A simple proposal for parallel implicit time evolution

Andy Wathen
Oxford University, UK

joint work with
Elle McDonald

Fox Prize 2015

- for a Numerical Analysis paper
- from researchers under 31 years old
- shortlisted presentations Monday June 22nd, 2015, University of Strathclyde, Glasgow
- entries should be submitted by January 31, 2015
- Google: Fox Prize 2015

Heat equation:

$$\frac{\partial u}{\partial t} - \nabla^2 u = f \quad \text{in } \Omega \times (0,T], \quad \Omega \subset \mathbb{R}^2 \text{ or } \mathbb{R}^3$$

$$u(\mathrm{x},0) = u_0(\mathrm{x}), \quad u \text{ given on } \partial\Omega$$

Finite elements in space (x), θ time stepping gives

$$M\frac{\mathbf{u}_k - \mathbf{u}_{k-1}}{\tau} + K\Big(\theta \mathbf{u}_k + (1-\theta)\mathbf{u}_{k-1}\Big) = \mathrm{f}_k$$

$M \in \mathbb{R}^{n\times n}$: SPD mass matrix (identity operator, but same sparsity as K

$K \in \mathbb{R}^{n\times n}$: SPD discrete (negative) Laplacian (stiffness matrix)

Rearranging:

$$\Big(M + \tau\,\theta\,K\Big)\mathbf{u}_k = \Big(M - \tau\,(1-\theta)\,K\Big)\mathbf{u}_{k-1} + \tau\,\mathbf{f}_k,$$

$$k = 1, 2, \ldots, N$$

$N\tau = T$

Recall for unconditional stability: $\frac{1}{2} \le \theta \le 1$

$\theta = 1$: backwards Euler, $\theta = \frac{1}{2}$: Crank-Nicolson

else need $\tau = \mathcal{O}(h^2)$: very small time steps for explicit method

$$\Big(M + \tau\,\theta\,K\Big)\mathbf{u}_k = \Big(M - \tau\,(1-\theta)\,K\Big)\mathbf{u}_{k-1} + \tau\,\mathbf{f}_k,$$

$$k = 1, 2, \ldots, N$$

Standard solution method:

Solve the N separate $n \times n$ linear systems sequentially for $k = 1, 2, \ldots, N$ e.g. by algebraic multigrid (we use HSL_MI20)

$\Rightarrow r = 5$ V-cycles for solution of each linear system to a relative residual tolerence of 10^{-6}

Hence if we (quite reasonably) regard 1 V-cycle as the main unit of work

$\Rightarrow Nr$ V-cycles sequentially for the overall solution

Alternative proposal for parallel computation:

Write all timesteps at one go (all-at-once method):

$$\mathcal{A}\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_N \end{bmatrix} = r.h.s$$

where $\mathcal{A}$ is the matrix

$$\begin{bmatrix} M+\tau\theta K & 0 & 0 & 0 \\ -M+\tau(1-\theta)K & M+\tau\theta K & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & -M+\tau(1-\theta)K & M+\tau\theta K \end{bmatrix}$$

and $r.h.s. = [M-\tau(1-\theta)K\mathbf{u}_0+\tau\,\mathbf{f}_1, \tau\,\mathbf{f}_2, \ldots, \tau\,\mathbf{f}_N]^T$

$$\boldsymbol{\mathcal{A}} = \begin{bmatrix} M+\tau\theta K & 0 & 0 & 0 \\ -M+\tau(1-\theta)K & M+\tau\theta K & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & -M+\tau(1-\theta)K & M+\tau\theta K \end{bmatrix}$$

$\boldsymbol{\mathcal{A}} \in \mathbb{R}^{L\times L}$, $L = Nn$

We propose to solve this huge linear system (for the solution at all time steps) by GMRES (or BICGSTAB) with block diagonal preconditioner

$$\boldsymbol{\mathcal{P}} = \begin{bmatrix} (M+\tau\theta K)_{MG} & 0 & 0 & 0 \\ 0 & (M+\tau\theta K)_{MG} & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ 0 & 0 & 0 & (M+\tau\theta K)_{MG} \end{bmatrix}$$

where $(M+\tau\theta K)_{MG}$ is one AMG V-cycle exactly as above

Theory: If we used

$$\mathcal{P}_{\text{exact}} = \begin{bmatrix} (\boldsymbol{M}+\boldsymbol{\tau\theta K}) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & (\boldsymbol{M}+\boldsymbol{\tau\theta K}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & (\boldsymbol{M}+\boldsymbol{\tau\theta K}) \end{bmatrix}$$

as preconditioner (no AMG approximation) then we would have

$$\mathcal{P}_{\text{exact}}^{-1}\boldsymbol{\mathcal{A}} = \begin{bmatrix} \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{J} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{J} & \boldsymbol{I} \end{bmatrix},$$

$$\boldsymbol{J} = (\boldsymbol{M}+\boldsymbol{\tau\theta K})^{-1}(-\boldsymbol{M}+\boldsymbol{\tau}(1-\boldsymbol{\theta})\boldsymbol{K})$$

