# Performance of a Compressible DNS code on latest GPU architectures

Maruthi N. H.
Engineering Mechanics Unit
JNCASR, Bangalore, India

With: Prof. Roddam Narasimha, Prof. S. M. Deshpande, Kishore S. Patel

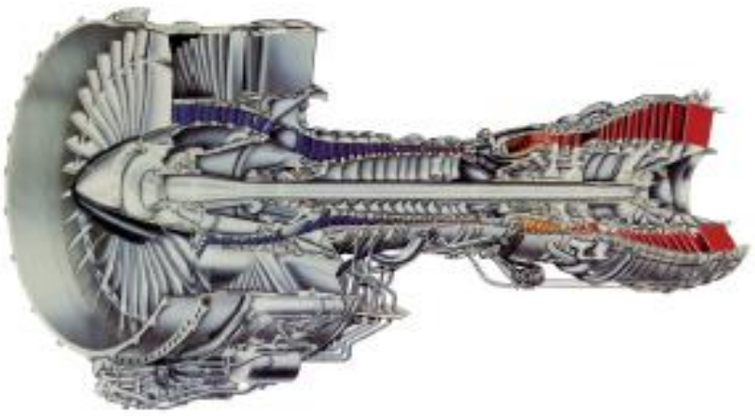
Rajesh Ranjan
The Ohio State University, Columbus, USA

S. R. Thejaswi, Bharatkumar Sharma
NVIDIA, Bangalore, India

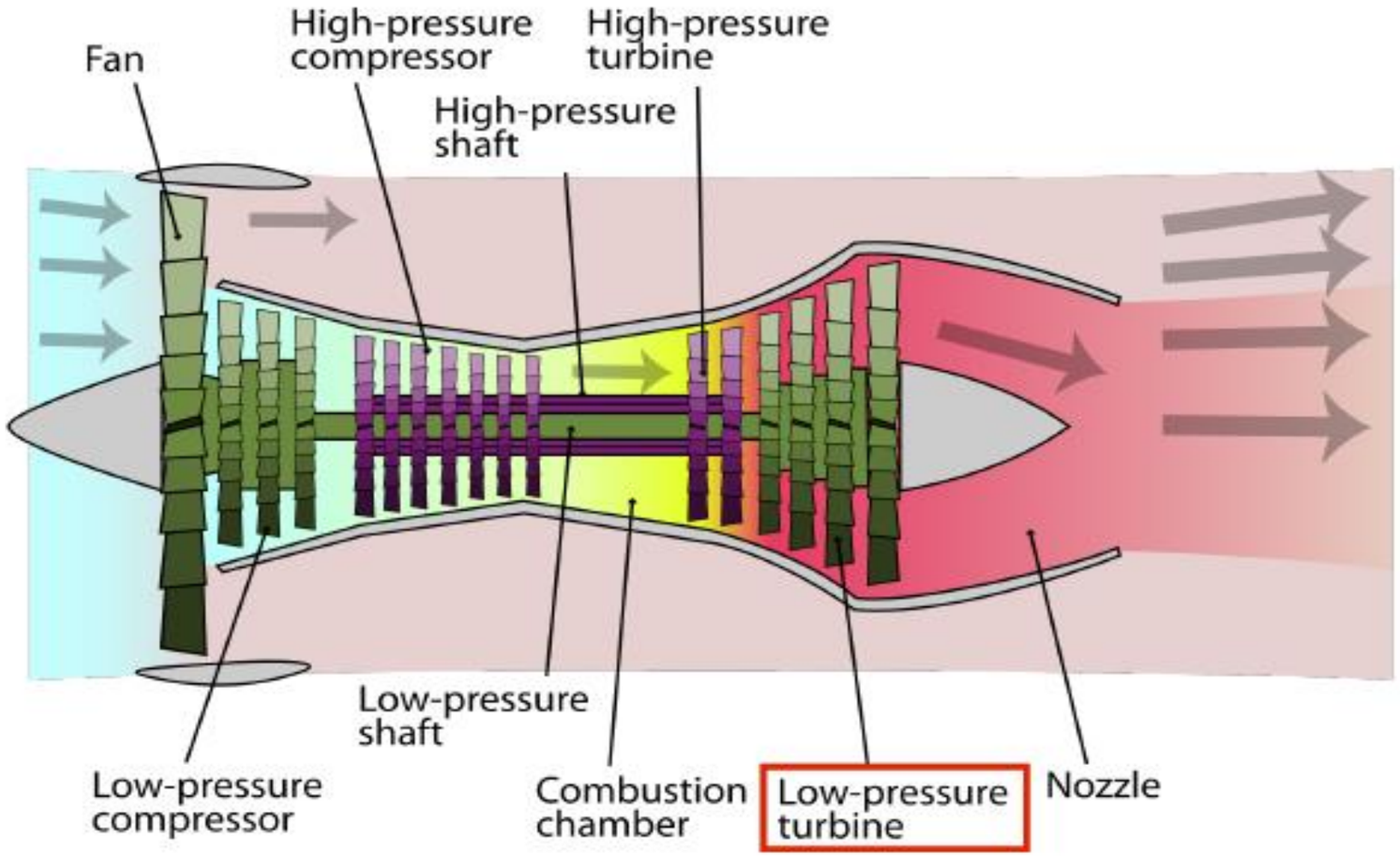
20th March 2019

GTC 2019
San Jose McEnery Convention Centre
San Jose, CA

# Gas Turbine Engine: Most Complex Piece of Mechanical Engineering



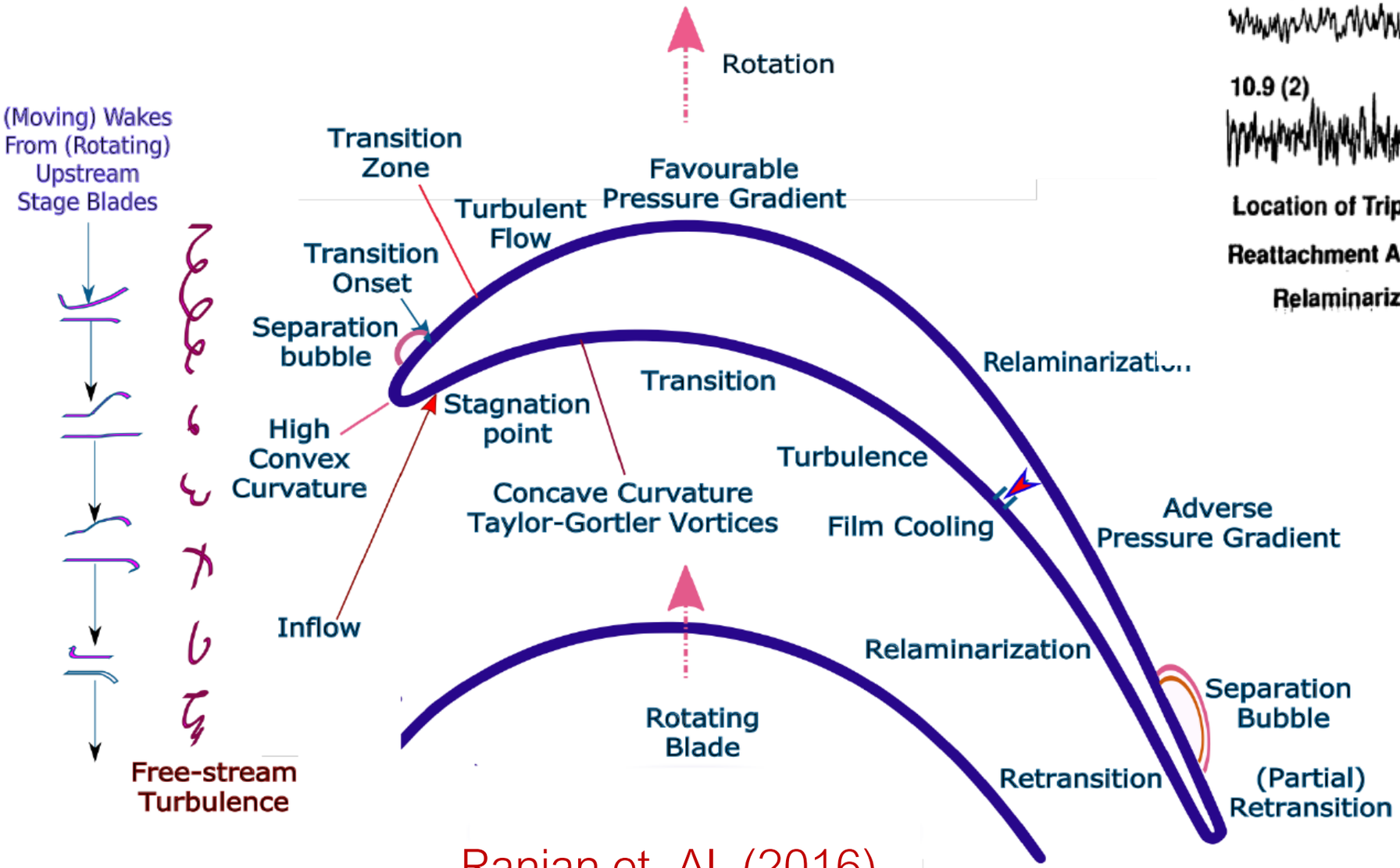
PW2000 engine (wiki)



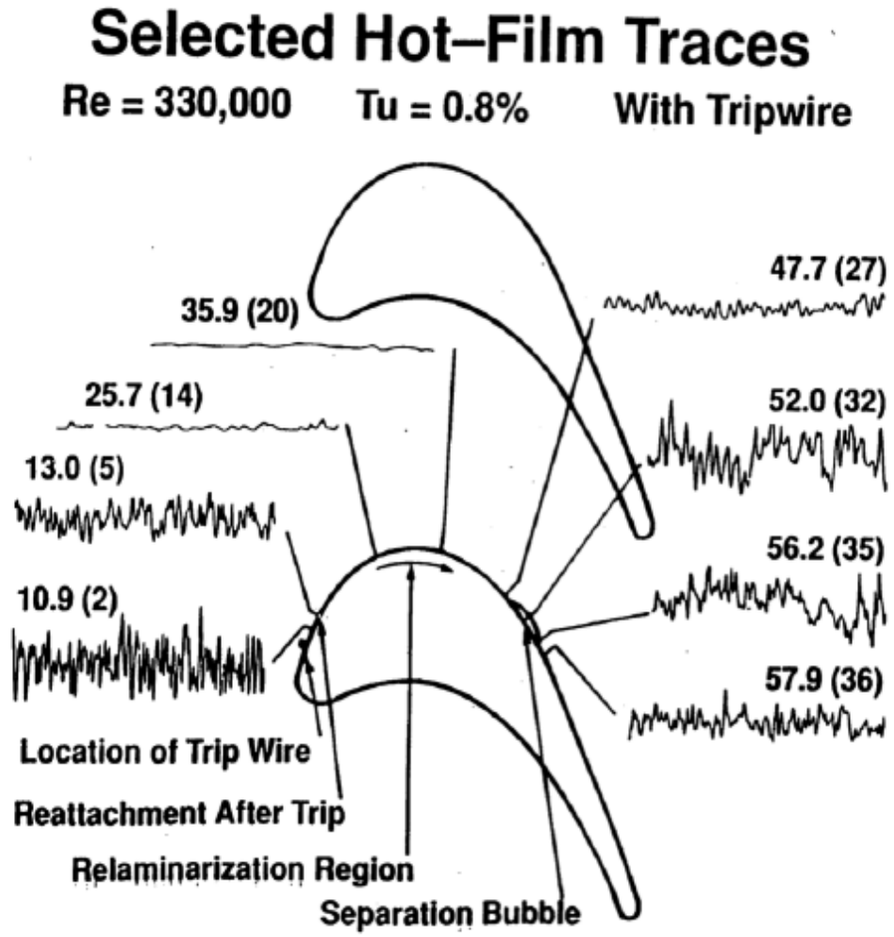

Schematic cross-section of high-bypass turbofan engine (wiki)

On Boeing 757, dual spool, bypass ratio 5.9:1
Thrust = 165-190kN
Dry weight = 3.22T

# The Flow Zoo on the Gas Turbine Blade




Ranjan et. Al. (2016)




Wisler (1993)

# What is a Complex Flow?

- ‘Complex’ flows have in general dynamics that is not understood
- Example: Turbulent flow in general.
  - Last great unsolved problem in physics
  - Equations known, solutions not, being beyond known maths: 1M$ prize from Clay Foundation
  - Turbulent phenomena beyond the reach of RANS models
  - Ex: transition from laminar to turbulent flow, relaminarization from turbulent to laminar, flow separation
- Fluid may also be complex: high speed, high temperature, multi-phase

  …

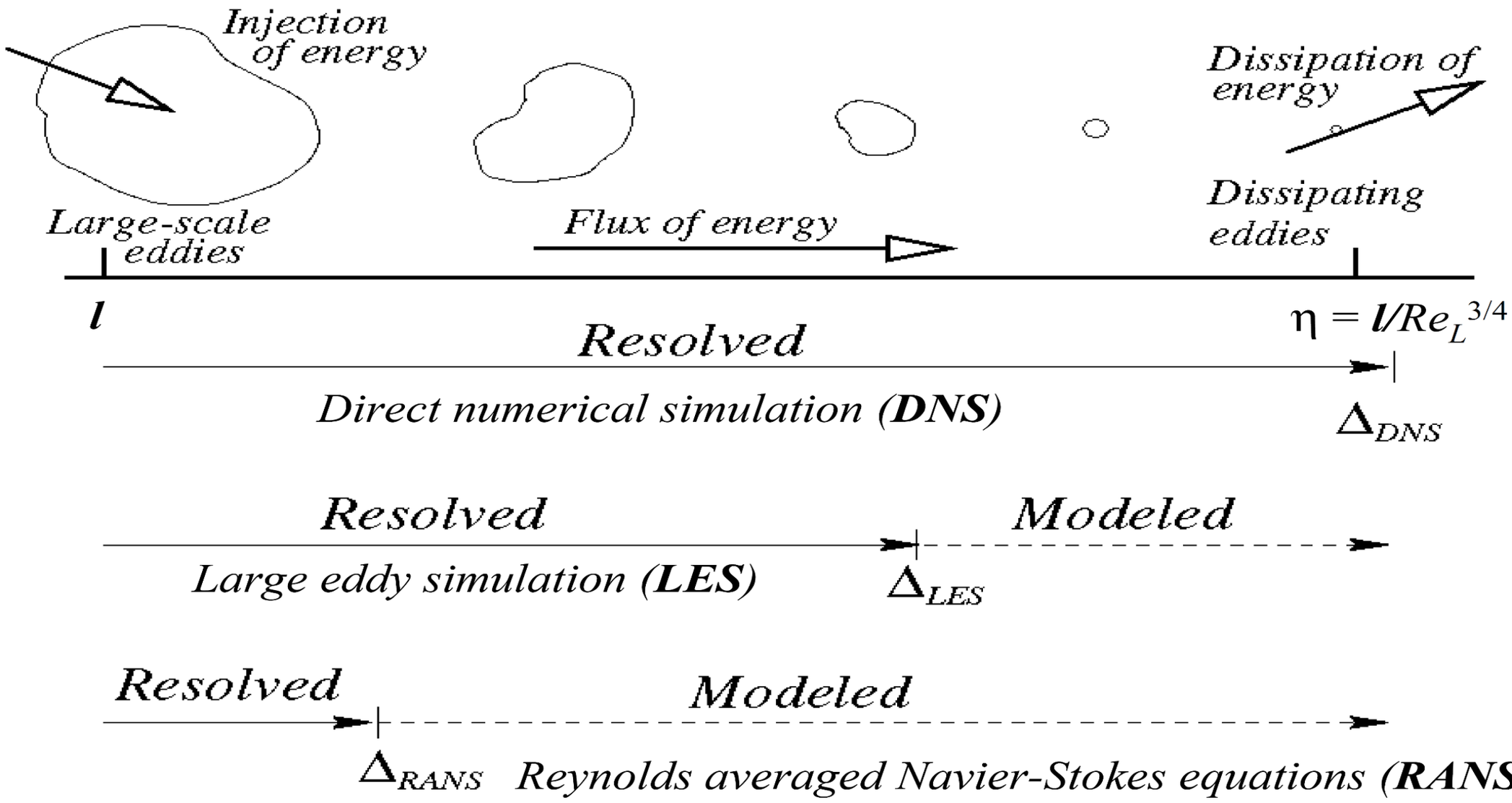
# The Grand Challenge of the Gas Turbine to CFD

- Gas turbine is the most sophisticated piece of mechanical engineering known
- Fluid is at high temperatures; flows at high speeds, is multi-phase, multi-component…
- Flow past blade subjected to:
  - High surface curvature (Prandtl's boundary layer theory, even with its first order correction, inadequate near leading edge)
  - Highly disturbed environment in the free-stream, with vortical (e.g. turbulence), acoustic (e.g. noise) and entropic (e.g. thermal inhomogeneities and fluctuations) components

# The Grand Challenge of the Gas Turbine to CFD

- High favourable and adverse pressure gradients (FPG, APG), inducing transition from laminar to turbulent flow and reverse, and flow separation at the surface, including separation bubbles…
- Shock/b.l. interactions…
- …and so on.
- Accurate predictions of such flows are beyond current RANS models; some hybrid or heterogenous models do better (e.g. LNS=LES+RANS near wall), but not yet satisfactory in comparison with DNS
- The modern turbine is a FLOW ZOO!

