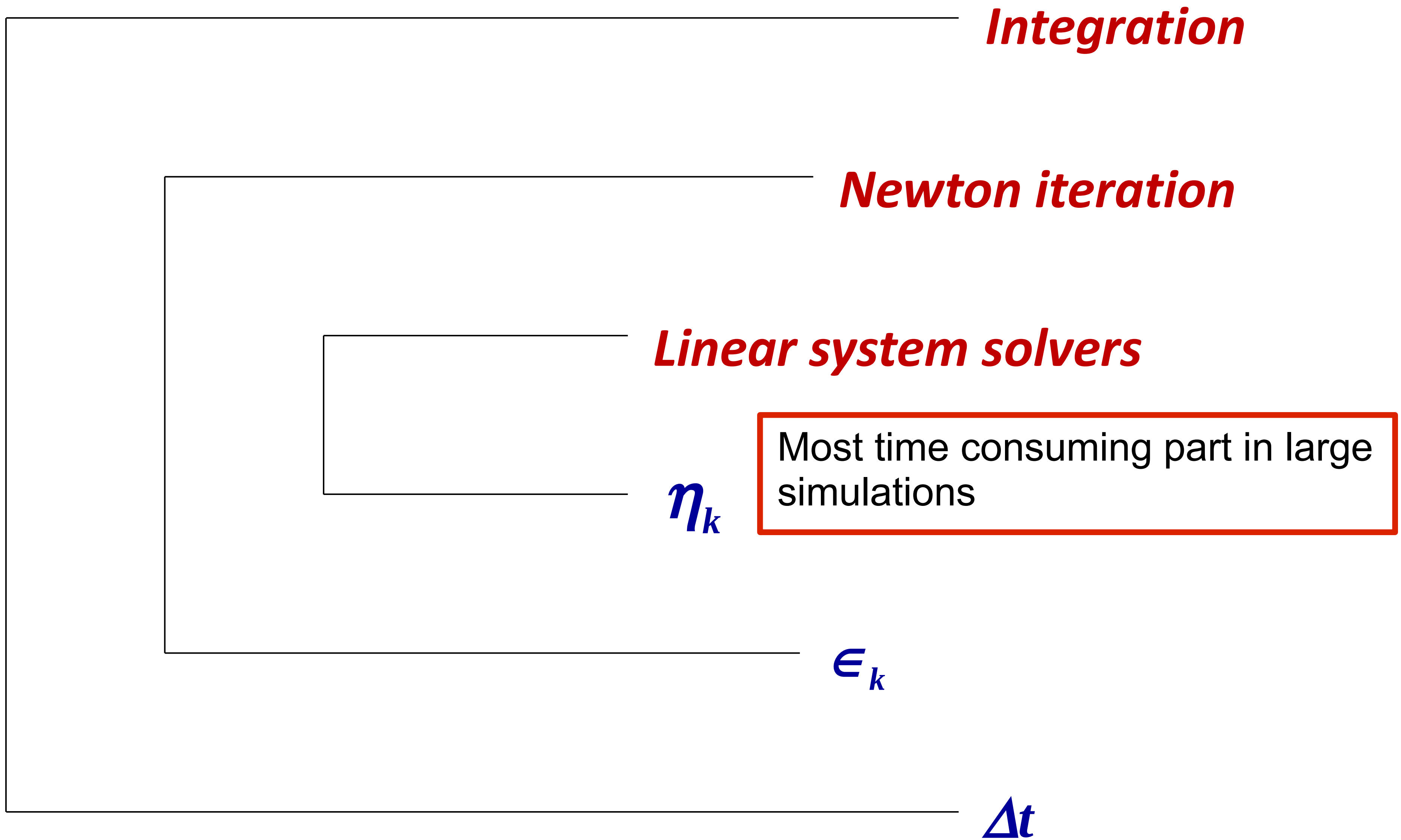
Sparse Matrix Computations in Arterial Fluid Mechanics

Murat Manguoğlu	Middle East Technical University, Turkey
Kenji Takizawa	Waseda University, Japan
Ahmed Sameh	Purdue University, USA
Tayfun Tezduyar	Rice University, USA

