中国科学院大学
夏季强化课程
2022

Fast Solvers for Large Algebraic Systems

Lecture 8. Robustness and adaptivity

Chensong Zhang, AMSS

http://lsec.cc.ac.cn/~zhangcs

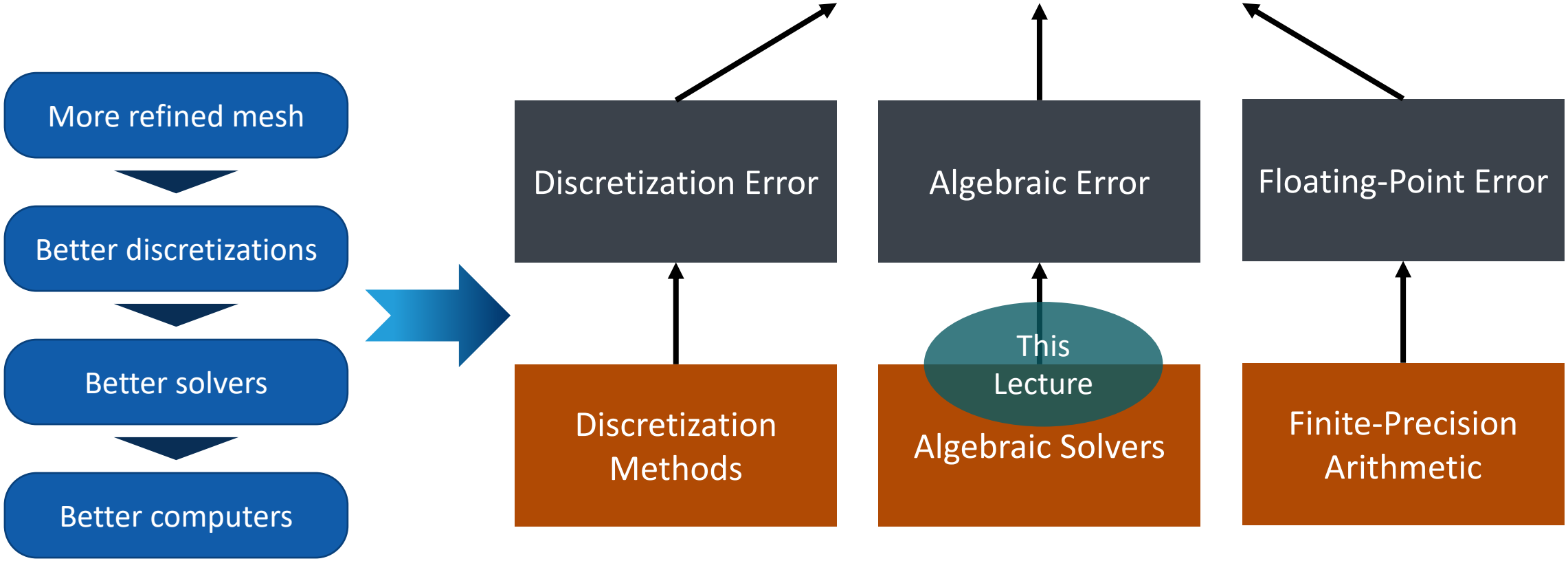
8

稳健性与自适应

Table of Contents

Sources of Error in Simulation

Approximation: $u(x) = U_h(x) + \mathcal{E}_{\text{dis}} + \mathcal{E}_{\text{alg}} + \mathcal{E}_{\text{fp}}$



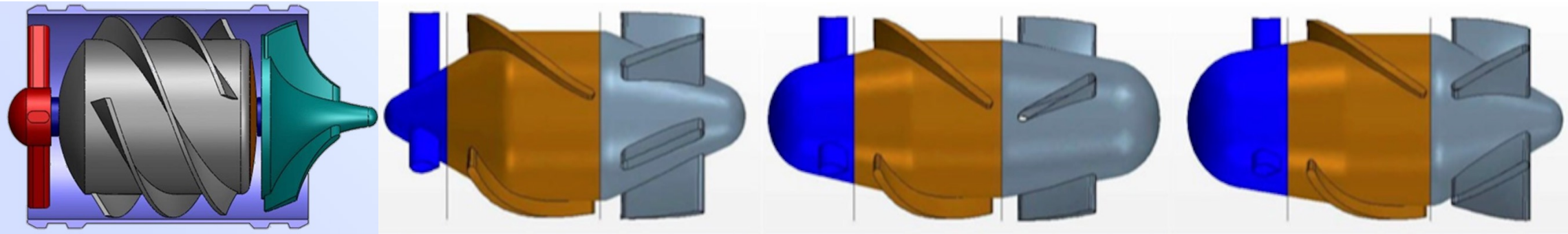
Introduction

Robustness is the Achilles’ heel of iterative solvers

Robustness Is Crucial

- In some (many) real applications, we need to solve thousands or more of linear systems from a PDE with a set of physical/discretization parameters
- Physical parameters could be nonlinear, heterogenous, anisotropic, degenerating
- Classical iterative methods (Weighted Jacobi, SOR, ...) introduce parameters

Shape Optimization for Artificial Heart Pumps



Source: W. Leng, C.-S. Zhang, P. Sun, et al., “Numerical simulation of an immersed rotating structure in fluid for hemodynamic applications”, Journal of Computational Science, 30, 79–89 (01/2019)

Robustness of Solvers

Parameters can affect convergence performance in a large extent ➔ Iterative methods are not robust

Q: What does "robustness" mean?

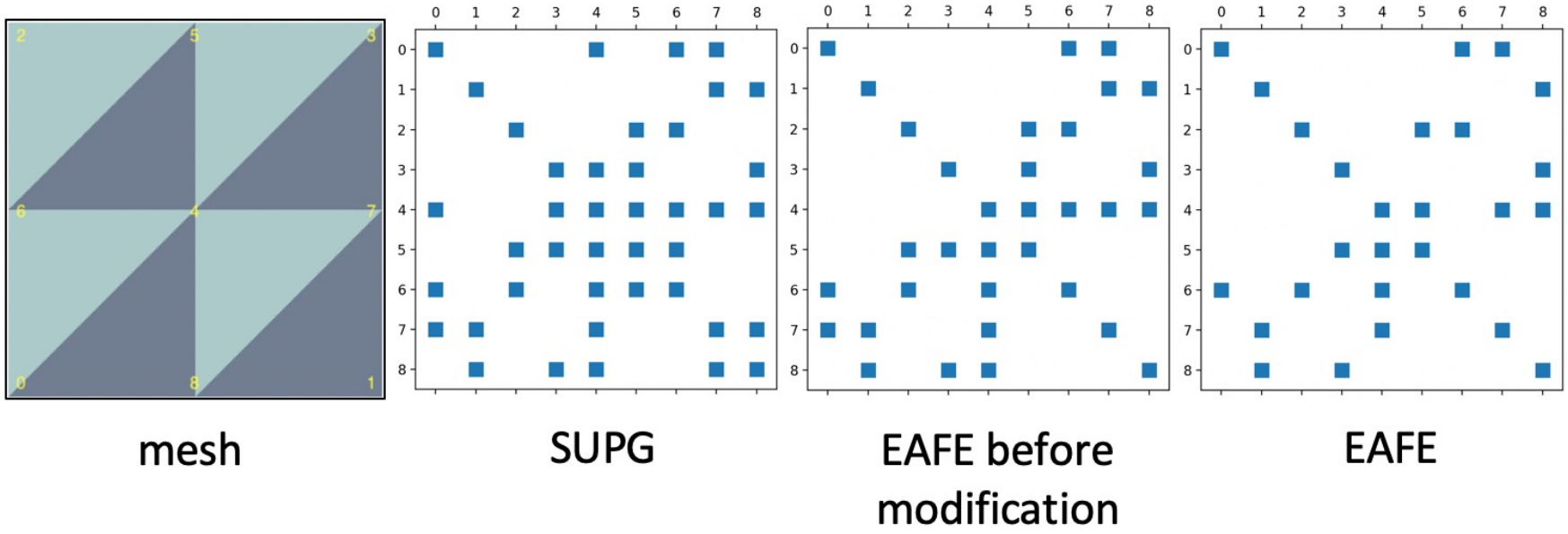
- Robustness of a system can be viewed as the property of being strong and healthy in constitution
- For solvers of linear algebraic systems, robustness has two meanings:
  - Breakdown-free and reliable for providing a solution; e.g.: GMRES, Robust ILU
  - Performance is resistant to perturbations of parameters

$$\max_{P_\alpha \in \mathcal{P}} \left\| \mathrm{Solve}(P_\alpha) \right\| \lesssim \varepsilon$$

- Solve is a solution algorithm (or a set of solution algorithms)
- $\mathcal{P}$ is a given set of problems (preferably parametrized)
- $\|\cdot\|$ is a reasonable performance measurement (0: best, ∞: fail)
- ε is a tolerance for worst performance compared with baseline
- $\lesssim$ refers to there might be a constant independent of parameter α

Algebraic Systems of Discretizations

- Different discretization methods lead to systems with different properties
  - Preserve positive-definiteness and symmetry at discrete level
  - Preserve maximum principle at discrete level
  - Sparsity pattern



SUPG and EAFE Methods

Solve the advection-diffusion problem using finite element:

$$u_t + b \cdot \nabla u - \mu \Delta u = 0$$

DoF	SUPG		EAFE		Sparsity Gain
	NNZ	Average NNZ	NNZ	Average NNZ	
66,049	454,161	6.88	329,217	4.98	28%
263,169	1,825,809	6.94	1,313,793	4.99	28%
1,050,625	7,321,617	6.97	5,249,025	5.00	28%
4,198,401	29,323,281	6.98	20,983,809	5.00	28%
4,913	46,941	9.55	32,657	6.65	30%
35,937	430,317	11.97	245,025	6.82	43%
274,625	3,680,781	13.40	1,897,025	6.91	48%
2,146,689	30,438,477	14.18	14,926,977	6.95	51%

