# Towards the Next-Generation CFD:

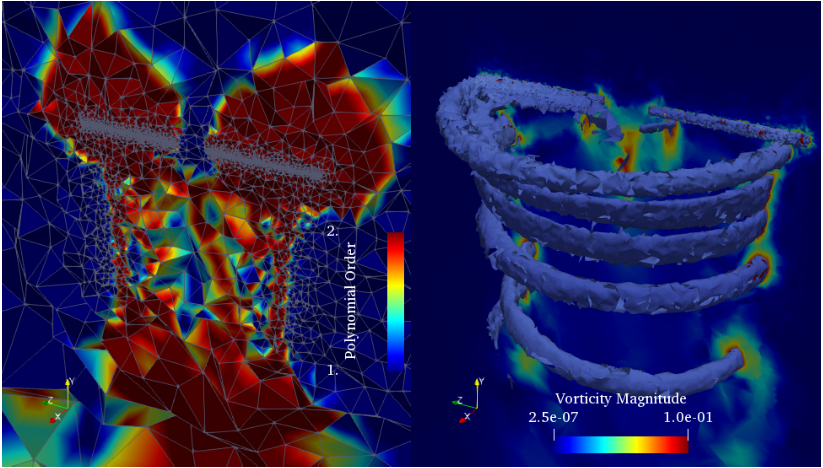
## High-order, fully *HP*-Adaptive Methods for 3-D Fluid Dynamics

Shu-Jie Li （李书杰）

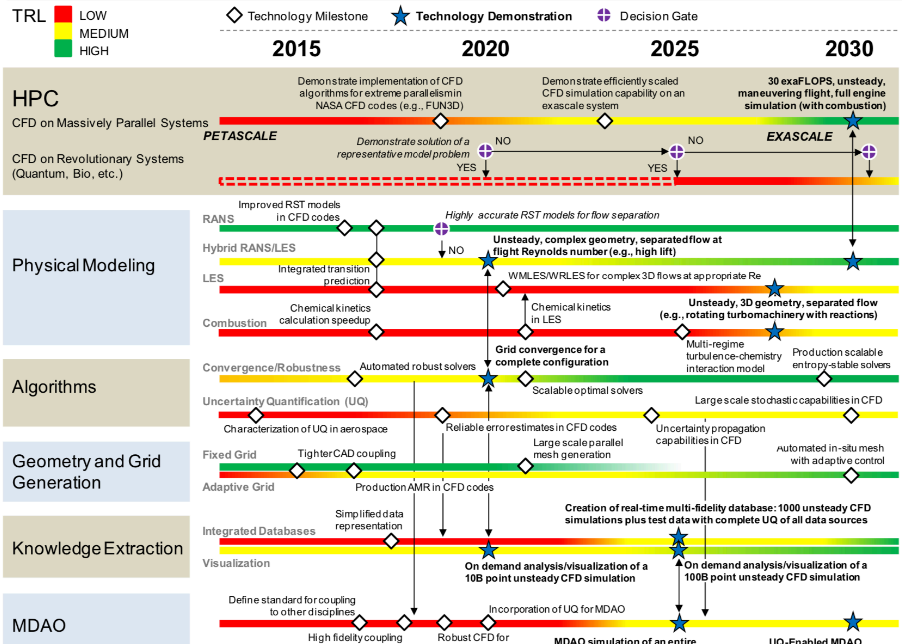
Mechanics Division, CSRC

# 基本情况

研究领域：计算流体力学（理论，算法，实现），气体动力学，超级计算



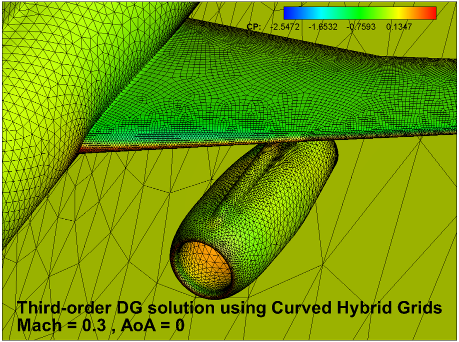
# 计算流体力学远景规划图（NASA CFD 2030）

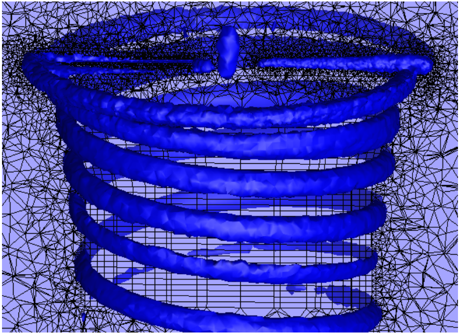


长期致力于研究高精度 CFD 计算所遇到的计算科学问题:

精度，效率，可用行，挑战性问题

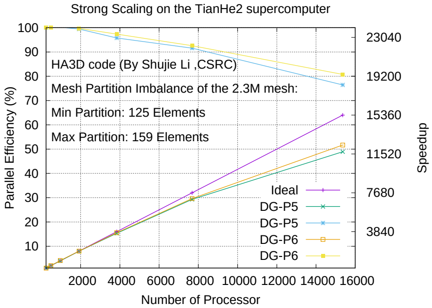
## Towards **CFD 2030** with **HA3D**:
## a High-order HP-adaptive Discontinuous Galerkin code
## ( by Shu-Jie Li， 2012-2019 )
## (H= 网格, P= 精度, HP 自适应计算 +HP 多重网格)



# HA3D: Parallel scalability

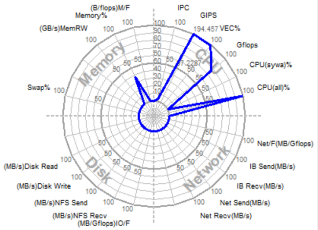
## Strong scaling :




## Weak scaling :

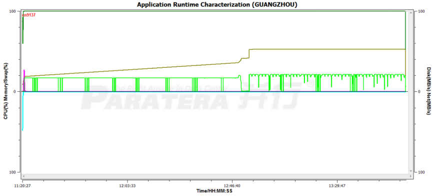
| 算例 | 计算规模 | 最小分区(单元数) | 最大分区(单元数) | 进程数 | 运行时间(秒) | 时间加速比(基于24核) | 效率(加速比/进程数) |
|---|---|---|---|---|---|---|---|
| nacas2 | 79872*8 | 3062 | 3470 | 24 | 428s | 1x | 100% |
| nacas3 | 79872*64 | 2882 | 3460 | 24*8 | 439s | 7.80x | 97.5% |
| nacas4 | 79872*512 | 2186 | 3492 | 24*64 | 436s | 62.83x | 98.2% |

## Vectorization :




## Memory usage :

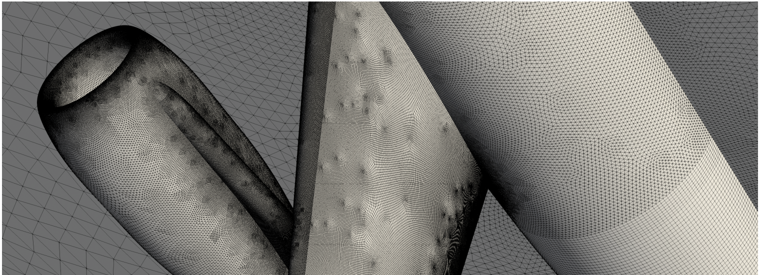
**1.4 整机运行详细信息：**



## **HA3D 的网格生成功能**：Shu-Jie Li, *Mesh curving and refinement based on cubic Bézier surface for high-order discontinuous Galerkin methods*, **Computational Mathematics and Mathematical Physics** ( 2019 in press, a RAS (俄罗斯科学院) journal ).

(部分发表在 Springer Book《Numerical Geometry, Grid Generation and Scientific Computing》第 15 章，纪念 Voronoi 诞辰 150 周年，莫斯科，2018 年 12 月)

论文中的 **9 亿**，**混合**四面体，六面体，金字塔的曲面网格生成



# 算法到应用存在巨大 GAP：例如美国 2030 年计划实现发动机整机数值模拟（带燃烧）


Stanford University

Center for Integrated Turbulence Simulations

About CITS | Research Projects | Simulation Gallery | Publications | Software

**Research Projects**
- RANS
- LES
- Integration
- Merrimac

CITS Home

**Research Projects**

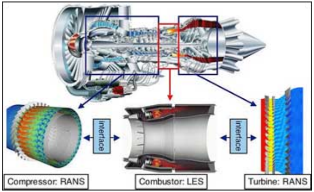
Simulation Framework | Software Engineering | Physics Modeling | Verification & Validation