# Why DNS?




DNS solves physical equations without appeal to any model.
For a typical aircraft simulation, an exascale supercomputer is required

# The Problem

- Blades operate in highly turbulent environment, often with a strong periodic component due to wakes from upstream rotor blades
- Strong pressure gradients and high surface curvature
- Blade Re ~$20\times10^3$ to $1\times10^6$–a modest but awkward range...
- …in which blade boundary layer may experience transition, relaminarization and separation - all still beyond current RANS models
- Spatial and temporal variation of surface heat transfer rates by factor ~10
- A 25% difference in heat transfer rates on a turbine blade can mean an order of magnitude difference to its life.
- “ 1% improvement in the efficiency of a low pressure turbine would result in a saving of $52,000 per year on a typical airliner." - Jahanmiri (2011)
- DNS possible at lower end of Re range (as of now)

# Experimental Studies in Open Literature

| Ref | Blade | Re | Remarks |
|---|---|---|---|
| Stadtmuller (2002a,b) | T106A & T106D | 60,000 & 500,000 | Experimental data at low Re for wake-induced transition. Separation at TE |
| Stieger et al. (2003); Stieger & Hodson (2004, 2005) | T106A | 160, 000 | At low Re and low FSTI, wake triggers K-H instability that breaks into turbulent flow, gets convected |
| Liu & Rodi (1994b,a) | MTU | 72,000 | BL on suction side goes transitional under sweeping wakes. Higher wake-passing freq. raises heat transfer |
| Volino (2002a,b, 2003) | Pak-B | 25,000-300,000 | Increasing Re, FSTI move transition upstream. No reattachment and no transition at low Re |
| Choi et al. (2004) | - | 15,700-105,000 | Flow-separation at LE enhanced as Re decreases, suppressed by increasing FSTI |
| Kumaran et al. (2014), NAL | STFE | 152,000-1000,000 | Transition and relaminarisation cycles |

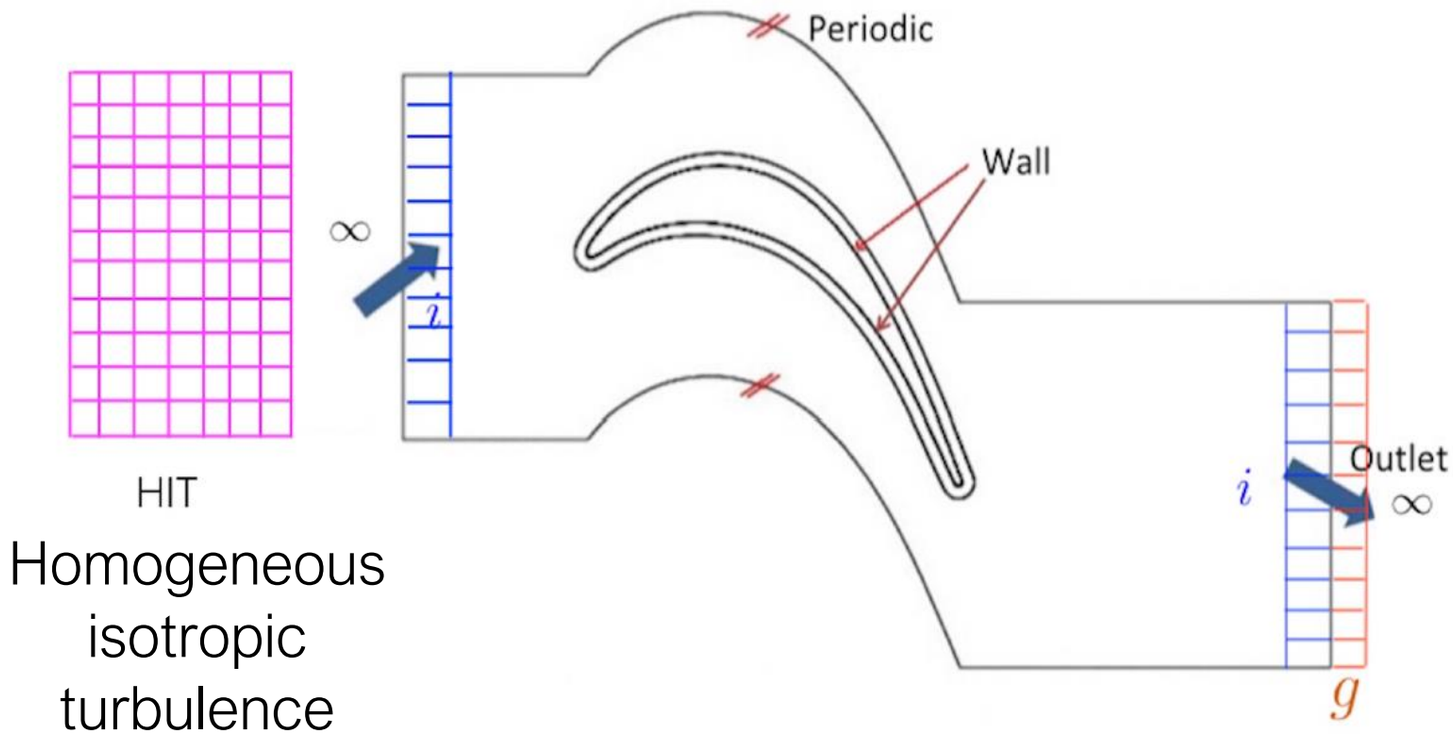
# Some DNS Studies in Literature

| Authors | Blade | Re | Type of Eqn | N×$10^6$ |
|---|---|---|---|---|
| Wu & Durbin (2001) | T106A | 148,000 | INS | 56 |
| Michelassi et al. (2002) | T106A | 51,800 | INS | 17 |
| Kalitzin et al. (2003) | T106A | 148,000 | INS | 85 |
| Wissink (2003) | T106A | 51,831 | INS | 17 |
| Wissink et al. (2006) | T106A | 51,800 | INS | 17 |
| Wissink & Rodi (2006) | MTU | 72,000 | INS | 93 |
| Ranjan et al. (2013[1], 2016) | T106A | 51,831 | CNS | 160 |
| Michelassi et al. (2015) | T106A | 59,634 | CNS | 18 |
| Garai et al. (2015) | T106A | 60,000 | CNS | 30.72 |
| Maruthi et al. (2017) | STFE | 152,000 | CNS | 93 |
| Maruthi et al. (2018) | STFE | 152,000 | CNS | 516 |

[1]First compressible solution

# DNS on T106A Blade

# Computational Domain and Simulation Parameters for T106A[1]




## Flow parameters

- $Re$ = 51831
- $M$ = 0.1 (inlet)
- $Pr$ = 0.71
- $\beta_1 = 45.50^0$ (inlet AoA)

## Simulation parameters

- CFL ≈ 1
- FSTI ≈ 0 – 10%
- Solution of previously simulated HIT (data due to Wu & Moin, JFM2009)
- Viscous padding near outflow
- 7-10 flow steps

| Sim | $N_f$ | $N_z$ | Grid Size |
|---|---|---|---|
| A | 384,164 | 64 | $25 \times 10^6$ |
| B | 740,088 | 64 | $47 \times 10^6$ |
| B1 | 740,088 | 128 | $95 \times 10^6$ |
| C | 1,257,162 | 128 | $161 \times 10^6$ |

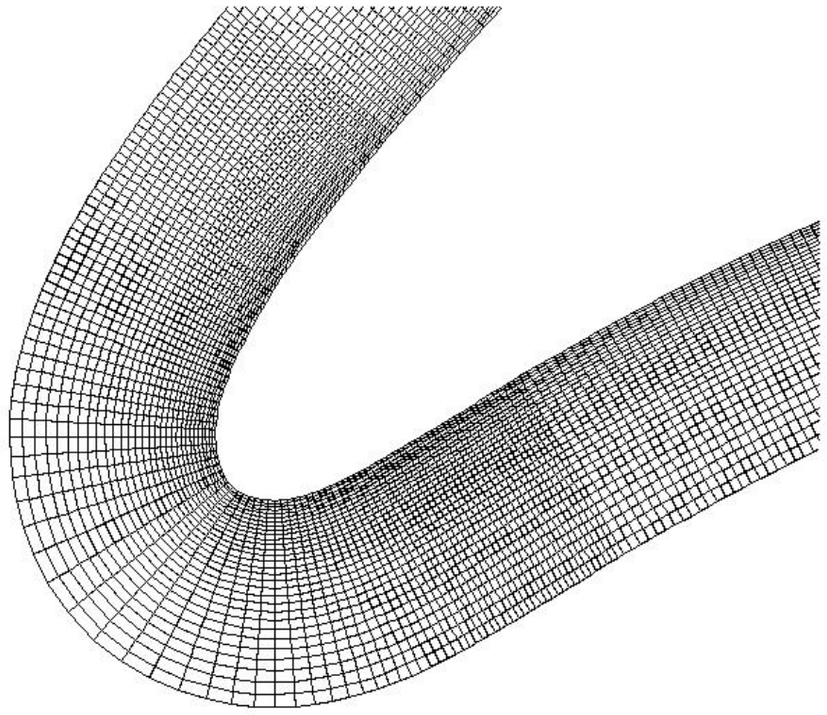
Overview of simulations

[1]Ranjan, SMD, RN, Com. Fluids, 2017

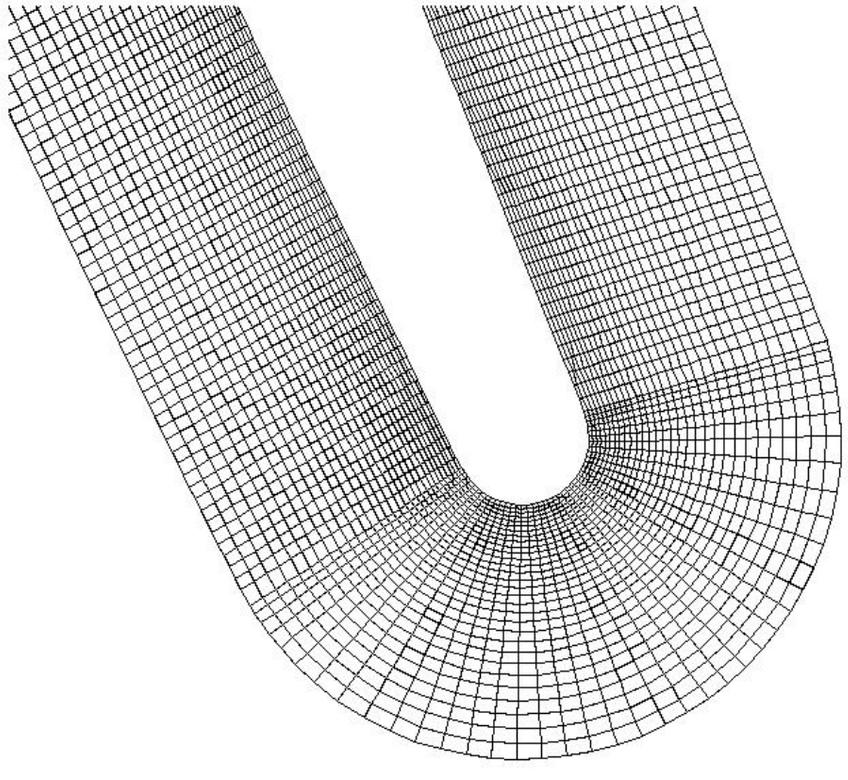
# High-resolution Compressible DNS on T106A

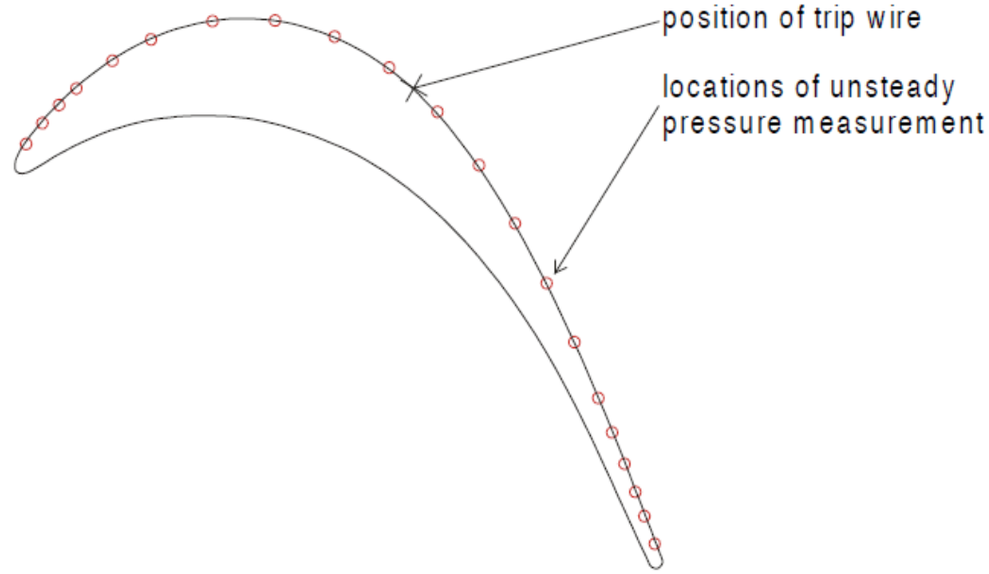
## Object of present DNS study

Leading edge



Trailing edge






Stieger(2002)

