmonic Ritz pair}) \]

- FGMRES-DR algorithm can become erroneous (implicitly restarted Arnoldi is corrupted)

- remedies are costly ...
A challenge for deflated restarts

A new challenge for deflation à la FGMRES-DR:

- upon single precision convergence

\[ r_c \parallel r_{hr} \quad (= My_{hr} - \lambda_{hr}y_{hr}, \quad (\lambda_{hr}, y_{hr}) \text{ harmonic Ritz pair}) \]

- FGMRES-DR algorithm can become erroneous (implicitly restarted Arnoldi is corrupted)

- remedies are costly . . .

- . . . and currently under investigation