AMG Performance, mu=1

$\mu = 10^0$	Mesh	SUPG					EAFE				
		Iter	Setup	Solve	Total	Comp	Iter	Setup	Solve	Total	Comp
Classic AMG	1	5	0.10	0.05	0.15	2.42	5	0.08	0.04	0.13	2.19
	2	5	0.48	0.25	0.74	2.43	5	0.39	0.20	0.59	2.19
	3	5	2.48	1.11	3.59	2.43	5	2.04	0.88	2.91	2.20
	4	5	11.86	5.14	17.01	2.43	5	9.79	3.92	13.71	2.20
Best AMG	1	11	0.07	0.07	0.15	1.43	11	0.07	0.07	0.13	1.59
	2	10	0.35	0.31	0.66	1.45	10	0.33	0.28	0.61	1.62
	3	10	1.78	1.33	3.11	1.46	10	1.65	1.19	2.85	1.64
	4	10	8.40	5.70	14.11	1.46	10	8.25	5.89	14.15	1.64
CAir2 AMG	1	8	0.19	0.09	0.28	2.79	8	0.16	0.07	0.24	2.74
	2	8	0.88	0.41	1.30	2.81	8	0.76	0.33	1.09	2.76
	3	8	4.21	1.92	6.14	2.82	8	3.64	1.59	5.23	2.78
	4	8	19.06	8.77	27.83	2.82	8	16.54	7.05	23.6	2.78

AMG Performance, mu=1e-2

$\mu = 10^{-2}$	Mesh	SUPG					EAFE				
		Iter	Setup	Solve	Total	Comp	Iter	Setup	Solve	Total	Comp
Classic AMG	1	4	0.12	0.05	0.17	2.43	4	0.09	0.04	0.14	2.19
	2	4	0.51	0.22	0.73	2.43	4	0.39	0.18	0.58	2.20
	3	4	2.49	0.93	3.42	2.43	4	2.07	0.72	2.80	2.20
	4	4	11.79	4.19	15.98	2.43	4	10.00	3.30	13.30	2.20
Best AMG	1	10	0.07	0.07	0.14	1.43	10	0.07	0.06	0.13	1.59
	2	10	0.35	0.31	0.66	1.45	10	0.32	0.28	0.60	1.62
	3	10	1.77	1.34	3.11	1.46	10	1.66	1.20	2.86	1.64
	4	9	9.03	5.15	14.19	1.46	9	8.05	4.69	12.75	1.64
CAir2 AMG	1	6	0.20	0.07	0.27	2.80	6	0.17	0.06	0.23	2.75
	2	6	0.89	0.31	1.20	2.81	6	0.77	0.26	1.03	2.77
	3	6	4.19	1.45	5.65	2.82	6	3.65	1.22	4.87	2.78
	4	6	19.21	6.82	26.04	2.82	6	16.67	5.52	22.20	2.78

AMG Performance , mu=1e-4

$\mu = 10^{-4}$	Mesh	SUPG					EAFE				
		Iter	Setup	Solve	Total	Comp	Iter	Setup	Solve	Total	Comp
Classic AMG	1	8	0.20	0.15	0.35	4.14	6	0.12	0.09	0.21	4.09
	2	7	0.85	0.60	1.46	4.07	6	0.70	0.43	1.14	4.50
	3	6	4.21	2.58	6.79	4.17	6	3.25	1.95	5.21	4.12
	4	5	15.73	7.40	23.14	3.36	6	12.18	6.34	18.52	3.01
Best AMG	1	14	0.07	0.07	0.14	1.43	14	0.12	0.14	0.26	2.99
	2	12	0.35	0.31	0.66	1.45	15	0.53	0.64	1.17	2.63
	3	14	1.77	1.34	3.11	1.46	13	1.70	1.79	3.52	1.69
	4	8	9.03	5.15	14.19	1.46	9	8.33	5.12	13.45	1.67
CAir2 AMG	1	5	0.71	0.17	0.88	11.03	5	0.41	0.10	0.52	8.58
	2	5	3.69	0.83	4.53	12.96	5	2.81	0.62	3.43	13.62
	3	6	15.75	4.07	19.83	11.43	6	11.87	3.00	14.88	11.79
	4	5	39.22	9.63	48.86	6.32	5	32.73	7.51	40.26	6.79

AMG Performance , mu=1e-6

$\mu = 10^{-6}$	Mesh	SUPG					EAFE				
		Iter	Setup	Solve	Total	Comp	Iter	Setup	Solve	Total	Comp
Classic AMG	1	X	0.15	3.50	3.66	3.88	7	0.10	0.10	0.21	3.76
	2	X	0.64	15.56	16.20	3.89	7	0.45	0.41	0.87	3.77
	3	X	2.95	71.35	74.31	3.89	7	2.21	1.91	4.13	3.85
	4	X	13.95	311.80	325.77	3.91	6	10.24	7.63	17.89	3.87
Best AMG	1	X	0.12	2.56	2.68	2.28	16	0.11	0.17	0.28	2.91
	2	X	0.56	12.14	12.71	2.40	16	0.51	0.76	1.27	2.94
	3	X	2.54	53.13	55.68	2.31	15	2.39	3.13	5.53	2.92
	4	X	12.73	202.35	215.09	2.26	15	11.45	14.03	25.49	2.92
CAir2 AMG	1	9	0.34	0.19	0.53	6.12	5	0.15	0.06	0.22	3.41
	2	8	1.41	0.73	2.15	5.97	5	0.60	0.29	0.89	3.25
	3	6	6.13	2.52	8.66	5.82	5	2.98	1.12	4.10	3.40
	4	6	26.76	11.24	38.02	5.68	7	14.05	6.90	20.96	3.35

Solver Performance for SUPG

SUPG Method										
μ	N	MUMPS	CAMG		CAMG+GMRES		AIR2		AIR2+GMRES	
		Time (s)	NumIt	Time (s)	NumIt	Time (s)	NumIt	Time (s)	NumIt	Time (s)
10^0	256X256	0.88	23	0.18	11	0.15	11	0.25	8	0.28
	512X512	5.02	23	0.89	10	0.66	10	1.14	8	1.31
	1024X1024	29.56	23	4.34	10	3.11	11	5.67	8	6.14
	2048X2048	203.17	22	17.63	10	14.11	10	25.22	8	27.83
10^{-2}	256X256	0.89	16	0.15	10	0.14	8	0.23	6	0.27
	512X512	4.85	16	0.73	10	0.66	8	1.06	6	1.21
	1024X1024	29.17	16	3.65	10	3.11	8	5.05	6	5.65
	2048X2048	203.34	15	16.48	9	14.19	7	22.57	6	26.04
10^{-4}	256X256	0.89	14	0.21	14	0.14	9	0.71	5	0.88
	512X512	5.47	13	0.85	12	0.66	48	7.46	5	4.53
	1024X1024	33.52	29	5.43	14	3.11	58	37.18	6	19.83
	2048X2048	265.11	9	12.63	8	14.19	19	54.71	5	48.86
10^{-6}	256X256	0.89	x	x	x	x	21	0.57	9	0.53
	512X512	5.18	x	x	x	x	13	1.94	8	2.15
	1024X1024	43.08	x	x	x	x	9	7.48	6	8.66
	2048X2048	371.99	x	x	x	x	8	32.55	6	38.02

Direct Solver

Preconditioning

Solvers Performance for EAFE

EAFE Method										
μ	N	MUMPS	CAMG		CAMG+GMRES		AIR2		AIR2+GMRES	
		Time (s)	NumIt	Time (s)	NumIt	Time (s)	NumIt	Time (s)	NumIt	Time (s)
10^0	256X256	0.93	23	0.18	11	0.13	11	0.21	8	0.24
	512X512	4.33	23	0.79	10	0.61	10	0.94	8	1.09
	1024X1024	25.28	23	3.68	10	2.85	11	4.79	8	5.23
	2048X2048	157.77	22	16.23	10	14.15	10	21.52	8	23.61
10^{-2}	256X256	0.93	16	0.13	10	0.13	8	0.19	6	0.23
	512X512	4.41	16	0.64	10	0.6	8	0.88	6	1.03
	1024X1024	26.08	16	3.03	10	2.86	8	4.28	6	4.87
	2048X2048	158.03	16	15.02	9	12.57	7	19.38	6	22.2
10^{-4}	256X256	0.98	x	x	14	0.26	7	0.39	5	0.52
	512X512	5.02	65	2.69	15	1.17	9	2.55	5	3.43
	1024X1024	29.66	13	3.06	13	3.52	11	11.91	6	14.88
	2048X2048	205.16	8	11.74	9	13.45	23	48.48	5	40.26
10^{-6}	256X256	0.91	x	x	16	0.28	12	0.24	5	0.22
	512X512	4.32	x	x	16	1.27	8	0.83	5	0.89
	1024X1024	23.27	x	x	15	5.53	9	3.97	5	4.11
	2048X2048	143.19	x	x	15	25.49	13	21.52	7	20.96

