北京
应用物理与计算数学研究所
Institute of Applied Physics
and Computational Mathematics

中物院
高性能数值模拟软件中心
CAEP
Software
Center for High Performance
Numerical
Simulation

解法器暑期强化课程前沿选讲，2022.7.3

稀疏线性解法器：离实际应用有多远？

徐小文
xwxu@iapcm.ac.cn
北京应用物理与计算数学研究所
中物院高性能数值模拟软件中心

提纲

- 应用需求与算法框架
- 解法器现状：离实际应用的距离
- 算法的挑战：以AMG为例
- 总结与展望

应用需求

目标：快速求解 $Ax=b$

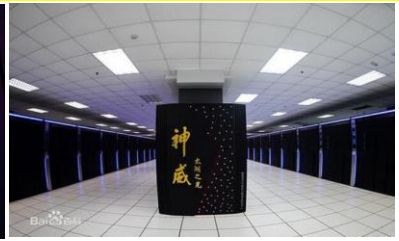
国际超级计算机TOP500排行榜“考题”

HPL(Linpack)：求解稠密线性代数方程组（1993-）

HPCG：求解稀疏线性代数方程组(Poisson方程离散系统)（2014-）

2013-2015

HPL: 61.4 PFlop/s (9)
HPCG: 0.58 PFlop/s (*13*)



2016-2017

93.0 PFlop/s (6)
0.48 PFlop/s (18)

2018-2019

148.6 PFlop/s (4)
2.9 PFlop/s (2)

2020-2021

442.0 PFlop/s (2)
16.9 PFlop/s (1)

2022-？

1102.0 PFlop/s (1)
HPCG: ？

浮点速度(Flop/s)：每秒执行的浮点运算次数。

应用需求

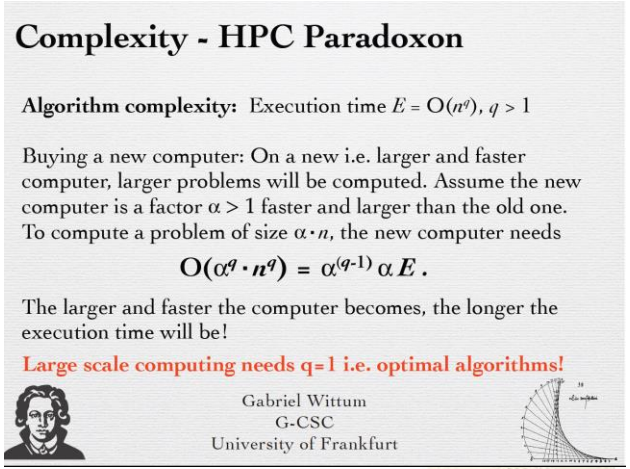
目标：快速求解 $Ax=b$

用户最关心：极小化到解时间 (极大化求解速度)

$$\frac{\text{Dof}}{\text{s}} = \frac{\text{Dof}}{\text{Flop}} \times \frac{\text{Flop}}{\text{s}}$$

1. 极大化算法速度(Dof/Flop)：选择一个好算法；
2. 极大化浮点速度(Flop/s)：选择一个好机器；选择一个好的并行实现方法/性能优化方法(针对基本操作，如SpMV)；
3. 极大化求解速度：算法与性能优化协同设计。

应用需求：(计算/机器)规模越大，对算法要求越高



Courtesy of G.Wittum
(中物院软件中心报告, 2016)

- Adaml定律(1967) → Gustafson 定律(1987)
- J.Gustafson, Reevaluating Amdahl's Law, CACM, 1988, 31(5): 532-533.
- J.Gustafson, et al, Development of Parallel methods for a 1024 processors hypercube, SIAM Sci. Comput., 1988, 9(4): 609-638.

$$E_\alpha^s = \frac{T_1}{T_\alpha \cdot p} \Rightarrow E_\alpha^w = \frac{T_1}{T_\alpha} \Rightarrow \overline{E}_\alpha^w = \frac{E_\alpha^w \cdot \alpha \cdot O(n^q)}{O(\alpha^q n^q)} = \frac{E_\alpha^w}{\alpha^{q-1}}$$

- Brandt's Golden Rule (1984): *The amount of computational work should be proportional to the amount of real physical changes in the computed solution.* Stalling numerical processes must be wrong. (ultimate upshot!)

应用需求：(计算/机器)规模越大，对算法要求越高

Complexity - HPC Paradoxon

Algorithm complexity: Execution time $E = O(n^q)$, $q > 1$

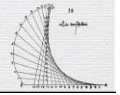
Buying a new computer: On a new i.e. larger and faster computer, larger problems will be computed. Assume the new computer is a factor $\alpha > 1$ faster and larger than the old one. To compute a problem of size $\alpha \cdot n$, the new computer needs

$$O(\alpha^q \cdot n^q) = \alpha^{(q-1)}\, \alpha\, E\,.$$

The larger and faster the computer becomes, the longer the execution time will be!

Large scale computing needs q=1 i.e. optimal algorithms!

Gabriel Wittum
G-CSC
University of Frankfurt



Horst Simon(LBNL), ISC2005, Heidelberg

The Top 10 Major Accomplishments in Supercomputing 1985 – 2005

- My own personal opinion
- Selected by "impact" and "change in perspective"

10) The TOP500 list
9) NAS Parallel Benchmark
8) The "grid"
7) Hierarchical algorithms: multigrid and fast multipole methods
6) HPCC initiative and Grand Challenge applications

- Adaml定律(1967) → Gustafson 定律(1987)
- J.Gustafson, Reevaluating Amdahl's Law, CACM, 1988, 31(5): 532-533.
- J.Gustafson, et al, Development of Parallel methods for a 1024 processors hypercube, SIAM Sci. Comput., 1988, 9(4): 609-638.

$$E_\alpha^s = \frac{T_1}{T_\alpha \cdot p} \Rightarrow E_\alpha^w = \frac{T_1}{T_\alpha} \Rightarrow \overline{E}_\alpha^w = \frac{E_\alpha^w \cdot \alpha \cdot O(n^q)}{O(\alpha^q n^q)} = \frac{E_\alpha^w}{\alpha^{q-1}}$$

- Brandt's Golden Rule (1984): *The amount of computational work should be proportional to the amount of real physical changes in the computed solution.* Stalling numerical processes must be wrong. (ultimate upshot!)

历史上重要的数值算法

Year	Development	Key early figures
263	Gaussian elimination	Hui, Lagrange, Gauss, Jacobi
1671	Newton's method	Newton, Raphson, Simpson
1795	Least-squares fitting	Gauss, Legendre
1814	Gauss quadrature	Gauss, Jacobi
1855	Adams ODE formulas	Adams, Bashforth
1895	Runge–Kutta ODE formulas	Runge, Heun, Kutta
1910	Finite differences for PDE	Richardson, Southwell, Courant, von Neumann, Lax
1936	Floating-point arithmetic	Zuse
1943	Finite elements for PDE	Courant, Feng, Argyris, Clough
1946	Splines	Schoenberg, de Casteljau, Bezier, de Boor
1947	Monte Carlo simulation	Ulam, von Neumann, Metropolis
1947	Simplex algorithm	Kantorovich, Dantzig
1952	Conjugate gradient iteration	Hestenes, Stiefel, Lanczos
1952	Stiff ODE solvers	Curtiss, Hirschfelder, Dahlquist, Gear
1954	Fortran	Backus
1958	Orthogonal linear algebra	Givens, Householder, Wilkinson, Golub
1959	Quasi-Newton iterations	Davidon, Fletcher, Powell, Broyden
1961	QR algorithm for eigenvalues	Rutishauser, Kublanovskaya, Francis, Wilkinson
1965	Fast Fourier transform	Gauss, Cooley, Tukey
1971	Spectral methods for PDE	Lanczos, Clenshaw, Orszag, Gottlieb
1971	Radial basis functions	Hardy, Askey, Duchon, Micchelli
1973	Multigrid iterations	Fedorenko, Brandt, Hackbusch
1976	EISPACK, LINPACK, LAPACK	Moler, Stewart, Smith, Dongarra, Demmel, Bai
1976	Nonsymmetric Krylov iterations	Vinsome, Saad, van der Vorst, Sorensen
1977	Preconditioned matrix iterations	van der Vorst, Meijerink
1977	MATLAB	Moler
1977	IEEE arithmetic	Kahan
1982	Wavelets	Morlet, Grossmann, Meyer, Daubechies
1984	LP interior-point methods	Khachiyan, Karmarkar, Megiddo
1987	Fast multipole method	Rokhlin, Greengard
1991	Automatic differentiation	Bischof, Carle, Griewank

N.Trefethen, *Numerical Analysis, in Princeton Companion to Mathematics, T. Gowers, 2006.*

$$Ax=b$$

高斯消元(GE, 236)

共轭梯度(CG,1952)

正交分解(SVD, 1958)

多重网格(MG, 1973)

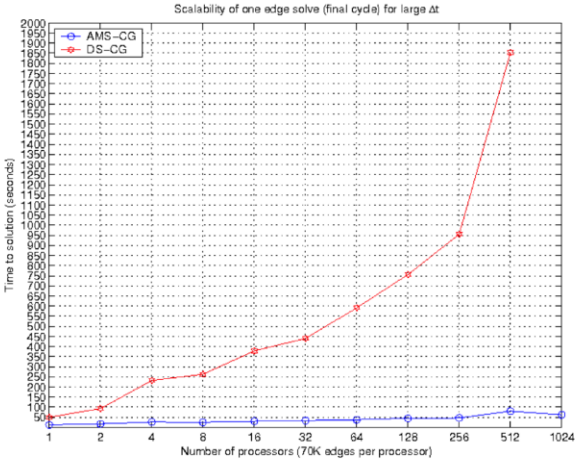
Linpack等(1976)

Krylov-GMRES(1976)

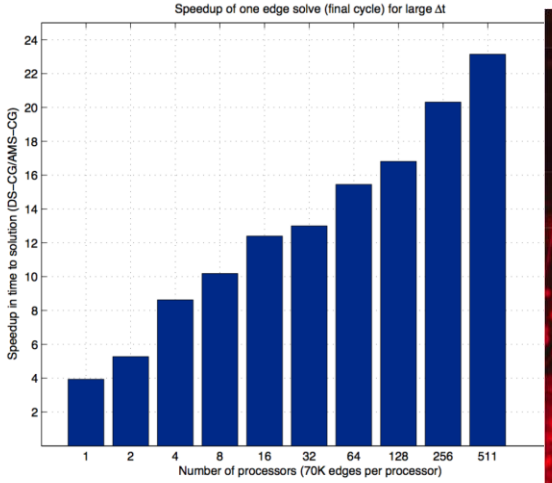
预条件算法(1977)

MATLAB(1977)

历史上重要的数值算法

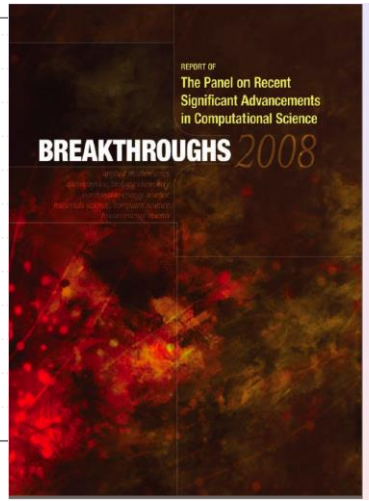


(a) Scalability



(b) Speedup

Figure 1: Scalability and Speedup of HX preconditioner



SciDAC Review, DOE, USA, March,2009.

Courtesy of Jinchao Xu's slides

AMS is a perfect example of how fundamental mathematical research can lead to important software advances in high-performance computing.

许进超教授的算法应该列入

许进超教授(PSU)：BPX预条件算法(1988)、子空间校正框架(1992)、HX/AMS预条件(2007) 。

主流算法框架：预条件迭代方法

$Ax = b$

Given x^0 ;

Do Until Convergence:

$x^{k+1} \leftarrow G(A, b, x^k)$;

Krylov methods: CG, GMRES, BiCGSTAB,…

G为迭代格式，一组基本运算的组合 G(SpMV,dot,axpy,...)

主流算法框架：预条件迭代方法

$M^{-1}Ax = M^{-1}b$, M^{-1} ($\approx A^{-1}$)为预条件子

Given x^0 ;

Do Until Convergence:

$x^{k+1} \leftarrow G(M^{-1}, A, b, x^k)$;

Krylov methods: CG, GMRES, BiCGSTAB,…

Preconditioners: JAC/G-S, ILU, GMG, AMG,DDM,…

G为迭代格式，一组基本运算的组合 G(precond,SpMV,dot,axpy,...)