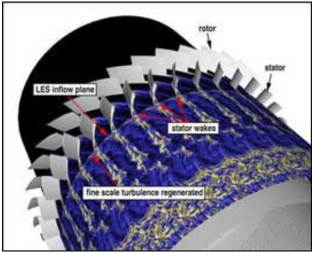
**Integrated Multicode Simulation Framework**

*CHIMPS, Coupler for High-performance Integrated Multi-Physics Simulations*

CITS overarching problem is the simulation of the complete aero-thermodynamic flow path through a jet engine. CITS simulation environment is based on Python and built around the Center's flagship codes, CDP (Large Eddy Simulation approach) and TFLO (based on Reynolds-averaged Navier-Stokes equations). It also includes a set of general-purpose interpolation and communication libraries (CHIMPS, Coupler for High-performance Integrated Multi-Physics Simulations).

Basic concept: a parallel, scalable, distributed module that automatically takes care of mesh / solution interpolations to exchange information between CFD solvers, Structral dynamics codes, etc. The code-interfacing is designed around a distributed "interpolation server". Participating codes "register" the required data and their locations and "provide" their local mesh and solutions to the other components in the integrated environment. Information exchange (volume/overlap, surface/minimum distance and failsafe) happens transparently to the user through straightforward API.



# High-order methods for 3-D Navier-Stokes?

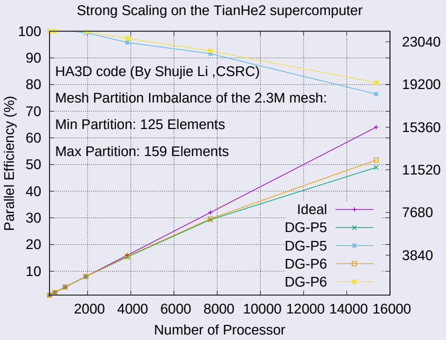
High order ( spatial ) methods are getting popular for the CFD community such as the Discontinuous Galerkin (DG), Flux Reconstrcution (FR) ...

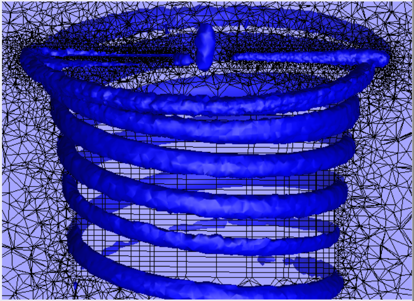
1. **PROCS:** spatially high accuracy
   - rich flow details
   - suitable for turbulence and aeroacoustic simulations (DNS, LES)
   - resolve boundary layer and vortex transportation
2. **CONS:** time marching could slow down the whole performance

**Total efficiency = Spatial discretization + TIME MARCHING**
计算速度，内存，鲁棒性，复杂度

# High-order solvers are expensive

**Towards arbitrarily high-order, 3-D turbulence simulations**
**HA3D**: highly parallel, high-order Discontinuous Galerkin code
( Author: Shu-Jie Li, 2012–present )



# Shortcomings of traditional time discretizations

**HA3D** has the following traditional options

- TVD-Runge Kutta, Jameson Runge Kutta (2-4th order)
- Point Jacobi
- Lower-Upper Symmetric Gauss Seidal (LU-SGS) and SGS
- Backword Differential Formula (BDF2)
- V-cycle *p*-multigrid (up to 11th order)
- Fully implicit restarted ILU-GMRES with exact Jacobians

# Why high-order methods so slow?

## An inefficient time marching leads to massive time steps

- Severe time step restriction for viscous flows, e.g., Discontinuous Galerkin : $\Delta t_{viscous} = CFL \frac{\Delta x^2}{(p+1)^2 \lambda_{viscous}}$
- The term $(p+1)^2$ makes the time step even worse
- $\Delta t_{p=9}$ is at least 100X smaller than $\Delta t_{p=0}$ and suffering a dramatically drop of CFL number
- At least, we want to get rid of the term $(p+1)^2$ to make DG more efficient