Direct Solver

Preconditioning

Robustness of Iterative Methods

Standard techniques for improving robustness of iterative methods

/02

Practical Request on Robustness

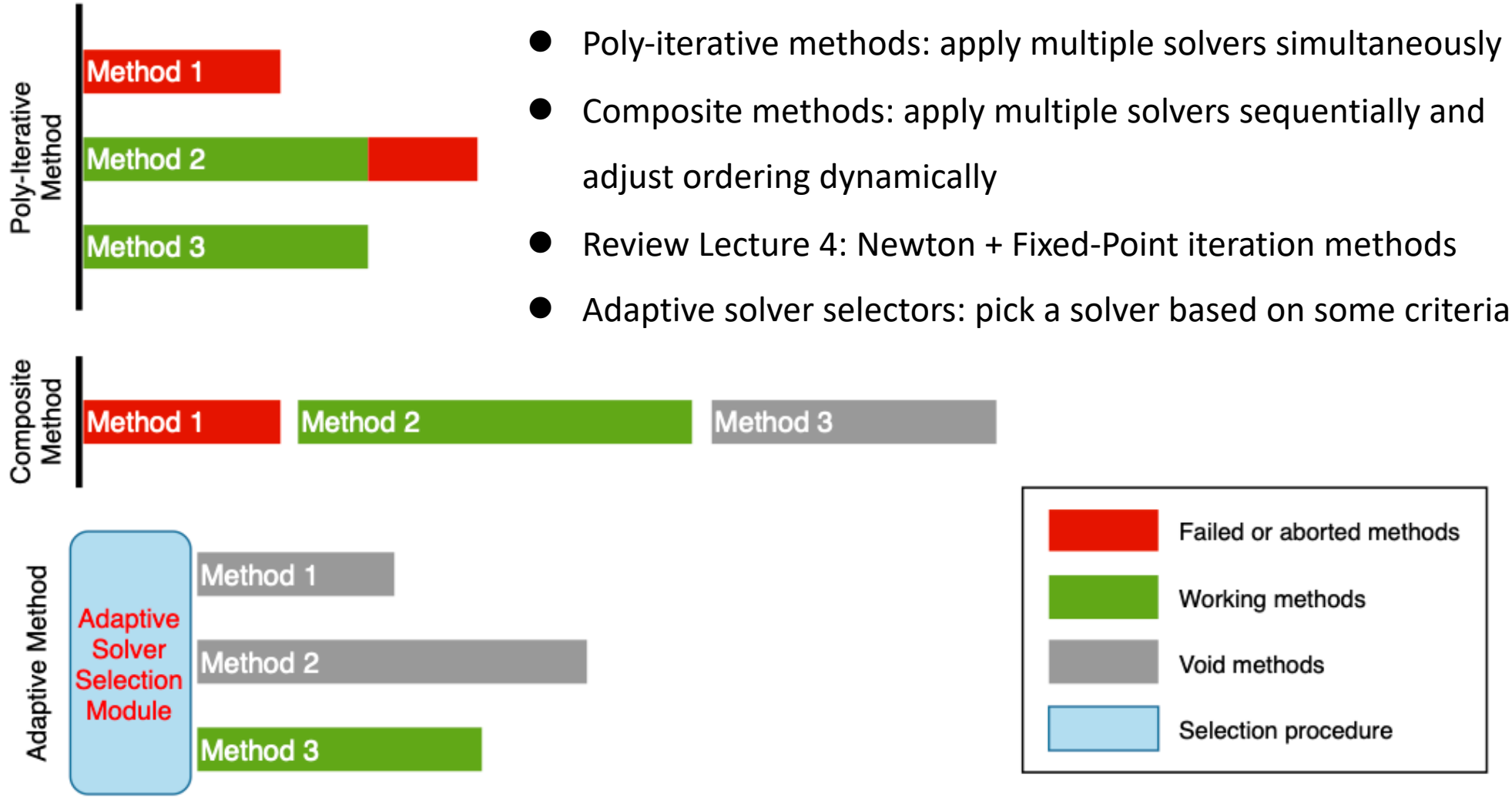
Find optimal available solver: $\operatorname{argmin}_{S_\beta \in \mathcal{S}} \max_{P_\alpha \in \mathcal{P}} \|S_\beta(P_\alpha)\| \lesssim \varepsilon$

- The easy way is, of course, to use solvers based on the direct methods; see Lectures 2 and 5
- We mainly discuss how to improve iterative methods:
  1. Improve theoretical understanding for simple model problems and construct preconditioners that are not sensitive to given parameters
  2. Combine appropriate iteration, precondition, and decoupling
  3. Provide an automatic / adaptive procedure to select solver or its parameters to assist simulation software

The unfortunate fact is: There is no well established theory, yet!

Combination of Iterative Methods

If one method does not work, try another one



- Poly-iterative methods: apply multiple solvers simultaneously
- Composite methods: apply multiple solvers sequentially and adjust ordering dynamically
- Review Lecture 4: Newton + Fixed-Point iteration methods
- Adaptive solver selectors: pick a solver based on some criteria

Extended Combined Preconditioners

- Extended the combined preconditioners to nonsymmetric problems
- Combine drop tolerance technique with ILU(k)
- Applied to multi-group radiation diffusion problems

Table 1: Number of iterations and wall time of ILU(0) and AMG.

	8000×6				16000×12				32000×24			
	Euclid		AMG		Euclid		AMG		Euclid		AMG	
	It	T_c	It	T_c	It	T_c	It	T_c	It	T_c	It	T_c
S_1	3	0.4	11	0.5	3	1.6	11	2.0	4	6.5	11	8.9
S_2	15	0.5	33	1.3	26	2.7	46	8.4	65	18.7	56	42.6
S_3	-	-	14	0.6	-	-	16	3.5	-	-	13	11.0
	4000×12				8000×24				16000×48			
	Euclid		AMG		Euclid		AMG		Euclid		AMG	
	It	T_c	It	T_c	It	T_c	It	T_c	It	T_c	It	T_c
M_1	2	3.3	14	14.2	2	13.8	35	179.6	3	60.6	48	1002.3
M_2	2	3.3	14	16.1	2	14.0	39	212.3	3	61.6	42	972.6
M_3	2	3.3	22	27.8	2	14.1	37	201.4	3	60.1	41	932.1
M_4	2	3.1	48	43.6	2	12.8	51	206.9	3	56.4	52	832.8
M_5	6	3.8	26	22.7	9	19.1	24	97.6	19	138.1	29	482.4
M_6	-	-	27	25.6	-	-	28	120.2	-	-	34	602.9

Table 2: Number of iterations and wall time of B_{co} (left) and $\tilde{B}_{co}$ (right).

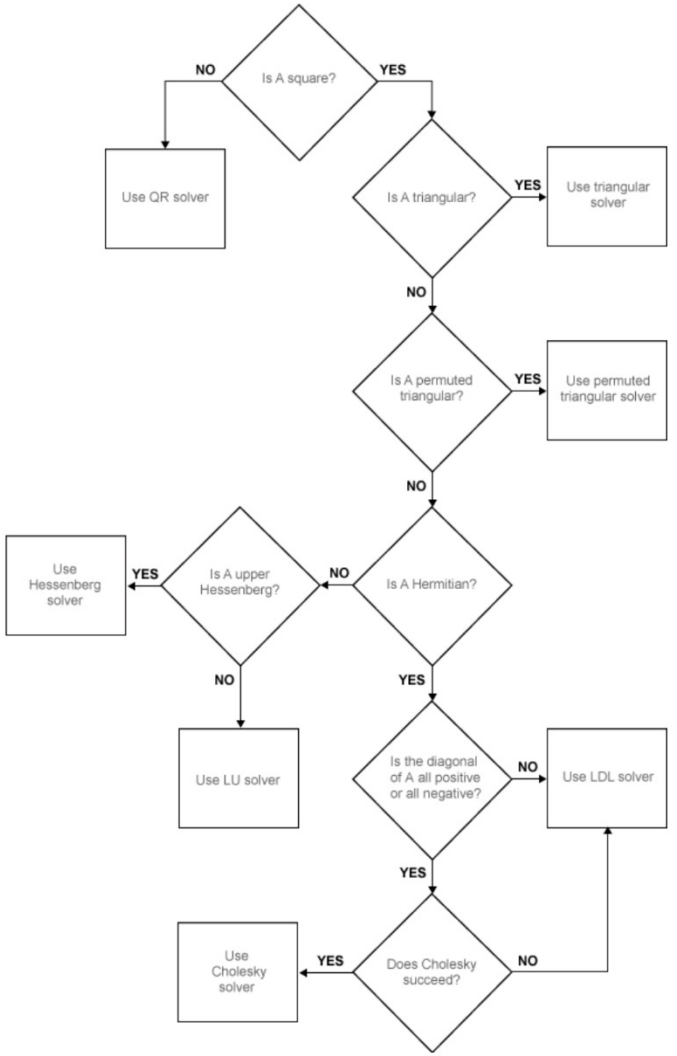
	8000×6				16000×12				32000×24			
	B_{co}		$\tilde{B}_{co}$		B_{co}		$\tilde{B}_{co}$		B_{co}		$\tilde{B}_{co}$	
	It	T_c	It	T_c	It	T_c	It	T_c	It	T_c	It	T_c
S_1	1	0.5	2	0.5	2	2.5	2	2.3	2	10.0	2	9.2
S_2	5	0.7	6	0.6	5	3.1	6	2.7	5	12.7	6	11.2
S_3	7	0.8	10	0.7	6	3.3	10	3.2	8	15.4	11	13.8
	4000×12				8000×24				16000×48			
	B_{co}		$\tilde{B}_{co}$		B_{co}		$\tilde{B}_{co}$		B_{co}		$\tilde{B}_{co}$	
	It	T_c	It	T_c	It	T_c	It	T_c	It	T_c	It	T_c
M_1	1	5.1	1	4.7	1	20.9	1	19.1	3	119.1	3	98.2
M_2	1	5.3	1	4.8	1	21.5	1	19.5	2	110.6	2	93.9
M_3	1	5.2	1	4.7	1	21.3	1	19.4	2	109.0	2	94.1
M_4	1	5.4	1	4.8	1	22.1	1	19.5	2	111.6	2	95.0
M_5	6	9.4	3	5.6	6	38.2	3	23.0	3	125.7	3	102.7
M_6	4	7.9	3	5.8	3	29.2	3	23.5	7	191.1	3	103.6

Source: Yue, X., Shu, S., Xu, X., & Zhou, Z. (2015). An Adaptive Combined Preconditioner with Applications in Radiation Diffusion Equations. Communications in Computational Physics, 18(5), 1313-1335

Matlab® Solver mldivide

Ask a sequence of questions about the coefficient matrix

- Dense or Sparse
- Is it a square matrix?
- Is it a triangular matrix?
- Is it a permuted triangular matrix?
- Is it a Hermitian matrix?
- Is it an upper Hessenberg matrix?
- Is the diagonal of the matrix positive?