- Most extensively studied LPT blade
- Used in PW2037 engine
- Test results from Universitat der Bundeswehr in Munchen, available as semi-open literature

# Governing N-S Equations for Compressible Flow

Continuity: $$\frac{\partial \rho}{\partial t} + \frac{\partial \left(\rho u_j\right)}{\partial x_j} = 0$$

Momentum: $$\frac{\partial (\rho u_i)}{\partial t} + \frac{\partial \left(\rho u_i u_j\right)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j}$$

Energy: $$\frac{\partial (\rho E)}{\partial t} + \frac{\partial \left(\rho u_j H\right)}{\partial x_j} = \frac{\partial \left(u_i \tau_{ij}\right)}{\partial x_j} - \frac{\partial q_j}{\partial x_j}$$

Equation of state: $p = \rho R T$

Energy: $$E = \frac{p}{\rho(\gamma - 1)} + \frac{1}{2}\sum_{i=1}^{3} u_i u_i$$

Enthalpy: $$H = E + \frac{p}{\rho}$$

# Governing N-S Equations for Compressible Flow

Constitutive equations:

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k}\right)$$

$$q_i = -k\frac{\partial T}{\partial x_i}$$

Sutherland's law for viscosity and thermal conductivity:

$$\frac{\mu}{\mu_{ref}} = \left(\frac{T}{T_{ref}}\right)^{1.5}\left(\frac{T_{ref} + S_\mu}{T + S_\mu}\right)$$

$$k = c_p\frac{\mu}{Pr}$$

$S_\mu$ and $S_k$ are constants for a given fluid

# N-S Equations in Integral form

$$\frac{\partial}{\partial t}\int_{\Omega}\vec{U}\,d\Omega+\oint_{\partial\Omega}\left(\vec{F}_c-\vec{F}_v\right)\mathrm{dS}=0$$

$$\vec{U}=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho E\end{bmatrix}\qquad \overrightarrow{F_c}=\begin{bmatrix}\rho V\\ \rho uV+n_x p\\ \rho vV+n_y p\\ \rho wV+n_z p\\ \rho VH\end{bmatrix}\qquad \overrightarrow{F_v}=\begin{bmatrix}0\\ n_x\tau_{xx}+n_y\tau_{xy}+n_z\tau_{xz}\\ n_x\tau_{yx}+n_y\tau_{yy}+n_z\tau_{yz}\\ n_x\tau_{zx}+n_y\tau_{zy}+n_z\tau_{zz}\\ n_x\theta_x+n_y\theta_y+n_z\theta_z\end{bmatrix}$$

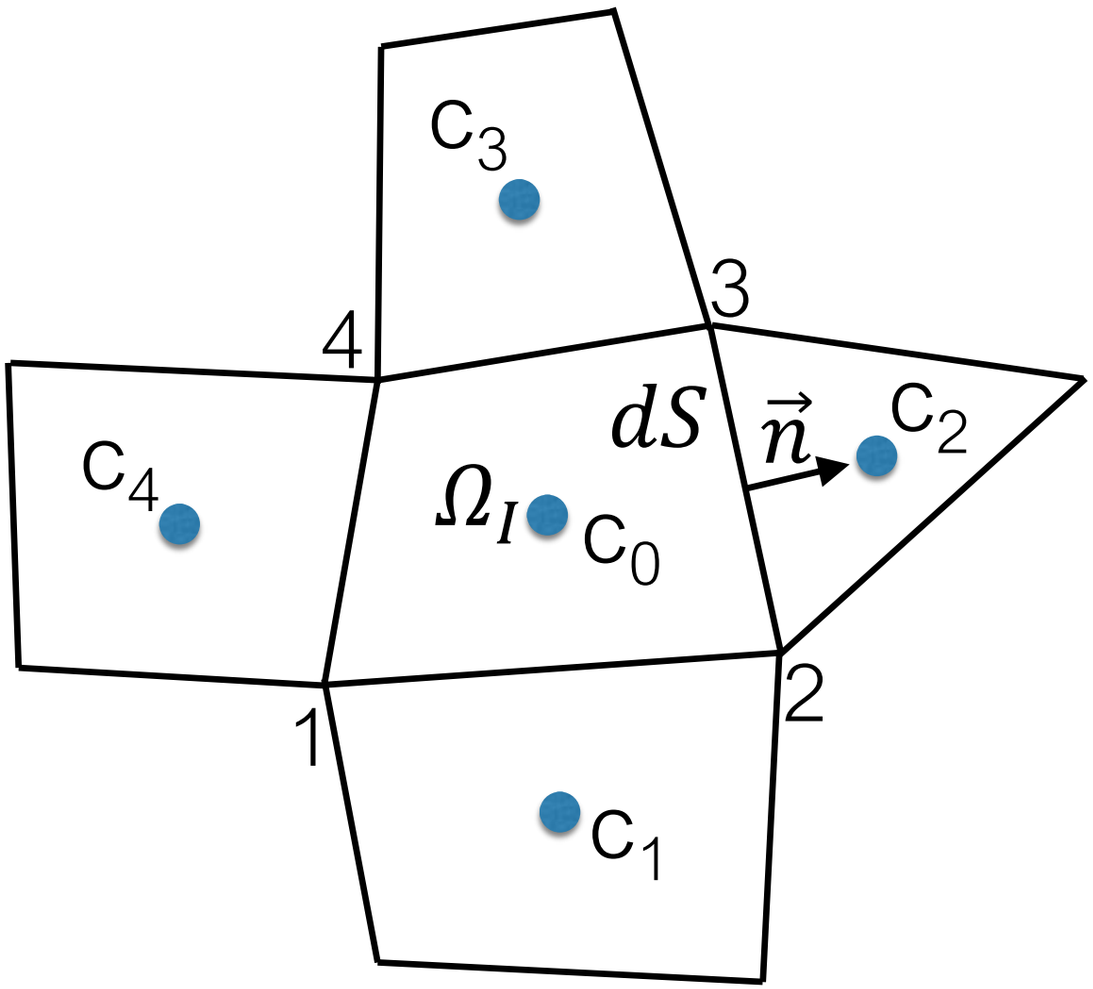
$$V=\vec{v}.\vec{n}=n_x u+n_y v+n_z w$$

$$\theta_x=u\tau_{xx}+v\tau_{xy}+w\tau_{xz}+k\frac{\partial T}{\partial x}$$

$$\theta_y=u\tau_{yx}+v\tau_{yy}+w\tau_{zz}+k\frac{\partial T}{\partial y}$$

$$\theta_x=u\tau_{zx}+v\tau_{zy}+w\tau_{zz}+k\frac{\partial T}{\partial z}$$

# Finite Volume Method (FVM)




Control volume of cell-centred scheme (2D)

$$\frac{\partial}{\partial t}\int_{\Omega}\vec{U}\,d\Omega = \Omega\frac{\partial\vec{U}}{\partial t}$$

$$\frac{\partial\vec{U}}{\partial t} = -\frac{1}{\Omega}\oint_{\partial\Omega}\left(\vec{F}_c - \vec{F}_v\right)\mathrm{dS}$$

If we consider particular control volume

$$\frac{d\vec{U}_I}{dt} = -\frac{1}{\Omega_I}\sum_{k=1}^{n_f}\left[\left(\vec{F}_c - \vec{F}_v\right)_k\right]\Delta S_k$$

$$\frac{d\vec{U}_I}{dt} = -\frac{1}{\Omega_I}\vec{R}_I$$

# Inviscid Flux Discretization

For discretising inviscid fluxes, we use kinetic energy preserving scheme due to A. Jameson (J. Sci. Com., 2008)

$$\overrightarrow{F_c} = \begin{bmatrix} \bar{\rho}\bar{V} \\ \bar{\rho}\bar{u}\bar{V} + n_x\bar{p} \\ \bar{\rho}\bar{v}\bar{V} + n_y\bar{p} \\ \bar{\rho}\bar{w}\bar{V} + n_z\bar{p} \\ \bar{\rho}\bar{V}\bar{H} \end{bmatrix}$$

$$\bar{V} = n_x\bar{u} + n_y\bar{v} + n_z\bar{w}$$

Here, bar denotes the average quantities.

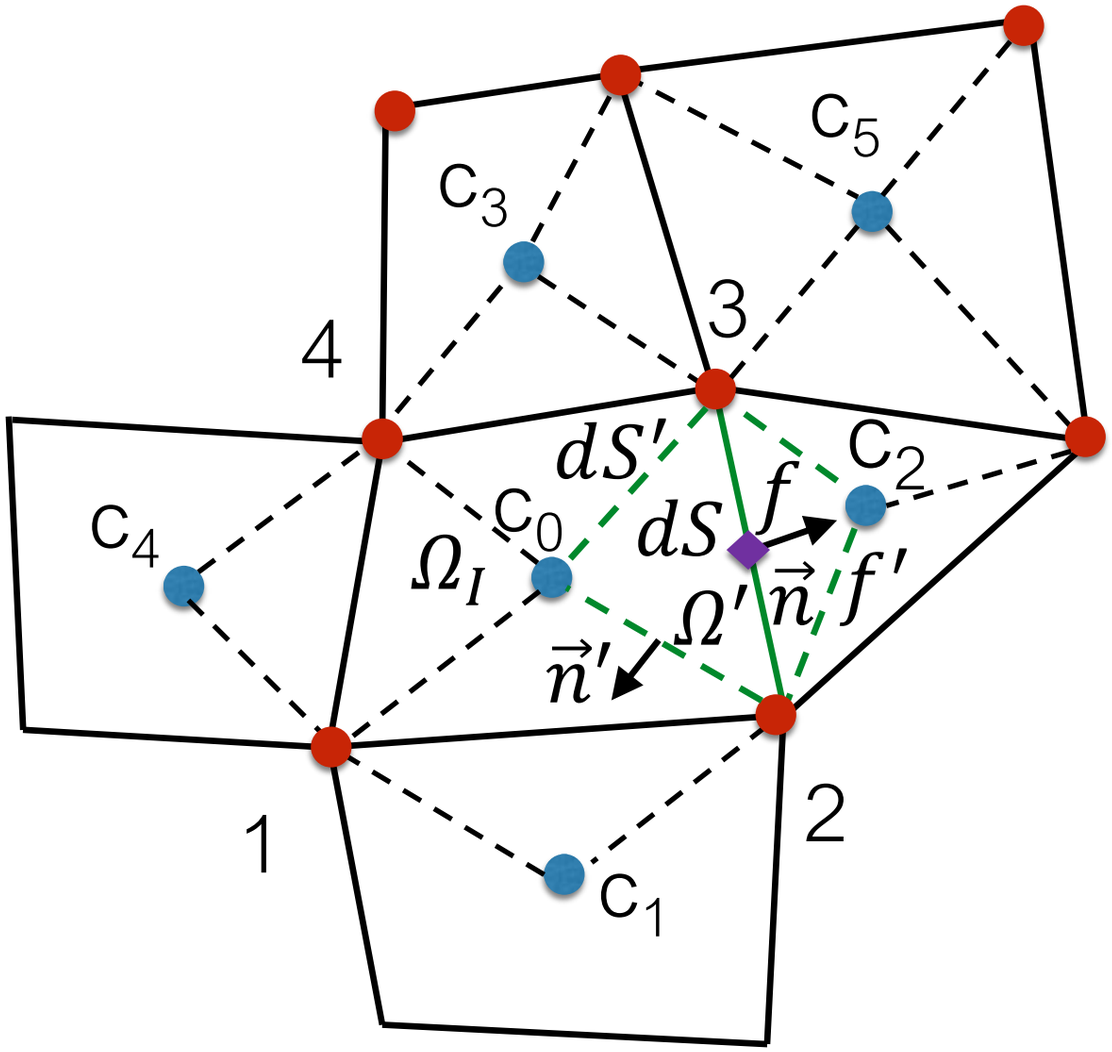
For example,

$$\bar{\rho}_{23} = \frac{1}{2}\left(\rho_{c_2} + \rho_{c_0}\right)$$

Similarly, other quantities can be obtained

# Viscous Flux Discretization

Calculation of the viscous terms needs velocity and temperature gradients at the faces of the control volume. In ANUROOP, the Green-Gauss (GG) method is used to calculate face gradients (Frink, AIAA 1994) .




Diamond path

The gradient of a quantity $\phi$ at a face $f$ (here face 2-3) using Green-Gauss method is given by

$$\int_{\Omega'} \nabla\phi_f \, d\Omega' = \oint_{\partial\Omega'} \phi \vec{n}' dS'$$

$$\nabla\phi_f = \frac{1}{\Omega'} \sum_{i=1}^{n_{f'}} \phi_{n_{f'}} \Delta S'_{n_{f'}}$$

# Viscous Flux Discretization

Any quantity $\phi$ at any vertex can be obtained from a weighted average

For example

$$\phi_n = \frac{\sum_i^N w_i \phi_i}{\sum_i^N w_i}$$

$w_i$ is the weight associated with cell $i$ surrounding vertex $n$

In ANUROOP code, we use Pseudo-Laplacian average

The Laplacian

$$L(\phi_n) = \sum_i^N w_i(\phi_i - \phi_n) = 0$$

Weights are given by

$$w_i = 1 + \Delta w_i$$

# Viscous Flux Discretization

Weights are found using the method of Lagrange multipliers.