关于预条件子：1948年，Turing在他关于直接法舍入误差分析的文章中首次使用Preconditioning一词。1968年，Evans首次将其用于方程组求解。

预条件子本质：高效求解残差方程

$$x^{k+1} = x^k + \varepsilon$$

$$\varepsilon \approx e^k = (x^* - x^k)$$

$$Ae^k = r^k = (b - Ax^k)$$ 残差方程

$$x^{k+1} = x^k + M^{-1}r^k$$ $\varepsilon = M^{-1}r^k$

Courtesy of Nick Higham

Turing's Paper

Rounding-Off Errors in Matrix Processes, *Quart. J. Mech. and Applied Math.*, 1948.

离实际应用的差距

当前解法器远远满足不了实际应用的需求

离Poisson方程有多远？

Poisson方程求解速度有多快？

回顾TOP500“考题”

HPL(Linpack)：求解稠密线性代数方程组 $Ax = b$ (1993-)
算法：高斯消元(GE)/ 矩阵LU分解
计算复杂度：O(n^3)

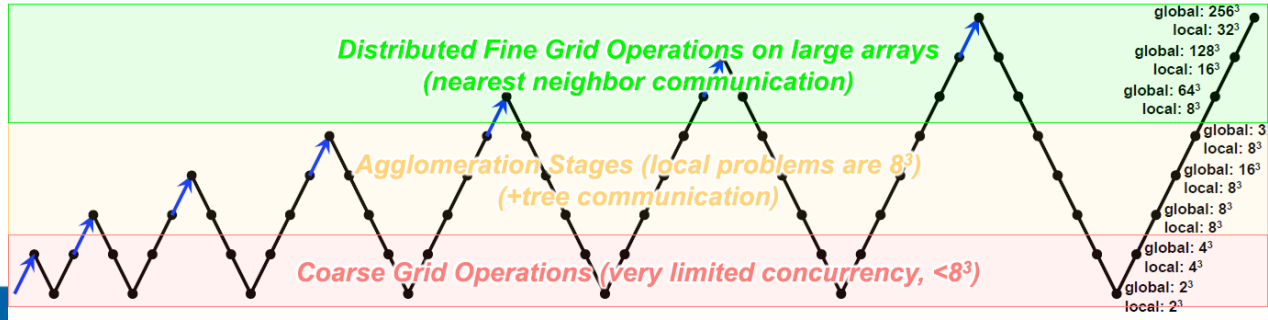
HPCG：求解3D-7pts离散Poisson方程 (2014-)
算法：多重网格方法(MG) + 共轭梯度方法(CG)
计算复杂度： $O(n + n_c^3)$

HPGMG：求解高阶FVM离散Poisson方程 (2014-)
算法：FMG方法 (在离散误差意义下的直接法)
计算复杂度：O(n)

S. William (LBNL), et al.
http://hpgmg.org

指标：
FLOP/s

$$\text{DOF/s} = \frac{\text{DOF}}{\text{FLOP}} \times \frac{\text{FLOP}}{\text{s}}$$



回顾TOP500“考题”

HPGMG: S.William, et al, SIAM NEWS, 2018(4).

midterm elections this fall.

https://www.nsf.gov/news/special_reports/big_ideas/

percent, while the Office of Multidisciplinary Activities (OMA) is projected to grow 197

https://www.nsf.gov/fy2019/toc.jsp

The High-Performance Geometric Multigrid: An HPC Performance Benchmark

By Samuel Williams, Mark F. Adams, and Jed Brown

The High-Performance LINPACK (HPL) Benchmark is the most widely-recognized metric for ranking high-performance computing (HPC) systems. It rose to prominence in the early 1990s, when its predicted ranking of a system correlated with the system's efficacy for full-scale applications. Computer system vendors sought designs that would increase HPL performance, which in turn improved overall application behavior.

Unfortunately, this is no longer the case; in fact, the opposite is now true. Although the HPL Benchmark continues to be an effective proxy for applications based on dense linear algebra, it has lost its proficiency as a proxy for many applications relevant to the HPC community. HPL rankings of computer systems, which utilize work-optimal algorithms with high bandwidth and low latency requirements, are not well-correlated to real application performance nowadays. Motived by HPL's increasing inapplicability, the High-Performance Geometric Multigrid (HPGMG) incorporates machine sensitivities that correlate well with the sensitivities of HPC applications.

HPGMG complements both HPL and the new High-Performance Conjugate Gradients (HPCG) Benchmark[1] [2], with more stress on the memory system and network fabric than HPL and HPCG respectively. The TOP500 list is currently adding new rankings for HPCG and HPGMG to complement the venerable HPL.

HPGMG Design Principles

The following design principles of HPGMG are discussed in the HPGMG 1.0 whitepaper [1]:

A benchmark must reflect improvements to computer systems that benefit our applications, and is essential for documenting future improvements to HPC systems. The metric must be

designed so that, as we optimize metric results for a particular platform, the changes will also lead to performance improvements realized by real

- Accurately reflect the characteristics of contemporary high-efficiency algorithms.
- Accurately reflect the principle challenges of future hardware design — a balanced combination of memory bandwidth, interconnect performance (both for small and large messages), computational performance, and reliability. It

by user-provisioning the hardware in very rate. of these areas. A machine designed for the sole purpose of performance on our metric should result in a "pretty good" machine for scientific and engineering applications.

See Geometric Mult

Rank	Site	System	10^9 DOF/s h, 2h, 4h	Parallelization MPI	OMP	ACC	DOF p Proce
1	National Supercomputing Center in Wuxi (China)	Sunway TaihuLight – Sunway MPP, SW26010 260C 1.45GHz, Sunway, NRCPC	1036 565 163	131072	1	1	32M
2	Department of Energy/Office of Science/Lawrence Berkeley National Laboratory/National Energy Research Scientific Computing Center (USA)	Cori – Cray XC40, Intel Xeon Phi 7250 68C 1.4GHz, Aries interconnect, Cray Inc.	859 376 87	65536	8	0	16M
3	Department of Energy/Office of Science/Argonne National Laboratory (USA)	Mira – BlueGene/Q, Power BQC 16C 1.60GHz, Custom interconnect, IBM	500 313 107	49152	64	0	36M
		(baseline)	395 286 107	49152	64	0	36M
4	Höchstleistungsrechenzentrum Stuttgart (Germany)	Hazel Hen – Cray XC40, Xeon E5-2680v3 12C 2.5GHz, Aries interconnect, Cray Inc.	495 411 221	15408	12	0	192I
5	Department of Energy/Office of Science/Oak Ridge National Laboratory (USA)	Titan – Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x, Cray Inc.	440 163 38.9	16384	4	1	32
		(CPU-only)	161 82.5 23.7	36864	8	0	
6	King Abdullah University of Science and Technology (Saudi Arabia)	Shaheen II – Cray XC40, Xeon E5-2698v3 16C 2.3GHz, Aries interconnect, Cray Inc.	326 287 175	12288	16	0	
7	Department of Energy/Office of Science/Lawrence Berkeley National Laboratory/National Energy Research Scientific Computing Center (USA)	Edison – Cray XC30, Intel Xeon E5-2695v2 12C 2.4GHz, Aries interconnect, Cray Inc.	296 246 127	10648	12	0	
8	Swiss National Supercomputing Centre (Switzerland)	Piz Daint – Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect, NVIDIA K20x, Cray Inc.	153 68.8 18.5	4096	8		
		(CPU-only)	85.1 62.6 24.7	4096	8		
9	Cyberscience Center, Tohoku University (Japan)	SX-ACE – 4C 1GHz, IXS NEC	73.8 45.2 15.6	4096	1		

Table 1. A compressed sampling of the High-Performance Geometric Multigrid (HPGMG) list of the world's largest s

[1] https://sinews.siam.org/Details-Page/e-high-performance-conjugate-gradients-hpcg-benchmark

2018.11 排行榜

No.1 K Computer

DOF = 5.971 万亿, 663552核(82944*8),

3.37s, 1243 GDOF/s

No.2 Sunway TaihuLight, TOP500-3

DOF = 4.194 万亿, 131072核,

4.05s, 1036 GDOF/s

No.7 Edison-Cray XC30(LLNL), TOP500-78

DOF = 1.362 万亿, 127776核(10648*12)

4.6s, 296 GDOF/s

实际应用离Piosson方程有多远？

三个应用实例：

（1）工业仿真：电子学系统级封装

（2）工程计算：大坝结构力学分析

（3）科学计算：激光聚变数值模拟

Poisson速度比(RP)：

在相同计算条件下(程序/机器/核数)，求解相同自由度的Poisson方程与实际应用问题，两者的最小到解时间之比.

$$\text{DOF/s} = \frac{\text{DOF}}{\text{FLOP}} \times \frac{\text{FLOP}}{\text{s}}$$