Sparse Days Meeting, Sept 6-7 2011
Toulouse, France

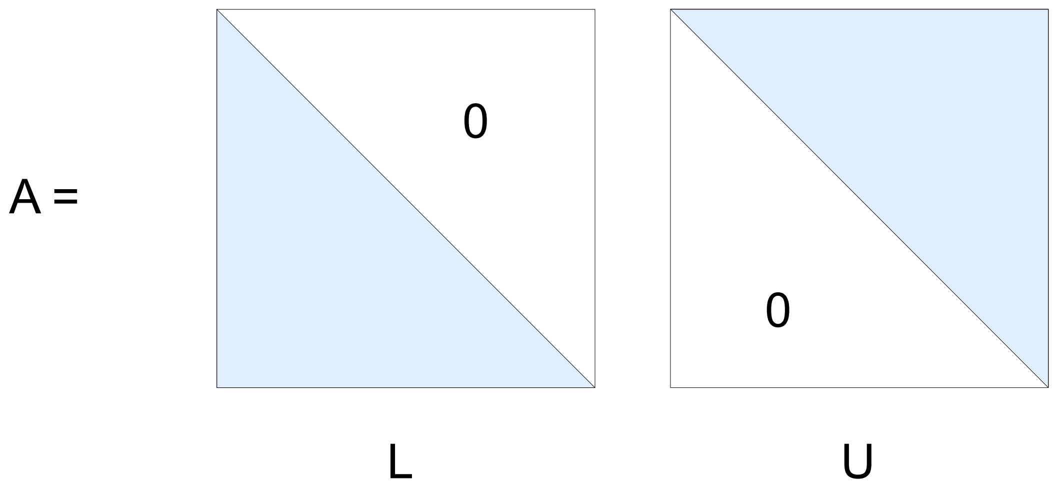
Target Computational Loop

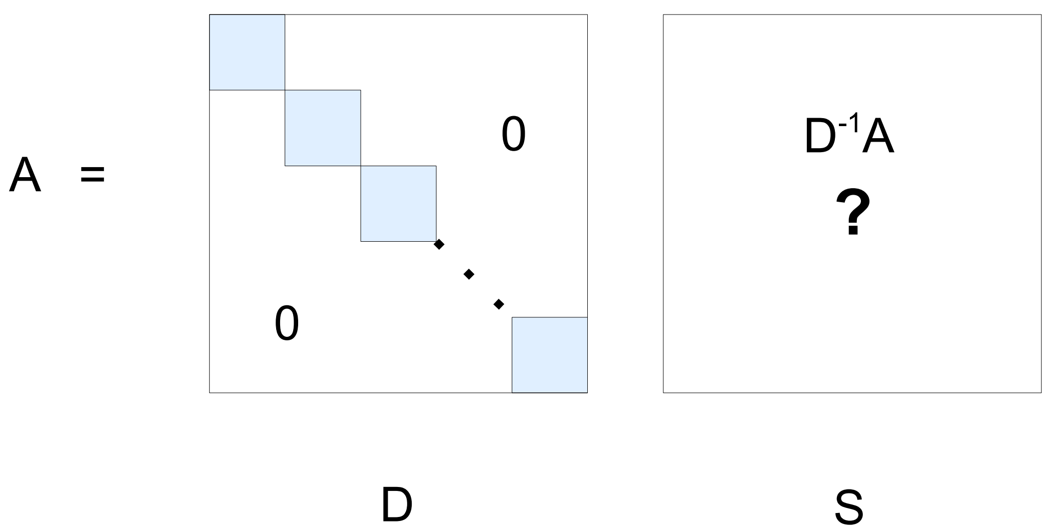


Design principles

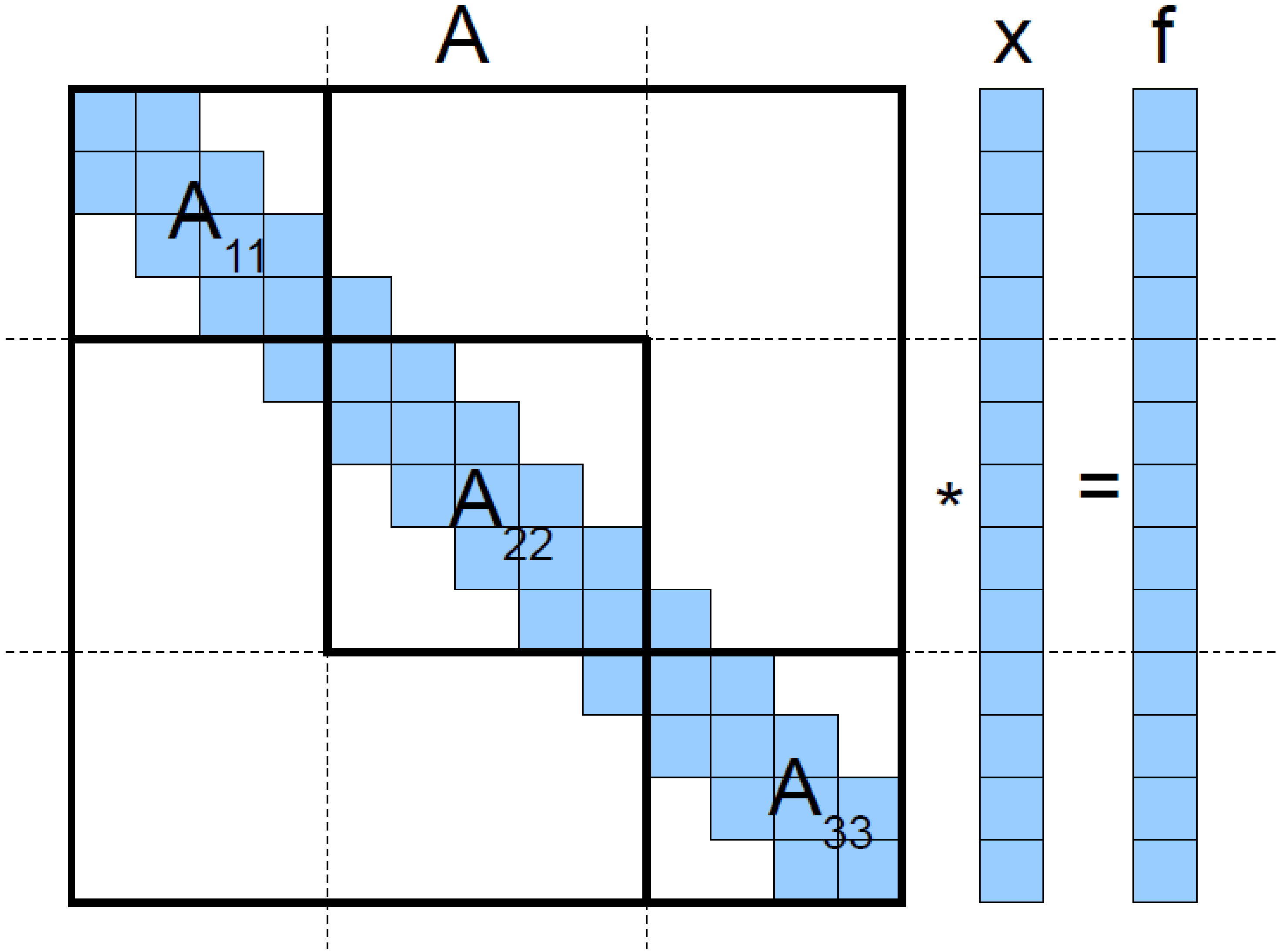
- ***Reducing memory references and interprocessor communications at the cost of extra arithmetic operations.***
- ***Allowing multiple levels of parallelism.***
- ***Creating many algorithms – versions vary depending on architectural characteristics of the underlying computing platform.***