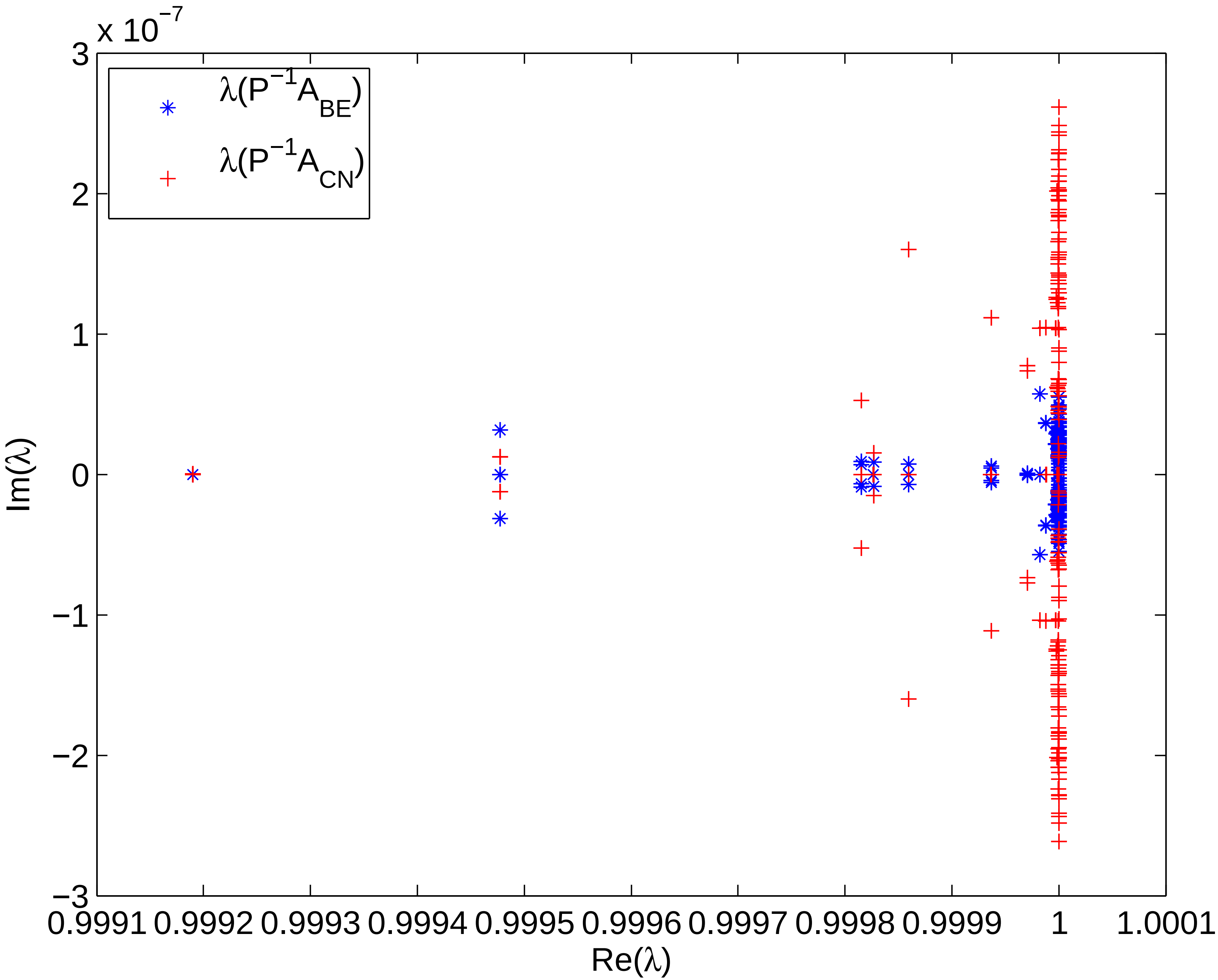
For

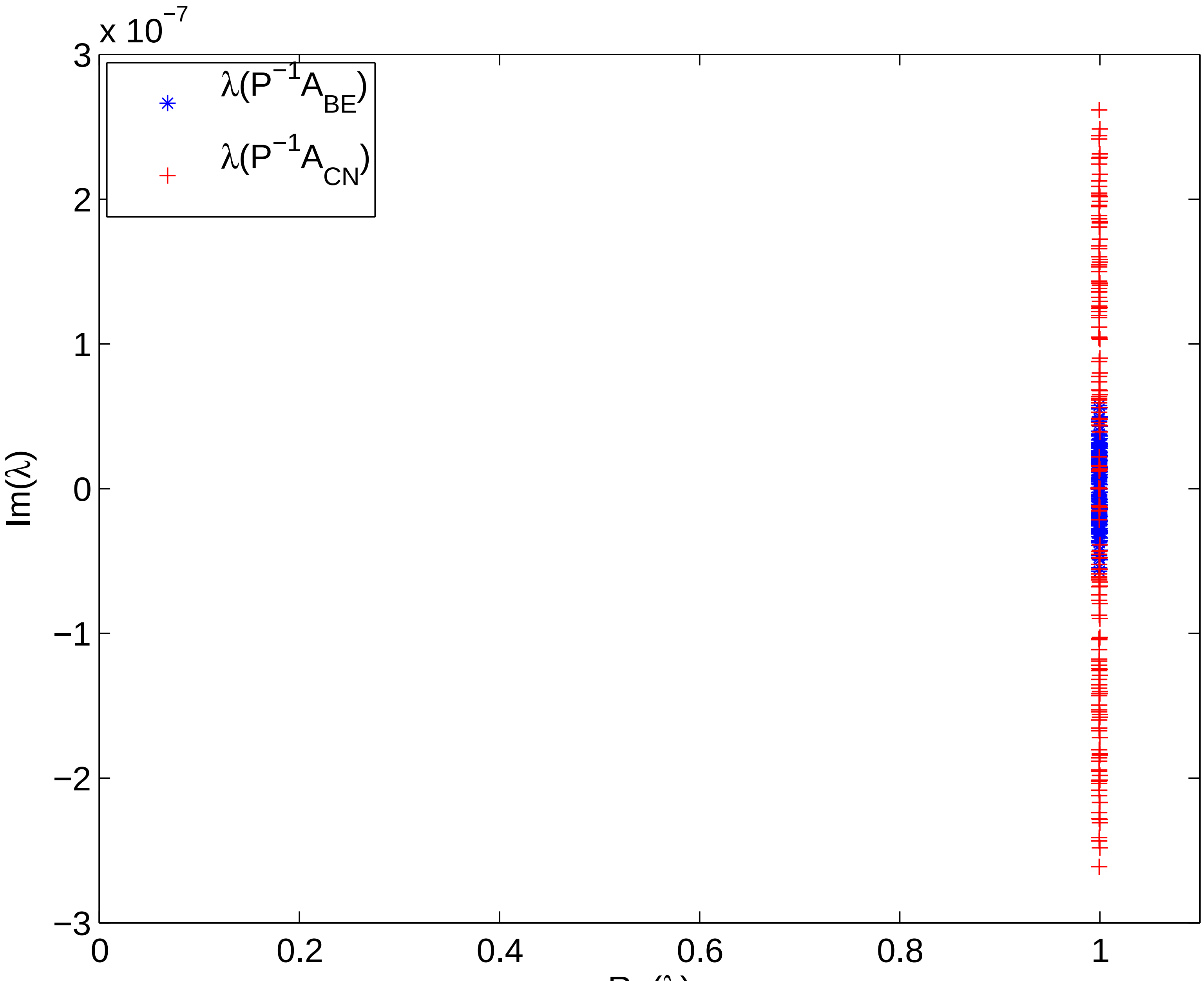
$$\mathcal{P}_{\text{exact}}^{-1}\mathcal{A} = \begin{bmatrix} \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{J} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{J} & \boldsymbol{I} \end{bmatrix},$$