Intel® oneAPI MKL Advisor

Intel® oneAPI Math Kernel Library Link Line Advisor

Introduction

The Intel® oneAPI Math Kernel Library (oneMKL) is designed to run on multiple processors and operating systems. It is also compatible with several compilers and third-party libraries, and provides different interfaces to the functionality. To support these different environments, tools, and interfaces, oneMKL provides multiple libraries to choose from.

To see which libraries are recommended for a particular use case, specify the parameters in the drop-down lists below.

Intel® oneAPI Math Kernel Library (oneMKL) Link Line Advisor v6.19 — Reset

Select Intel® product:	oneMKL 2022
Select OS:	Linux*
Select programming language:	C/C++
Select compiler:	Intel® oneAPI DPC++/C++
Select architecture:	Intel® 64
Select dynamic or static linking:	Static
Select interface layer:	C API with 32-bit integer
Select threading layer:	OpenMP threading
Select OpenMP library:	Intel® (libiomp5)
Enable OpenMP offload feature to GPU:	☐
Select cluster library:	☐ Parallel Direct Sparse Solver for Clusters (BLACS required) ☐ Cluster Discrete Fast Fourier Transform (BLACS required) ☐ ScaLAPACK (BLACS required) ☐ BLACS
Select MPI library:	<Select MPI>
Select the Fortran 95 interfaces:	☐ BLAS95 ☐ LAPACK95
Link with Intel® oneMKL libraries explicitly:	☑
Link with DPC++ debug runtime compatible libraries:	☐

Use this link line:

```
-Wl,--start-group ${MKLROOT}/lib/intel64/libmkl_intel_lp64.a
${MKLROOT}/lib/intel64/libmkl_intel_thread.a ${MKLROOT}/lib/intel64/libmkl_core.a -Wl,--end-group
-liomp5 -lpthread -lm -ldl
```

LAPACK Search Engine

LAPACK

Home Help Netlib Search Sitemap

Access Individual Routines

- Driver Routines
- Computational Routines
- Utility Routines
- Testing Routines
- Timing Routines

Download Libraries and Packages

Documentation

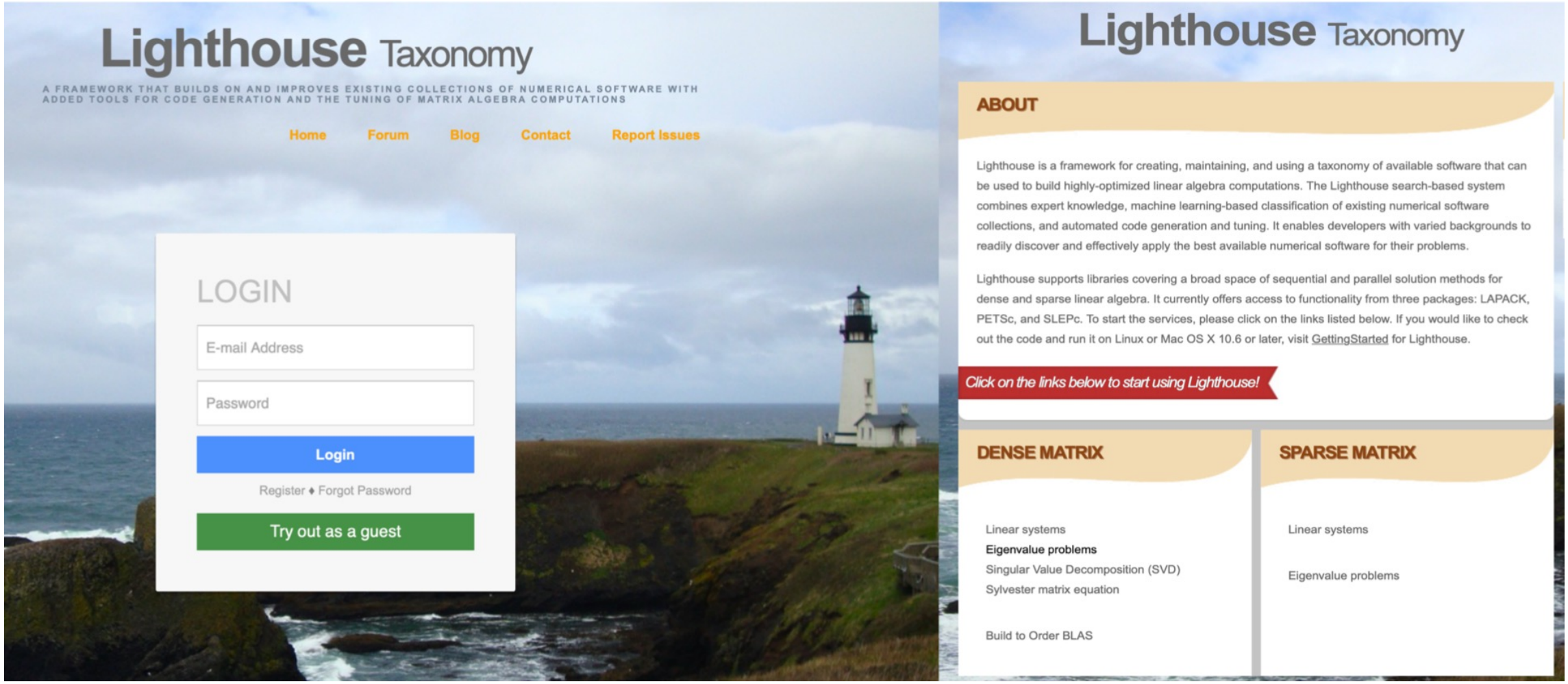
Search

Note: You can use two engines for your search:

1. Netlib's Entire Repository Search
2. Search for LAPACK Routine by Name (applet):

Out of date, for many years!

Lighthouse Project



A Weaker Request on Robustness

For any problem $P_\alpha \in \mathcal{P}$, find a solver $S_\beta \in \mathcal{S}$ such that $\|S_\beta(P_\alpha)\| \lesssim \varepsilon$

Parameters $\xrightarrow{\text{analysis}}$ Feature (fixed) $\cup$ Free (variable) $\xrightarrow{\text{tuning}}$ Performance!

Problems	Algorithms	Resources
• Type of PDE (system)	• Krylov methods	• Num of nodes
• Physical parameters	• Preconditioning	• Num of processes
• Discretization	• Decoupling	• Num of threads
•	•	• .....

Question: How to find a "good" set of problem (easy-to-compute) features?

Problem Feature Parameters

Feature names	
avgnnzprow	right-bandwidth
avgdistfromdiag	symmetry
n-dummy-rows	blocksize
max-nnzeros-per-row	diag-definite
lambda-max-by-magnitude-im	lambda-max-by-magnitude-re
ellipse-cy	nnzup
ruhe75-bound	avg-diag-dist
nnz	left-bandwidth
lambda-min-by-magnitude-im	lambda-min-by-magnitude-re
norm1	sigma-min
upband	n-struct-unsymm
colours	diagonal-average
diagonal-dominance	dummy-rows
ritz-values-r	symmetry-snorm
symmetry-fanorm	symmetry-fsnorm
lambda-max-by-real-part-im	lambda-max-by-real-part-re
lambda-max-by-im-part-re	lambda-max-by-im-part-im
col-variability	trace-abs
ritz-values-c	nnzeros
diag-zerostart	loband
positive-fraction	trace
min-nnzeros-per-row	diagonal-sign
row-variability	nrows
colour-offsets	n-colours
relsymm	diagonal-variance
departure	nnzlow
n-nonzero-diags	sigma-max
dummy-rows-kind	kappa
n-ritz-values	colour-set-sizes
sigma-diag-dist	symmetry-anorm
ellipse-ax	ellipse-ay
ellipse-cx	lee95-bound
normInf	normF
nnzdia	trace-asquared

Full feature set comprising of 68 features computed using Anamod

Feature learning (selection & extraction)

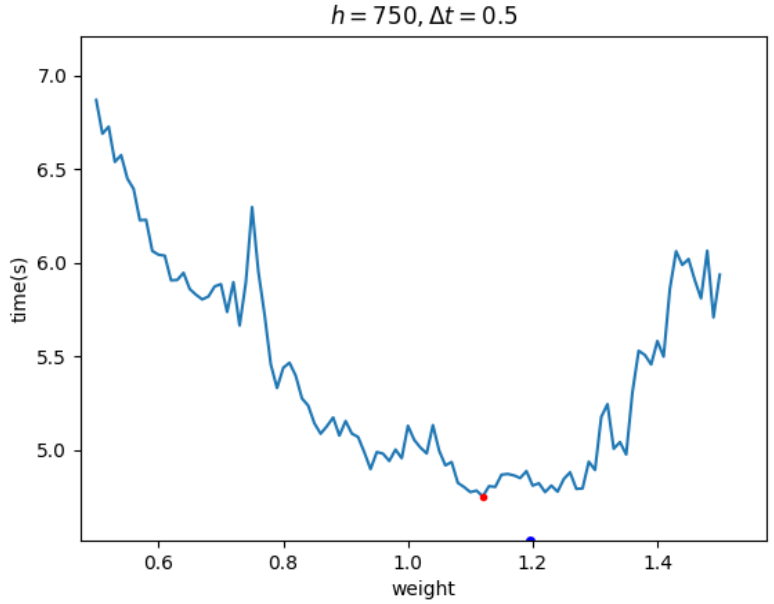
- Performance of ML models is largely affected by the choice of features
- A comprehensive feature set may include all entries of A (however, it is too complicated)
- The choice of features is usually problem-dependent
- Obtaining features is usually costly and needed in both offline and online steps
- Training takes a lot of time if too many features are selected
- Analytical and/or empirical results should be used to select features