# Outside Explicit and Implicit, the Exponential Schemes. Predictor-Corrector EXPonential time integrator (PCEXP)

1. 任意时间步长，低绝对误差，时间准确，收敛快


- **S.J. Li**, et.al, *Adaptive Exponential Time Integration of the Navier-Stokes Equations.*. (***AIAA***-2020 submitted)
- **S.J. Li**, Efficient p-Multigrid Method based on Exponential Time Discretization for Compressible Steady Flows. arXiv:1807.01151
- **S.J. Li**, et.al, An exponential time-integrator scheme for steady and unsteady inviscid flows. ***Journal of Computational Physics*** 365 (2018) 206–225
- **S.J. Li**, et.al, *Exponential Time-Marching method for the Unsteady Navier-Stokes Equations.*. (***AIAA***-2019-0907)
- **S.J. Li**, et.al, *Fast Time Integration of* **Navier-Stokes** *Equations with an Exponential-Integrator Scheme*. (***AIAA***-2018-0369)
- **S.J. Li**, et.al, *Explicit large time stepping with a second-order exponential time integrator scheme for unsteady and steady flows* , (***AIAA***-2017-0753)

# Basic ideal of Exponential schemes

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{R}(\mathbf{u}) \tag{1}$$

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{J}_n\mathbf{u} + \underbrace{\mathbf{R}(\mathbf{u}) - \mathbf{J}_\mathbf{n}\mathbf{u}}_{\mathbf{N}(\mathbf{u})}, \qquad \mathbf{J}_\mathbf{n} = \frac{\partial \mathbf{R}(\mathbf{u})}{\partial \mathbf{u}} \tag{2}$$

$$\mathbf{u}_{n+1} = e^{\Delta t \mathbf{J}_n}\,\mathbf{u}_n + e^{\Delta t \mathbf{J}_n}\int_0^{\Delta t} e^{-\tau \mathbf{J}_n}\;\underbrace{\mathbf{N}(\mathbf{u}(\mathbf{x}, t_n + \tau))}_{\text{numerical approximation}}\; d\tau \tag{3}$$

# Basic ideal of Exponential schemes

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{R}(\mathbf{u}) \tag{4}$$

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{J}_n\mathbf{u} + \underbrace{\mathbf{R}(\mathbf{u}) - \mathbf{J}_\mathbf{n}\mathbf{u}}_{\mathbf{N}(\mathbf{u})}, \qquad \mathbf{J}_\mathbf{n} = \frac{\partial \mathbf{R}(\mathbf{u})}{\partial \mathbf{u}} \tag{5}$$

$$\mathbf{u}_{n+1} = e^{\Delta t \mathbf{J}_n}\mathbf{u}_n \text{ , } \textbf{EXACT} \text{ for linear equations} \tag{6}$$

# The PCEXP scheme
S.-J. Li et al. AIAA-2017-0753, JCP-365(2018) 206-225

$$\mathbf{u}^* = \mathbf{u}_n + \Delta t \, \Phi_1(\Delta t \mathbf{J}_n) \mathbf{R}(\mathbf{u}_n) \qquad (7)$$

$$\mathbf{u}_{n+1} = \mathbf{u}^* + \frac{1}{2} \Delta t \, \Phi_1(\Delta t \mathbf{J}_n) \left[ (\mathbf{N}(\mathbf{u}^*) - \mathbf{N}(\mathbf{u}_n) \right] \qquad (8)$$

where $\Phi_1$ is a matrix function defined as

$$\Phi_1(\Delta t \mathbf{J}) = \frac{1}{\Delta t} \int_0^{\Delta t} e^{(\Delta t - \tau)\mathbf{J}} \, d\tau \qquad (9)$$

$$\stackrel{??}{=} \frac{\mathbf{J}^{-1}}{\Delta t} \left[ \exp\left( \Delta t \mathbf{J} \right) - \mathbf{I} \right] \qquad (10)$$

$$\text{Matrix Exponential : } \exp(\Delta t \mathbf{J}) := \sum_{m=0}^{\infty} \frac{(\Delta t \mathbf{J})^m}{m!} \qquad (11)$$