the minimum polynomial is $(1 - s)^N$, so GMRES would terminate (in exact arithmetic) in N iterations

We observe that $(M+\tau\theta K)_{MG}$ is spectrally so close to $(M+\tau\theta K)$ that convergence to a tolerance much less than the discretization error is achieved in N iterations also with $\mathcal{P}$ as preconditioner.

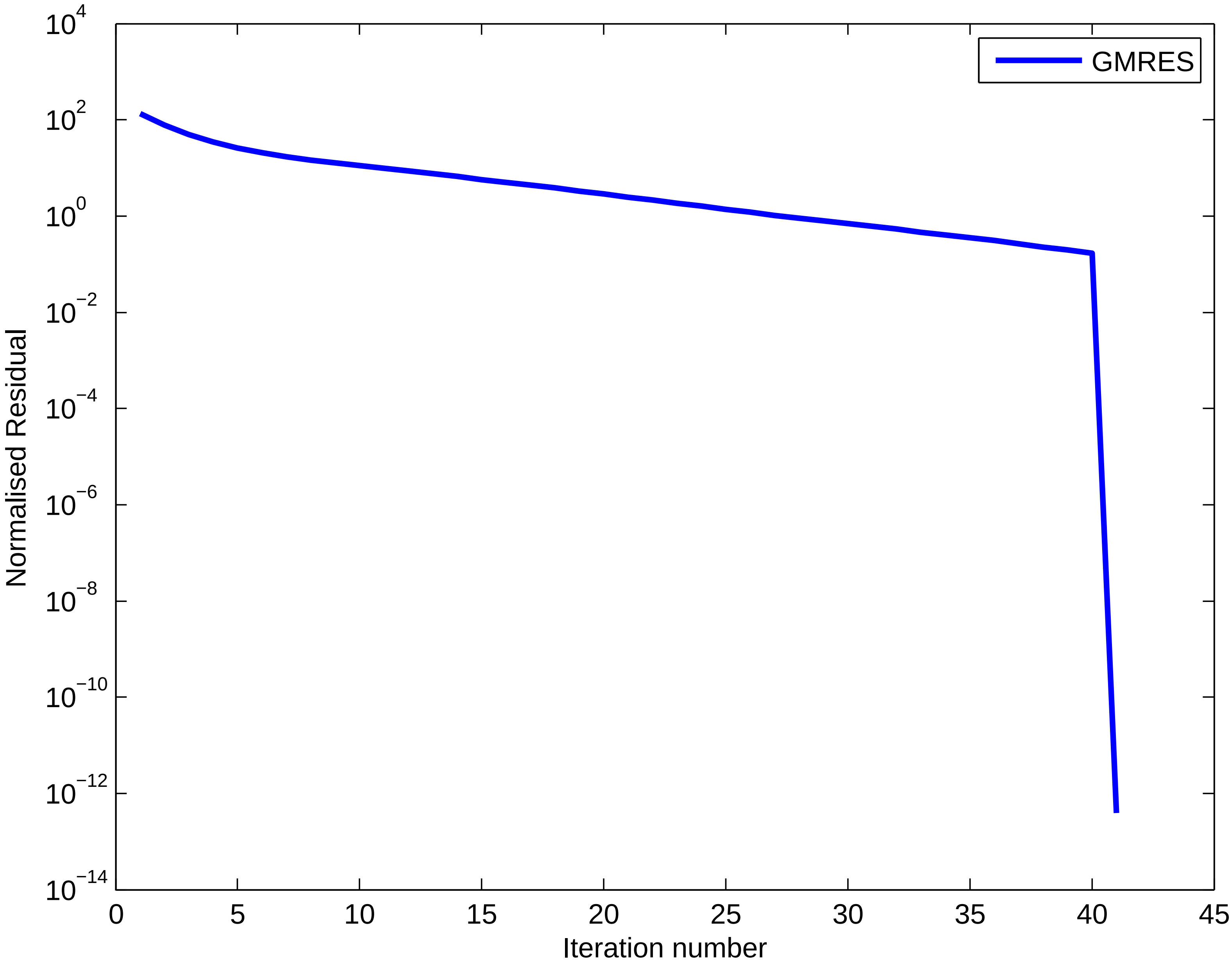
For N=5:


x 10^−7
3
2
1
0
−1
−2
−3
λ(P^−1A_BE)
λ(P^−1A_CN)
Im(λ)
0.9991 0.9992 0.9993 0.9994 0.9995 0.9996 0.9997 0.9998 0.9999 1 1.0001
Re(λ)



x 10^−7
3
2
1
0
−1
−2
−3
λ(P^−1A_BE)
λ(P^−1A_CN)
Im(λ)
0
0.2
0.4
0.6
0.8
1
Re(λ)

For N=40:



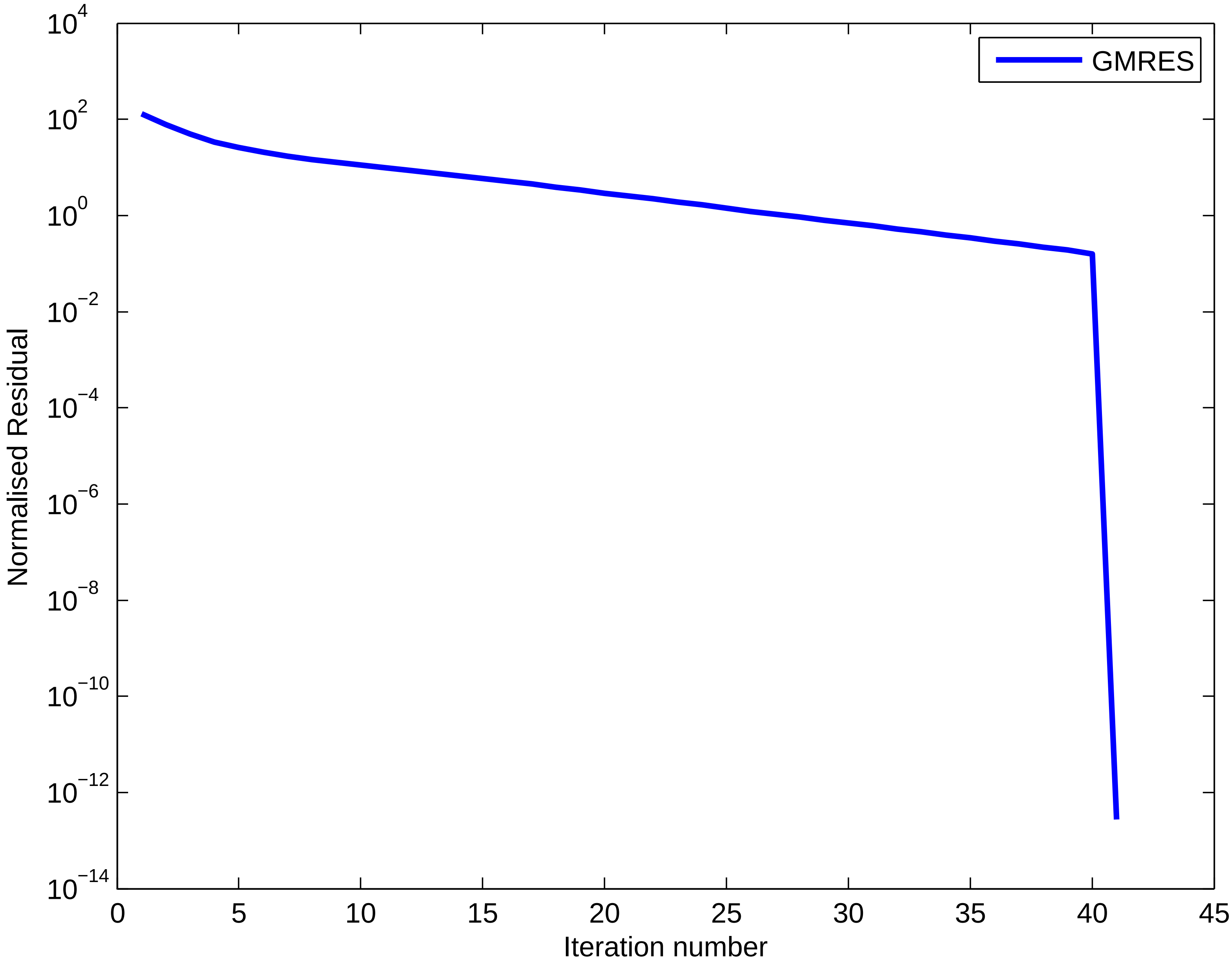
GMRES
Normalised Residual
Iteration number
10^{4}
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
0
5
10
15
20
25
30
35
40
45

Thus: N V-cycles for each of N GMRES iterations—hence N^2 ($> Nr$) overall.