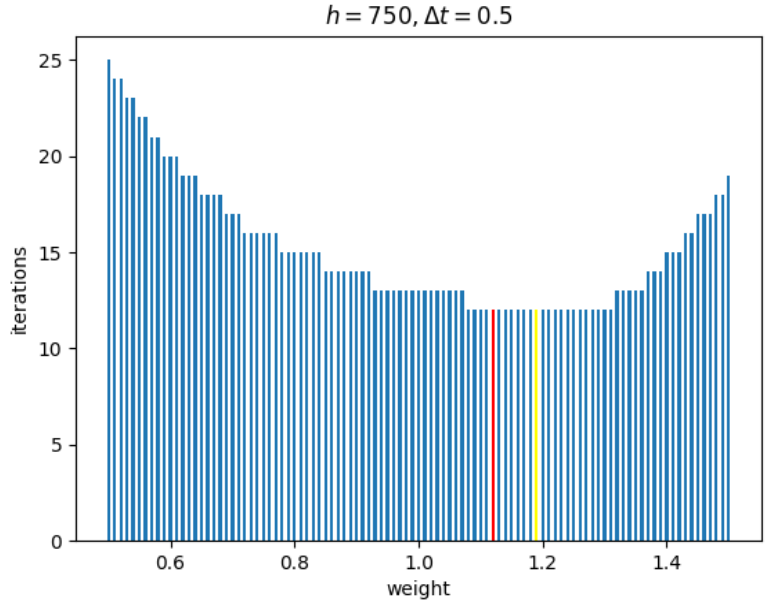
Free Parameters: Weight for Jacobi

Weighted Jacobi method $\vec{u}^{\text{new}} = \vec{u}^{\text{old}} + \omega D^{-1}(\vec{f} - A\vec{u}^{\text{old}})$

Question: How to find a good weight in practice?

- Minimize smoothing factor by LFA (local mode analysis) [Brandt 1994]
- Minimize spectral radius bound of iterative matrix [Yang 2004]
- It is complicated to determine ω for practical applications: Trial and error!





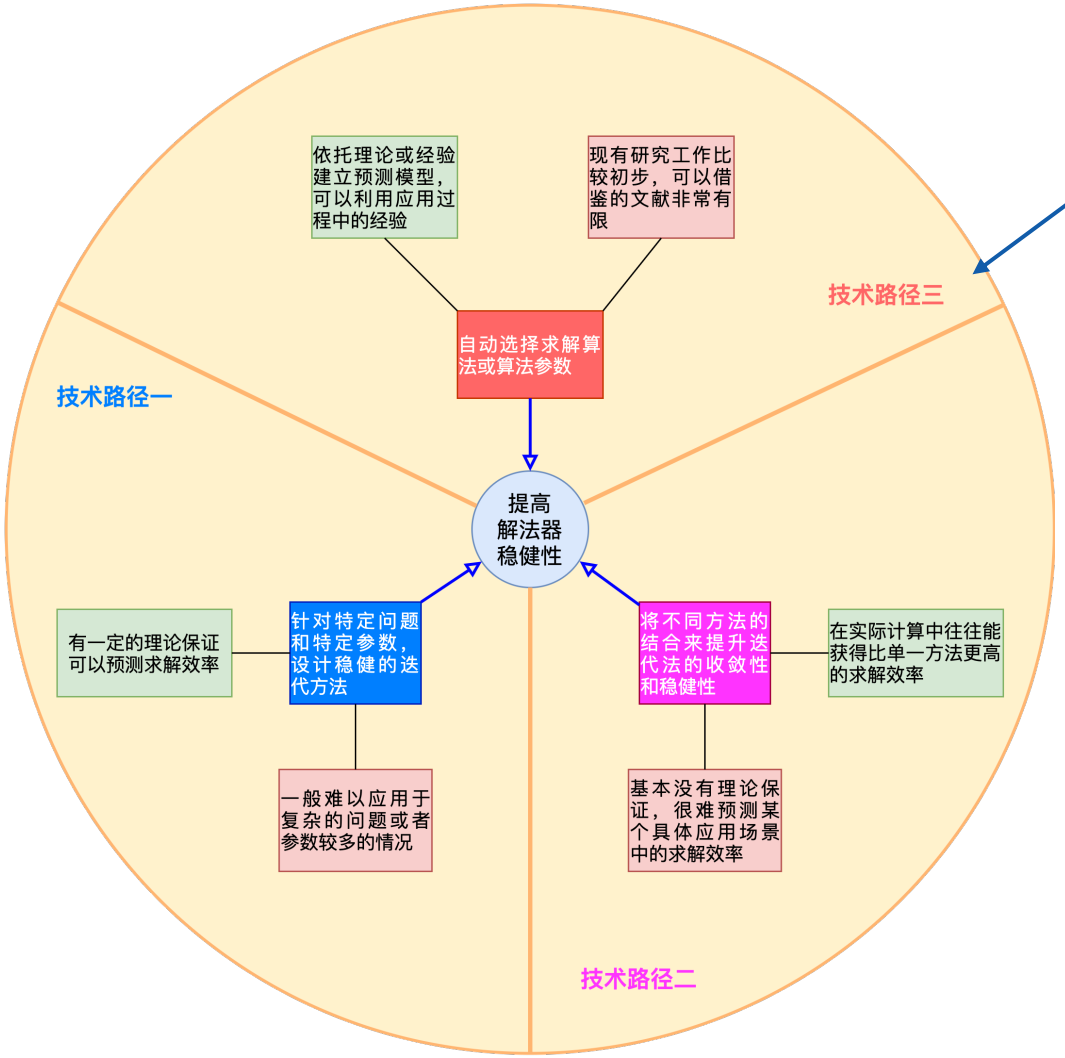
Adaptive Solvers

Adaptive iterative methods via machine learning

/03

A Short Summary on Classical Methods

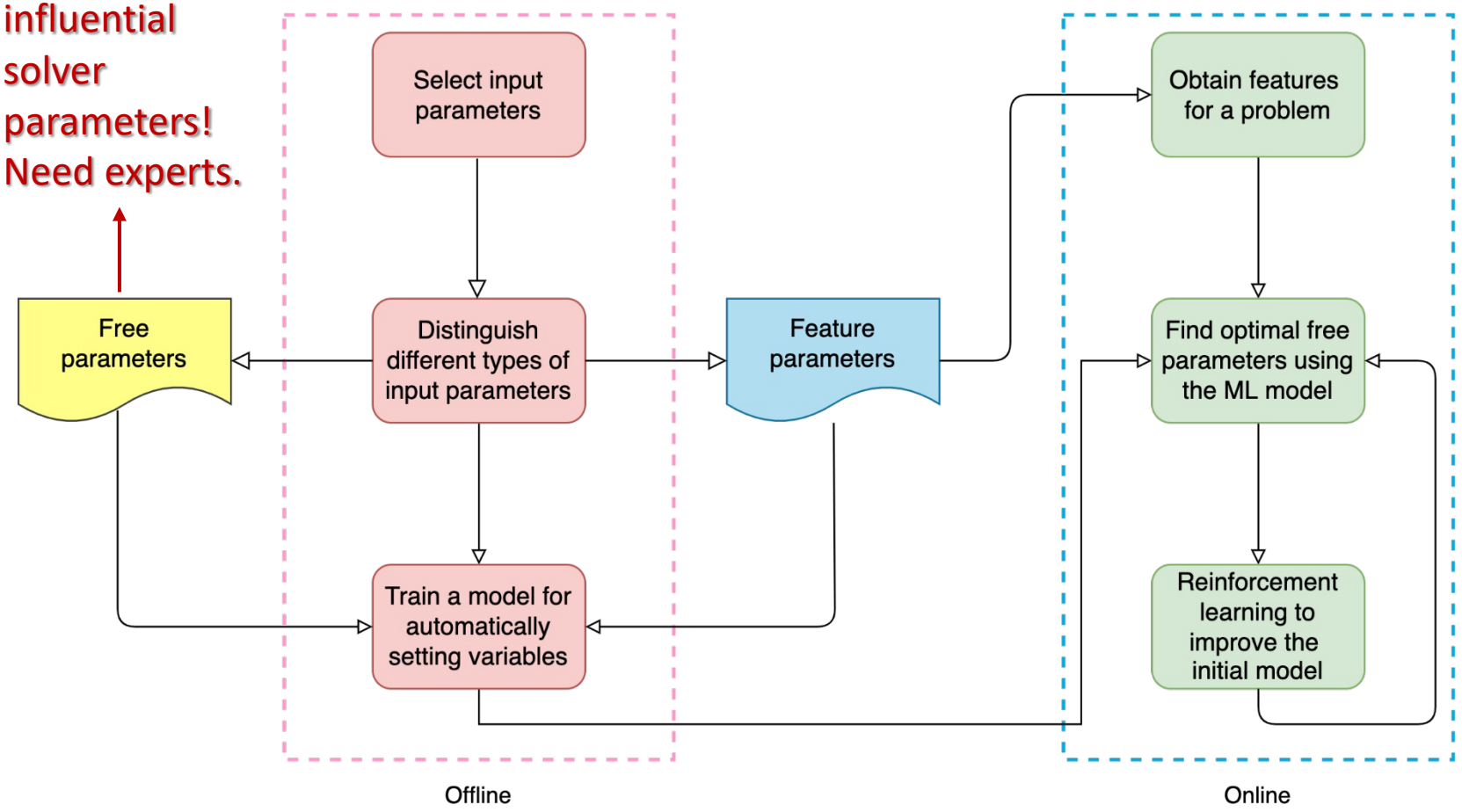
- 迭代法中有很多可调参数，这些参数对求解效率有很大影响
  - 同一个方程，选择不同的参数，会对求解效率有很大影响
  - 不同的方程，往往需要不同的参数，固定的参数无法达到最优效果
- 在实际应用场景中，如油藏数值模拟、计算流体、形状优化
  - 往往需要求解成千上万次不同的线性代数方程组
  - 这些问题有类似的性质，但不完全相同，需要使用不同的可调参数
  - 需要有一个自适应的策略，自动地选择合适的可调参数



技术有一定的通用性，利于推广应用，但非常不成熟

Constructing Adaptive Solvers

Need to find influential solver parameters! Need experts.