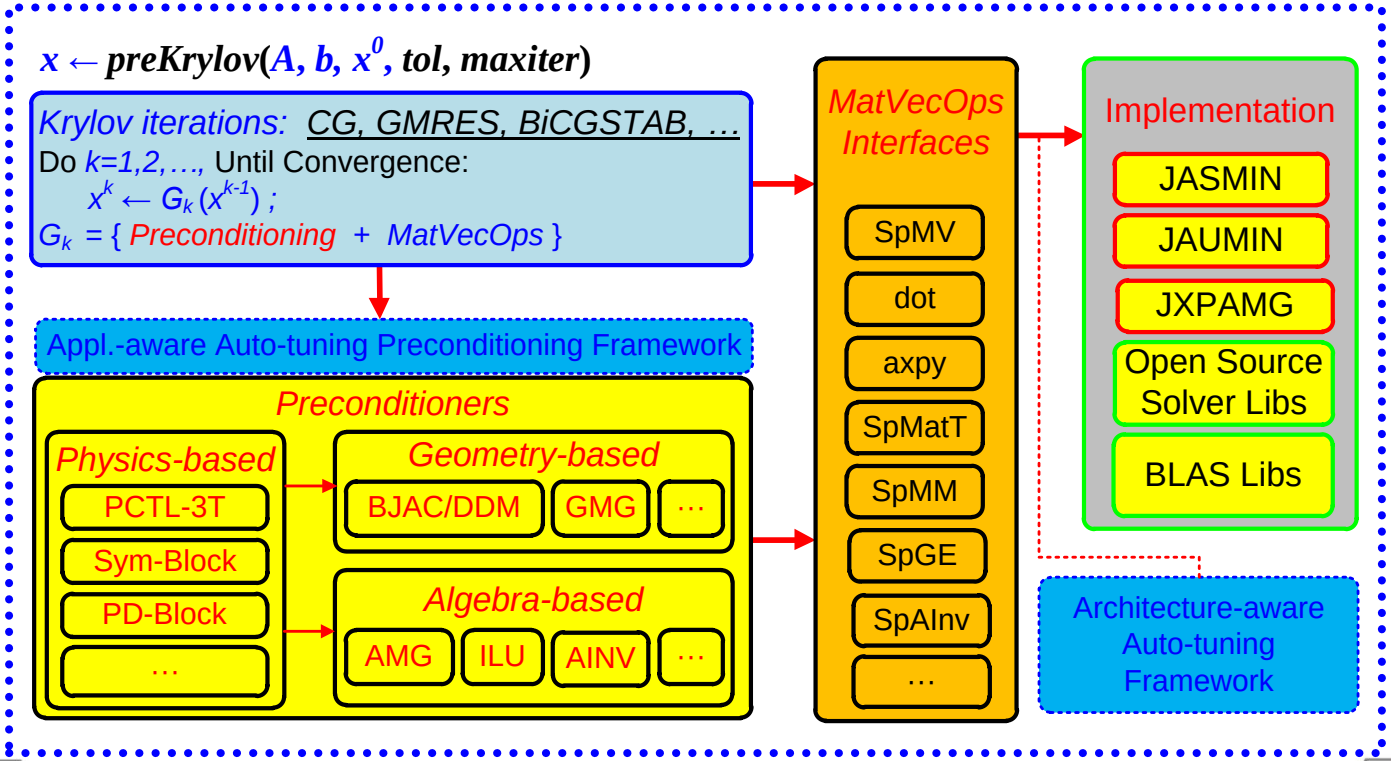
九所某机器：DOF = 2.6亿, 8000核

HPGMG：0.09s, 2.8 GDOF/s

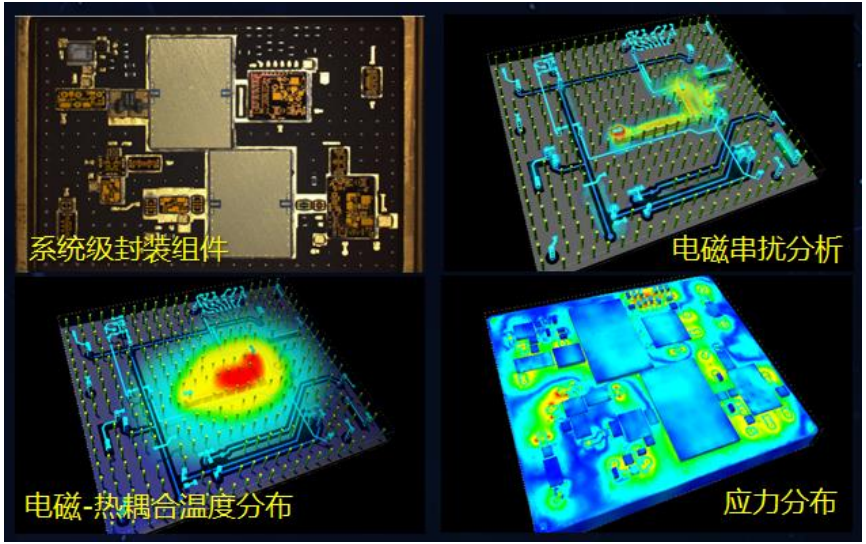
JPSOL+JXPAMG：69iters, 2.5s, 0.1GDOF/s

实际应用离Piosson方程有多远？

JPSOL: Parallel SOLver based on JASMIN/JAUMIN



例一：电子学系统级封装模拟

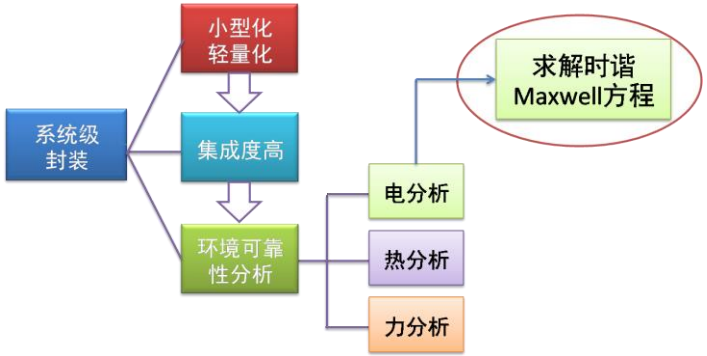


2016年，三星公司生产的Note7手机发生多起爆炸，导致全球召回，损失巨大。

爆炸主要原因：手机集成电路功率电源与散热结构设计缺陷。

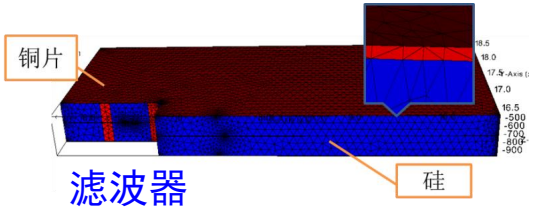
硅通孔布局优化、信号完整性分析、功率分配网络优化、射频参数优化。

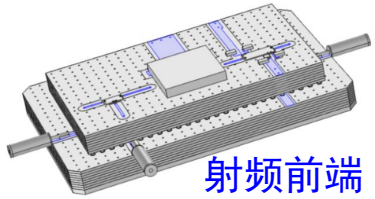
例一：电子学系统级封装模拟



$$
\begin{cases}
curl\,\dfrac{1}{\mu_r}curl\,E - k_0^{\,2}\varepsilon_r E = 0 \quad in \quad \Omega \\
\vec{n}\times\dfrac{1}{\mu_r}curl E - i\,k_g(\vec{n}\times E)\times\vec{n} = \chi \quad on \quad \Gamma_1 \\
\vec{n}\times\dfrac{1}{\mu_r}curl E - i\,k_g(\vec{n}\times E)\times\vec{n} = 0 \quad on \quad \Gamma_2 \\
\vec{n}\times E = 0 \quad on \quad \Gamma_r
\end{cases}
$$

$$
k_g \in (k_0,\quad k_0\sqrt{max(Re(\varepsilon_r))}),\quad \Gamma_r = \partial\Omega\setminus(\Gamma_1\cup\Gamma_2)
$$





- 大量薄片部件，实体多，实体间尺度跨越大，结构复杂。材料分布复杂，求解大规模散射问题的方法失效。
- 不同的实体材料差异巨大，材料分布混乱。材料间物理参数尺度变化大。
- 单元二面角太大或太小，长扁四面体，薄片结构附近单元尺寸变化剧烈。
- 多端口，多频率点，需反复求解线性系统。
- 复数系统，一个半正定算子减去一个正定算子导致方程不定。

例一：电子学系统级封装模拟

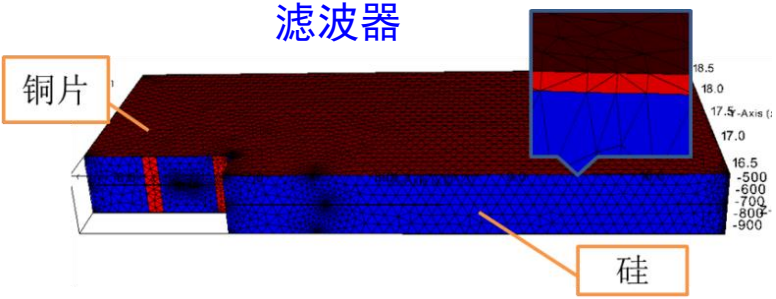
- COMSOL (GMG, 直接法)
- HFSS (DDM, 直接法)
- Feko (DDM, 直接法)
- JEMS-CDS (中物院软件中心)

求解策略	有效算法集@JPSOL
基于原复数矩阵	LU，ASM_GMRES
基于等价实数矩阵	LU，M(LU)_GMRES，M_d(ASM)_GMRES M_d(HX)_GMRES , M_d(HX)_CG

胡少亮，徐小文等．系统级封装应用中时谐Maxwell 方程大规模计算的求解算法：现状与挑战．计算物理，38(2)，2021

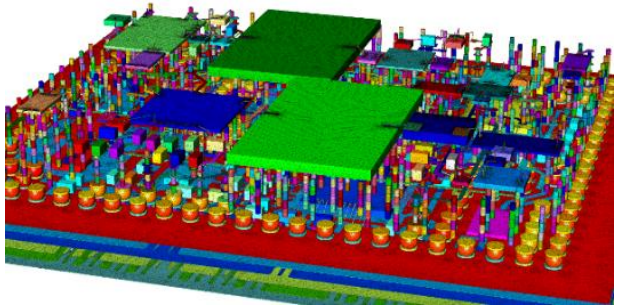
例一：电子学系统级封装模拟

滤波器



射频前端：

2111个实体，尺寸跨度0.025-20mm



DOFs	直接法	预条件迭代法
1,765,074	0.047	M(LU) 0.100
13,957,912	0.003	M_d (HX) 0.023
111,017,824	~0	M_d (HX) 0.019

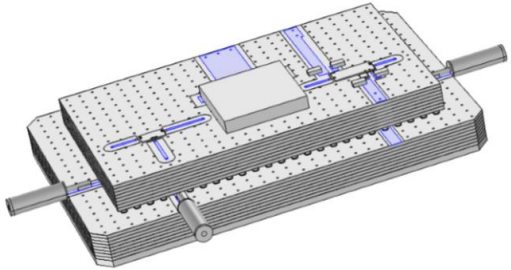
Poisson速度比

DOFs	直接法	预条件迭代法
12,502,683	0.021	M(LU) 0.098
99,071,137	~0	M_d (HX) 0.009

胡少亮，徐小文等．系统级封装应用中时谐Maxwell 方程大规模计算的求解算法：现状与挑战．计算物理，38(2)，2021

例一：电子学系统级封装模拟

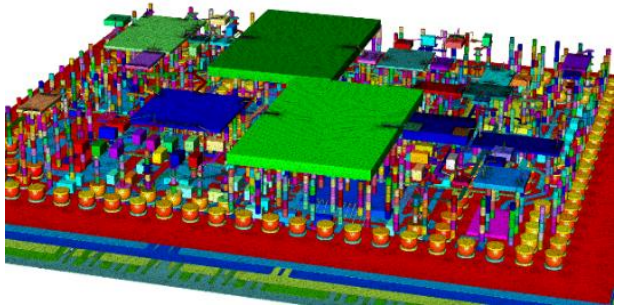
射频前端：4800个实体，尺寸跨度0.01-10mm　　材料：空气、硅、铜、HTCC和Teflon



DOFs	直接法	预条件迭代法
7,134,040	0.0015	M_d(HX) 0.006
56,939,637	~0	M_d(HX) 0.002

射频前端：

2111个实体，尺寸跨度0.025-20mm

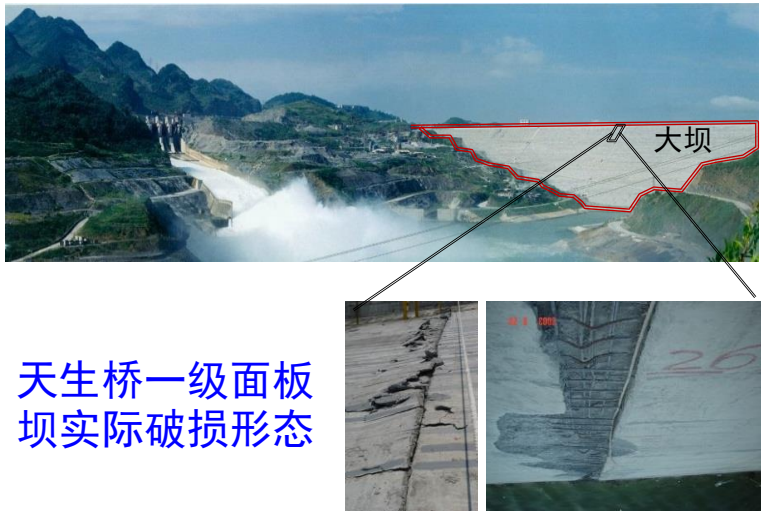


Poisson速度比

DOFs	直接法	预条件迭代法
12,502,683	0.021	M(LU) 0.098
99,071,137	~0	M_d (HX) 0.009

胡少亮，徐小文等．系统级封装应用中时谐Maxwell 方程大规模计算的求解算法：现状与挑战．计算物理，38(2)，2021

例二：大坝结构力学分析



天生桥一级面板坝实际破损形态

力-热模型

$$\begin{cases} \rho\ddot{u}_i + \gamma\dot{u}_i - \sigma_{ij,j} = f_i \\ \rho c\dot{\theta} + Kd\alpha_L\theta_0\dot{\epsilon}_{kk} - \kappa\alpha\dot{\theta}_{ii} = \rho s \end{cases}$$