Cost function for this optimization problem is

$$C = \sum_{i=1}^{N} (\Delta w_i)^2$$

Subjected to constraints

$$L(x_n) = \sum_{i}^{N} w_i (x_i - x_n) = 0$$

$$L(y_n) = \sum_{i}^{N} w_i (y_i - y_n) = 0$$

$$L(z_n) = \sum_{i}^{N} w_i (z_i - z_n) = 0$$

# Viscous Flux Discretization

The resulting weights

$$\Delta w_i = \lambda_x(x_i - x_n) + \lambda_y(y_i - y_n) + \lambda_z\,(z_i - z_n)$$

The solution to the optimization problems yields the following Lagrange multipliers

$$-R_x\left(I_{yy}I_{zz} - I_{yz}^2\right) + R_y\left(I_{xy}I_{zz} - I_{xz}I_{yz}\right)$$

$$\lambda_x = \frac{\begin{array}{c} - R_z\left(I_{xy}I_{yz} - I_{yy}I_{xz}\right) \\ \hline I_{xx}\left(I_{yy}I_{zz} - I_{yz}^2\right) - I_{xy}\left(I_{xy}I_{zz} - I_{xz}I_{yz}\right) \\ + I_{xz}\left(I_{xy}I_{yz} - I_{yy}I_{xz}\right) \end{array}}$$

$$\lambda_y = \frac{\begin{array}{c} R_x\left(I_{xy}I_{zz} - I_{xz}I_{yz}\right) - R_y\left(I_{xx}I_{zz} - I_{xz}^2\right) \\ + R_z\left(I_{xx}I_{yz} - I_{xy}I_{xz}\right) \\ \hline I_{xx}\left(I_{yy}I_{zz} - I_{yz}^2\right) - I_{xy}\left(I_{xy}I_{zz} - I_{xz}I_{yz}\right) \\ + I_{xz}\left(I_{xy}I_{yz} - I_{yy}I_{xz}\right) \end{array}}$$

# Viscous Flux Discretization

$$\lambda_z = \frac{\begin{array}{c} -R_x\left(I_{xy}I_{yz} - I_{yy}I_{xz}\right) + R_y\left(I_{xx}I_{yz} - I_{xy}I_{xz}\right) \\ - R_z\left(I_{xx}I_{yy} - I_{xy}^2\right) \end{array}}{\begin{array}{c} I_{xx}\left(I_{yy}I_{zz} - I_{yz}^2\right) - I_{xy}\left(I_{xy}I_{zz} - I_{xz}I_{yz}\right) \\ + I_{xz}\left(I_{xy}I_{yz} - I_{yy}I_{xz}\right) \end{array}}$$

$$R_x = \sum_{i=1}^{N}(x_i - x_n)\,, \qquad R_y = \sum_{i=1}^{N}(y_i - y_n), \qquad R_x = \sum_{i=1}^{N}(z_i - z_n)$$

$$I_{xx} = \sum_{i=1}^{N}(x_i - x_n)^2\,, \qquad I_{yy} = \sum_{i=1}^{N}(y_i - y_n)^2, \qquad I_{zz} = \sum_{i=1}^{N}(z_i - z_n)^2$$

# Viscous Flux Discretization

$$I_{xy} = \sum_{i=1}^{N} (x_i - x_n)(y_i - y_n)$$

$$I_{xz} = \sum_{i=1}^{N} (x_i - x_n)(z_i - z_n)$$

$$I_{yz} = \sum_{i=1}^{N} (y_i - y_n)(z_i - z_n)$$

These weights are computed entirely from geometric information

# Time Stepping Scheme

In case of explicit scheme

$$\Delta\vec{U}_I^n = -\frac{\Delta t_I}{\Omega_I}\vec{R}_I^n$$

Here, we use the strong stability preserving Runge-Kutta-3 method given as

$$\vec{U}_I^0 = \vec{U}_I^n$$
$$\vec{U}_I^1 = \vec{U}_I^0 - \frac{\Delta t_I}{\Omega_I}\vec{R}_I^0$$
$$\vec{U}_I^2 = \frac{3}{4}\vec{U}_I^0 + \frac{1}{4}\vec{U}_I^1 - \frac{1}{4}\frac{\Delta t_I}{\Omega_I}\vec{R}_I^1$$
$$\vec{U}_I^{n+1} = \frac{1}{3}\vec{U}_I^0 + \frac{2}{3}\vec{U}_I^1 - \frac{2}{3}\frac{\Delta t_I}{\Omega_I}\vec{R}_I^2$$

# Time Step

$$\Delta t_I = \frac{\Omega_I}{(\Lambda_{inv} + B\,\Lambda_v)_I}$$

$$(\Lambda_{inv})_I = \sum_{k=1}^{n_f} (|\boldsymbol{v}.\boldsymbol{n}| + c)\,\Delta S_k$$

$$(\Lambda_v)_I = \frac{1}{\Omega_I} \sum_{k=1}^{n_f} max\left(\frac{4}{3\rho}, \frac{\gamma}{\rho}\right)\left(\frac{\mu}{Pr}\right)\,(\Delta S_k)^2$$

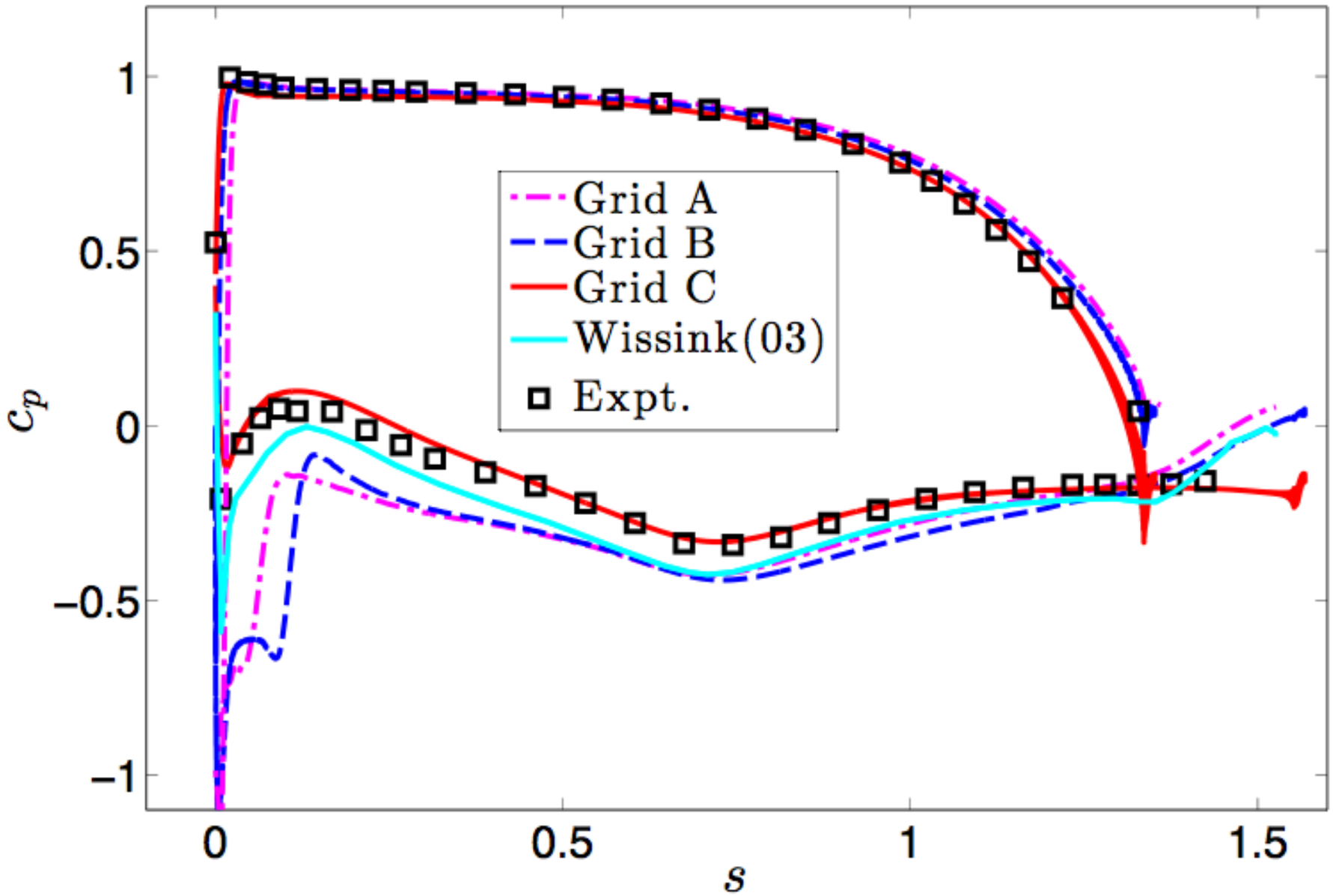
- $\Lambda_{inv}$ and $\Lambda_v$ represents the inviscid and viscous spectral radii
- $\boldsymbol{v}.\boldsymbol{n}$ is the normal velocity on the face $k$ of a cell $I$ and $c$ is the speed of sound
- Constant $B$ is taken as 4, which is recommended for central solvers

# DNS Code ANUROOP[1]

| | |
|---|---|
| Type of equations | Compressible Navier-Stokes |
| Dimensionality | 3D |
| Discretization methodology | Finite volume |
| Inviscid flux reconstruction | Second order kinetic energy preserving central difference scheme |
| Viscous flux reconstruction | Green-Gauss method |
| Time discretization | Explicit (SSPRK-3) |
| Element type | Hexahedra, prisms, tetrahedra (Unstructured) |
| Mesh partitioning | METIS |
| Computer architecture | CPU, GPU (NVIDIA) |

[1]Rajesh Ranjan, Roddam Narasimha, S. M. Deshpande

# T106A Blade Simulations: Pressure Distribution[1]




Mean pressure co-efficient along arc length

Pressure side $c_p$: robust, benign
Suction side $c_p$: sensitive to resolution

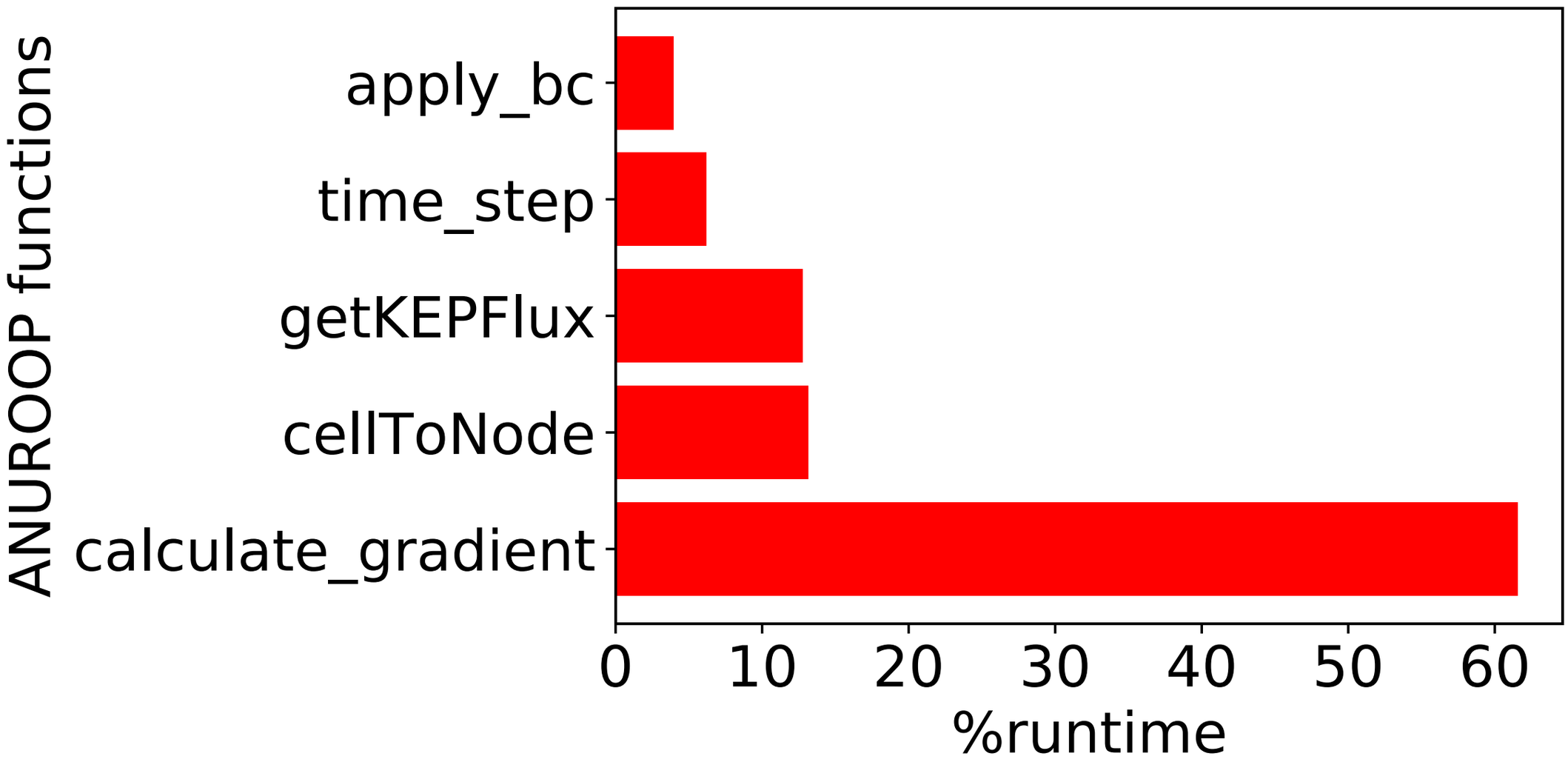
Wissink (2003): 17; Grid A: 25; Grid B: 47; Grid C: 161

[1]Ranjan, SMD, RN, Com. Fluids, 2017

# Observations of ANUROOP from Computational Point of View

- ANUROOP being an unstructured code has lot of non-coalesced memory accesses
- Both memory and compute intensive
- Major part of the code is parallelizable, and hence suitable for acceleration on GPUs

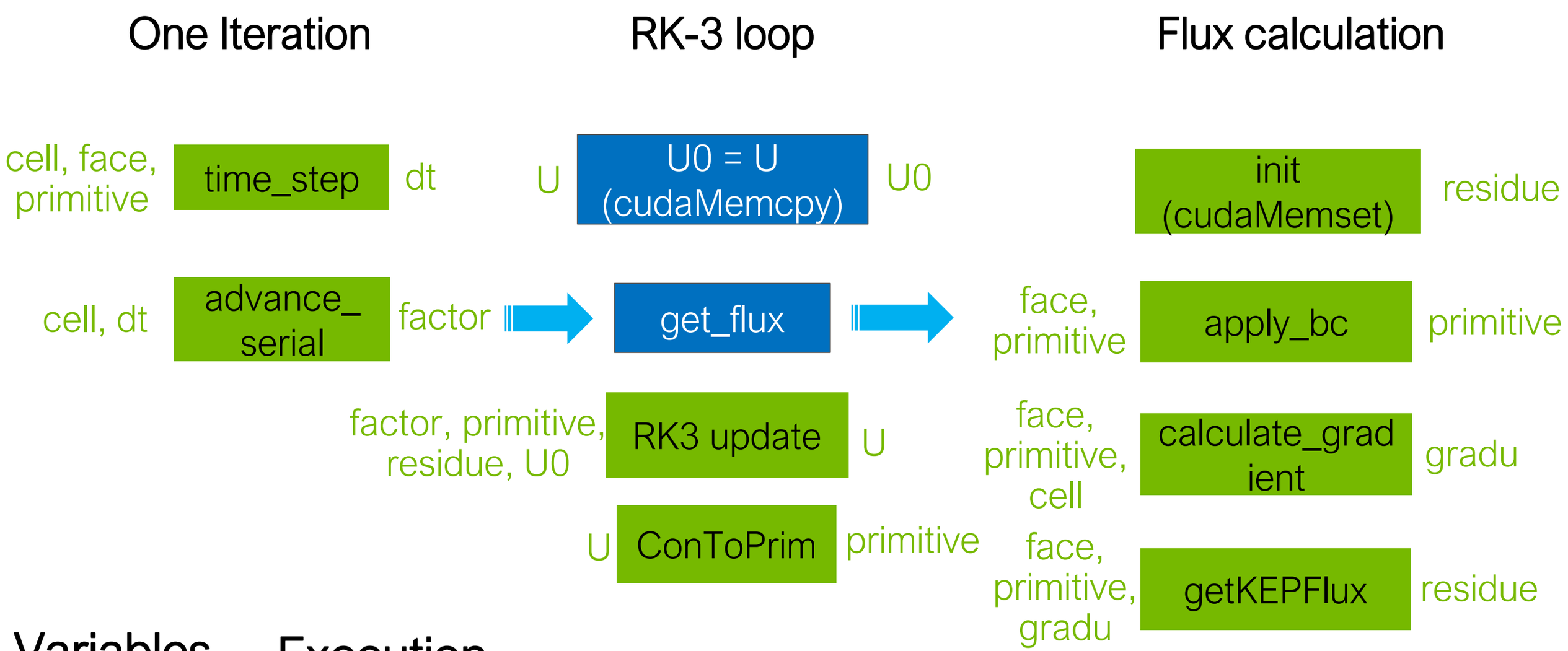
# Profiling of Baseline C++ Code

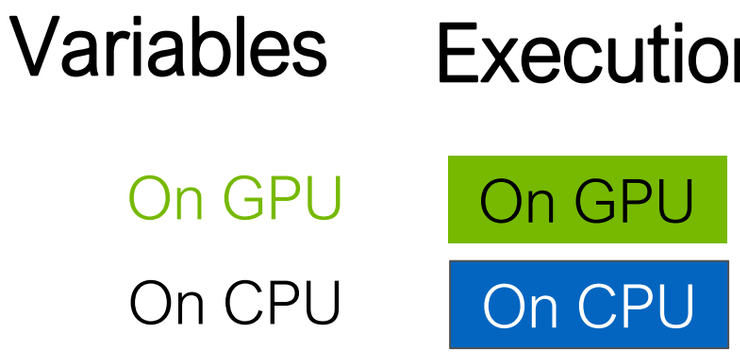


# Conversion of Baseline C++ Code to CUDA

- Identify hot-spots
  - Convert those module to CUDA
- Eliminate CPU⇔GPU copies of variables
  - New GPU kernels implemented
- Main bottleneck kernels on GPU were optimized
  - AoS was converted to SoA
- Modules leading to race conditions were identified
  - New kernel was written for GPU
- Most of the CPU⇔GPU communications are eliminated
  - Entire iteration runs on GPU