DS factorization vs LU factorization

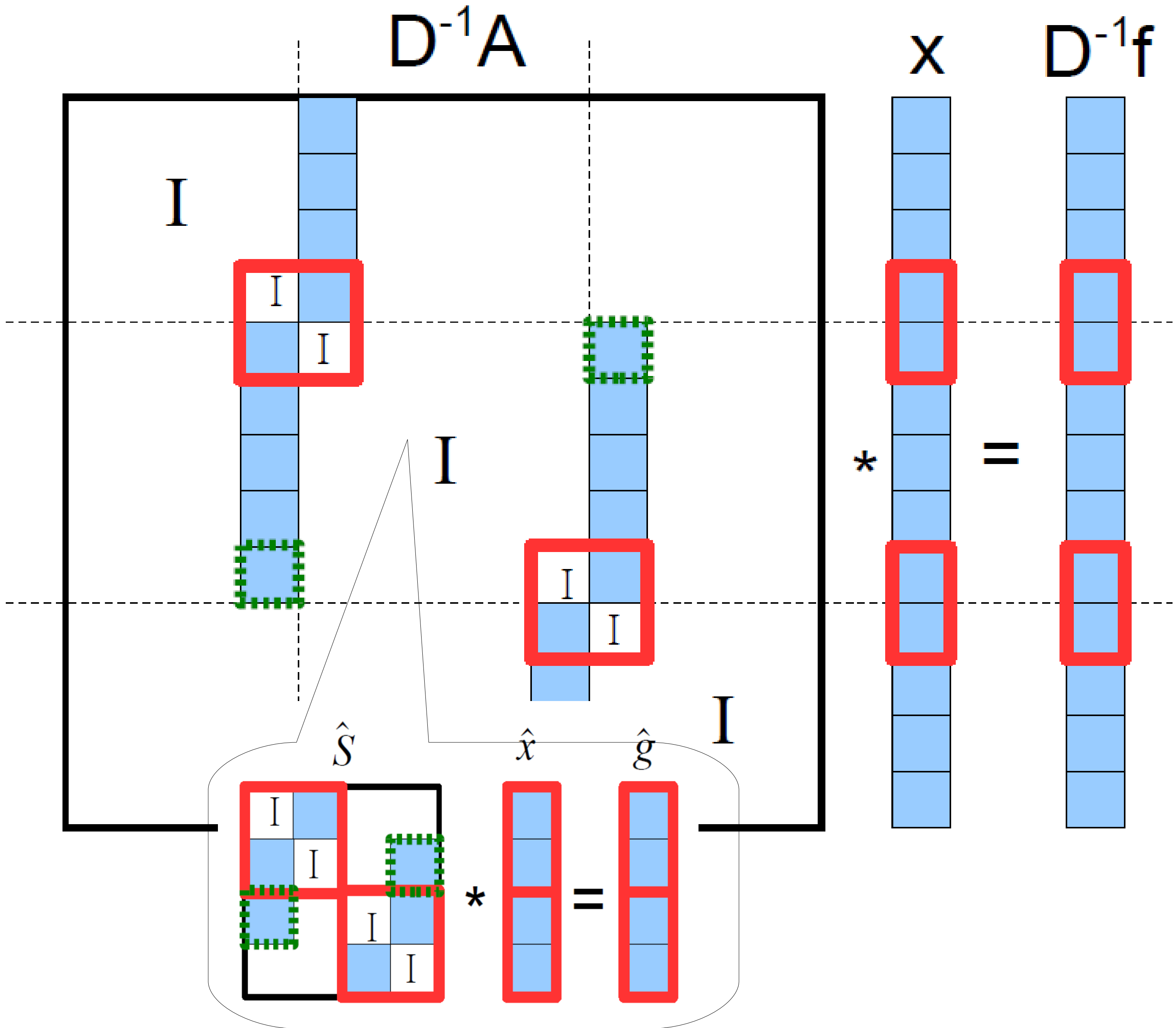




Parallel solution of banded linear systems (SPIKE algorithm)



A. H. Sameh, D. J. Kuck, On stable parallel linear system solvers, J. ACM 25 (1) (1978) 81–91.



D-1A
x
D-1f
I
I
I
I
I
I
I
*
=
Ŝ
x̂
ĝ
I
I
I
I
*
=

The banded SPIKE scheme: $A\,x = f$

Use an outer Krylov subspace method if any Ak is factored via diagonal boosting

BiCGstab

Solve $Pz = r$; (via D-S factorization))

SPIKE vs. MKL

SPIKE vs. Lapack & ScaLapack

A Matrix-Market collection of banded systems (n > 10,000)

filename	*kl*	*ku*	*N*	*~cond*
1- misc/CYLSHELL/s3dkq4m2.csr	614	614	90449	N/A
2- misc/CYLSHELL/s3dkt3m2.csr	614	614	90449	N/A
3- SPARSKIT/FIDAP/fidap035.csr	244	247	19716	4.33E+12
4- SPARSKIT/DRIVCAV/e40r0000.csr	451	451	17281	2.19E+08
5- SPARSKIT/DRIVCAV/e40r5000.csr	451	451	17281	2.20E+10
6-Harwell-Boeing/bcsstruc3/bcsstk25.csr	292	292	15439	1.28E+13
7-Harwell-Boeing/bcsstruc2/bcsstk18.csr	1243	1243	11948	6.49E+11
8-Harwell-Boeing/bcsstruc2/bcsstk17.csr	521	521	10974	1.95E+10

MKL vs SPIKE on Clovertown

	MKL 1 CPU	MKL 2 CPUs	SPIKE-TA3 2 CPUs
Avg. time (normalized)	1	8.5	0.4
Relres	$O(10^{-3})$ to $O(10^{-10})$	$O(10^{-2})$ to $O(10^{-10})$	$O(10^{-5})$ to $O(10^{-11})$