接触模型

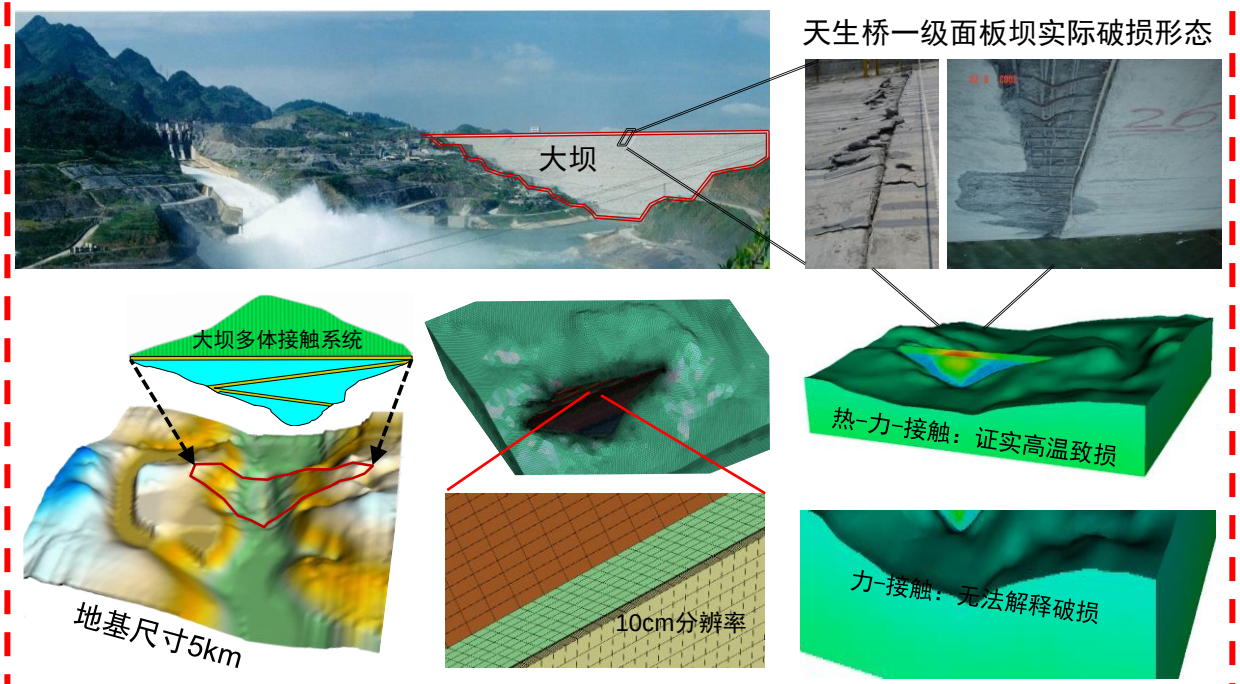
$$\begin{cases} p_n(\mathrm{x}) \le 0, & g(\mathrm{x}) \ge 0, & p_n(\mathrm{x})g(\mathrm{x}) = 0 \\ \omega(\mathrm{x}) \le 0, & \beta(\mathrm{x}) \ge 0, & \omega(\mathrm{x})\beta(\mathrm{x}) = 0 \end{cases}$$

$$g(\mathrm{x}) = -\mathrm{n}\cdot(\mathrm{x}^s - \mathrm{x}^m)$$

- 多物理耦合、接触非线性：水利工程中挑战性计算问题。
- 复杂构型、几何保真：高质量几何建模与网格生成。
- 隐式求解：载荷步-非线性接触迭代

例二：大坝结构力学分析

广西隆林天生桥面板堆石坝：坝高178米，坝顶宽1168米



应用软件：PANDA
计算规模：11亿Dofs,
天河二号：16000 CPU核
计算时间：37分钟

Poisson速度比 = 0.0022

首次通过数值模拟手段定量分析和证实了混凝土升温膨胀效应是导致该坝频繁发生面板挤压破损的主要原因，得到设计单位认可。

Rong Tian, Mozhen Zhou, Jingtao Wang, et al, A challenging dam structural analysis: large-scale implicit thermo-mechanical coupled contact simulation on Tianhe-II, Computational Mechanics, DOI 10.1007/s00466-018-1586-5, 2018.

例二：大坝结构力学分析

The numerical algorithms mentioned in Sect. 4 have been implemented into our in-house software PANDA [86,87]. PANDA is built upon a HPC middleware called JAUMIN [88–91] in which the parallel linear solver package is called JPSOL [81]. The architecture of PANDA is illustrated in Fig. 10. The major goal of PANDA has been set to bridge the complex engineering mechanics analyses to China's post-petaflops supercomputers. PANDA includes four modules: PANDA_StaVib for static and modal analyses based on the finite element method, PANDA_Impact for dynamic analy-

We compare the preconditioner proposed in Sect. 5.3 with the other typical preconditioners available in the well-known external open source code PETSc [101] by solving the thermo-mechanical-contact coupled problem of 118,608 dofs. The preconditioners provided by PETSc are tested using default parameters (this may deteriorate the performance of PETSc to some extents). JPSOL is tested using its own BiCGStab solver with the AMG preconditioner proposed in Sect. 4.2.

Table 1 Converge performance of linear solvers and preconditioners

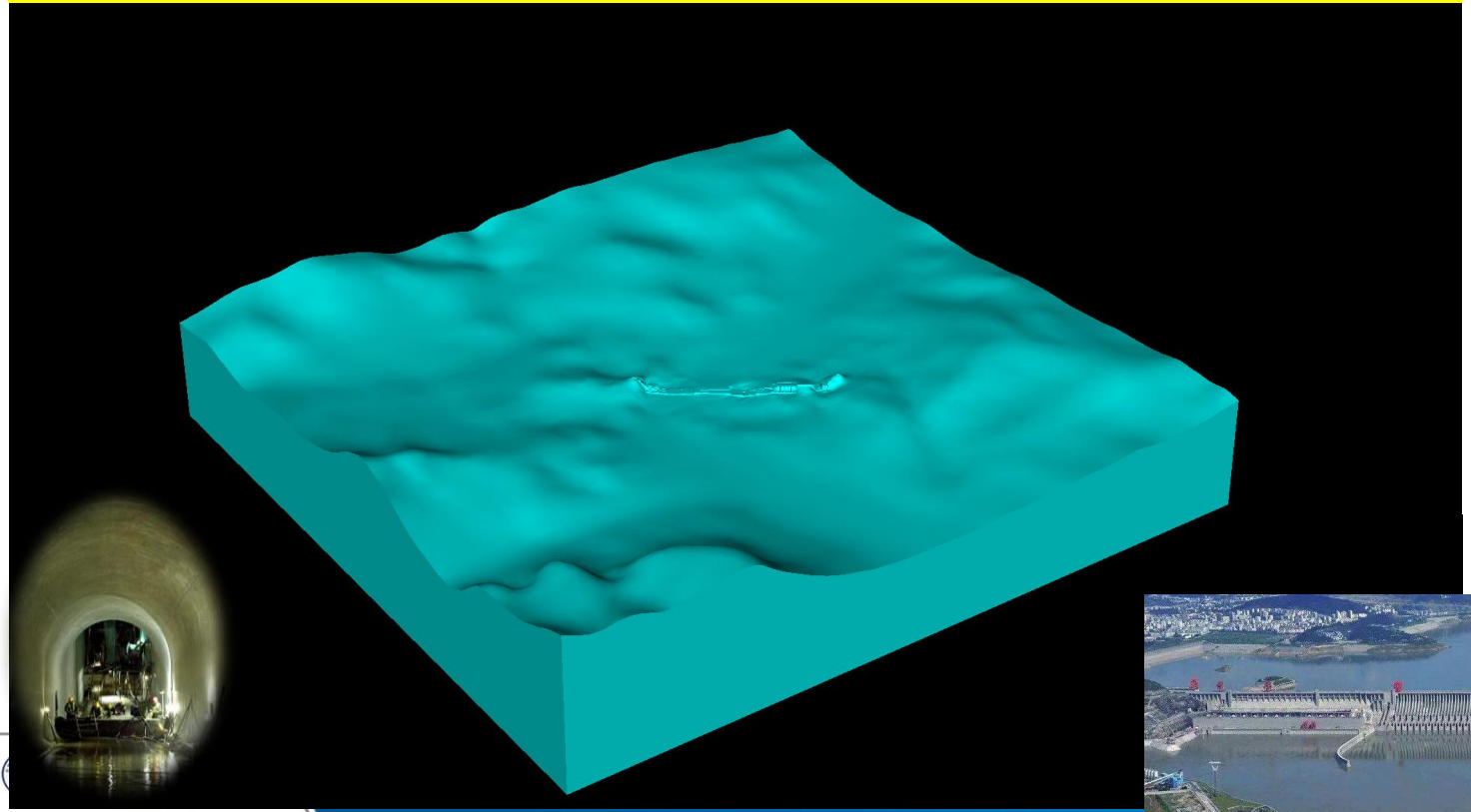
	Preconditioner	CG	GMRES	BiCGStab
PETSc	None	×	⊗	⊗
	ILU	×	×	×
	GAMG	×	×	×
	Jacobi	×	⊗	⋆(1)
	SOR	×	⊗	⋆(2)
	SOR Eisenstat	×	⊗	⋆(3)
	Hypre	×	⋆(4)	⋆(5)
JPSOL	AMG	×	⋆(6)	⋆(7)

(1) Converged at 14,381 iterations within 126.2 s
(2) Converged at 4062 iterations within 83.6 s
(3) Converged at 3714 iterations within 93.8 s
(4) Converged at 1730 iterations within 254.5 s
(5) Converged at 625 iterations within 185.8 s
(6) Converged at 759 iterations within 56.7 s
(7) Converged at 209 iterations within 35.2 s

Rong Tian, Mozhen Zhou, Jingtao Wang, et al, A challenging dam structural analysis: large-scale implicit thermo-mechanical coupled contact simulation on Tianhe-II, Computational Mechanics, DOI 10.1007/s00466-018-1586-5, 2018.

例二：大坝结构力学分析

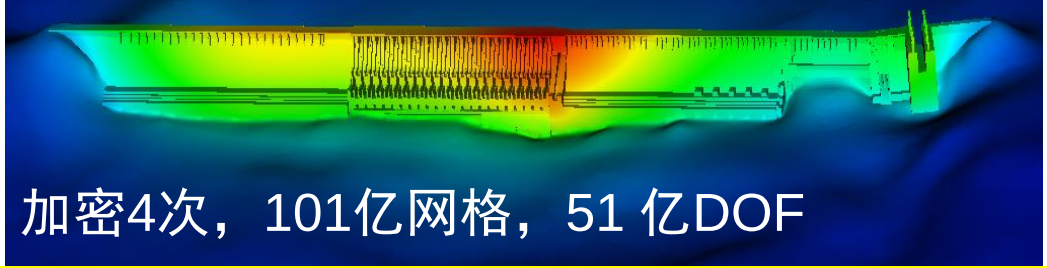
三峡大坝亚米分辨率结构力学分析：10公里地基-整个坝体-80个坝段-130纵缝、80横缝-廊道-0.25m排水孔。



- 模型：三峡大坝线弹性静力学分析。含地基和详细坝段，考虑全局重力和水压载荷，底部使用固定边界条件。
- 中国水利水电科学研究院、中物院数值模拟软件中心
- 应用软件: PANDA

例二：大坝结构力学

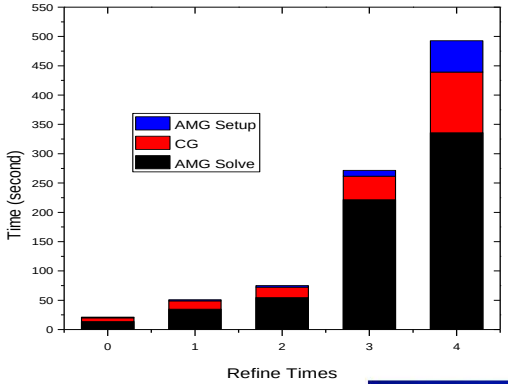
初始四面体网格146万 自由度

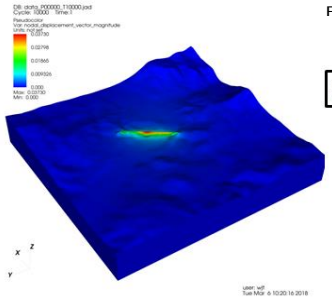


加密4次，101亿网格，51 亿DOF

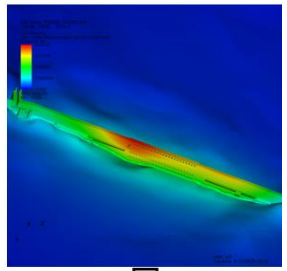
PANDA@JAUMIN；AMG-CG@JXPAMG-JPSOL
2018/3：48000核@天河二号，6.5小时，982迭代;
2018/5：49152核@九所机器，25分钟，593迭代;
2019/10：49152核@九所机器，7.3分钟，593迭代;

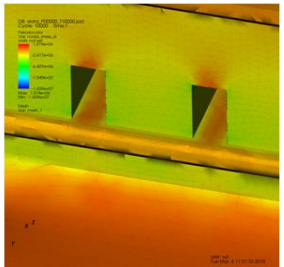
3DPoisson: DOF 58.3亿, 49152核, 15.1s.
Poisson速度比：0.034