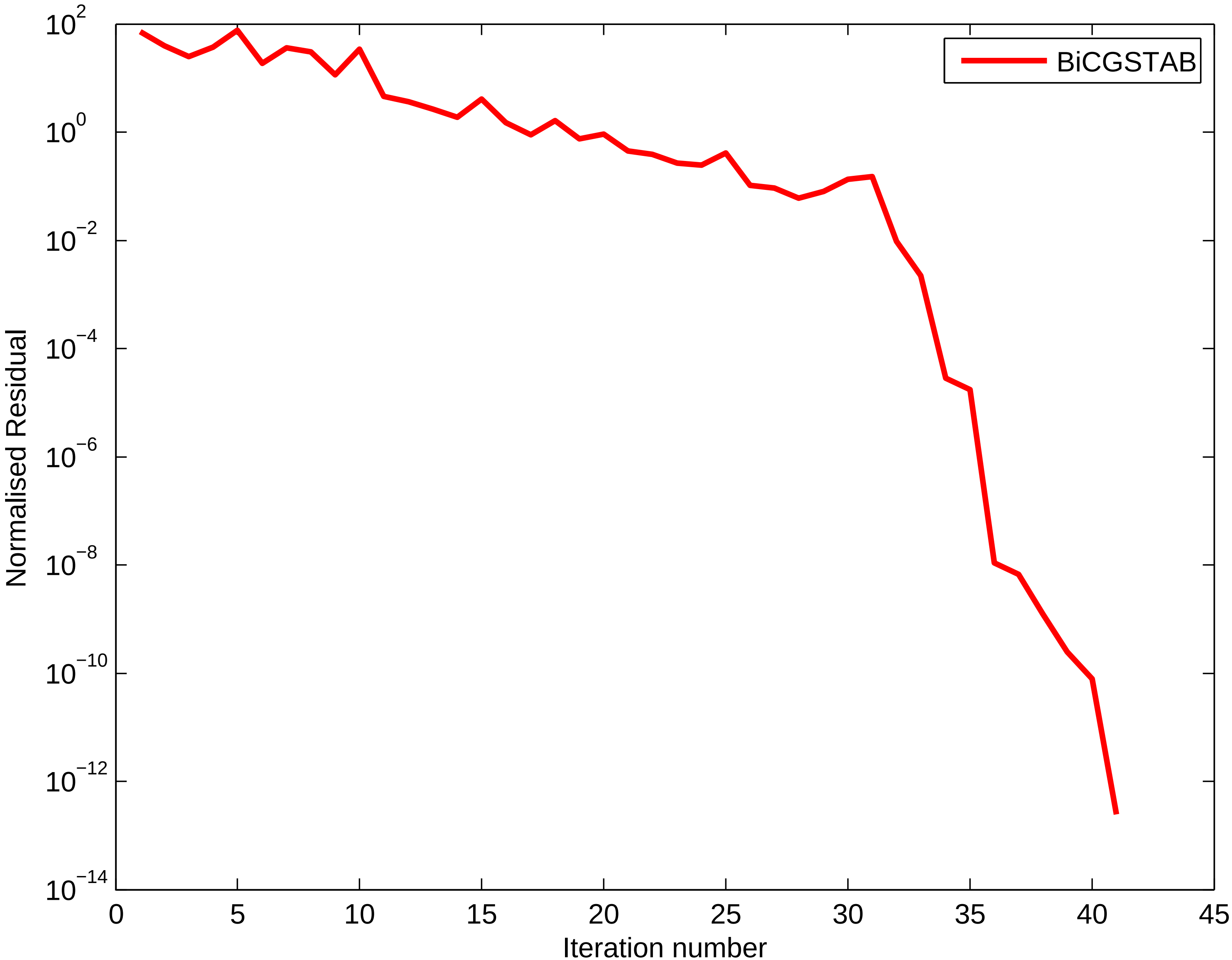
BUT with N processors, solution with $\mathcal{P}$ is (embarrasingly) parallel—block diagonal $\Rightarrow$ independent computation.