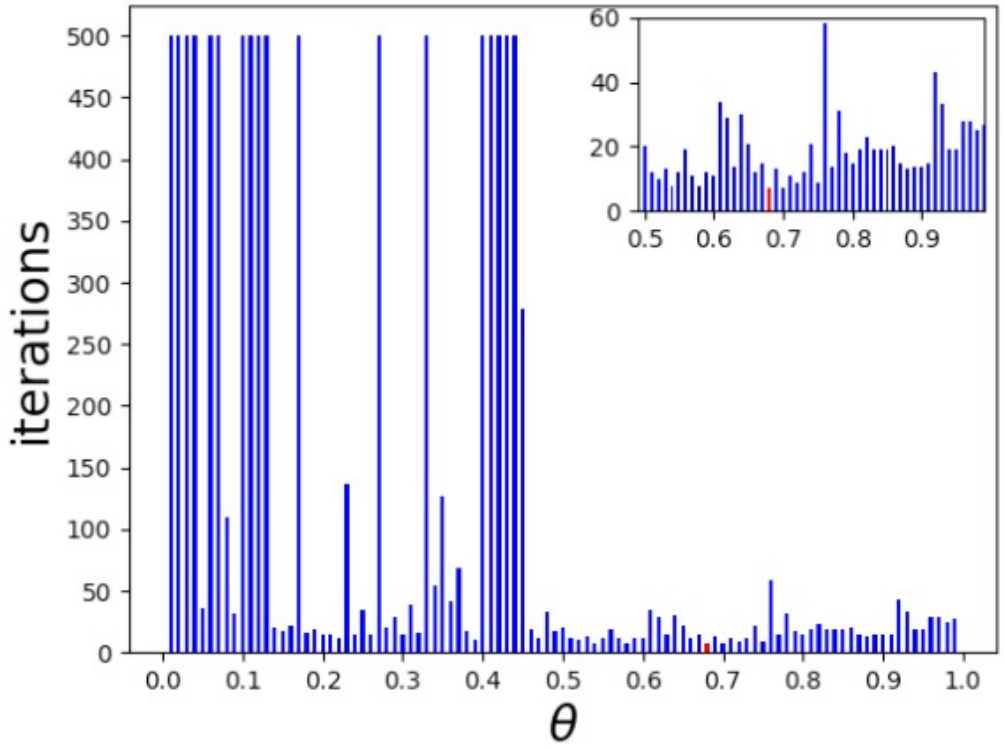
Keys to success:

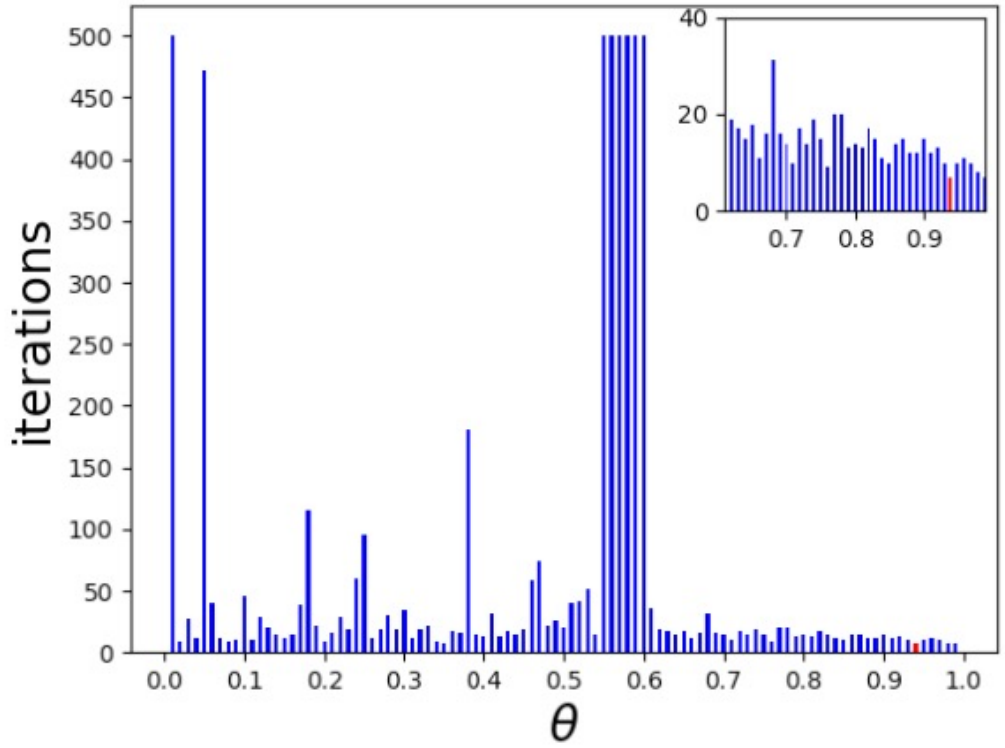
- Look at preconditioners, instead of simple Krylov subspace methods
- Look at the powerful preconditioners, like ILU, DDM, Multigrid, or combination of these methods …

Free Parameters: Coupling Threshold

Strong coupling threshold: $-a_{i,j} \geqslant \theta_{\text{str}} \left| \min_k a_{i,k} \right|$

- $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ is the corresponding graph of A
- $\mathsf{S}_i := \{ j \in \mathsf{N}_i : \ j \ \text{ strongly coupled to } \ i \}$, $\mathsf{S}_i^T := \{ j \in \mathsf{V} : \ i \in \mathsf{S}_j \}$ set affected by i





Numerical Results Matches Theory

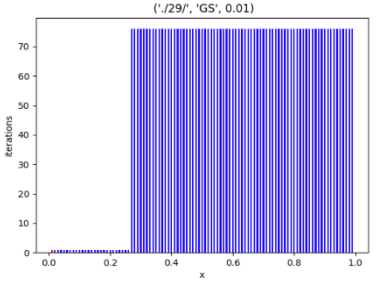
Convergence Analysis (X. Xu and C.S. Zhang, 2022)

$$1 - \frac{\Delta_2}{K_{\mathrm{TG}}} \leq \|E_{\mathrm{PTG}}\|_A \leq \max\left\{1 - \frac{\Delta_1}{K_{\mathrm{TG}}}, \Delta_2 - 1\right\}$$

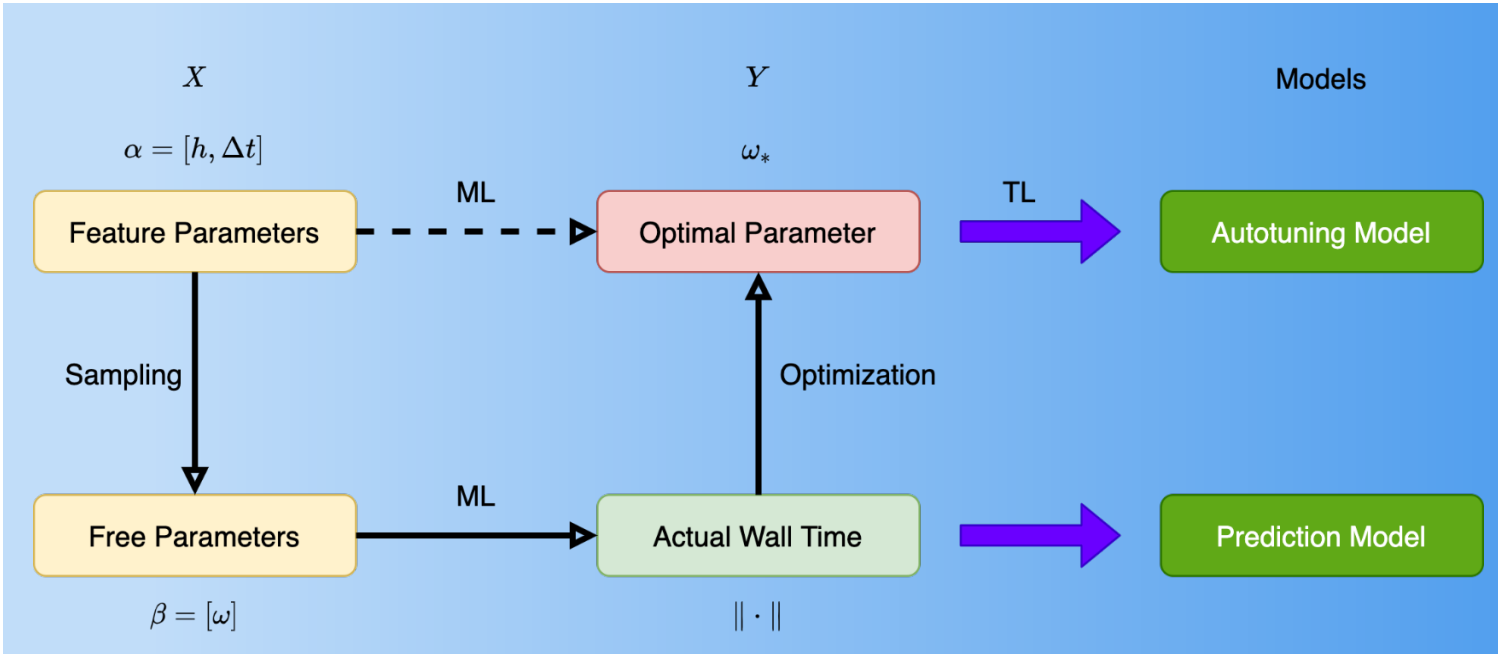
Smoother	Cycle Type	Convergence Rate	Y. Notay 2007	Xu & Zhang 2022
Gauss-Seidel	V-cycle	**0.876**	**N/A**	**0.955**
	W-cycle	**0.556**	**0.859**	**0.812**
Weighted Jacobi (0.5)	V-cycle	**0.905**	**N/A**	**0.993**
	W-cycle	**0.639**	**Fail**	**0.979**