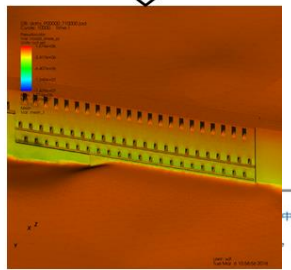


整体位移

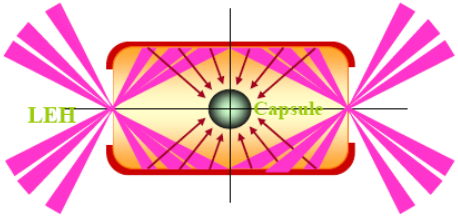


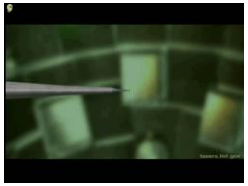


局部应力



例三：激光聚变数值模拟

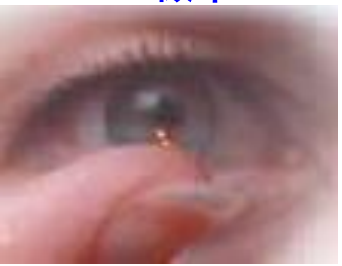


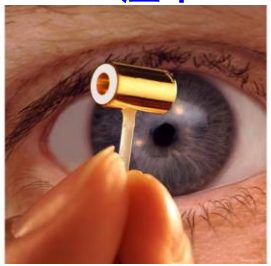


1 微米

1厘米

10米





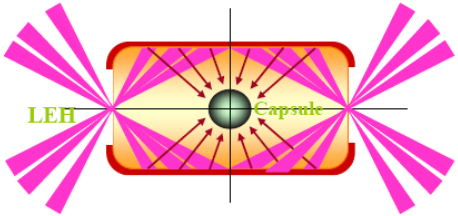
3个足球场

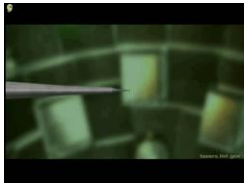
美国国家点火装置(NIF)@LLNL：

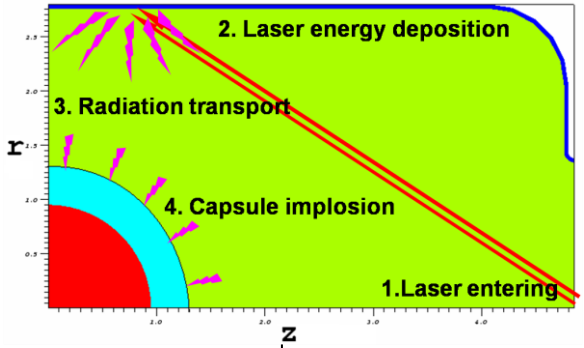
激光192束，能量1.8MJ。

2021年8月8日：NIF获得了超过1.3MJ的聚变放能，将 3个足球场大小的激光束，聚焦到直径为 6mm大小的靶丸上，产生了人类头发丝直径大小的点火热斑，在0.1ns内放出了1亿亿瓦特的聚变功率。 NNSA局长称这一结果无与伦比，使人类处于聚变点火门槛。

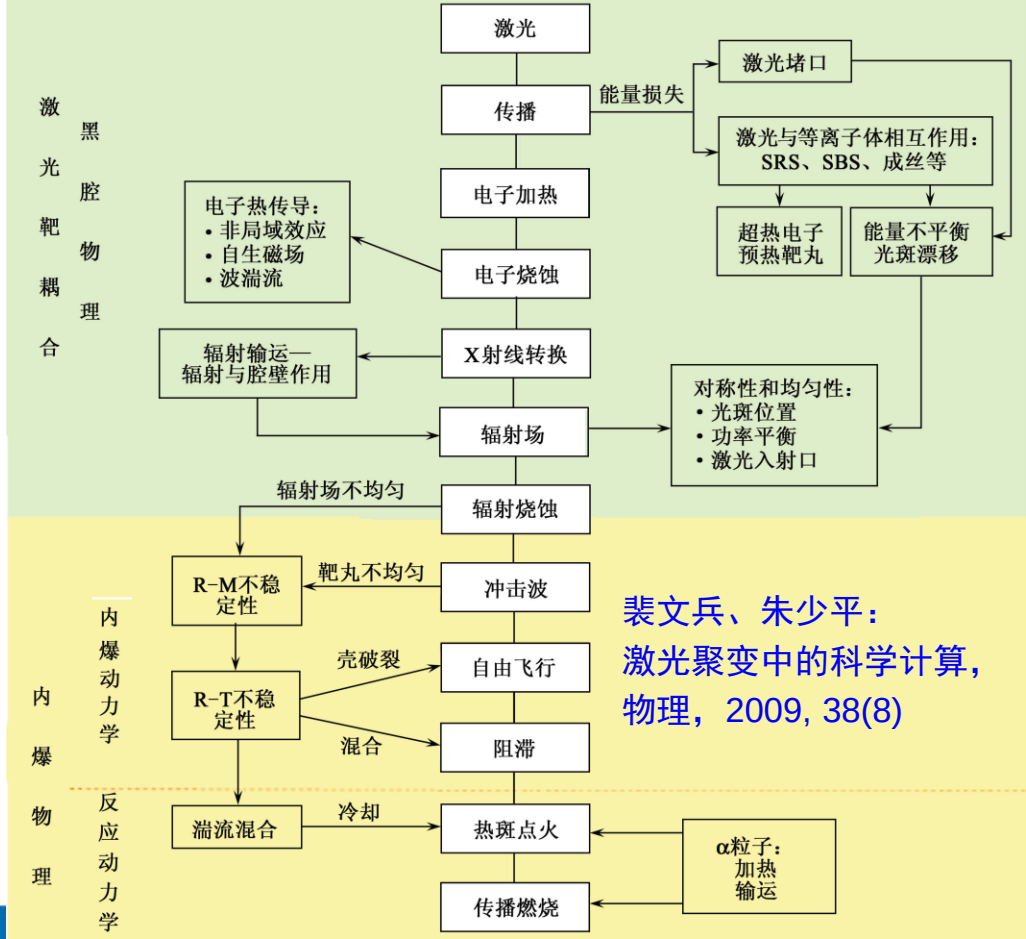
例三：激光聚变数值模拟





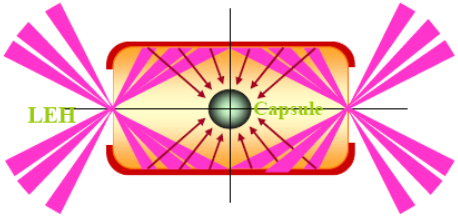


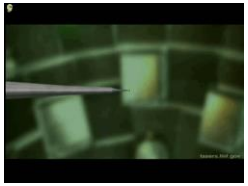
涉及学科：辐射流体力学、粒子输运、等离子体物理、原子物理、热核反应动力学等。

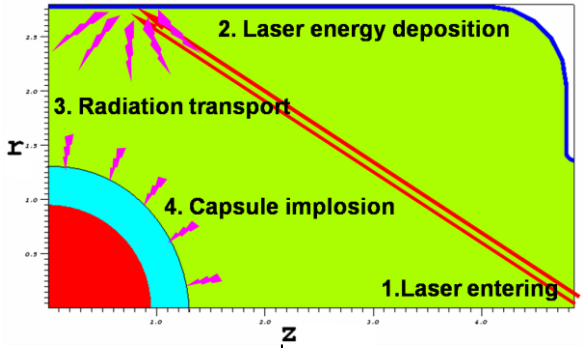


裴文兵、朱少平：
激光聚变中的科学计算，物理，2009, 38(8)

例三：激光聚变数值模拟



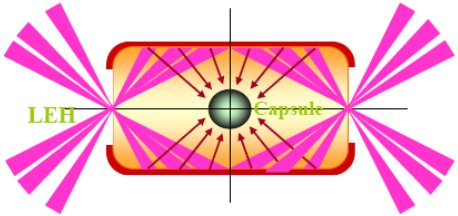


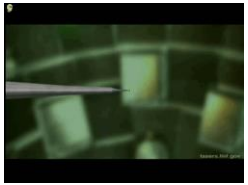


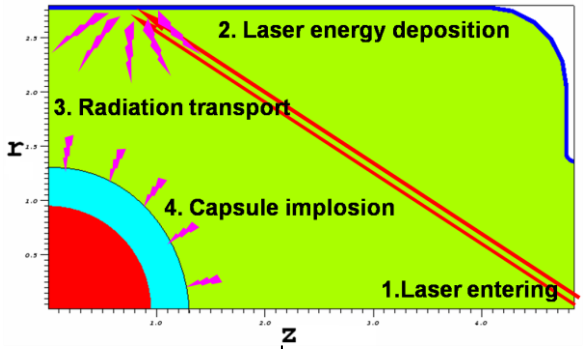
涉及学科：辐射流体力学、粒子输运、等离子体物理、原子物理、热核反应动力学等。

- 问题特点
  - 多介质、大变形、极端物理状态
  - 多时空尺度（1微米～亚厘米、fs-100ns）
  - 多物理过程（非平衡、强耦合）
- 模拟要求
  - 实际构型
  - 定量正确
- 软件要求
  - 高精细建模
  - 高分辨率高精度离散
  - 高效率运行

例三：激光聚变数值模拟







涉及学科：辐射流体力学、粒子输运、等离子体物理、原子物理、热核反应动力学等。

$$\frac{\partial \rho}{\partial t}+\nabla\cdot(\rho\vec{u})=0$$

辐射流体力学（RHD）

$$\frac{\partial \rho\vec{u}}{\partial t}+\nabla\cdot(\rho\vec{u}\vec{u})+\nabla p=0$$

$$\frac{\partial E_g}{\partial t}+\nabla\cdot(E_g\vec{u})+p_g\nabla\cdot\vec{u}+\nabla\cdot\vec{F}_g=c\left(\sigma_{Bg}E_{pg}(T_e)-\sigma_{pg}E_g\right)+S_g$$

$$\frac{\partial \rho\varepsilon_e}{\partial t}+\nabla\cdot(\vec{u}\rho\varepsilon_e)+p_e\nabla\cdot\vec{u}+\nabla\cdot\vec{F}_e=-c\sum\nolimits_g\left(\sigma_{Bg}E_{pg}(T_e)-\sigma_{pg}E_g\right)+\omega_{ie}(T_i-T_e)$$

$$\frac{\partial \rho\varepsilon_i}{\partial t}+\nabla\cdot(\vec{u}\rho\varepsilon_i)+p_i\nabla\cdot\vec{u}+\nabla\cdot\vec{F}_i=-\omega_{ie}(T_i-T_e)$$

$g=1,2,\ldots,G$　**G**: 辐射能群数

$$A=\begin{bmatrix}\mathbf{A_r} & \mathbf{D_{re}} & 0\\ \mathbf{D_{er}} & A_e & D_{ei}\\ 0 & D_{ie} & A_i\end{bmatrix}\qquad \mathbf{A_r}=\begin{pmatrix}A_1 & & 0\\ & \ddots & \\ 0 & & A_G\end{pmatrix}$$

$$\mathbf{D_{re}}=(D_{1e},\ldots,D_{Ge})^T$$

$$\mathbf{D_{er}}=(D_{e1},\ldots,D_{eG})$$

例三：激光聚变数值模拟

- ICF典型模型@JASMIN/JAUMIN (解法器：JPSOL-JXPAMG)

RHD模型	DOF	CPU核数	求解时间(平均)	Poisson速度比	Poisson求解时间 (自由度)
2D3T(ALE)	30万	10	0.39秒	0.36	0.14秒(30.2万)
2D (20群ALE)	10万	10	0.14秒	0.29	0.04秒(10.2万)
2D3T(Euler)	2097万	1024	0.63秒	0.33	0.20秒(2116万)
3D3T(Euler)	17.3亿	16384	137.5秒	0.088	12.1秒(17.9亿)
3D(20群Euler)	21亿	32000	604.4秒	0.014	8.4秒(21.0亿)
3D(64群ALE)	7040万	2100	571.2秒	0.008	4.5秒 (7147万)

解法器算法与性能优化竞赛(SolverChallenge)

解法器快速算法及应用研讨会(Solver)
https://www.solver-conference.cn/

2021
邀请函
第一届线性解法器算法与性能优化竞赛
(SolverChallenge)

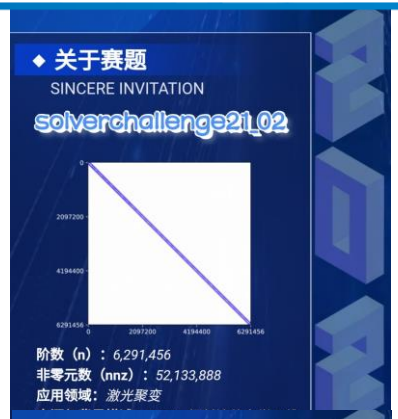
第一届
(SolverChallenge21)