Solving Ax=f

Step 1: *weighted matrix reordering (symmetric/nonsymmetric)*

Step 2: *extract a "banded" preconditioner,M, and solve the system:*

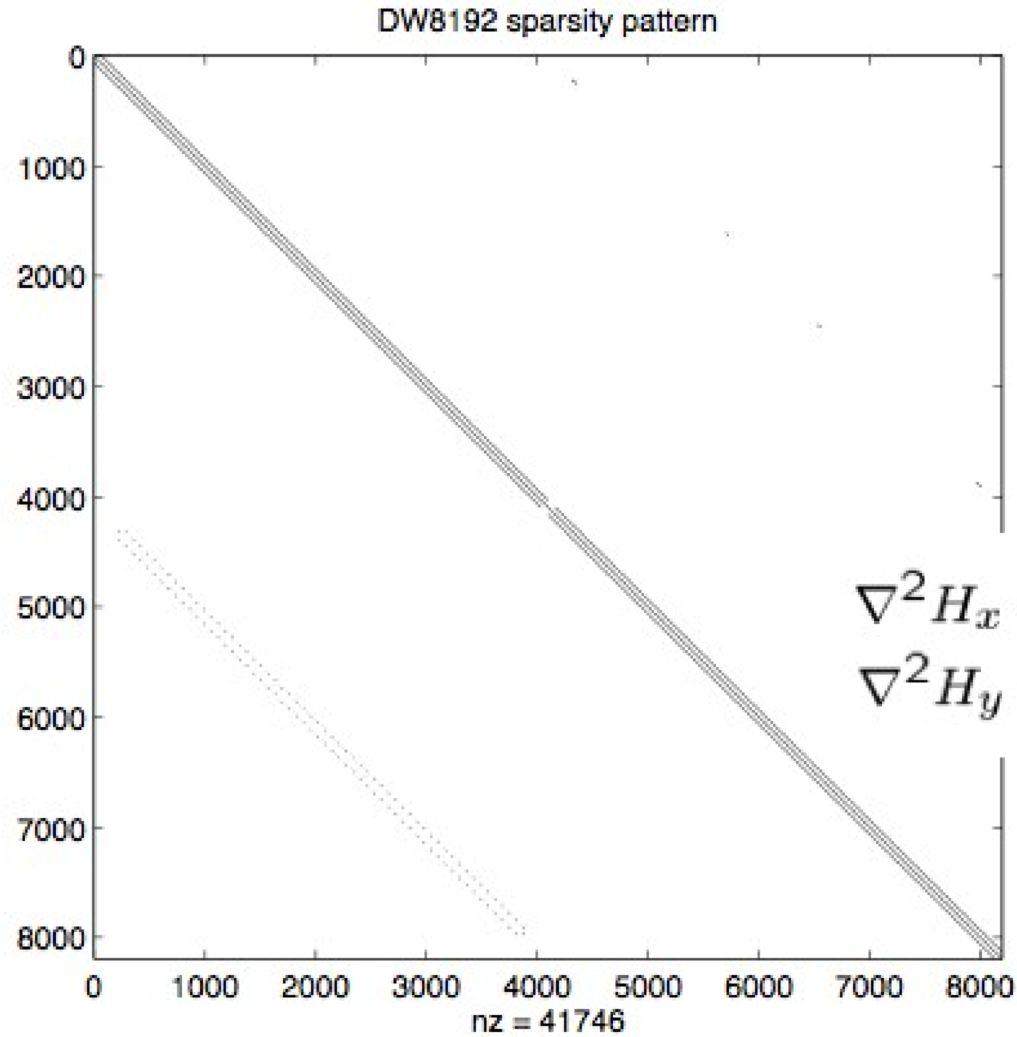
BiCGstab

(or any other Krylov subspace method)

Solve Mz = r using DS fact. ***(M is "banded")***

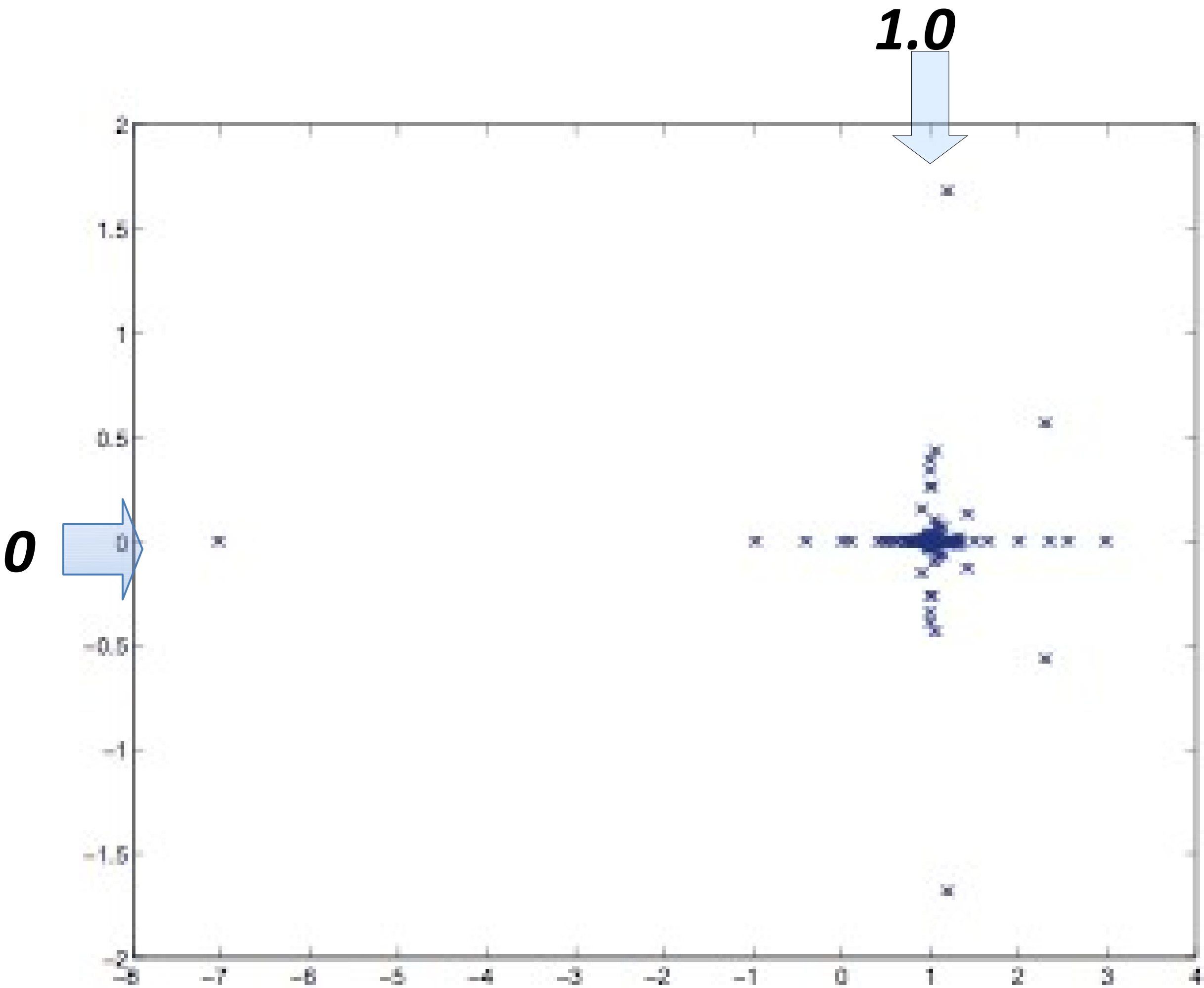
Manguoglu M, Koyuturk M, Sameh A, Grama A, SIAM Journal on Scientific Computing, 32(3), pp.1201-1216, 2010

