UKAEA NEPTUNE - Preconditioning

Sue Thorne (PI), Vassil Alexandrov,
Anton Lebedev and Emre Sahin

STFC Hartree Centre

sue.thorne@stfc.ac.uk
vassil.alexandrov@stfc.ac.uk
anton.lebedev@stfc.ac.uk
emre.sahin@stfc.ac.uk

Linear solvers

Discretisation of the PDE => a very large linear system ***Ax=b*** must be solved (or series)

$$\boldsymbol{A} = \begin{bmatrix} A_{1,1} & \cdots & A_{1,n} \\ \vdots & \ddots & \vdots \\ A_{n,1} & \cdots & A_{n,n} \end{bmatrix}$$

$A_{i,i}$ represents discretisation of *i*-th PDE

Iterative methods preferable but convergence slow: $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(k)} \rightarrow \boldsymbol{x}$

Science and Technology Facilities Council

Linear solvers (2)

Want to (approximately) solve $\boldsymbol{Ax=b}$

Instead find $\boldsymbol{P}$ and/or $\boldsymbol{Q}$ and solve $\boldsymbol{By=c}$, where $\boldsymbol{B= P^{-1}AQ^{-1}}$, $\boldsymbol{c= P^{-1}b}$ and $\boldsymbol{x=Q^{-1}y}$

Choose $\boldsymbol{P}$/$\boldsymbol{Q}$ that

- improve rate of convergence
- but "cheap" to generate and apply preconditioner

From SD1D Example (BOUT++)

- Good preconditioner for $A_{1,1}$ is not a good preconditioner for $\boldsymbol{A}$ when n>1

Our approach

$$\boldsymbol{A} = \begin{bmatrix} A_{1,1} & \cdots & A_{1,n} \\ \vdots & \ddots & \vdots \\ A_{n,1} & \cdots & A_{n,n} \end{bmatrix}$$

Identification of key PDEs and literature search to assess suitability of known preconditioners

- PDE-based (multigrid,…)
- Sparse approximate inverses (Markov Chain Monte Carlo Matrix Inverse,…)

Development of implicit-factorization preconditioners

- Make use of block structure and preconditioners for $A_{i,i}$
- Comparison of implicit-factorization versus SPAI applied to A

Guide NEPTUNE Programme in choice of preconditioner

UKRI Science and Technology Facilities Council

Key deliverables

- Literature surveys
  - [D1.1] Elliptic
  - [D2.1] Hyperbolic
  - [D3.1] Sparse approximate inverse

- [D4.1] Implicit-factorization preconditioner report
  - Proposed preconditioners
    - Loosely-coupled
    - Strongly-coupled
  - Theoretical convergence results
  - Reduced subspace iteration methods

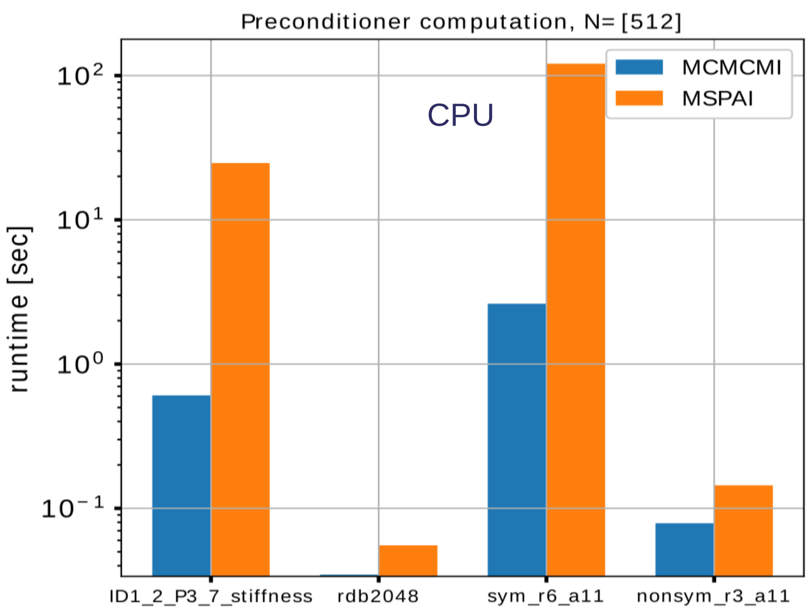
- [D6.1-6.4] Comparison report and [D5.1-5.2] prototype code
  - Recommendations