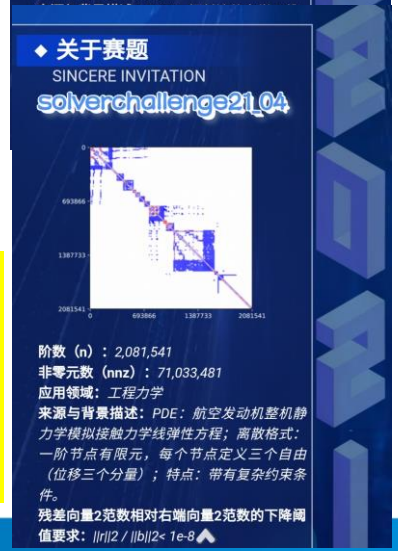
➤7月30日-8月1日
➤新疆·克拉玛依
SINCERELY INVITE YOU

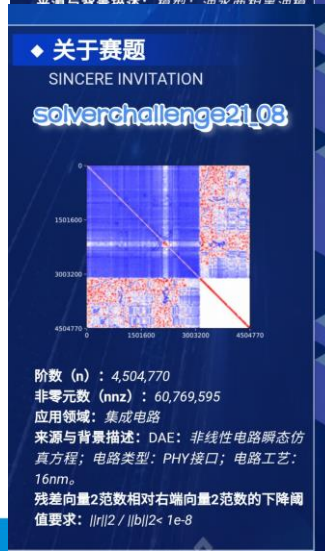
由于疫情原因，会议及竞赛答辩推迟到12月26-27日在北京举行。

10个竞赛题目
来自5个行业领域
激光聚变
工程力学
油藏模拟
电子学系统
集成电路

Solver22：
7月20日-24日，重庆。
SolverChallenge22：
已经启动，7月23日揭晓。







解法器挑战：以AMG算法为例

为什么是AMG？

目标：快速求解 $Ax=b$

支撑实际应用大规模计算

用户友好（通用性、健壮性）
适应批量应用（行业-领域）

计算复杂度低、并行度高
适应不同机器架构（同构-异构）

代数多重网格
(AMG)

多重网格：是一种多级结构算法，具有最优计算复杂度性质。

Algebraic Multigrid (AMG) for $Ax=b$

- **AMG softwares/libraries**
  - BoomerAMG in Hypre, LLNL (开源)
  - MueLu in Trilinos, SNL (开源)
  - GAMG in PETSc, LBNL (开源)
  - PyAMG, University of Illinois (开源)
  - LAMG, LANL
  - FAMG in UG, G-CSC, Frankfurt
  - JXPAMG in JAS/UMIN, IAPCM & XTU
  - SAMG, Fraunhofer SCAI (商业)
  - AGMG, Universite libre de Bruxelles (商业)
  - AmgX, NVIDIA (商业, ANYSIS/FLUENT)
  - SMS-AMG, VINAS Ltd (商业)

首个并行AMG软件(1998-)

AMG效率与问题/机器相关。

首个AMG商业软件(1995-)

Geometric Multigrid (GMG)

Ae = r : Divide and Conquer Rule

e = Low frequency + High frequency

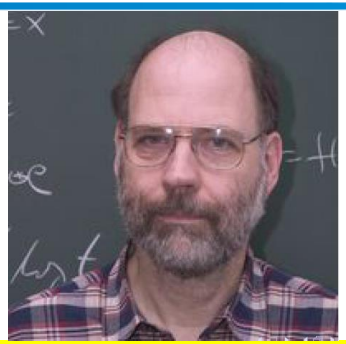
粗网格校正

光滑子

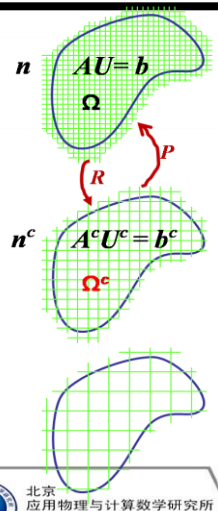
$M^{-1} = P\,(A^c)^{-1}R \cdot S$

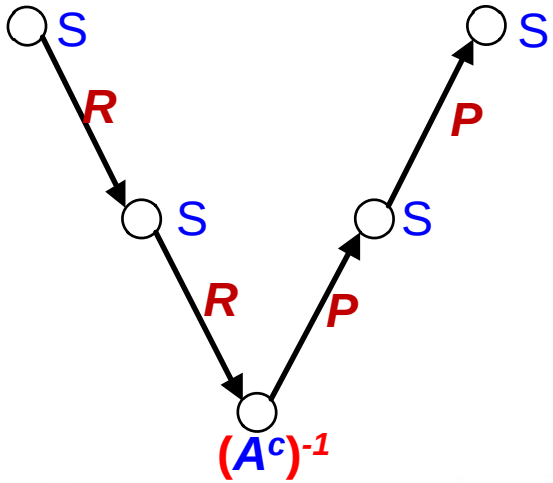
插值和限制算子

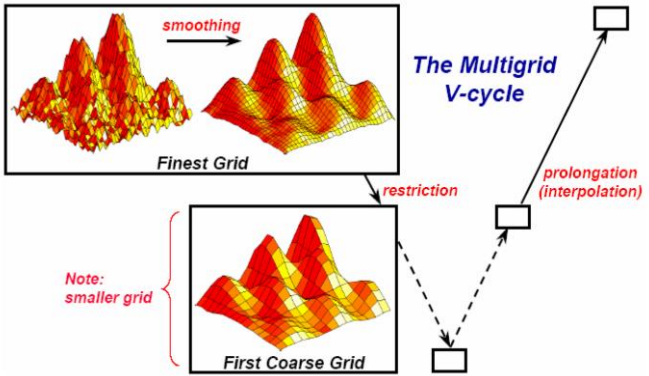
GMG：实用化问题



Fedorenko, Brandt, Hackbusch,1973







GMG -> AMG

Ae = r : Divide and Conquer Rule

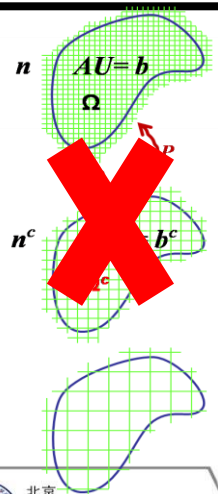
e = Low frequency + High frequency

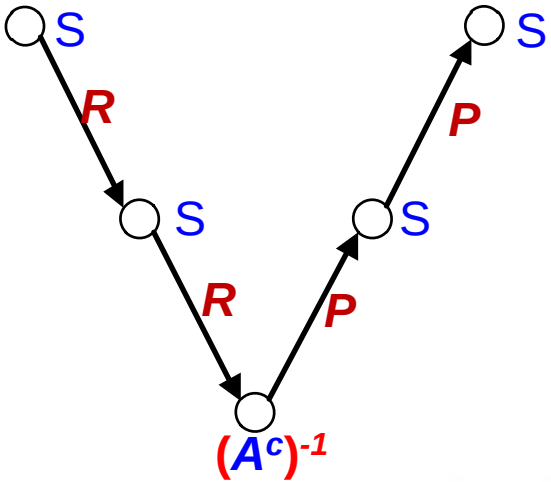
粗网格校正

光滑子

$M^{-1} = P\,(A^c)^{-1}R \cdot S$

插值和限制算子





GMG -> AMG

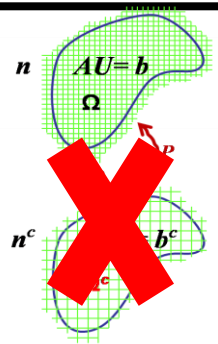
Ae = r : Divide and Conquer Rule

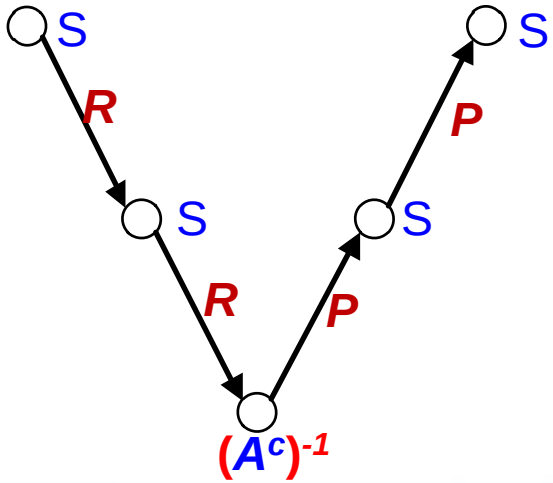
e = Low frequency + High frequency

粗网格校正 光滑子

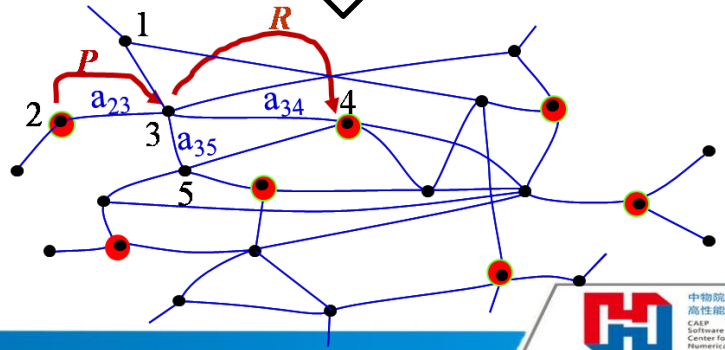
$M^{-1} = P\,(A^c)^{-1}R \cdot S$

插值和限制算子





$$A = \left(a_{ij}\right)_{n\times n}$$



GMG -> AMG

Ae = r : Divide and Conquer Rule

e = Low frequency + High frequency

粗网格校正 $M^{-1} = P\,(A^c)^{-1}R$ 光滑子 $\cdot\, S$

插值和限制算子

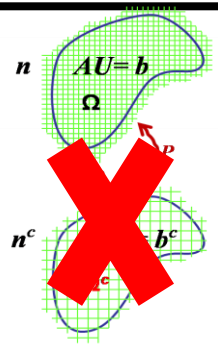
C-AMG

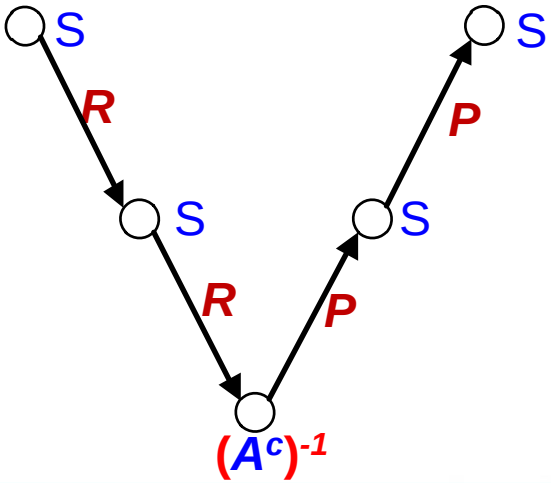
Brandt, Ruge, McCormick,1982

Setup Phase：粗化+粗网格矩阵

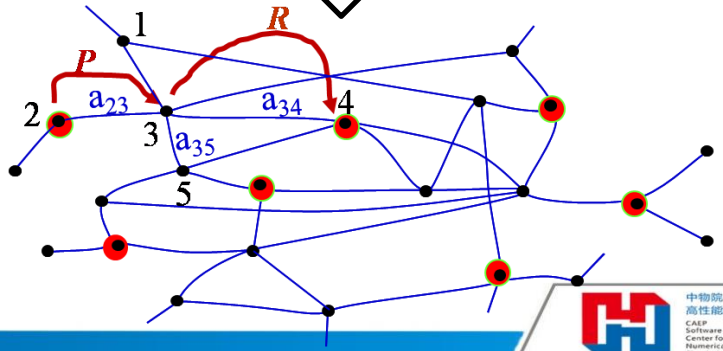
- $\Omega = \Omega^c \cup \Omega^f$ (C/F Splitting)
- $P: \Omega^c \rightarrow \Omega$ (Interpolation)
- $A^c = RAP$ (Galerkin method)