Computational Electromagnetics UFL: DW8192



Discretization of the Helmholtz equation (2D):

$$
\begin{aligned}
\nabla^2 H_x + k^2 n^2(x,y) H_x &= \beta^2 H_x, \\
\nabla^2 H_y + k^2 n^2(x,y) H_y &= \beta^2 H_y.
\end{aligned}
$$



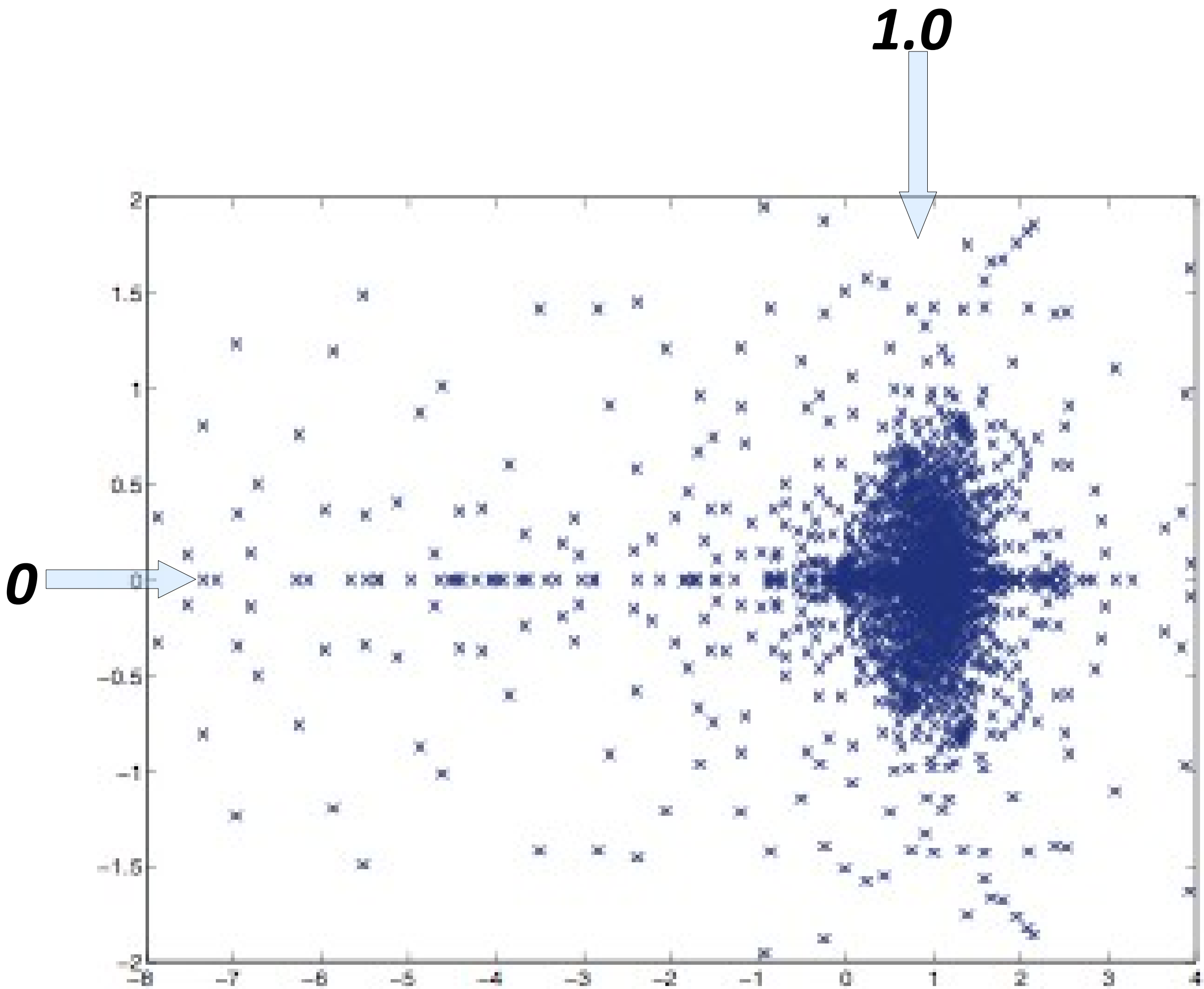
System based on sparse matrix DW8192:

- *n = 8192*
- *nnz = 41,746*
- *κ = O(107)*

Spectrum of $M^{-1}A$

Weighted matrix reordering + narrow-banded preconditioner: M

- $\varepsilon = 10^{-4}$
- half-bandwidth $\beta \leq 50$



System based on sparse matrix DW8192:

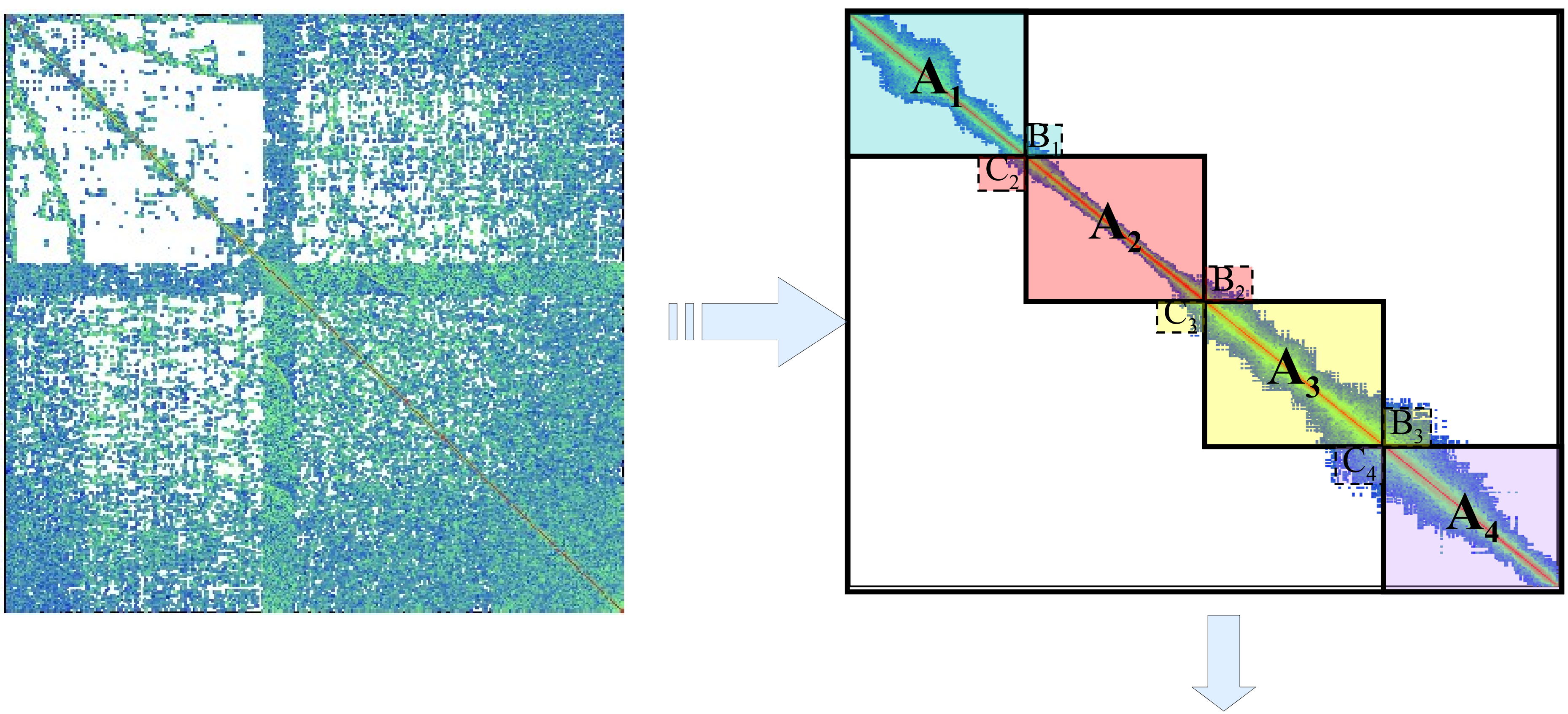
- *n = 8192*
- *nnz = 41,746*
- *κ = O(107)*

Spectrum of $M^{-1}A$

MC64 + ILUT Preconditioner: P

- *20% fill-in per row*
- *rel. drop tol = 10-1*

reordering with weights and extracting the preconditioner large bandwidth



PSPIKE

M. Manguoglu, A. Sameh, O. Schenk, A parallel hybrid sparse linear system solver, LNCS - Proceedings of EURO-PAR09 5704 (2009) 797–808.