## How to compute $\Phi_1$ times the vector such as $\mathbf{R}$

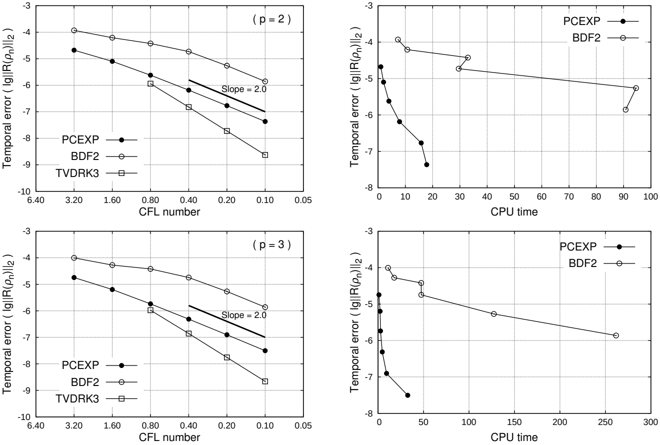
$$\begin{aligned} \mathbf{u}^* &= \mathbf{u}_n + \Delta t\,\Phi_1(\Delta t\mathbf{J}_n)\mathbf{R}(\mathbf{u}_n) & (12)\\ \mathbf{u}_{n+1} &= \mathbf{u}^* + \frac{1}{2}\Delta t\,\Phi_1(\Delta t\mathbf{J}_n)\left[(\mathbf{N}(\mathbf{u}^*) - \mathbf{N}(\mathbf{u}_n)\right] & (13)\end{aligned}$$

### Krylov subspace projection

$$\Phi_1(\Delta t\mathbf{J})\mathbf{R} = \mathbf{J}^{-1}\frac{e^{\Delta t\mathbf{J}} - \mathbf{I}}{\Delta t}\mathbf{R} = \left(\mathbf{I} + \frac{(\Delta t\mathbf{J})}{2!} + \frac{(\Delta t\mathbf{J})^2}{3!} + \frac{(\Delta t\mathbf{J})^3}{4!} + \cdots\right)\mathbf{R},$$

$$K_m(\mathbf{J},\mathbf{R}) = span\{\mathbf{R}, \mathbf{J}\mathbf{R}, \mathbf{J}^2\mathbf{R}, \cdots, \mathbf{J}^{m-1}\mathbf{R}\}.$$

# PCEXP for unsteady flows: accuracy and cost




(a) Convergence order (b) CPU time

图: Vortex transportation on a $24 \times 24$ uniform mesh with $\mathrm{CFL} = 0.1 \times 2^n$, $0 \leq n \leq 5$. Left: Temporal convergence of PCEXP, BDF2, and TVDRK3 schemes. Right: Evolution of the errors versus CPU time for PCEXP and BDF2 schemes. From top to bottom: $0 \leq p \leq 3$.

# Comparison: TVDRK3 and BDF2 v.s. PCEXP

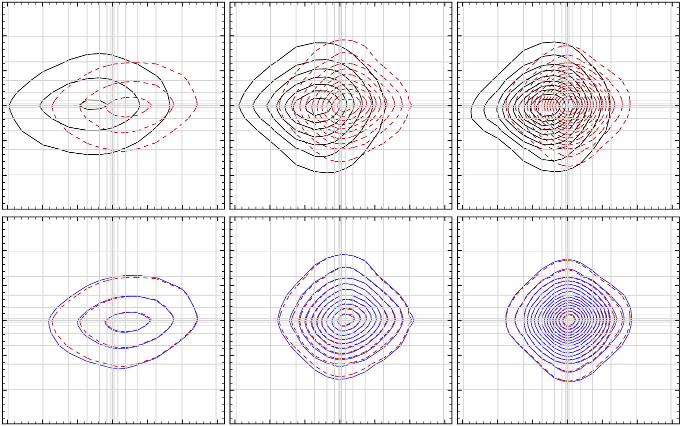


图: Vortex transport on a $24 \times 24$ stretched mesh. Density Contours of 15 iso-lines in the range of [0.99691, 0.99974]. Top: BDF2 (black solid lines) *versus* TVDRK3 (red dashed lines). Bottom: PCEXP (blue solid lines) *versus* TVDRK3 (red dashed lines). From the left to the right: $p = 1$, 2, and 3.

# Unsteady flows: speedup gain

表: Vortex transportation on $24 \times 24$ stretched mesh in one period. The $t_\mathrm{P}$, $t_\mathrm{B}$, and $t_\mathrm{T}$ are the CPU times (in minutes) corresponding to the PCEXP, BDF2, and TVDRK3 schemes, respectively.