Construct a test finite element matrix on 12×12 grid

- $\theta = 0.26$: Num Iter = 08, Actual Rate = 0.2498, Theoretical Rate = 0.2500
- $\theta = 0.27$: Num Iter = 74, Actual Rate = 0.9477, Theoretical Rate = 0.9477



Adaptive Iterative Methods

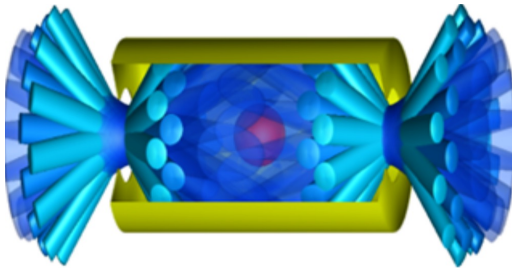


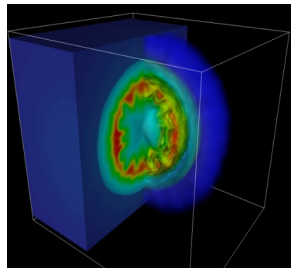
- 迭代法稳健性

$$\max_{P_\alpha \in \mathcal{P}} \left\| \mathrm{Solve}(P_\alpha) \right\| \lesssim \varepsilon$$

- 自适应迭代法
  - 分析影响稳健性的主要因素
  - 构建更准确的性能预测模型
  - 提出自适应和参数调整策略

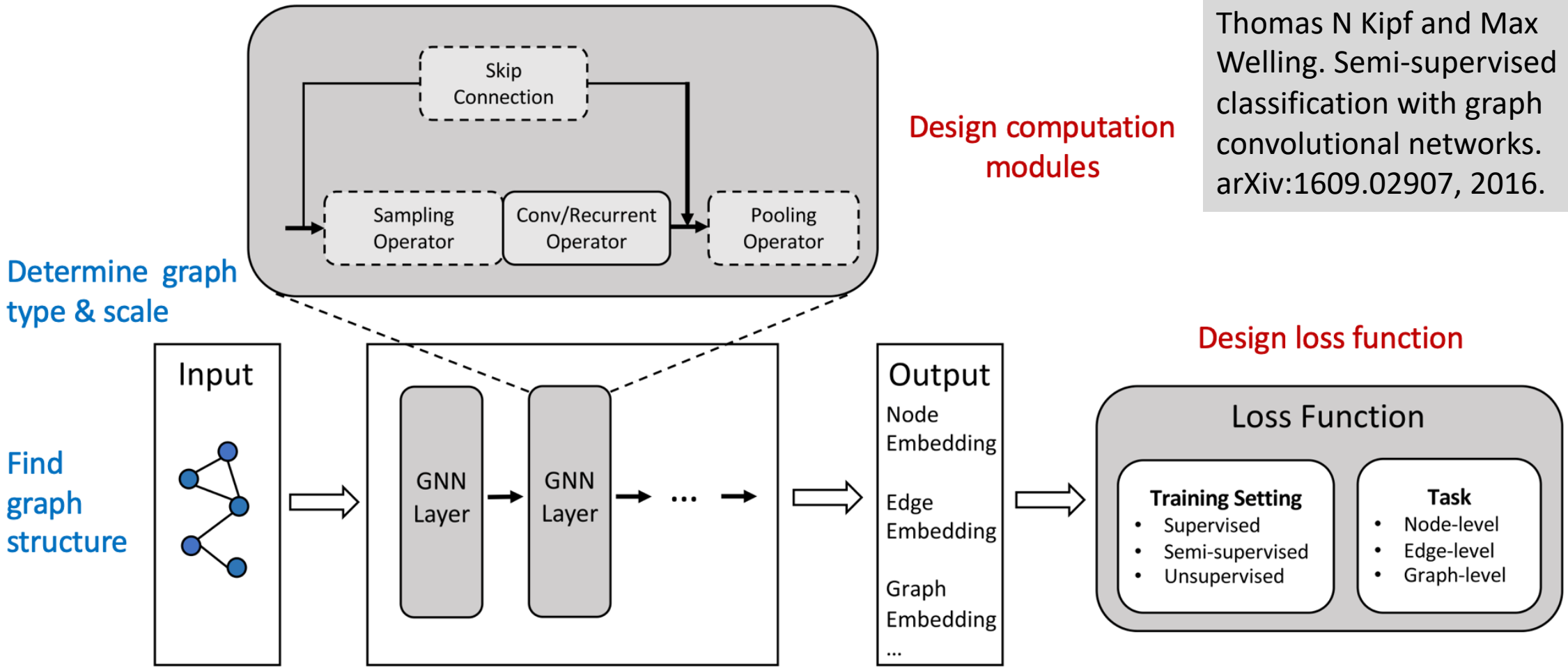
3D ICF Demo Example: 120 training problems and 10 test problems





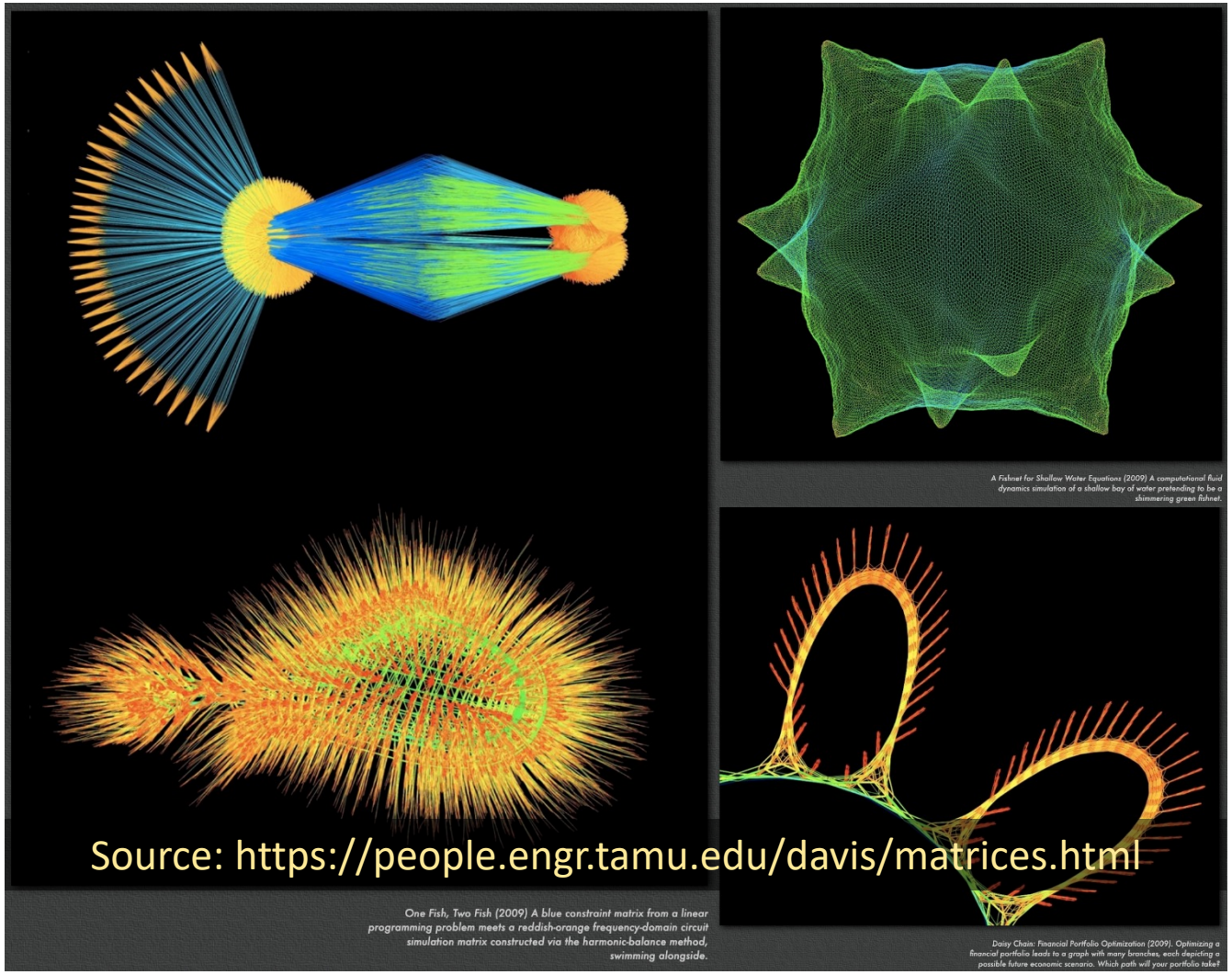
Number of Nodes		Average Number of Iterations			
Graph	Subgraph	$\theta = 0.25$	$\theta = 0.5$	Optimal	AutoAMG
6.29M	4.91M	51.3	466.6	**23.1**	**27.7**