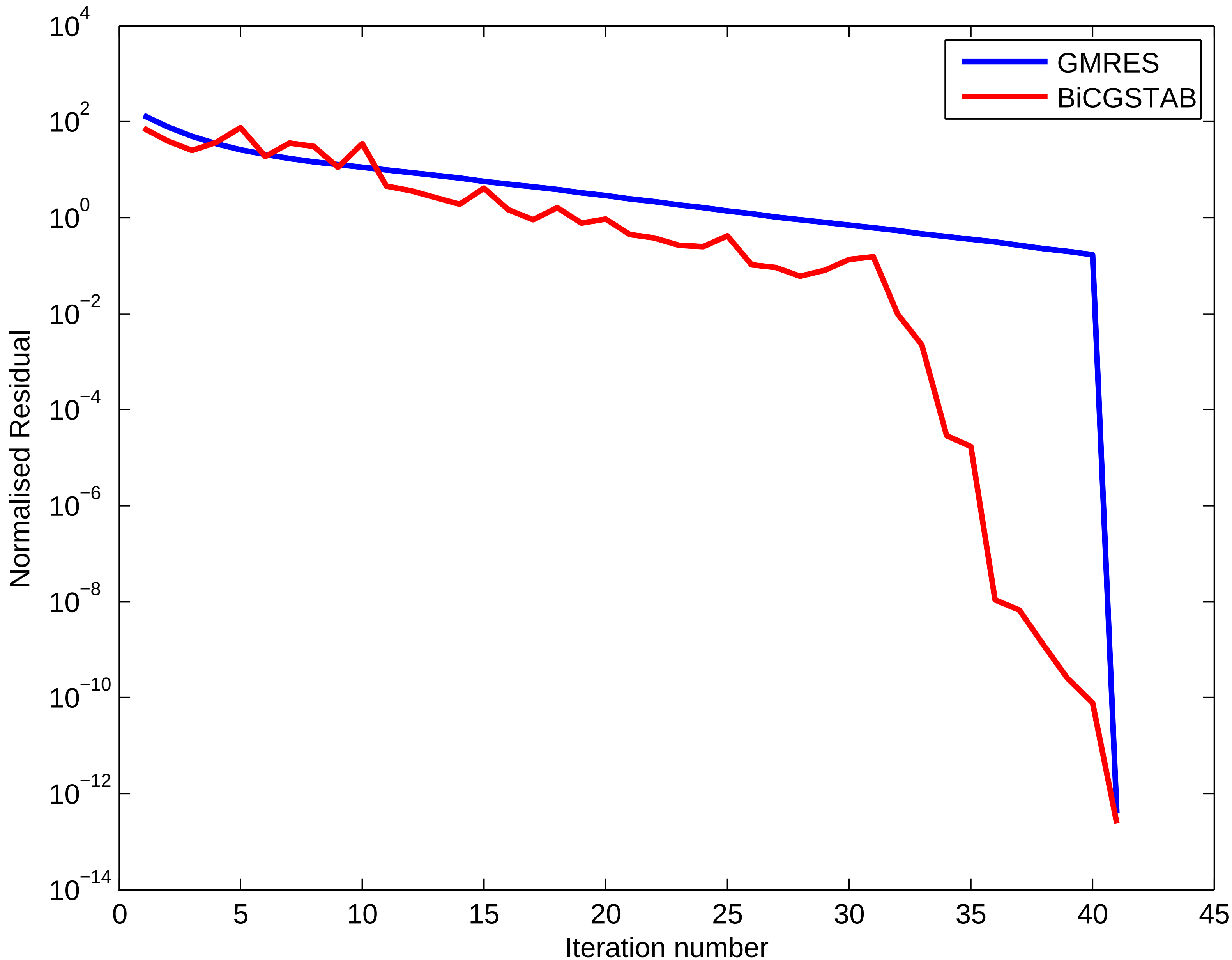
Thus parallel effort is $N < Nr$ (= sequential effort).

For another problem:

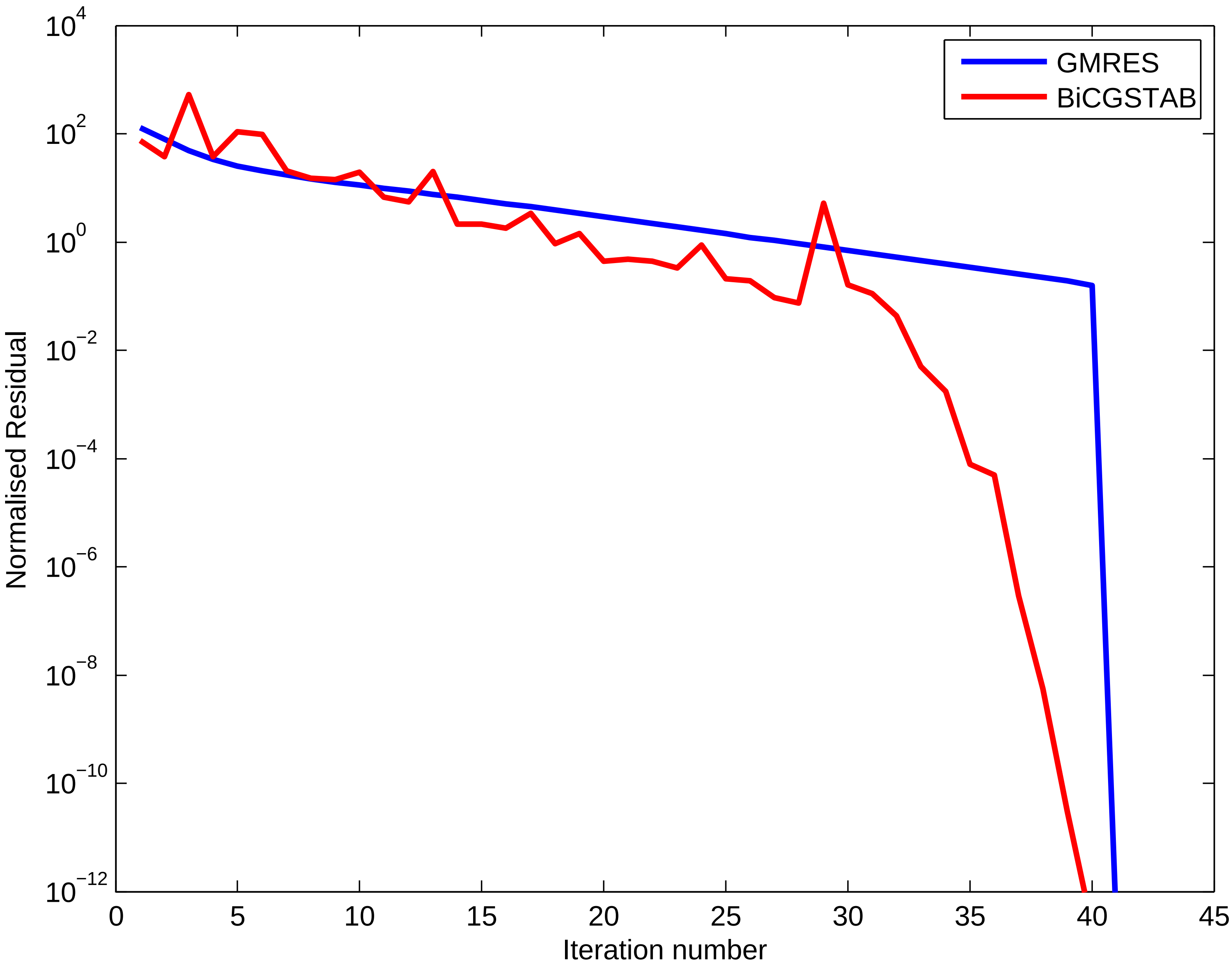


Large N means increasing work (of Arnoldi orthogonalisation) with GMRES : use BICGSTAB :




GMRES
BiCGSTAB
Normalised Residual
Iteration number
10^4
10^2
10^0
10^−2
10^−4
10^−6
10^−8
10^−10
10^−12
10^−14
0
5
10
15
20
25
30
35
40
45

For another problem:



Backwards Euler

h	τ	N	DoF	GMRES	BiCGStab
2^{-3}	2^{-3}	40	3240	40	38
2^{-4}	2^{-4}	80	23120	80	78
2^{-5}	2^{-5}	160	174240	160	157
2^{-3}	2^{-3}	40	3240	40	38
2^{-4}	2^{-4}	40	11560	40	40
2^{-5}	2^{-5}	40	43560	40	43
2^{-6}	2^{-6}	40	169000	43	44
2^{-7}	2^{-7}	40	665640	45	45
2^{-8}	2^{-8}	40	2641960	46	45

Crank-Nicholson

h	τ	N	DoF	GMRES	BiCGStab
2^{-3}	2^{-5}	32	2592	33	35
2^{-4}	2^{-6}	64	18496	66	68
2^{-5}	2^{-7}	128	139392	132	138
2^{-3}	2^{-5}	32	2592	34	35
2^{-4}	2^{-6}	32	9248	35	34
2^{-5}	2^{-7}	32	34848	37	35
2^{-6}	2^{-8}	32	135200	39	36
2^{-7}	2^{-9}	32	532512	40	38
2^{-8}	2^{-10}	32	2113568	38	39

Summary

For a simple linear PDE problem our proposal should achieve

N work on N processors (N^2 work on 1 processor)