| | PCEXP | | | BDF2 | | | TVDRK3 | | |
|---|---|---|---|---|---|---|---|---|---|
| $p$ order | Steps | CFL | $t_\mathrm{P}$ | Steps | CFL | $t_\mathrm{B}/t_\mathrm{P}$ | Steps | CFL | $t_\mathrm{T}/t_\mathrm{P}$ |
| $p=0$ | 12 | 1000.0 | 0.03 | 12 | 1000.0 | 3.90 | 9696 | 1.2 | 48.97 |
| $p=1$ | 35 | 1000.0 | 1.10 | 35 | 1000.0 | 1.13 | 29106 | 1.2 | 10.70 |
| $p=2$ | 52 | 1000.0 | 12.10 | 52 | 1000.0 | 1.00 | 48507 | 1.2 | 4.98 |
| $p=3$ | 84 | 1000.0 | 76.50 | 84 | 1000.0 | 1.60 | 67893 | 1.2 | 3.80 |

# Unsteady viscous solution with PCEXP v.s. Analytical solution ( rotating concentric cylinder flow )

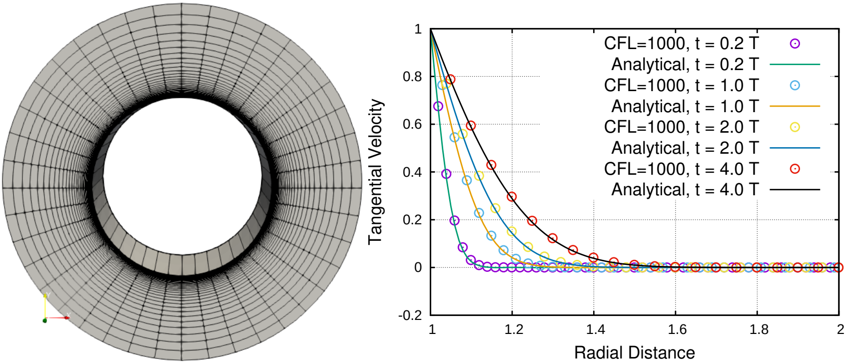


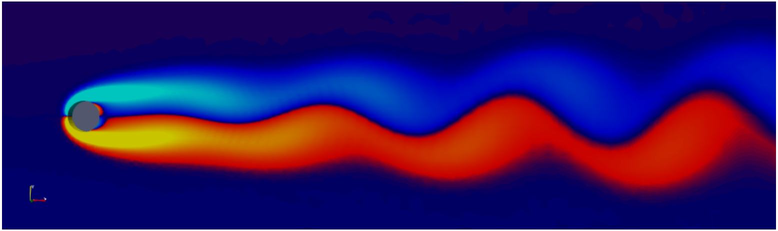

图: Quasi-2D large-aspect-ratio mesh and velocity profiles comparsion ($40 \times 40 \times 1 = 1600$ cells )

# Unsteady viscous solution with PCEXP v.s. Analytical solution ( rotating concentric cylinder flow )

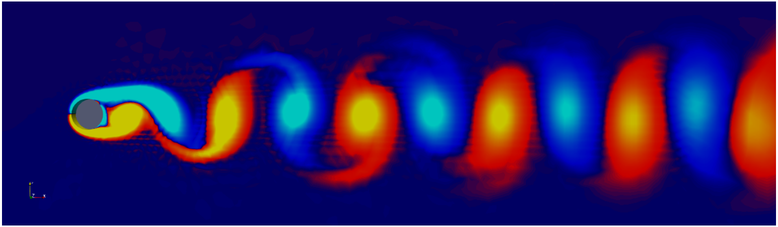
表: Results statistics of the PCEXP scheme for the rotating concentric cylinder flow.

| Time ( $T = r_0^2/\nu$) | 0.2 $T$ | 1.0 $T$ | 2.0 $T$ | 4.0 $T$ |
|---|---|---|---|---|
| #Time steps | 16 | 79 | 158 | 315 |
| CFL | $10^3$ | $10^3$ | $10^3$ | $10^3$ |
| $L_2$ Error (%) | 0.28 | 0.24 | 0.26 | 0.29 |

# Unsteady laminar flows past a circular cylinder



(a) Re $= 60$



(b) Re $= 180$

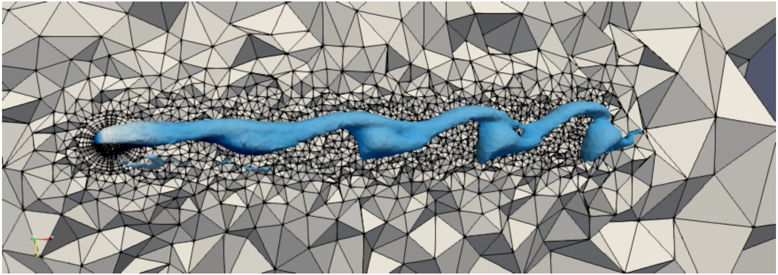
图: Vorticity field for the laminar flow past a circular cylinder with CFL $= 500$.