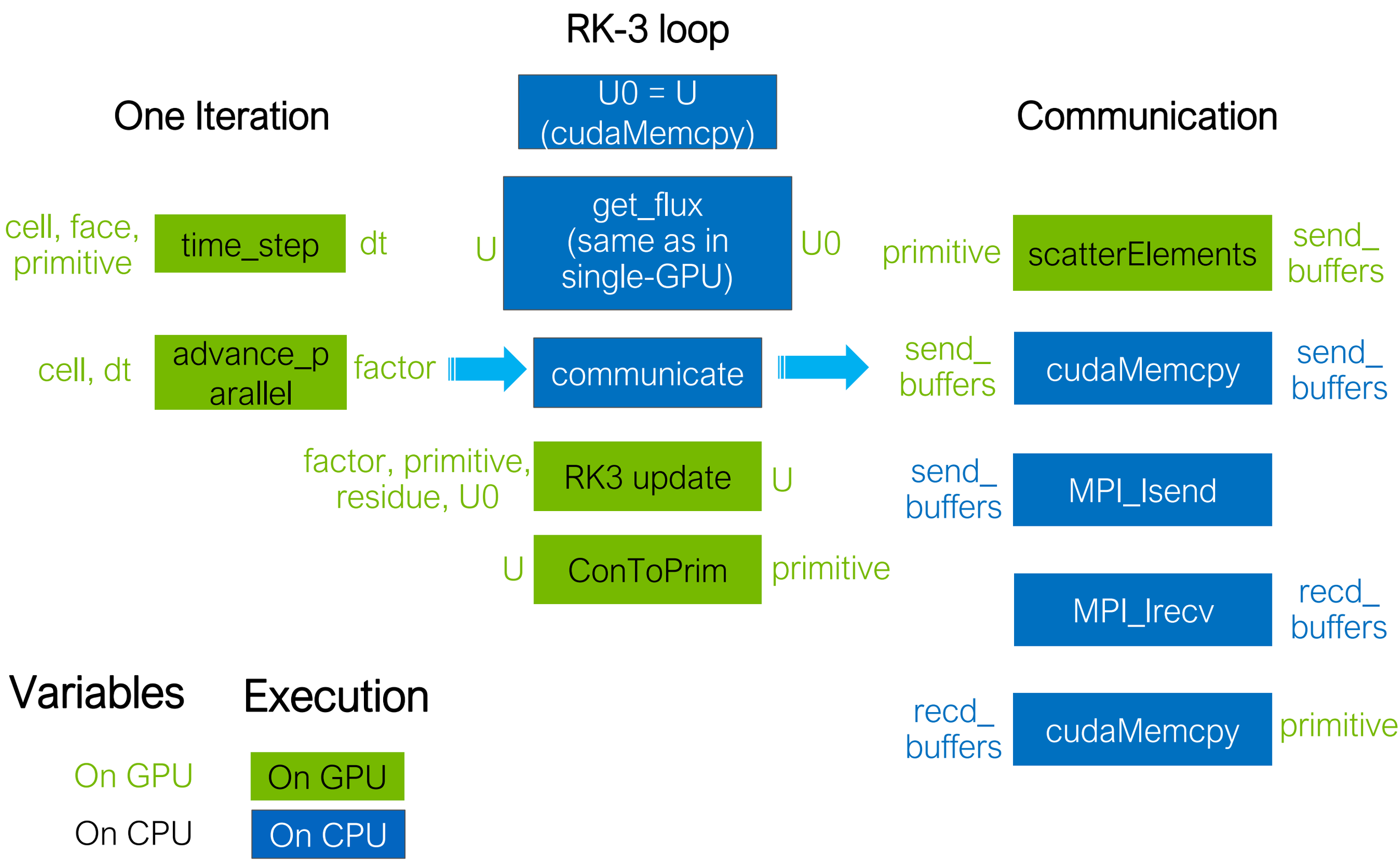
# ANUROOP: Single-GPU Flow of Work




Functions mentioned in the green box are executed on GPUs

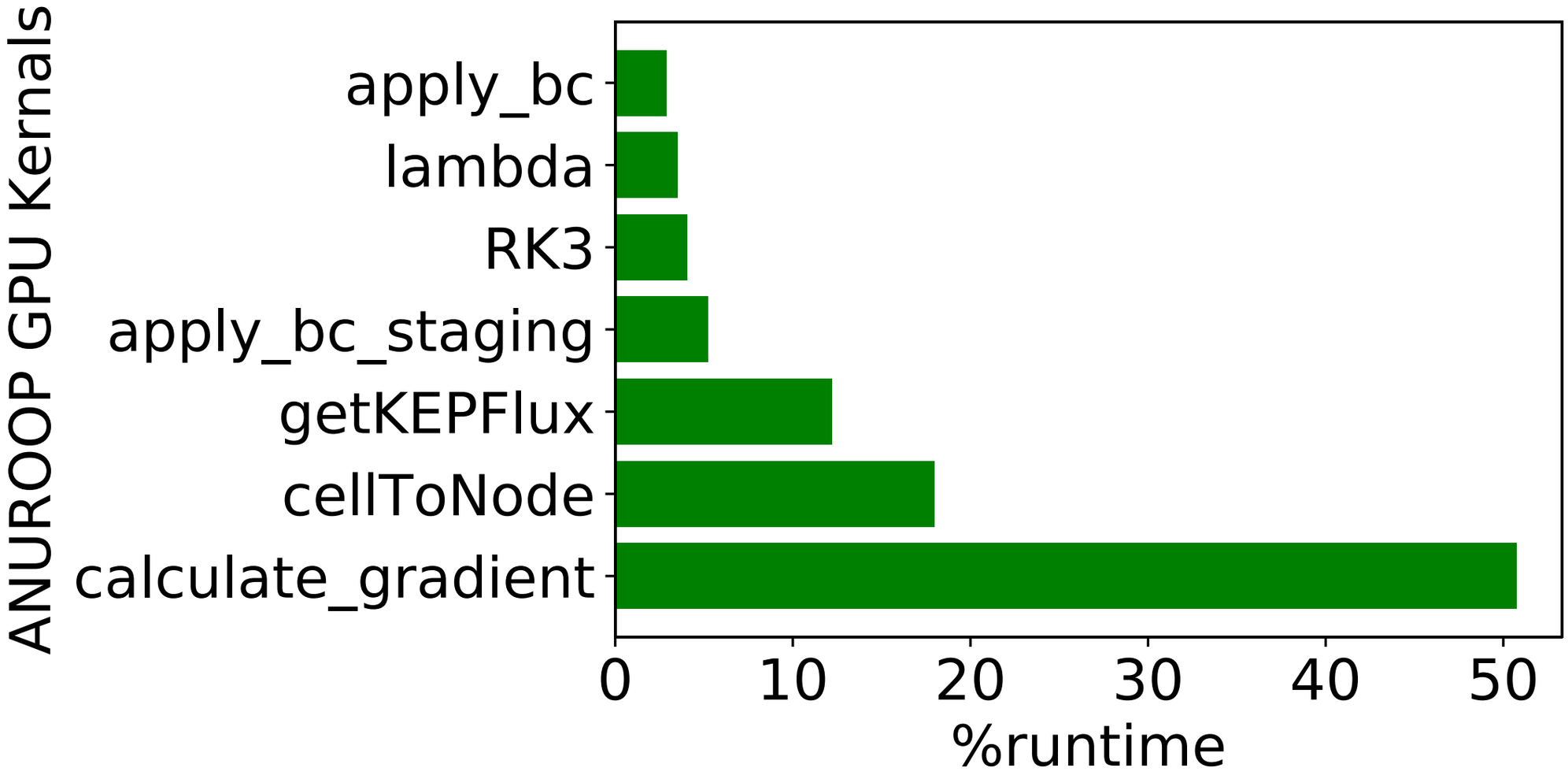
# ANUROOP: Multi-node Multi-GPU Flow of Work




Functions mentioned in the green box are executed on GPUs

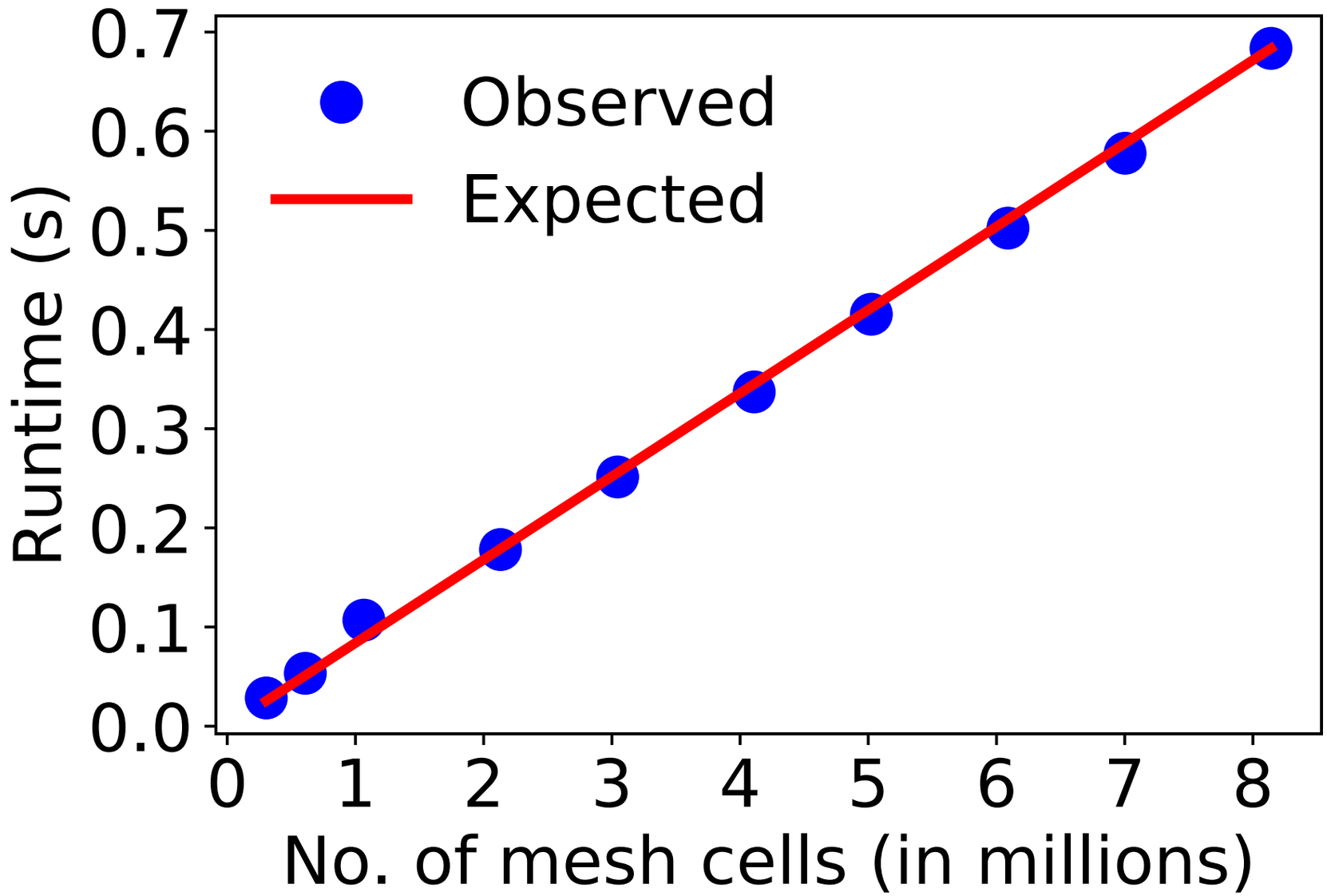
# GPU Accelerated ANUROOP

- CUDA version of ANUROOP is developed in collaboration with NVIDIA
- Most of the baseline code is parallelized with CUDA. So CPU is used only to control and schedule the job