  $(R = P^T)$





$$A = \left(a_{ij}\right)_{n\times n}$$



GMG -> AMG

Ae = r : Divide and Conquer Rule

e = Low frequency + High frequency

粗网格校正 $M^{-1} = P\,(A^c)^{-1}R$ · S 光滑子

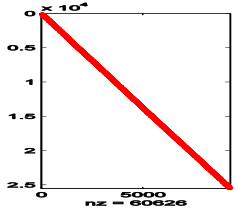
插值和限制算子

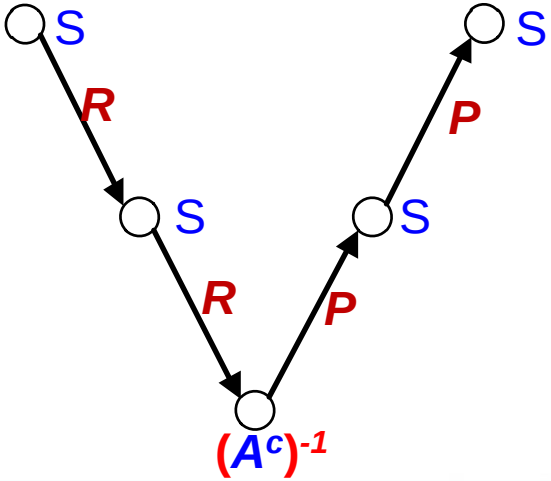
C-AMG

Brandt, Ruge, McCormick,1982

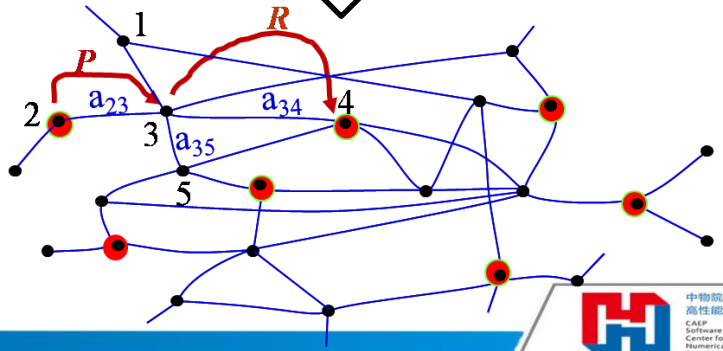
Setup Phase：粗化+粗网格矩阵

- $\Omega = \Omega^c \cup \Omega^f$ (C/F Splitting)
- $P: \Omega^c \rightarrow \Omega$ (Interpolation)
- $A^c = RAP$ (Galerkin method) ($R = P^T$)





$$A = \left(a_{ij}\right)_{n\times n}$$

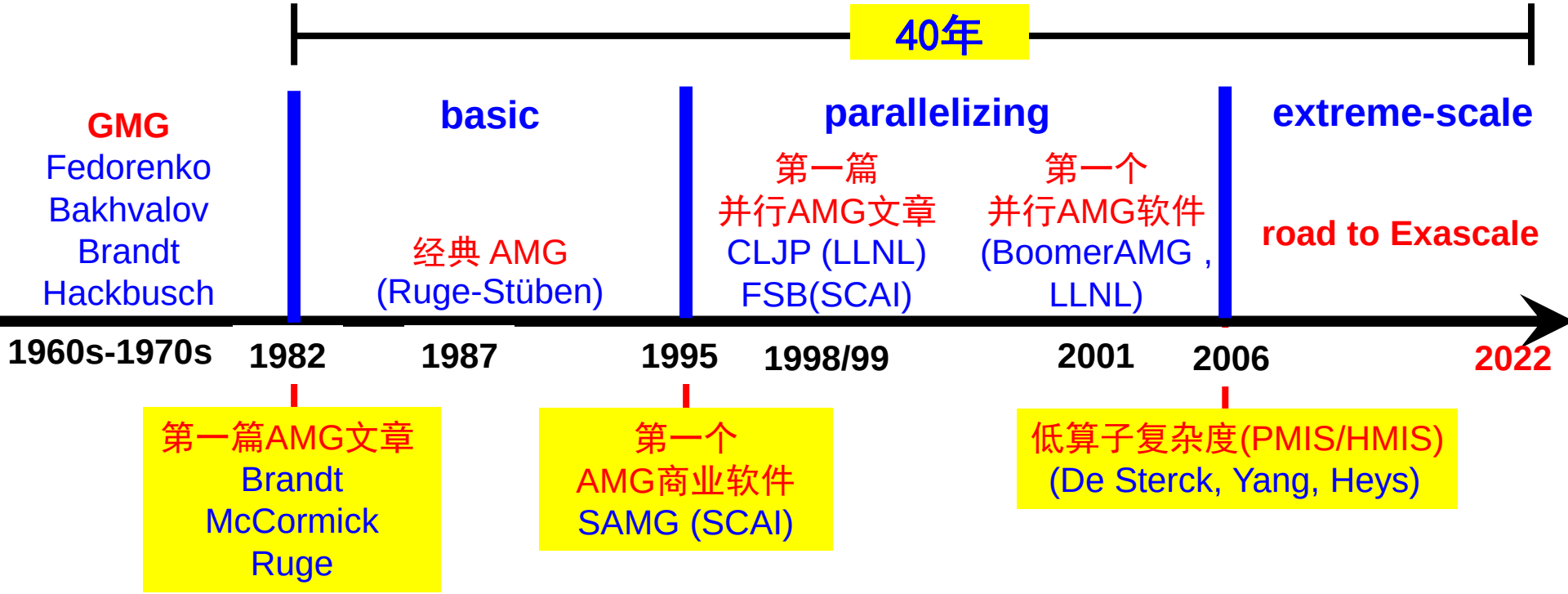


Algebraic Multigrid (AMG) for $Au=b$

- **Algorithm variants**
  - Classical AMG (C-AMG), 1980s
  - SA-AMG / UA-AMG, 1990s
  - AMGe / ρAMGe, αAMG / αSA-AMG, 2000s
  - Bootstrap AMG / Lean AMG / AI-AMG, 2010s
  - Parallel Coarsening: RS-type, CLJP, FSB, Relaxed-type, PMIS, …..
- **Applications**
  - radiation-hydrodynamics, oil/ground reservoir, structural mechanic, electro-magnetic environment, semi-conductor device, automotive design, circuit design, QCD, etc.
  - elliptic/parabolic, Navier-Stokes, helmholtz, Maxwell, elasticity, drift-diffusion, eigenvalue, non-PDE system (Markov chain, image/graph analysis, etc)
- **Roles**
  - Preconditioner for the problems close to M-matrices. (Combined with Krylov)
  - Component for complicated problems (e.g. HX/AMS for Maxwell, PCTL for RHD)

Algebraic Multigrid (AMG) for $Au=b$

- 40年:三个主要阶段，应用驱动的典范。



Algebraic Multigrid (AMG) for $Au=b$

- **相关文献推荐**

AMG 综述文章:

- Ruge & Stüben (1987), *Algebraic multigrid*, in Multigrid, McCormick, SIAM, 73-130.
- Stüben (2001), *An introduction to AMG*, in Multigrid, Trottenberg, Academic Press, 413-532.
- Falgout (2006), *An introduction to algebraic multigrid*, CiSE, 8: 24-33.
- Xu & Zikatanov(2017), *Algebraic multigrid methods*, Acta Numerica, 591-721.
- 徐小文(2019)，*并行代数多重网格算法：大规模计算应用现状与挑战*，数值计算与计算机应用, 青年评述，40(4): 243-260.

多重网格入门文献:

- W.Briggs, V.Henson, S.McCormick, *A multigrid tutorial,* 2nd Edition, SIAM, 2000.
- U.Trottenberg, C.Oosterlee, A.Schuller, *Multigrid*, Academic Press, 2001.
- 谷同祥，徐小文等，*迭代方法和预处理技术(下册) 第2章“多层预处理子”*，科学出版社，2015.

AMG的效率问题

AMG efficiency issues

Ae = r : Divide and Conquer Rule

e = Low frequency + High frequency

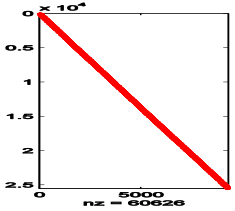
粗网格校正 ↑ $M^{-1} = P\,(A^c)^{-1}R \cdot S$ ↑ 光滑子

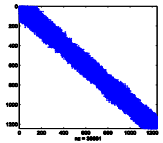
插值和限制算子

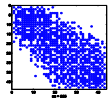
C-AMG

Setup Phase：粗化+粗网格矩阵

- $\Omega = \Omega^c \cup \Omega^f$ (C/F Splitting)
- $P: \Omega^c \rightarrow \Omega$ (Interpolation)
- $A^c = RAP$ (Galerkin method)
  ($R = P^T$)







AMG efficiency issues

Ae = r : Divide and Conquer Rule

e = Low frequency + High frequency

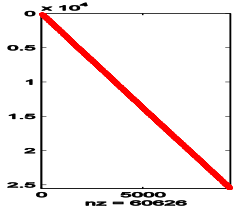
粗网格校正 ↑ $M^{-1} = P\,(A^c)^{-1}R \cdot S$ ↑ 光滑子

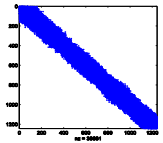
插值和限制算子

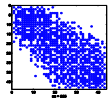
C-AMG

Setup Phase：粗化+粗网格矩阵

- $\Omega = \Omega^c \cup \Omega^f$ (C/F Splitting)
- $P: \Omega^c \rightarrow \Omega$ (Interpolation)
- $A^c = RAP$ (Galerkin method)
  ($R = P^T$)







关键：收敛速度(*nits*) 与 算子复杂度(C_A) 之间权衡。

$$C_A = \frac{\sum_l |A^l|}{|A^0|}$$

AMG efficiency issues

Ae = r : Divide and Conquer Rule

e = Low frequency + High frequency

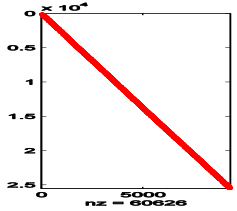
粗网格校正 ↑ $M^{-1} = P\,(A^c)^{-1}R$ ↑ 光滑子 $\cdot\ S$

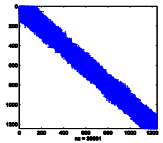
插值和限制算子

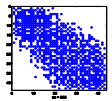
C-AMG

Setup Phase：粗化+粗网格矩阵

- $\Omega = \Omega^c \cup \Omega^f$ (C/F Splitting)
- $P: \Omega^c \rightarrow \Omega$ (Interpolation)
- $A^c = RAP$ (Galerkin method) ($R = P^T$)







关键：收敛速度(*nits*) 与 算子复杂度(C_A) 之间权衡。

$$C_A = \frac{\sum_l |A^l|}{|A^0|}$$

约束：低算子复杂度是大规模计算的必要条件（尤其是三维问题）。

- De Sterck, Yang, Heys, Reducing complexity in parallel AMG preconditioners, SIMAA, 2006.