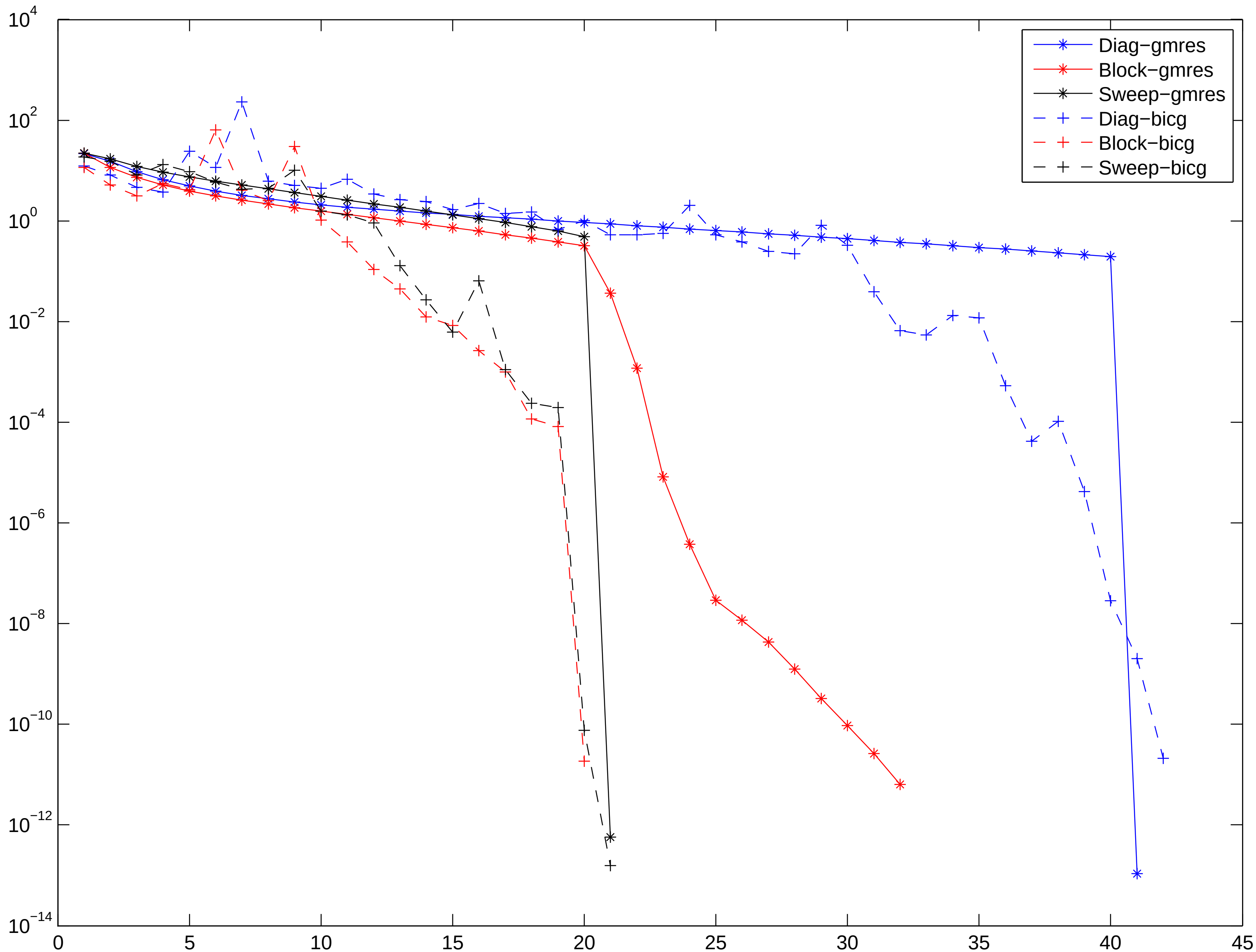
compared to

Nr work for the standard sequential algorithm

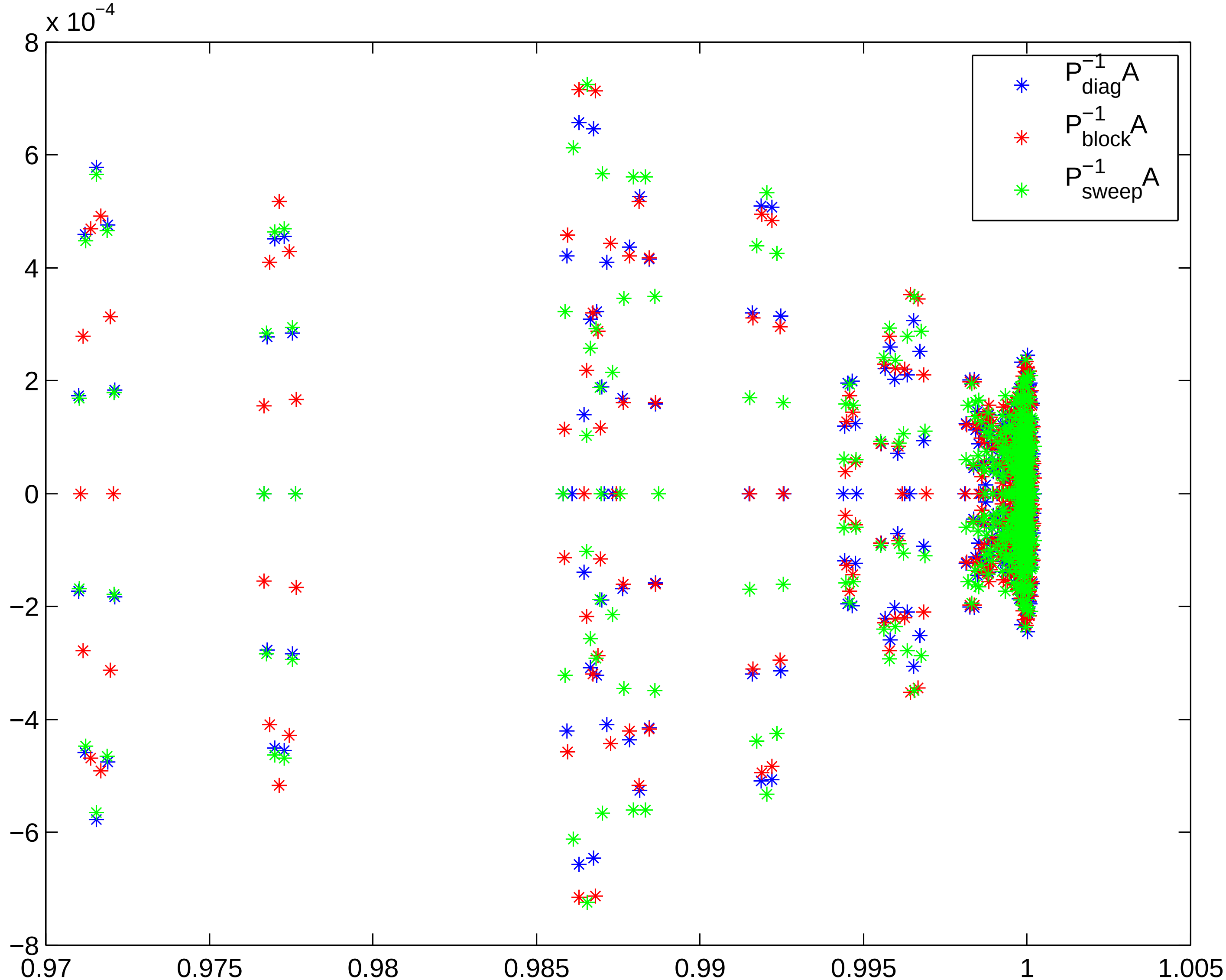
$\frac{1}{2}Nr$ work (?) for Parareal (?)

Acknowledgement

This work is partially supported by Award No. KUK-C1-013-04 made by King Abdullah University of Science and Technology (KAUST)



Diag−gmres
Block−gmres
Sweep−gmres
Diag−bicg
Block−bicg
Sweep−bicg
10^4
10^2
10^0
10^−2
10^−4
10^−6
10^−8
10^−10
10^−12
10^−14
0
5
10
15
20
25
30
35
40
45


x 10^{-4}
8
6
4
2
0
−2
−4
−6
−8
0.97
0.975
0.98
0.985
0.99
0.995
1
1.005
$P_{diag}^{-1}A$
$P_{block}^{-1}A$
$P_{sweep}^{-1}A$