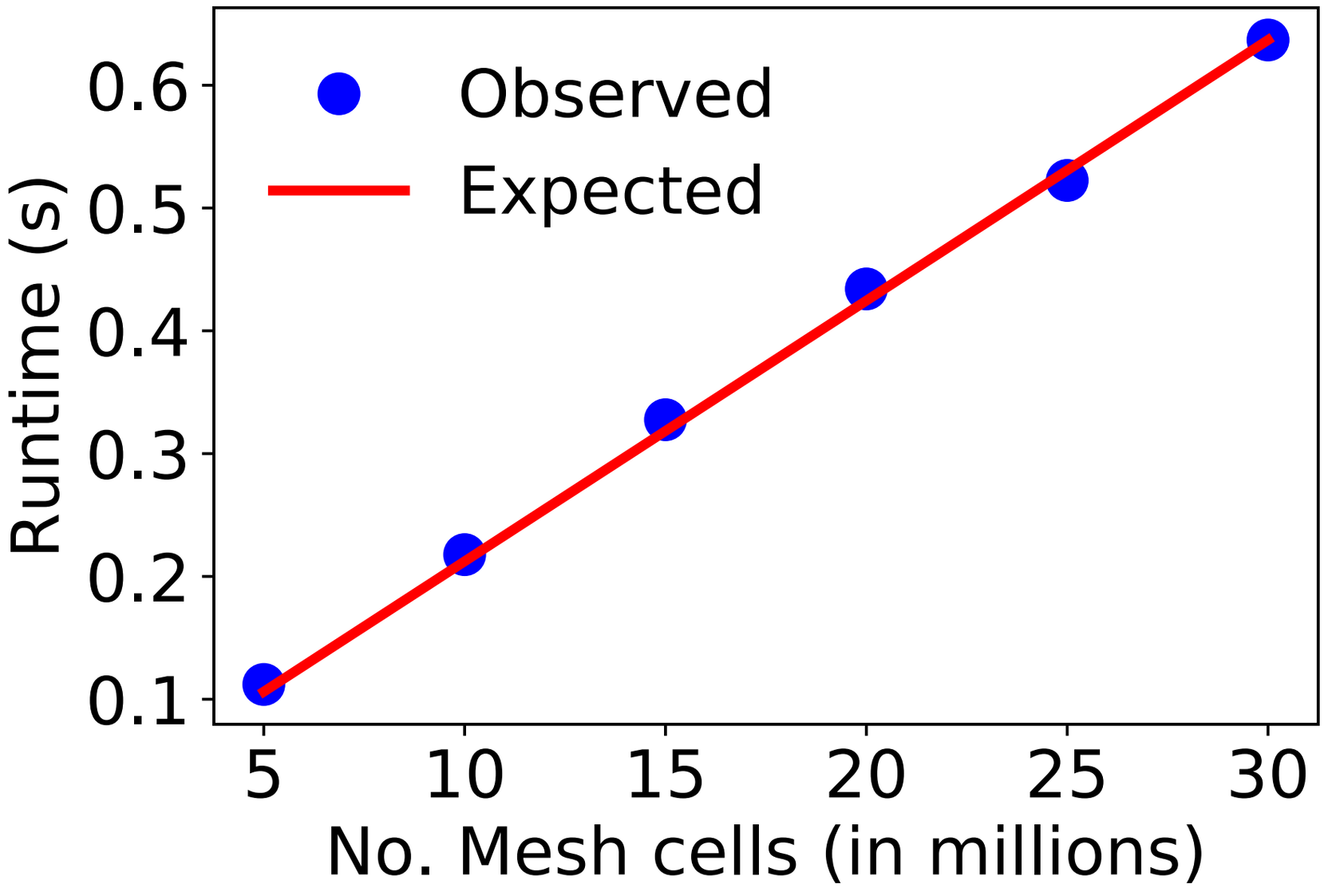

N. H. Maruthi et al., GPU acceleration of a DNS code for gas turbine blade simulations, CSS, IISc, March 2017.

# Performance on Single P100



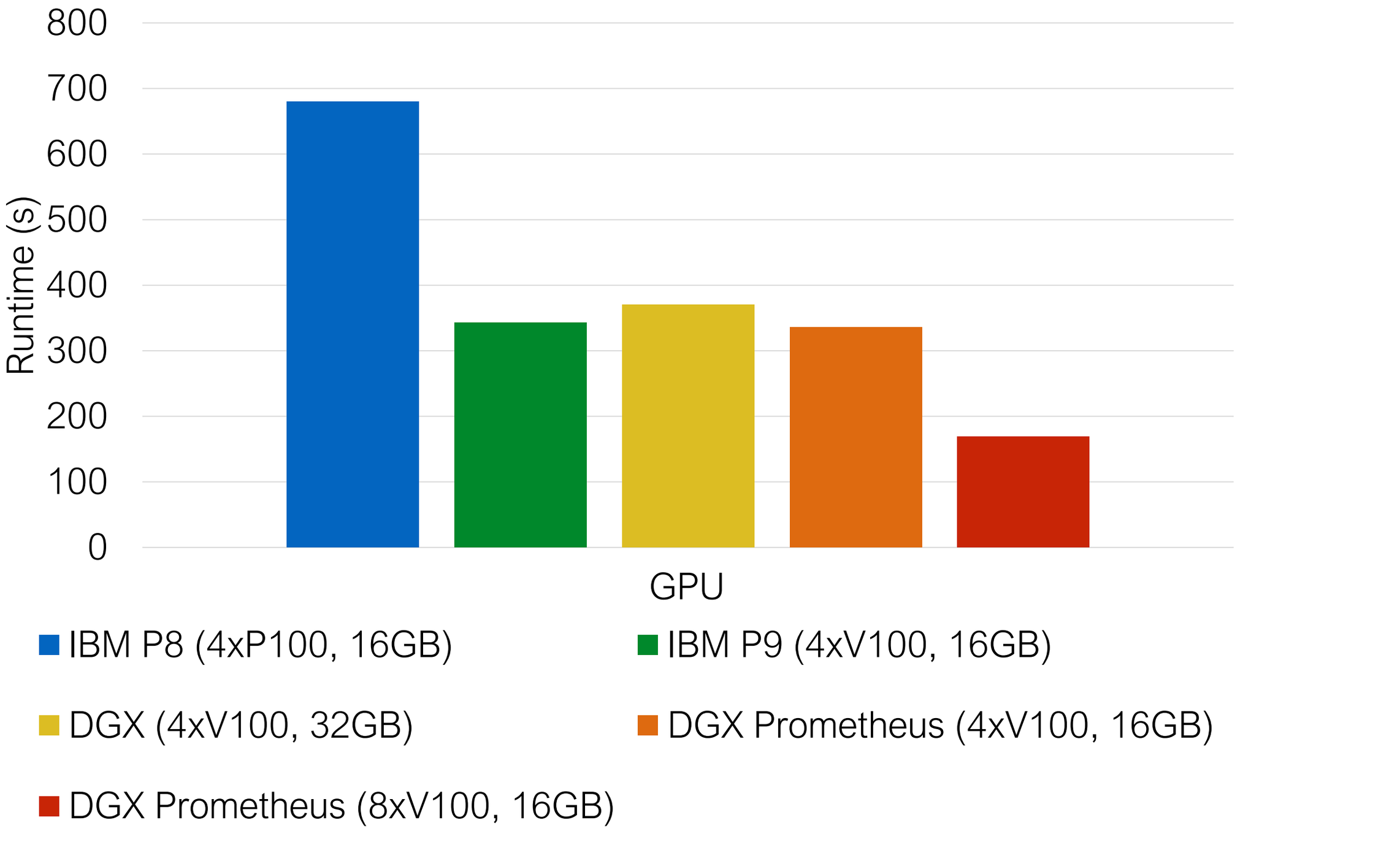

High Bandwidth Memory (HBM) on P100 provides faster memory accesses, and hence the performance degradation as the number of cells are decreased is negligible

# Performance on One node of IBM Minsky (4 P100)



# Performance on V100 in Comparison with P100

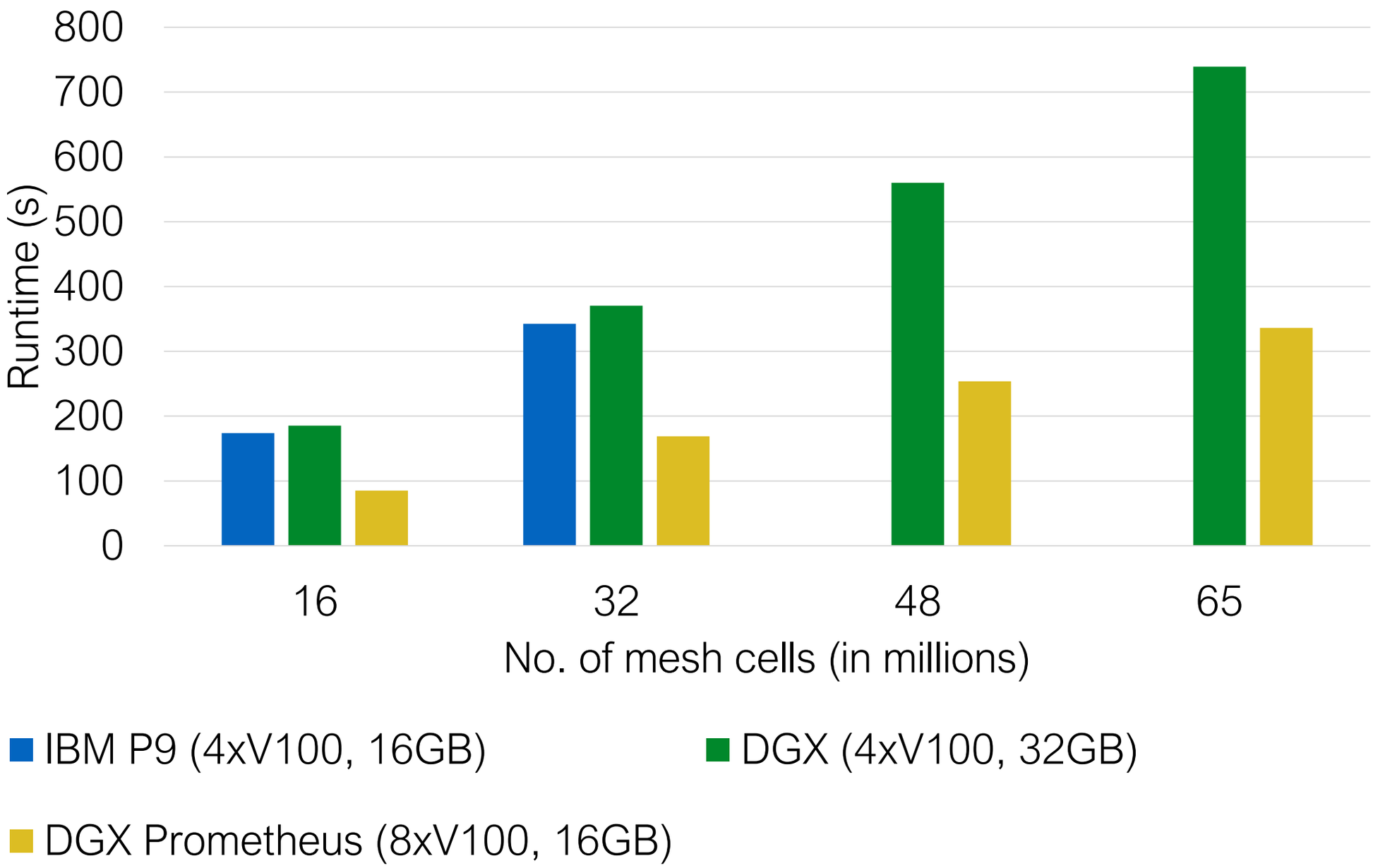
## Benchmarking on one-node: 1000 iterations, 32 million mesh cells




This work was done during GPU hackathon held at IISER Pune, India

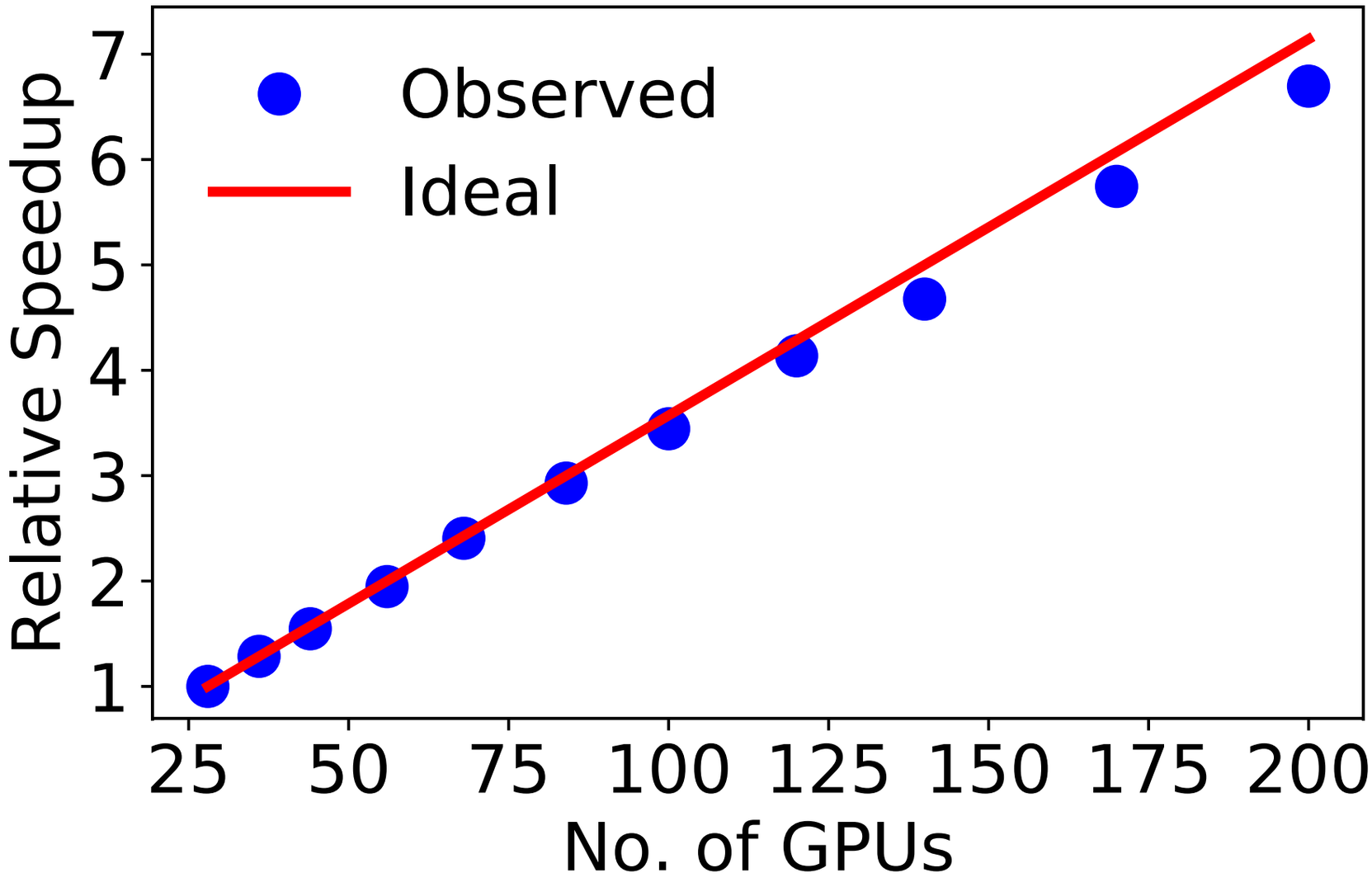
# Performance on V100 GPUs

## Benchmarking on one-node: 1000 iterations



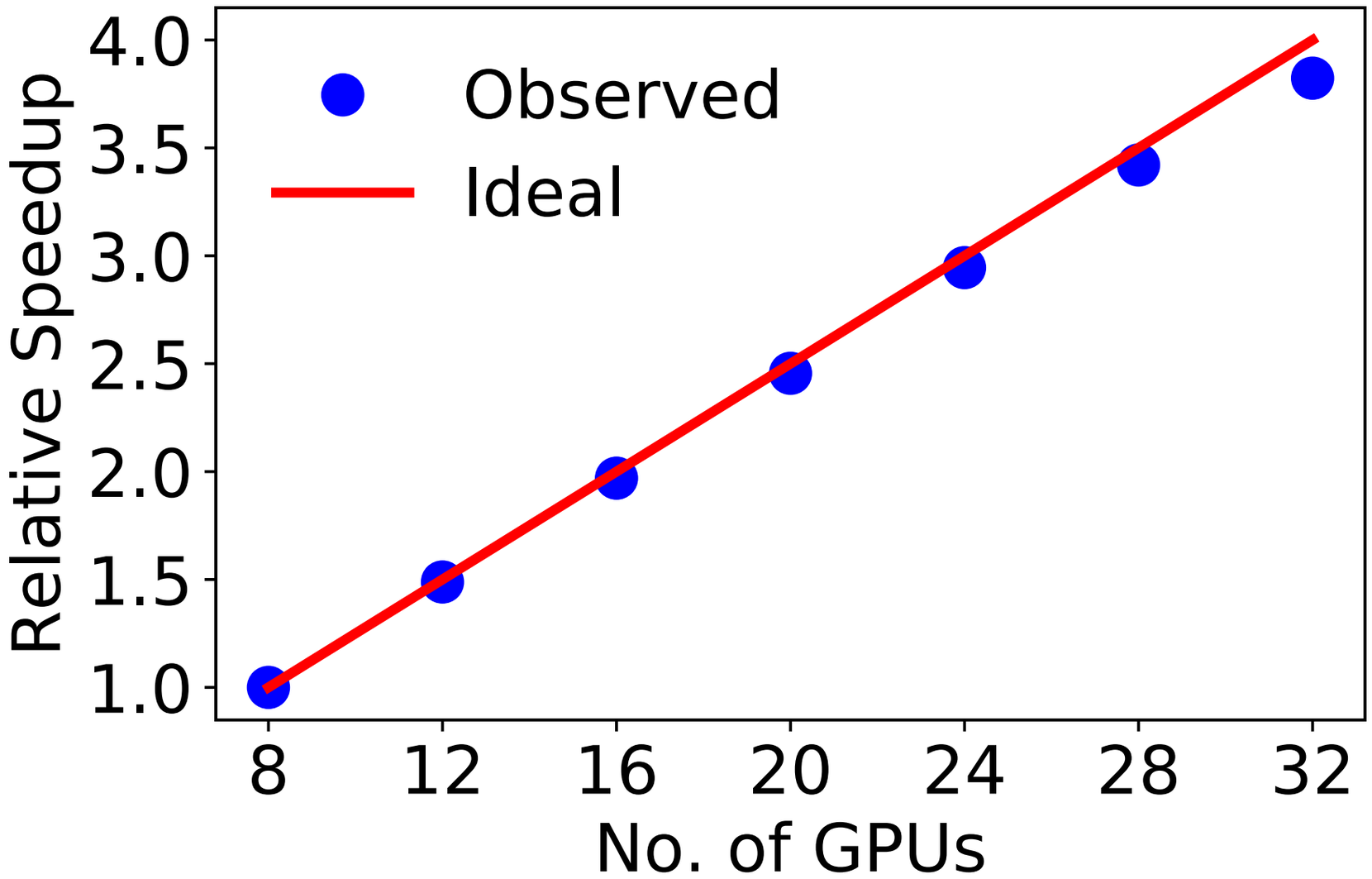
# Strong Scaling on Dhruva Cluster (K20)