# Unsteady laminar flows past a circular cylinder

表: Results statistics for the laminar flow past a circular cylinder.

| Re | 60 | 80 | 100 | 120 | 140 | 160 | 180 |
|---|---|---|---|---|---|---|---|
| Present | 0.139 | 0.155 | 0.168 | 0.178 | 0.185 | 0.191 | 0.197 |
| Experiments | 0.135 | 0.152 | 0.164 | 0.173 | 0.181 | 0.186 | N/A |
| Karniadakis | 0.153 | 0.168 | 0.178 | 0.185 | 0.192 | 0.197 | 0.203 |

# Flow past a sphere at $\mathrm{Ma} = 0.3$, $\mathrm{Re} = 300$



(a) Q contour colored by the velocity magnitude

(b) Velocity magnitude contour on the cut slice at $z = 0$

# Flow past a sphere at Ma = 0.3, Re = 300

| Method | #Cell | $S_{order}$ | $T_{order}$ | $C_d$ | $\Delta C_d$ | $St$ |
|---|---|---|---|---|---|---|
| Present | 80093 | $3rd$ | $2nd$ | 0.674 | 0.0033 | 0.133 |
| Gassner | 160000 | $4th$ | $4th$ | 0.673 | 0.0031 | 0.131 |
| Johnson | 428442 | $2nd$ | $2nd$ | 0.656 | 0.0035 | 0.137 |
| Haga | 54312 | $4th$ | $3rd$ | 0.670 | 0.0032 | 0.131 |

表: Results statistics of the PCEXP scheme for the flow past a sphere at Re = 300.

# Turbulent flow past a square cylinder at Ma $= 0.15$, Re $= 22000$

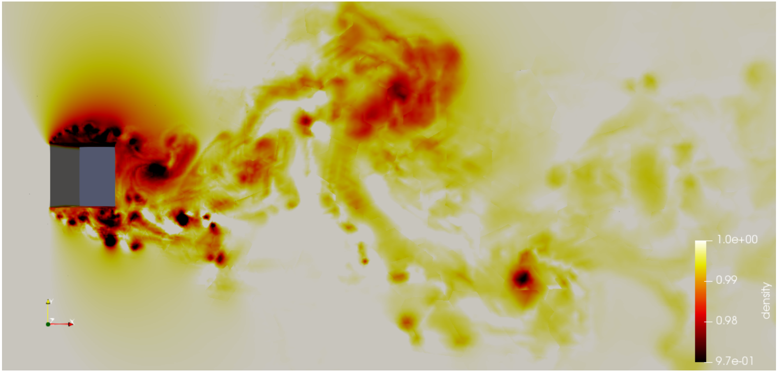


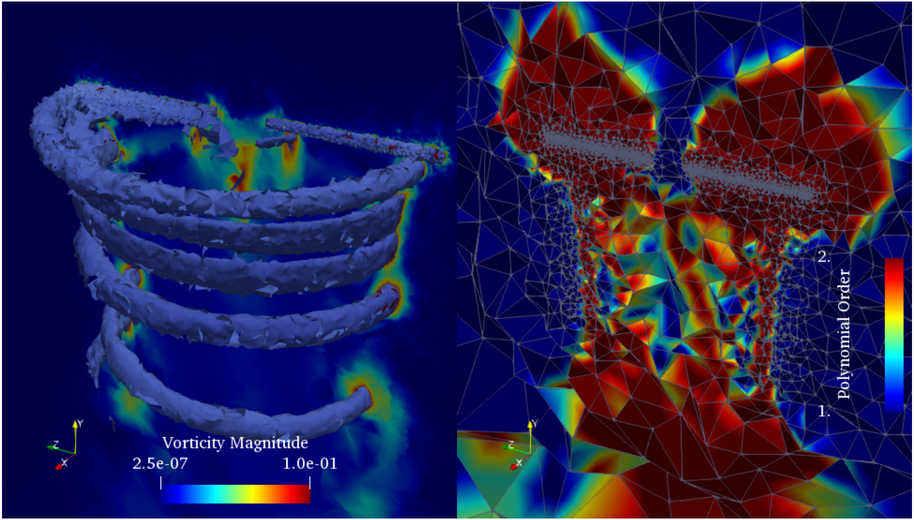

图: Turbulent flow field for the flow past a square cylinder: Ma $= 0.15$, Re $= 22000$, DG $p = 4$.