2^{nd} approach for solving $Ax=f$

Step 1: *extract a "sparse" preconditioner, M, and solve the system:*

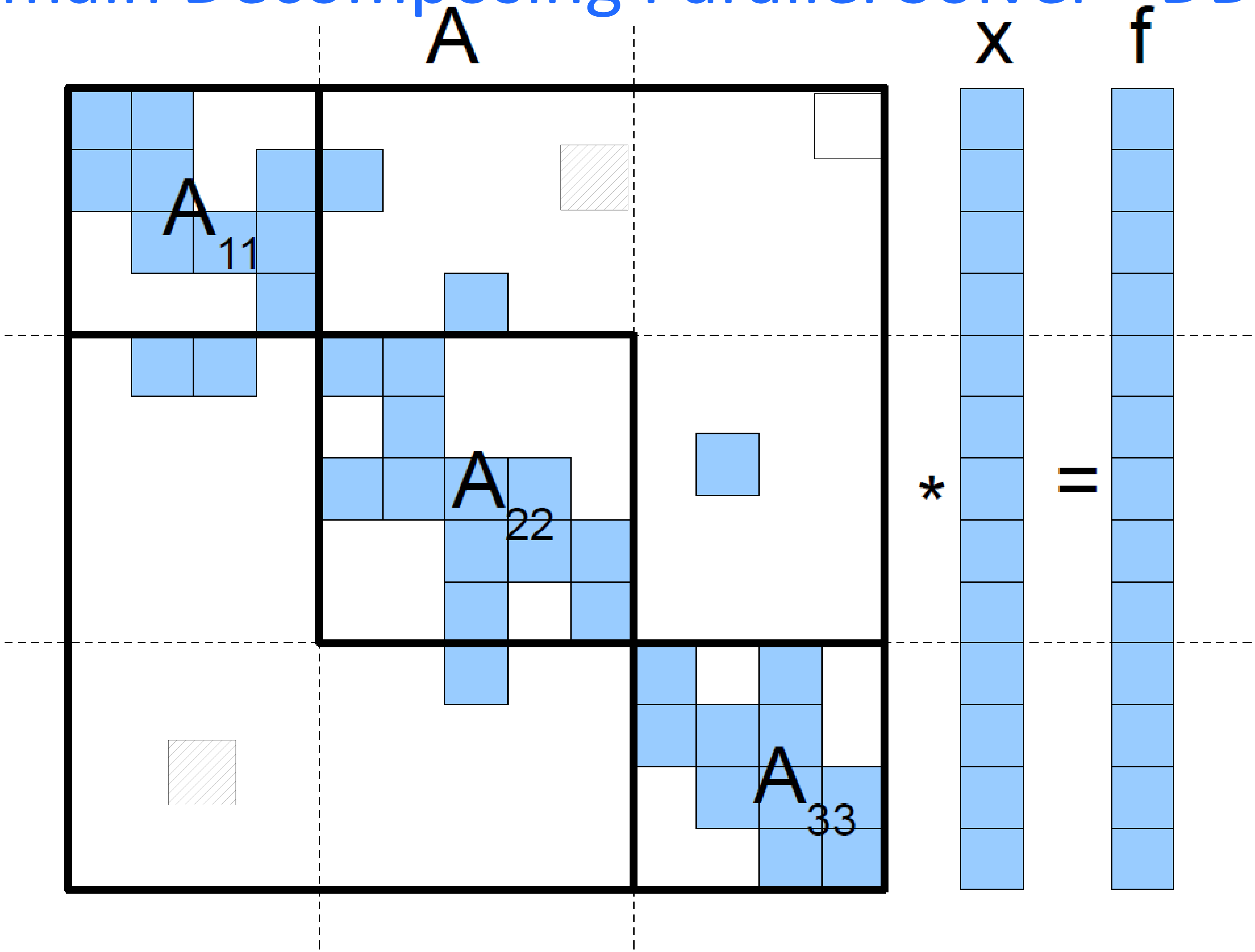
BiCGstab

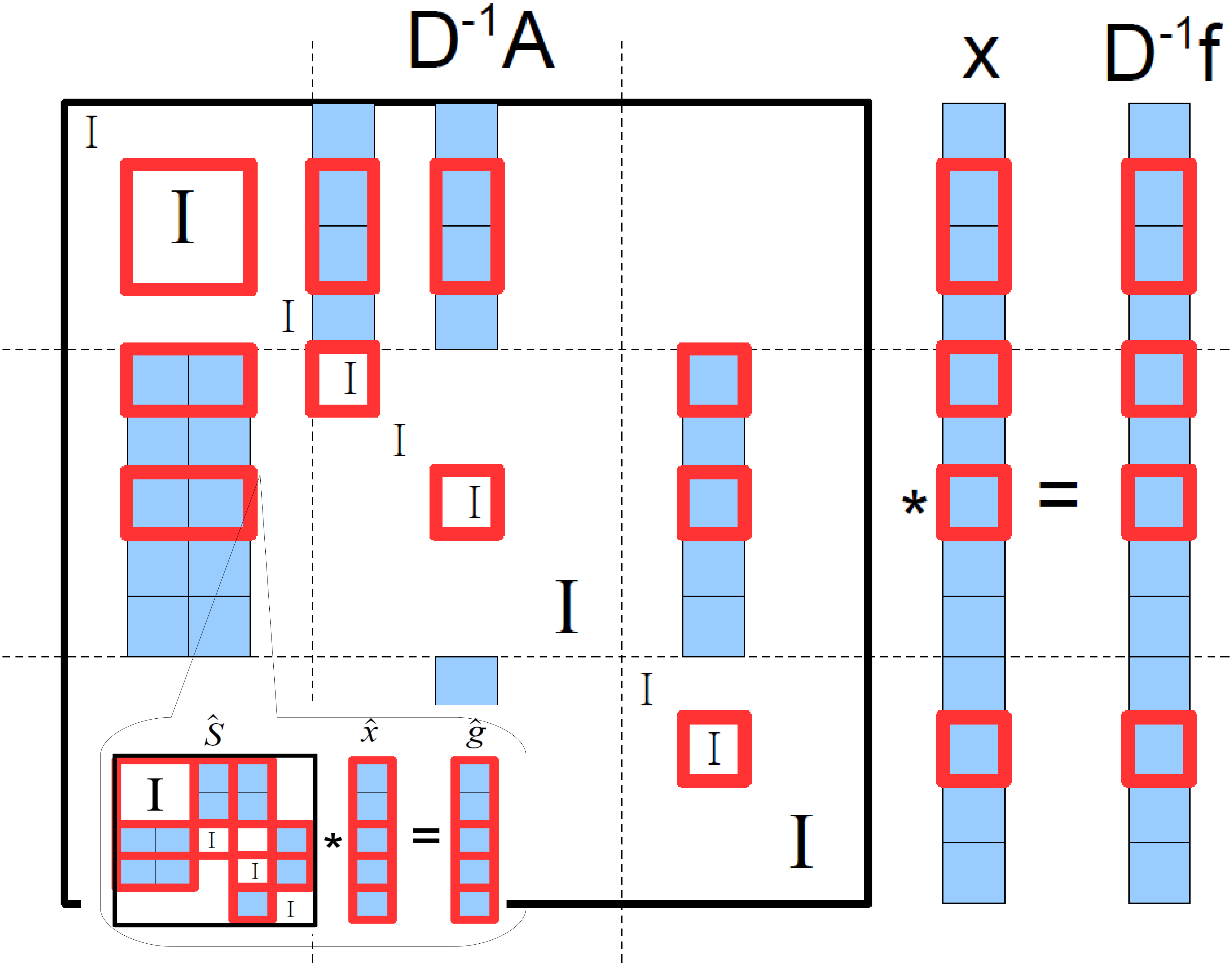
(or any other Krylov subspace method)

Solve $Mz = r$ using DS factorization

Manguoglu M, Journal of Computational and Applied Mathematics, doi:10.1016/j.cam.2011.07.017

Parallel solution of sparse linear systems (Domain Decomposing Parallel Solver - DDPS)