47 Million cells, and 1000 iterations



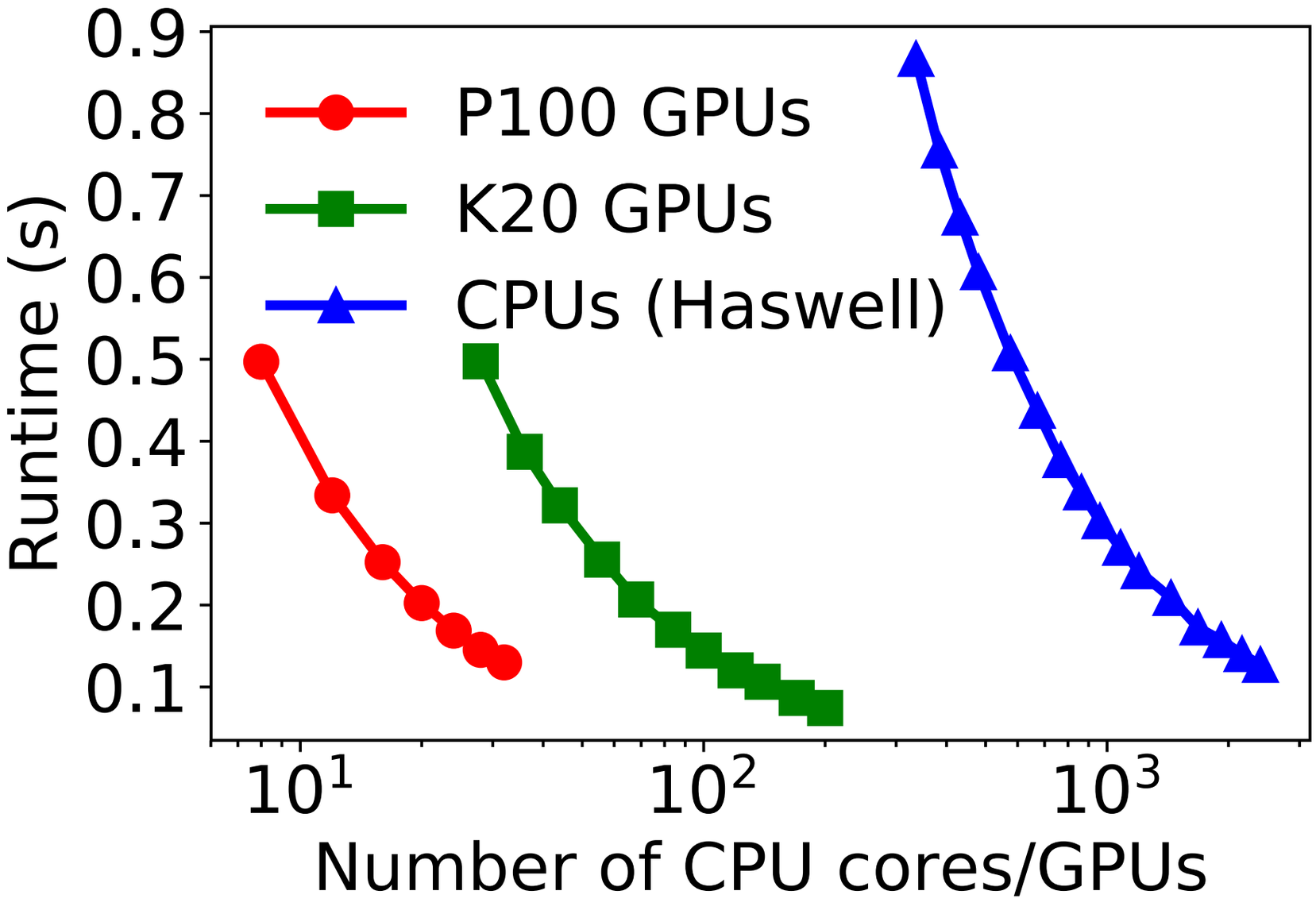
# Strong Scaling on Minsky Cluster (P100)

## 47 Million cells, and 1000 iterations



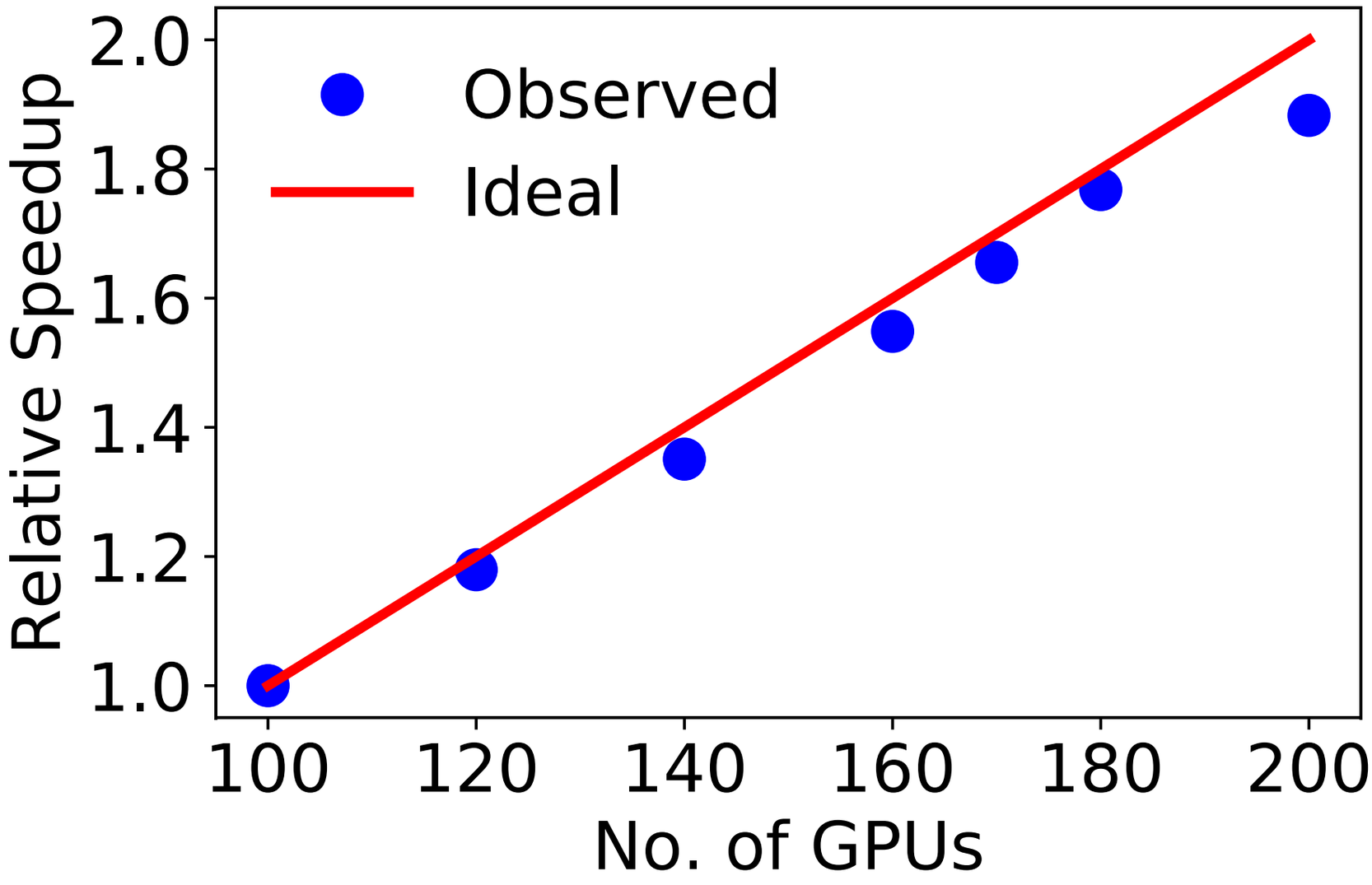
# Comparison of Performance of GPU vs CPU

## 47 Million cells, and 1000 iterations



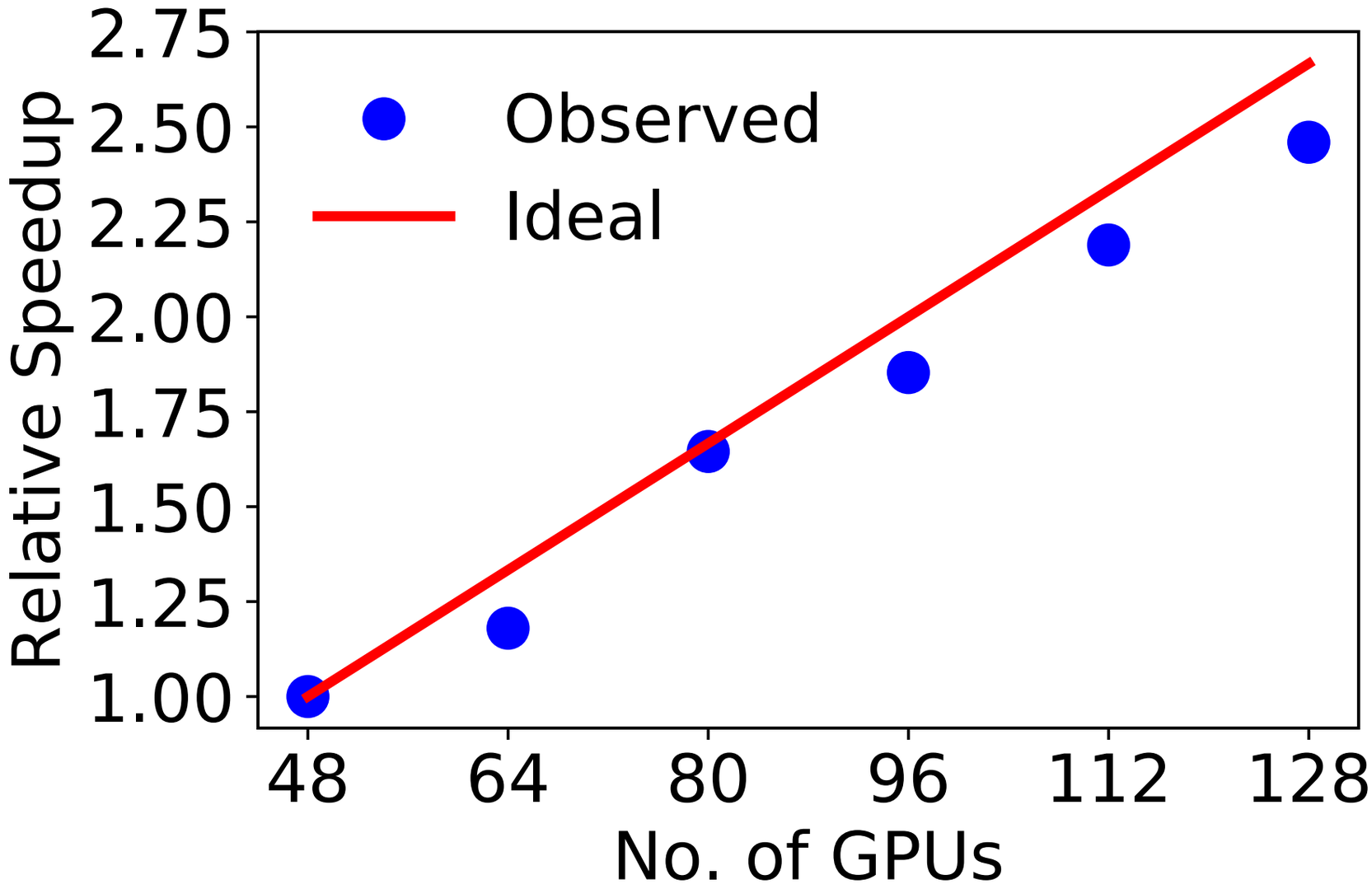
# Strong Scaling on Dhruva (K20)