AMG efficiency issues

- 问题1: 低算子复杂度导致算法收敛速度退化。

$-\nabla\cdot(\kappa\nabla u)=f$ **Poisson-like equ., AMG-GMRES(20), PMIS coarsening, *tol* = 10^{-8}**

mesh_size	#cores	*k*				
		constant	*anisotropic*	*one-jump*	*three-jump*	*random(1-10^3)*
64^3	1	19	20	22	25	21
1024^3	4096	91	50	416	515	553

AMG efficiency issues

- 问题1：低算子复杂度导致算法收敛速度退化。

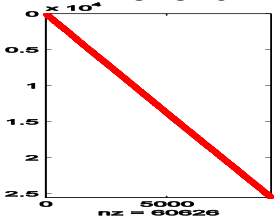
$-\nabla\cdot(\kappa\nabla u) = f$ **Poisson-like equ., AMG-GMRES(20), PMIS coarsening, *tol* = 10^{-8}**

mesh_size	#cores	*k*				
		constant	*anisotropic*	*one-jump*	*three-jump*	*random(1-10³)*
64^3	1	19	20	22	25	21
1024^3	4096	91	50	416	515	553

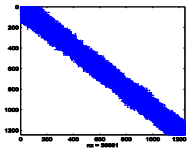
- 问题2：粗网格层复杂计算/通信模式导致浮点效率低和性能波动大。

$A^c = RAP$

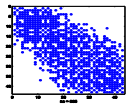
Level 0



Level 2



Level 3



粗网格层：
- 访存/通信模式复杂；
- 通信计算比高；

AMG efficiency issues

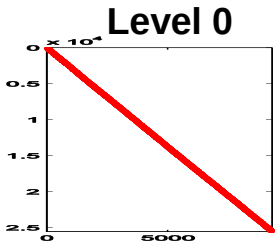
- 问题1：低算子复杂度导致算法收敛速度退化。

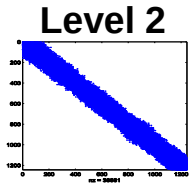
$-\nabla\cdot(\kappa\nabla u)=f$ **Poisson-like equ.**, **AMG-GMRES(20), PMIS coarsening, *tol* = 10^{-8}**

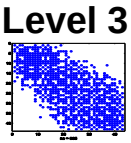
mesh_size	#cores	*k*				
		constant	*anisotropic*	*one-jump*	*three-jump*	*random(1-10³)*
64^3	1	19	20	22	25	21
1024^3	4096	91	50	416	515	553

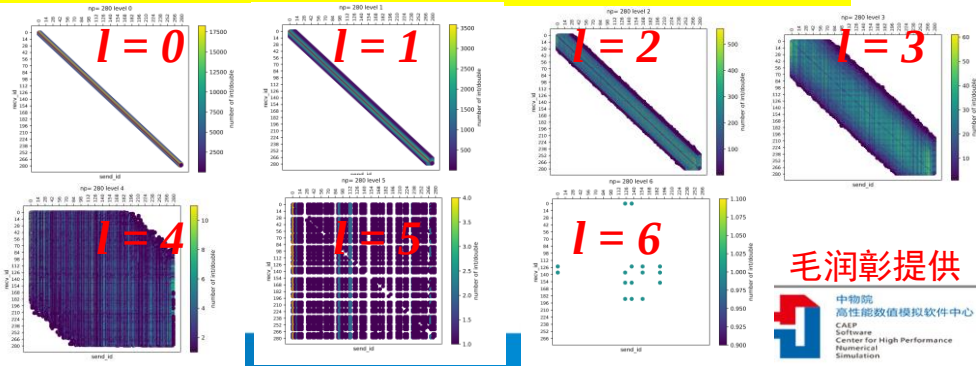
- 问题2：粗网格层复杂计算/通信模式导致浮点效率低和性能波动大。

$A^c = RAP$







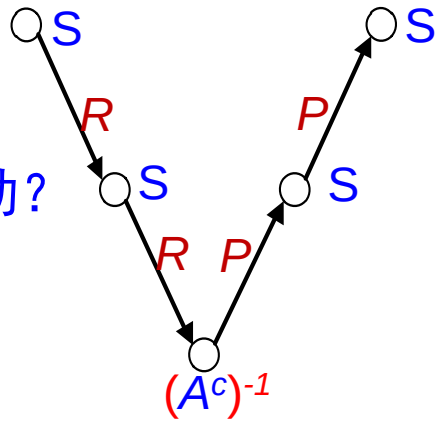


毛润彰提供

AMG efficiency issues

4个挑战

1. 复杂问题(3D)：如何同时确保低算子复杂度和快速收敛？
2. 大规模并行：如何避免/降低粗网格层并行效率的损失？
3. 基本组件(***RAP, SpMV***)：如何提升结点内浮点效率 (多核/异构)？
   - ① ***Setup(l =1:L) { RAP }***
   - ② ***Cycle(l =1:L) { SpRr, SpAx, SpPe }***
4. 实际应用场景：如何避免/降低AMG导致的机器性能波动？



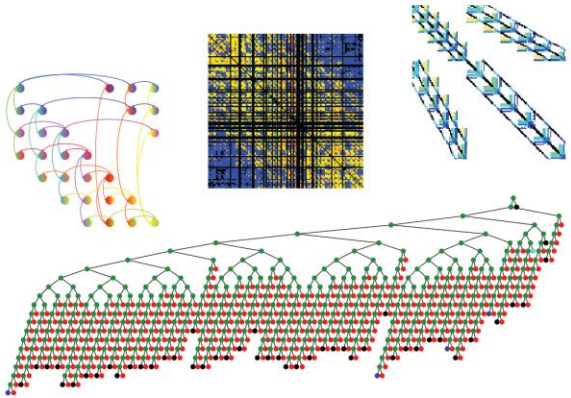
徐小文(2019)，并行代数多重网格算法：大规模计算应用现状与挑战，
数值计算与计算机应用，青年评述，40(4): 243-260.

总结与展望

总结与展望

Report on the Workshop on Extreme-Scale Solvers: Transition to Future Architectures

March 8-9, 2012, Washington, D.C.



U.S. Department of Energy
Office of Advanced Scientific Computing Research

U.S. DEPARTMENT OF ENERGY

一个例子：Charon程序，AMG+GMRES

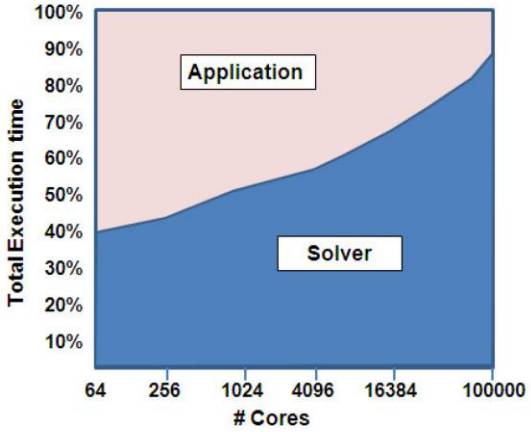
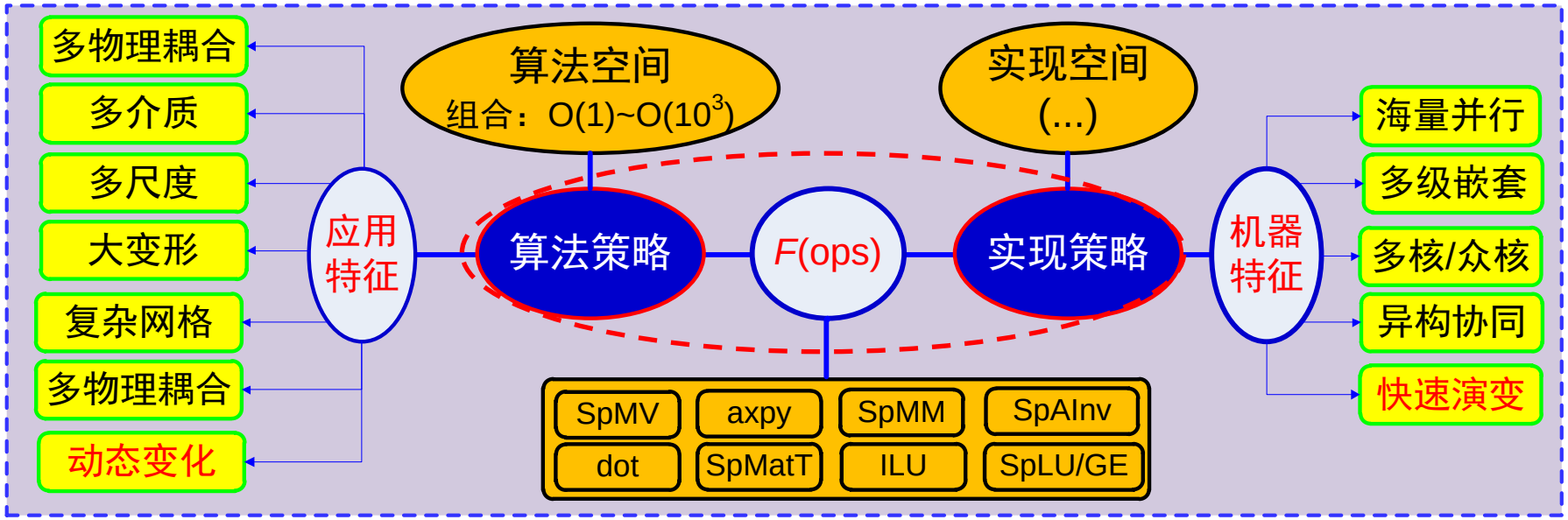


Figure 1: Comparison of time spent in the multiphysics application Charon vs. time spent in the linear solver (algebraic multigrid with GMRES), on 64 to 100,000 cores of a Cray XE6 (Cielo) using weak scaling. As core count and problem size increase, the fraction of time spent in the solver also increases. Although we use a state-of-the-art multilevel preconditioner, this trend is typical for many applications. The problem setup time is nearly constant from 64 to 100,000 cores, but solver time steadily increases. (Data courtesy of Paul Lin)

今天，这个现象依然非常普遍！

中物院 高性能数值模拟软件中心 CAEP Software Center for High Performance Numerical Simulation

展望：如何匹配应用特征与机器特征



$$\frac{\text{Dof}}{\text{s}} = \frac{\text{Dof}}{\text{Flop}} \times \frac{\text{Flop}}{\text{s}}$$

实际应用场景下，如何确定最优算法策略与实现策略？

算法的差距：与Poisson方程的距离 (HPGMG,O(n))

激光聚变($n=10^9$)，RP =0.01-0.02

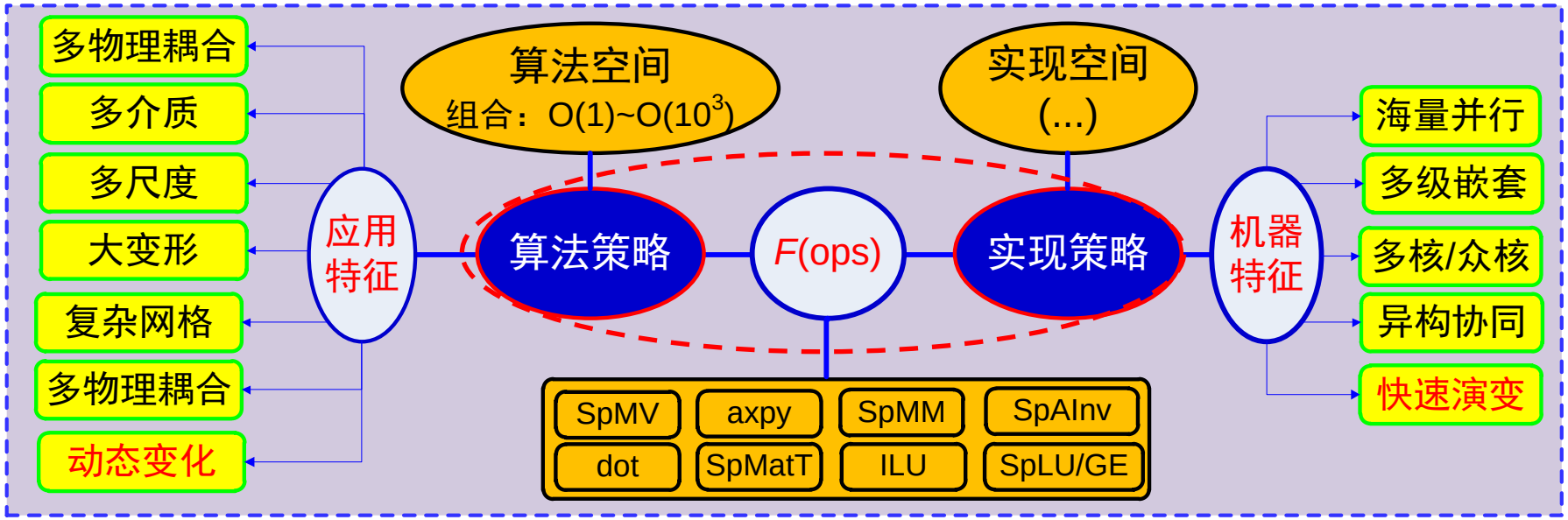
结构力学($n=10^9$)，RP =0.02-0.03

电子学系统($n=10^8$)，RP =0.009

实际模拟，相差2个量级以上。

浮点效率的差距？(SpBLAS, HPCG)

展望：如何匹配应用特征与机器特征



$$\frac{\text{Dof}}{\text{s}} = \frac{\text{Dof}}{\text{Flop}} \times \frac{\text{Flop}}{\text{s}}$$

实际应用场景下，如何确定最优算法策略与实现策略？

智能型算法框架：实现算法空间到特征空间的映射。

- 自动调优机制：能够在模拟过程实现算法、组件和参数的自动调整，基于特征建模与分析。
- 人工智能/机器学习：基于学习模型进行预测。

北京应用物理与计算数学研究所
中物院高性能数值模拟软件中心

谢谢！

http://www.iapcm.ac.cn
http://www.caep-scns.ac.cn