## 多体运动计算：动态自动网格剖分（h remeshing）

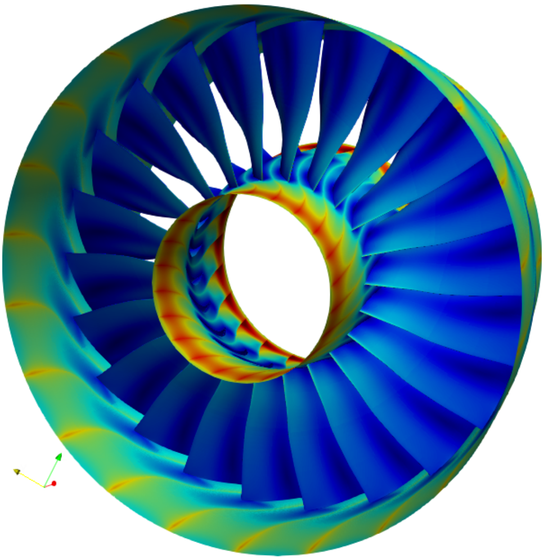
三维网格**实时劈分，边交换，拓扑优化，自适应全结合**

初步实现：弹体发射

# p 高阶自适应算法应用在螺旋桨：自动捕捉需要高精度计算的区域（AIAA paper 2020, submitted）

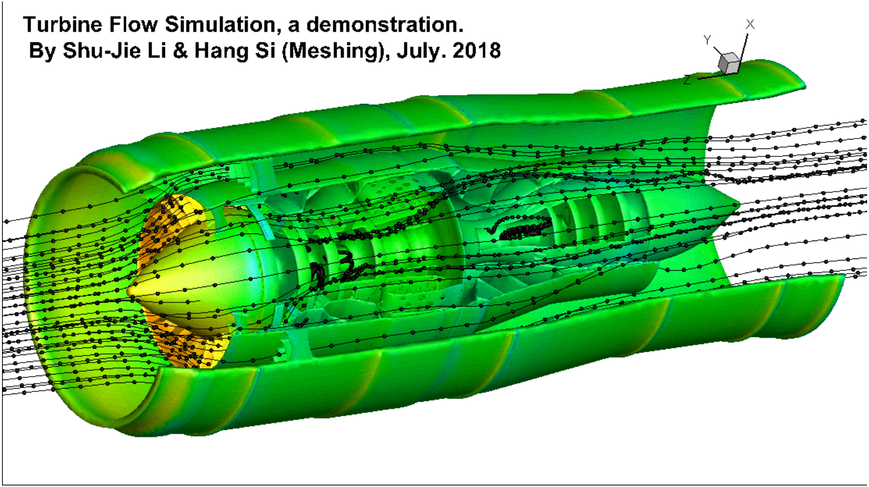


# 单级叶片，周期边界的高阶快速计算 (by Shu-Jie Li, 08-16-2019) （指数时间 + 间断有限元）



# 中物院与中国航空发动机集团全面战略合作子方向

复杂几何边界上的湍流大涡模拟



## Summary

1. 在世界范围内首次实现了 hp 自适应指数时间推进高精度求解三维流动问题
2. 支持旋转，多体运动，网格，精度（hp）等全套自适应及多级加速算法，MPI/OPENMP/OPENACC 混合并行
3. 发展了一整套三维任意精度阶, 工业级流动模拟方法（包含曲面网格生成，大规模计算，远程 IO 并行可视化）

## Ongoings

1. 发展旋转流动的高精度先进计算方法


- S.-J. Li. arXiv:1807.01151 (p-multigrid exponential for steady )
- S.-J. Li et al. AIAA-2019-0907 (3-D unsteady NS)
- S.-J. Li et al. AIAA-2018-0369 (3-D steady NS)
- S.-J .Li et al. JCP, 365(2018) 206-225 (3-D steady, unsteady inviscid)
- S.-J. Li et al. AIAA-2017-0753 (3-D steady, unsteady inviscid)



**Thanks to Prof. Pierre Lallemand , Z.J. Wang, Li-Shi Luo, Hang Si.**