D-1A
x
D-1f
I
I
I
I
I
I
I
I
I
I
*
=
Ŝ
x̂
ĝ
I
I
I
I
*
=

DDPS variations

DDPS-D : use the direct solver Pardiso for $D^{-1}A$

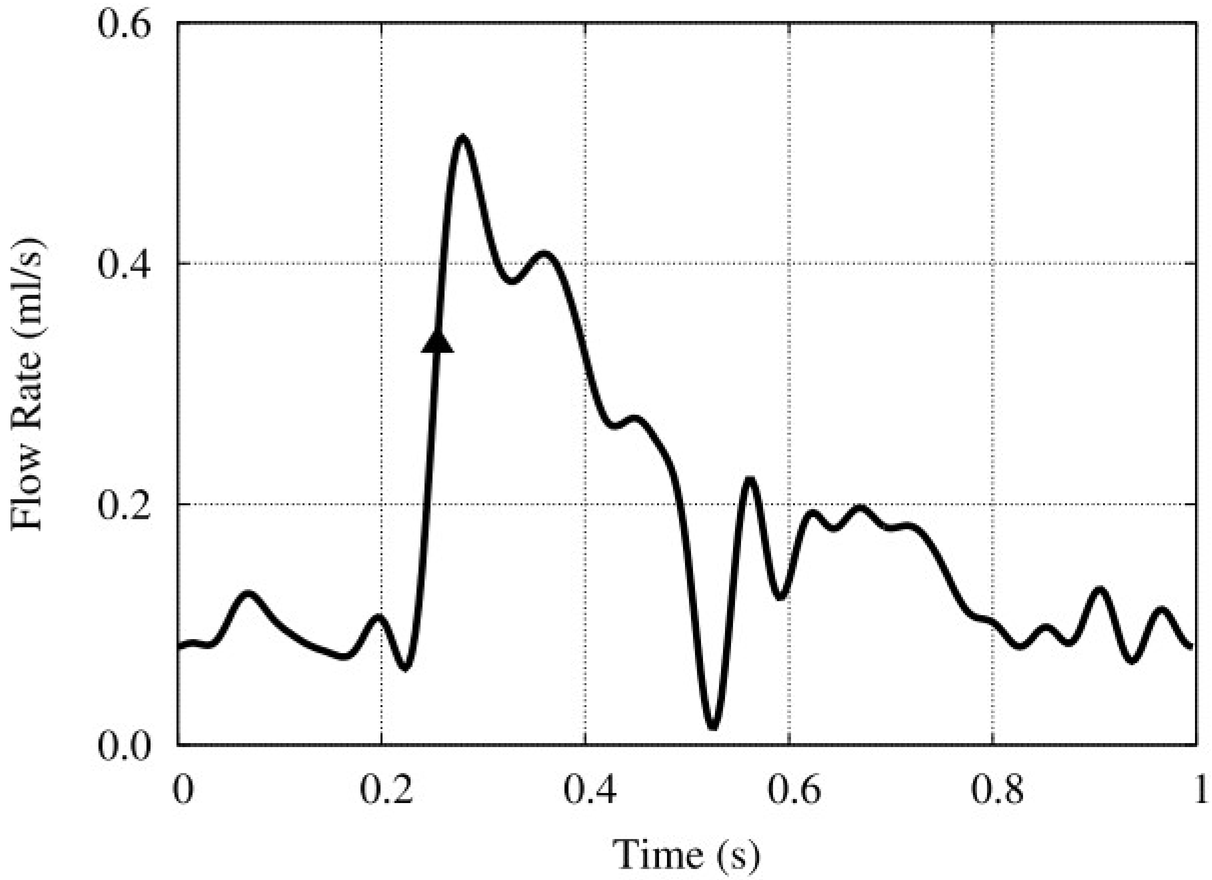
DDPS-I1: use ILUT(10^{-1}, 5) for $D^{-1}A$

DDPS-I3: use ILUT(10^{-3}, 5) for $D^{-1}A$

DDPS-I4: use ILUT(10^{-5},10) for $D^{-1}A$

In all cases the outer stopping tolerance is 10^{-7} and inner stopping tolerance is 10^{-5}

Flow field at time level 78 of the FSI computation. Stream ribbons are colored by the velocity magnitude, with the colors ranging from blue (low velocity) to red (high velocity).



Volumetric flow rate, with the triangle representing the instant (time level 78+0.5) when the test data is extracted.

Linear systems to solve at each nonlinear iteration has 1,399,566 unknowns and the coefficient matrix 167,638, 284 nonzeros

Computing platform

cluster which consists of two Intel X5670 processors per node (total 12 cores per node) and 24 GB of ram.

1st nonlinear iteration solution time (in seconds)

p	DDPS-D	DDPS-I1	DDPS-I2	DDPS-I3	Diag
2	-	F	F	1001.6	F
4	-	F	F	562.6	F
8	-	F	F	344.6	F
16	397.1	F	F	227.3	F
32	164.6	F	F	106.3	F
64	80.9	F	F	61.3	F
128	40.0	F	F	37.0	F
256	33.0	F	F	F	F

3rd nonlinear iteration solution time (in seconds)

p	DDPS-D	DDPS-I1	DDPS-I2	DDPS-I3	Diag
2	-	F	F	1127.1	F
4	-	F	F	582.7	F
8	-	F	F	408.7	F
16	398.2	F	F	228.1	F
32	178.1	F	F	113.3	F
64	76.3	F	F	63.8	F
128	45.4	F	F	35.6	F
256	23.1	F	F	F	F

Robustness of DDPS-D
- systems from UF sparse matrix collection-

#	name	n	application
1	atmosmodl	1,489,752	computational fluid dynamics
2	hvdc2	189,860	power network analysis
3	language	399,130	directed weighted graph
4	ohne2	181,343	semiconductor device simulation
5	rajat31	4,690,002	circuit simulation
6	thermomech_dk	204,316	thermal simulation
7	tmt_unsym	917,825	electromagnetic simulation
8	torso3	259,156	2d/3d problem
9	xenon2	157,464	metarial science

Robustness of DDPS-D
- number of iterations -

Matrix #	DDPS-D	ILUPACK
1	21.5	26
2	260	**F**
3	6	4
4	**F**	**F**
5	106.5	**F**
6	752.5	31
7	212	**F**
8	8.5	5
9	67	**F**

Conclusions

- Solution of large sparse linear systems required for arterial FSI modelling at every nonlinear iteration

- DDPS-D is scalable and robust involving inner-outer iterations and hence flexible

Thank you!