Designing GNN Networks



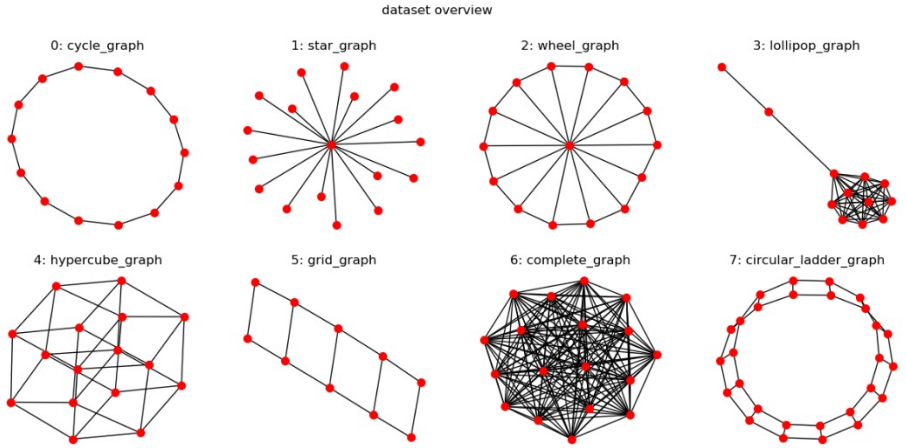
Source: Jie Zhou et al. "Graph neural networks: A review of methods and applications", AI Open, 2020, AI Open, 57-81

From Matrices to Graphs



Source: https://people.engr.tamu.edu/davis/matrices.html

Graph Structure Classification

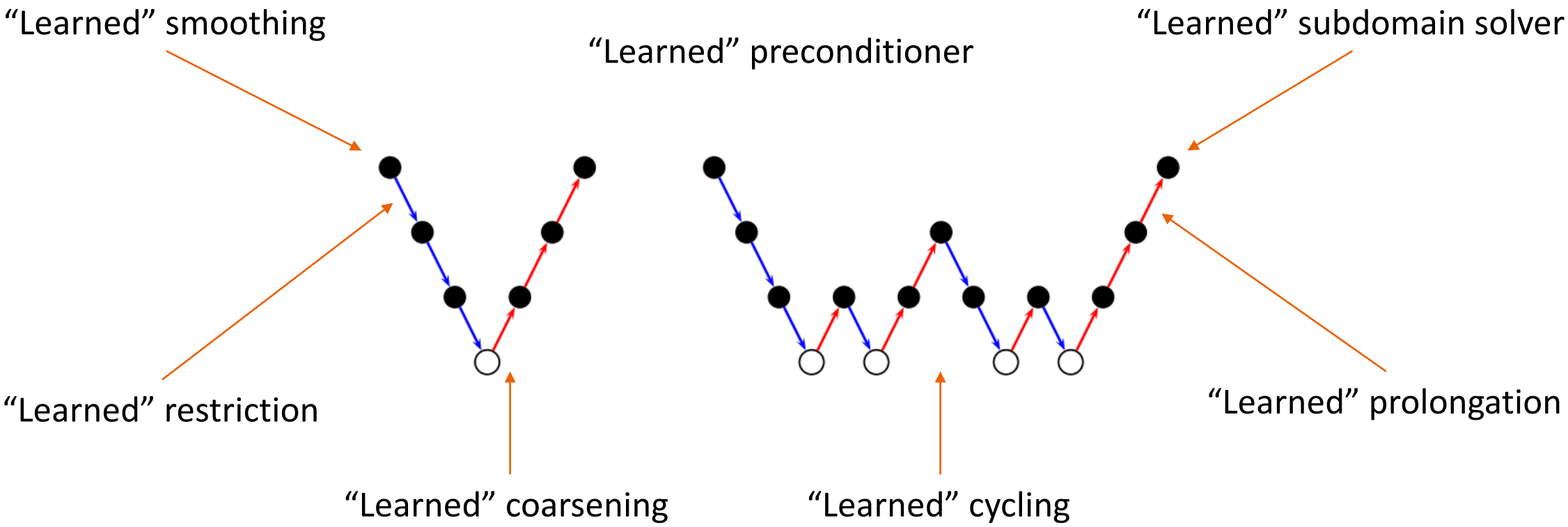


Source: https://www.dgl.ai/blog/2019/01/25/batch.html

- Using GNN to classify graph structures (learn features)
- Numerical test: 320 (8x40) training graphs, 80 (8x10) testing graphs
- Classification accuracy: about 71.25%

Learning Solver Components

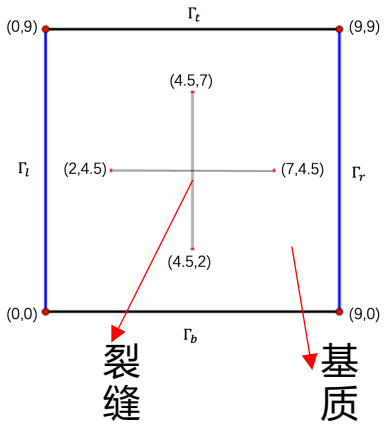
There are many examples on learning solver components by machine learning (deep learning)

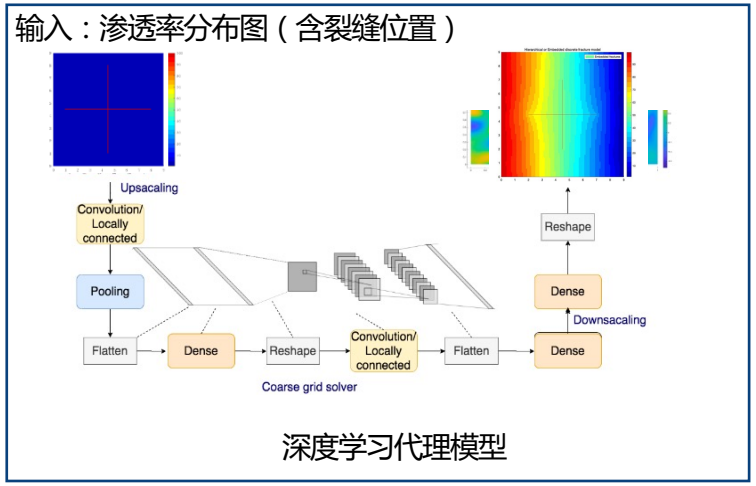


Yuyan Chen, Bin Dong, and Jinchao Xu. “Meta-MgNet: Meta multigrid networks for solving parameterized partial differential equations”. In: Journal of Computational Physics 455 (2022)

Learning Proxy Models

- **裂缝性油气藏储集层**
  - 裂缝分布随机性强，非均质性差别大
  - 孔隙率高低不一
  - 非裂缝处渗透率低，但在裂缝处的渗透率极高
- **在裂缝性油气藏模拟中，有如下难点**
  - 没有准确的裂缝模型：裂缝的渗透率、孔隙度、密度等难确定
  - 裂缝处的网格处理复杂，计算量大
- **目标：建立代理模型，对于不同裂缝的位置与渗透率，快速求解油藏压力分布**





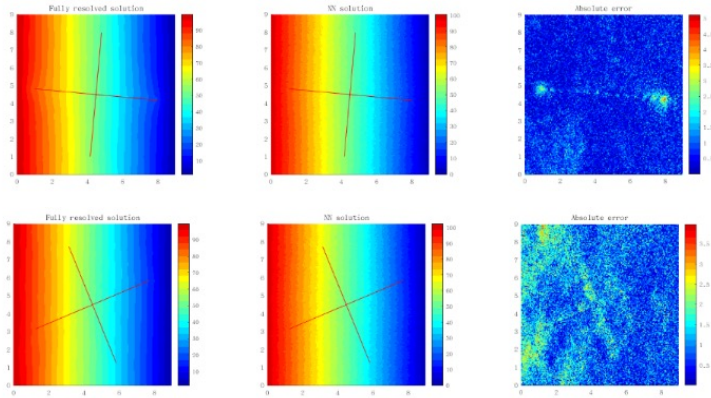
考虑如下带裂缝的多孔介质流问题

$$\begin{cases} -\nabla\cdot(\kappa\nabla p)=0, & \text{in } \Omega,\\ p=10, & \text{in } \Gamma_l,\\ p=1, & \text{in } \Gamma_r,\\ \dfrac{\partial p}{\partial n}=0, & \text{in } \Gamma_b\cup\Gamma_t. \end{cases}$$

- 裂缝位置随机分布
- 渗透率κ的取值分基质和裂缝两部分，基质相对渗透率远小于裂缝处。如：
  - 裂缝处：1000
  - 基质处：（0，10）的随机数

结果：上下两个例子裂缝位置不同、渗透率不同

有限体积数值解(左图)；深度学习代理模型解(中图)；两者误差(右图)



Reading and Thinking

MULTITASK AND TRANSFER LEARNING FOR AUTOTUNING EXASCALE APPLICATIONS

WISSAM M. SID-LAKHDAR, MOHSEN MAHMOUDI AZNAVEH, XIAOYE S. LI, AND JAMES W. DEMMEL

ABSTRACT. Multitask learning and transfer learning have proven to be useful in the field of machine learning when additional knowledge is available to help a prediction task. We aim at deriving methods following these paradigms for use in autotuning, where the goal is to find the optimal performance parameters of an application treated as a black-box function. We show comparative results with state-of-the-art autotuning techniques. For instance, we observe an average $1.5x$ improvement of the application runtime compared to the OpenTuner and HpBandSter autotuners. We explain how our approaches can be more suitable than some state-of-the-art autotuners for the tuning of any application in general and of expensive exascale applications in particular.

https://arxiv.org/pdf/1908.05792.pdf

- Are the algebraic solvers robust in your application?
- What solvers do you use? Do you think they are good enough?
- What strategies for improving robustness will fit your application the best? Why?
- How do you want to improve robustness?

Contact Me

- Office hours: Mon 14:00—15:00
- Walk-in or online with appointment
- zhangcs@lsec.cc.ac.cn
- http://lsec.cc.ac.cn/~zhangcs

My sincere gratitude to:

Xiaowen Xu, Haifeng Zou, Lian Zhang, Rui Li

中国科学院大学
夏季强化课程
2022

Fast Solvers for Large Algebraic Systems

Chensong Zhang, AMSS

http://lsec.cc.ac.cn/~zhangcs

Release version 2022.07.10