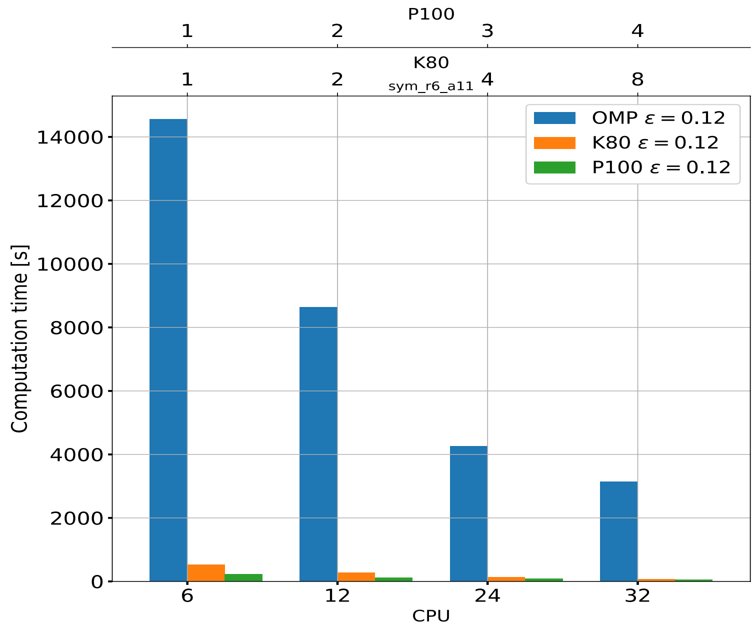
Markov Chain Monte Carlo for Approximate Matrix Inversion (MCMCMI)

- Sparse approximate inverse preconditioners are based on forming a sparse approximation $\boldsymbol{G} \approx \boldsymbol{A}^{-1}$
  - Modified Sparse Approximate Inverse is most widely used

- MCMCMI preconditioner
  - The main idea of the method is to employ Neumann series which is evaluated stochastically using Markov Chains to compute an approximate inverse of a diagonally dominant matrix.
  - CPU parallelization combining MPI and OpenMP (OMP)
  - Multiple GPU

Markov Chain Monte Carlo for Approximate Matrix Inversion (MCMCMI)





MATRIX SET.

Matrix	Dimension	Non-zeros	Sparsity
ID1_2_P3_7_stiffness	$514,369 \times 514,369$	8,702,911	0.003%
nonsym_r3_a11	$20,930 \times 20,930$	638,733	0.15%
rdb2048_noL	$2,048 \times 2,048$	12,032	0.29%
sym_r6_a11	$1,314,306 \times 1,314,306$	36,951,316	0.02%

Science and Technology Facilities Council

Lebedev, Alexandrov “On advanced Monte Carlo methods for linear algebra on advanced accelerator architectures” ScalA2018

Implicit-factorization preconditioners

$$\boldsymbol{A} = \boldsymbol{A}^T = \begin{bmatrix} H_{1,1} & H_{1,2} & A_{1,3} \\ H_{2,1} & H_{2,2} & A_{2,3} \\ A_{3,1} & A_{3,2} & C \end{bmatrix}$$

Family 2

$$\boldsymbol{P} = \begin{bmatrix} 0 & 0 & A_{1,3} \\ 0 & G_{2,2} & A_{2,3} \\ G_{3,1} & 0 & G_{3,3} \end{bmatrix} \begin{bmatrix} B_{1,1} & 0 & B_{1,3} \\ 0 & B_{2,2} & 0 \\ B_{3,1} & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & G_{1,3} \\ 0 & G_{2,2} & 0 \\ A_{3,1} & A_{3,2} & G_{3,3} \end{bmatrix}$$

$$G_{3,1} = B_{3,1}^T, \qquad G_{3,3} + G_{3,3}^T + G_{3,1}B_{1,1}G_{3,1}^T = C$$

Total of 14 families derived with constrained optimization as application

Science and Technology Facilities Council

Implicit-factorization preconditioners

PDE-constrained optimization problem

Example 5.1. *Let* $\Omega = [0,1]^m$, *where* $m = 2, 3$, *and consider the problem*

$$\min_{u,f} \frac{1}{2}||u - \hat{u}||^2_{L_2(\Omega)} + \beta||f||^2_{L_2(\Omega)}$$

$$\text{s.t.} \quad -\nabla^2 u = f \text{ in } \Omega \tag{5.1}$$

$$u = \hat{u}|_{\partial\Omega} \text{ on } \partial\Omega \tag{5.2}$$

Implicit-factorization preconditioner using multigrid within some sub-blocks

h	3n	backslash	MINRES ($\mathcal{P}_{D2}$)	MINRES ($\mathcal{P}_{D3}$)	PPCG ($\mathcal{P}_{C2}$)	PPCG ($G = diag(A)$)
			Convergence tolerance $= 10^{-4}$			
2^{-2}	81	0.001	0.15 (10)	0.15 (11)	0.02 (5)	0.01 (6)
2^{-3}	1029	0.13	0.34 (14)	0.48 (23)	0.10 (5)	0.18 (6)
2^{-4}	10125	18.9	2.74 (14)	4.36 (23)	1.58 (5)	18.8 (5)
2^{-5}	89373	—	26.5 (14)	44.1 (24)	15.7 (4)	—
			Convergence tolerance $= 10^{-8}$			
2^{-2}	81	0.001	0.053 (12)	0.060 (17)	0.03 (7)	0.01 (10)
2^{-3}	1029	0.13	0.25 (18)	0.45 (31)	0.15 (8)	0.34 (14)
2^{-4}	10125	18.9	3.76 (18)	7.29 (37)	2.21 (8)	36.4 (13)
2^{-5}	89373	—	44.0 (18)	85.3 (37)	25.6 (8)	—

Thank you!