## 240 Million cells, and 1000 iterations



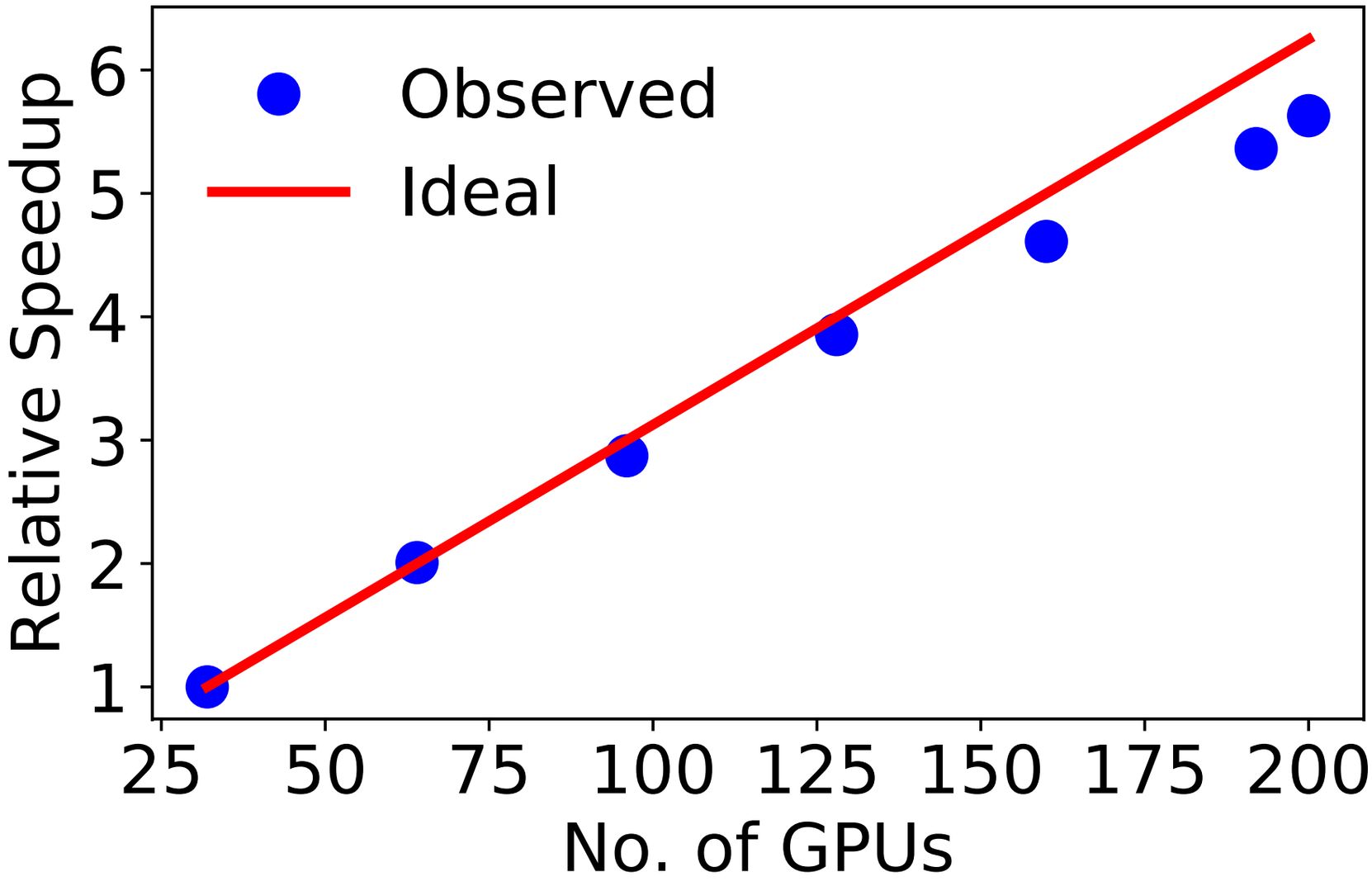
# Strong Scaling on IBM Minsky (P100)

## 240 Million cells, and 1000 iterations



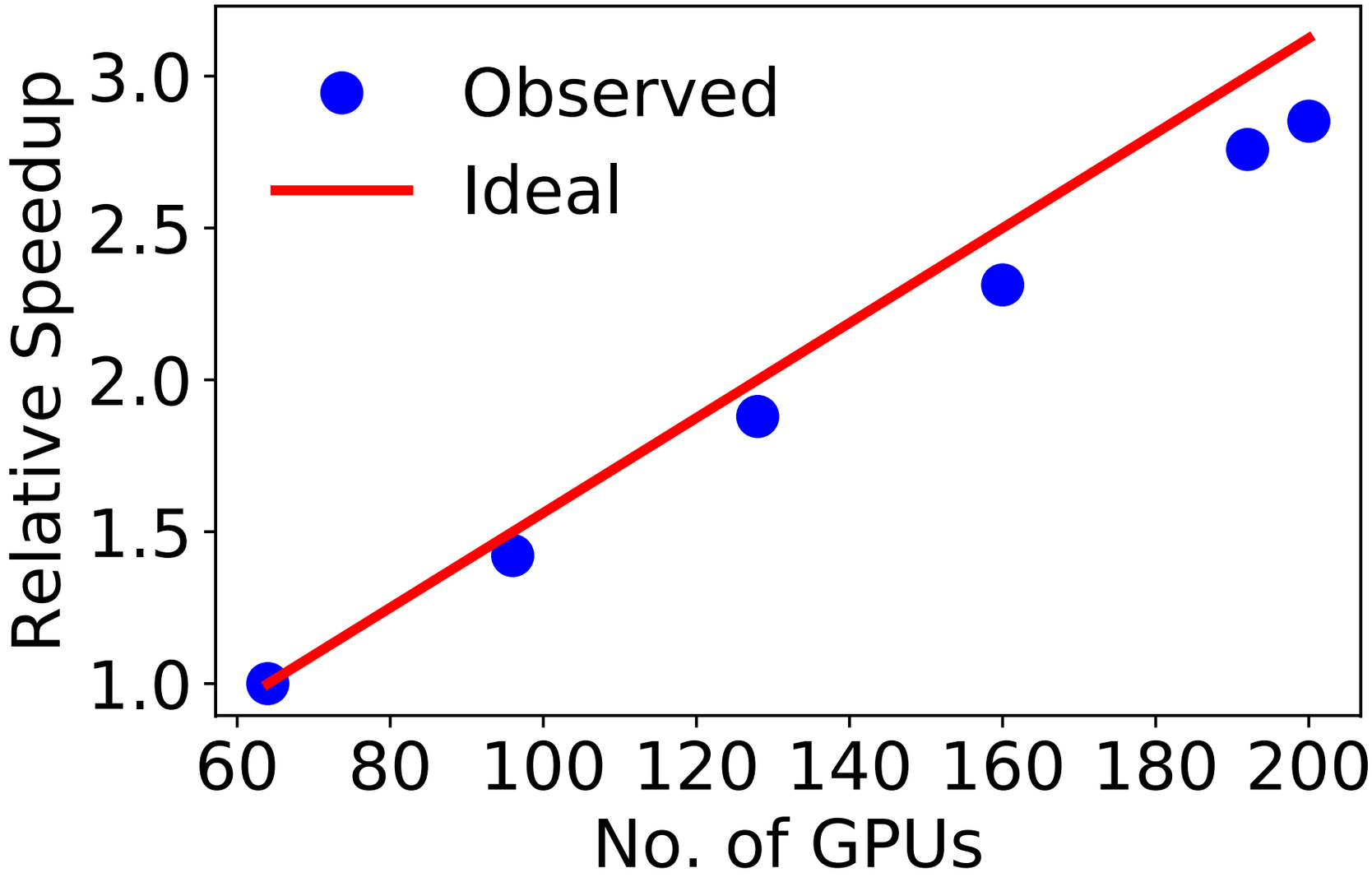
# Strong Scaling on Prometheus (DGX-V100)

## 240 Million cells, and 1000 iterations



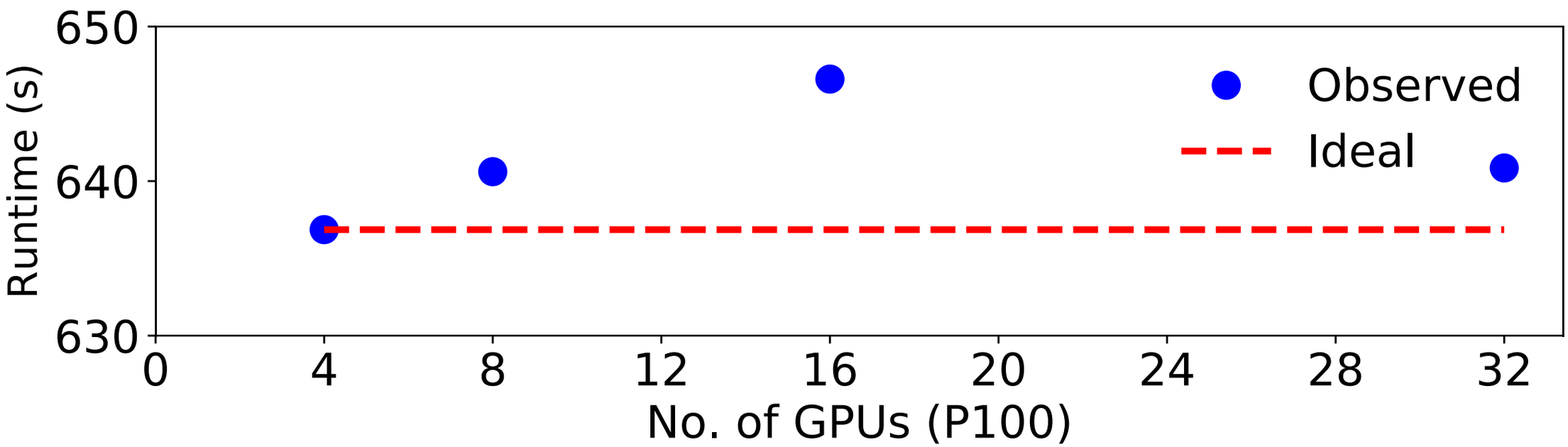
# Strong Scaling on Prometheus (DGX-V100)

## 516 Million cells, and 1000 iterations

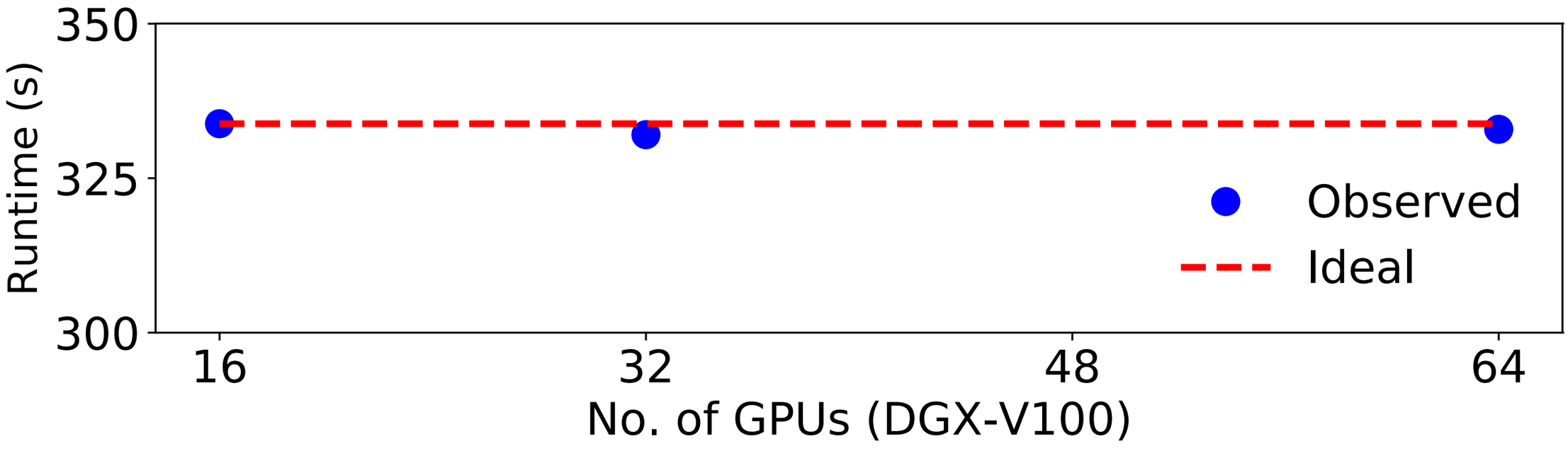


# Weak Scaling

IBM Minsky cluster (P100), 7.5M cells on a GPU

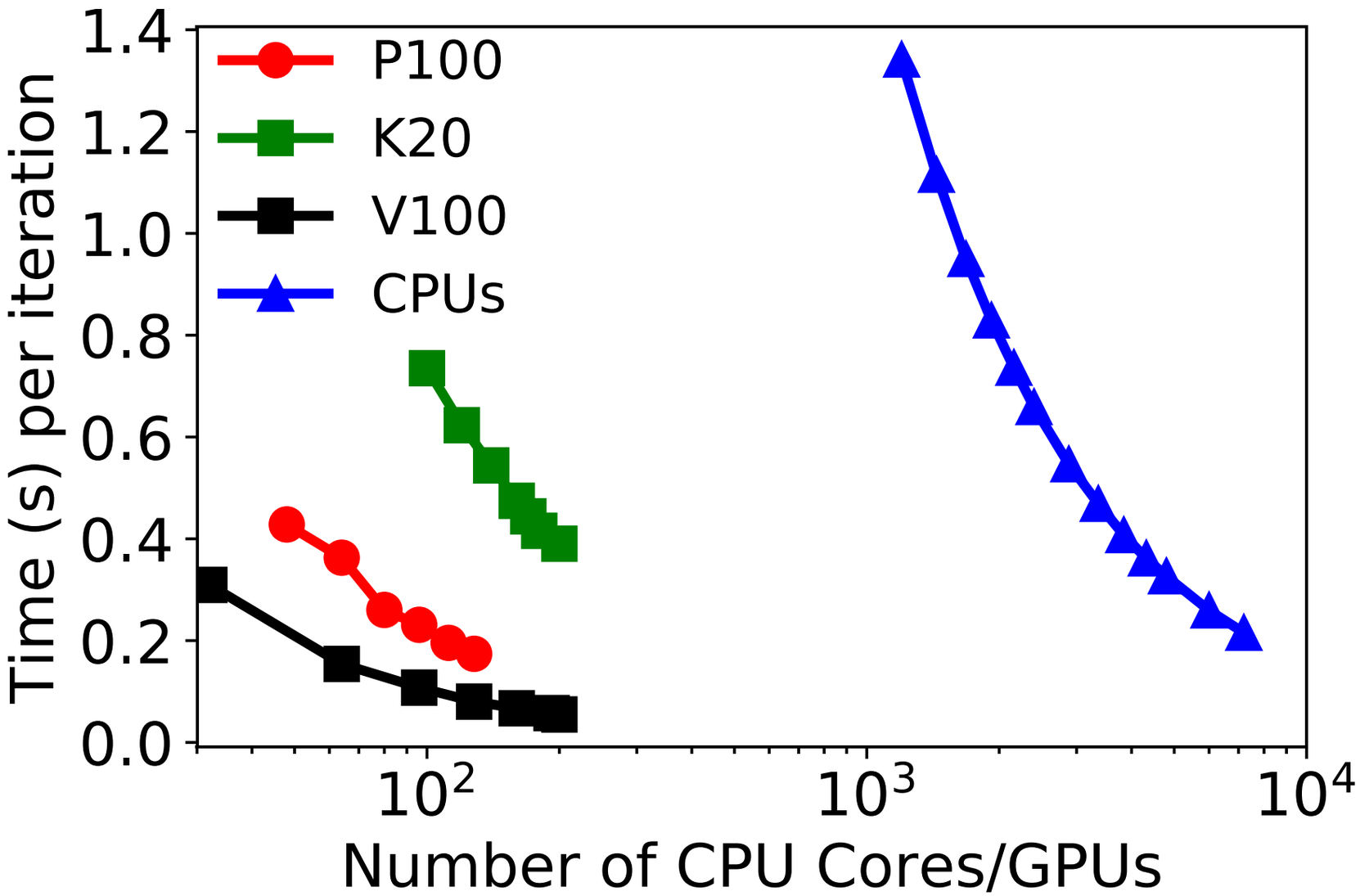



Prometheus cluster (DGX-V100), 8M cells on a GPU



# ANUROOP: Comparison of Performance of GPU vs CPU

## Strong scaling: 240 Million cells, and 1000 iterations

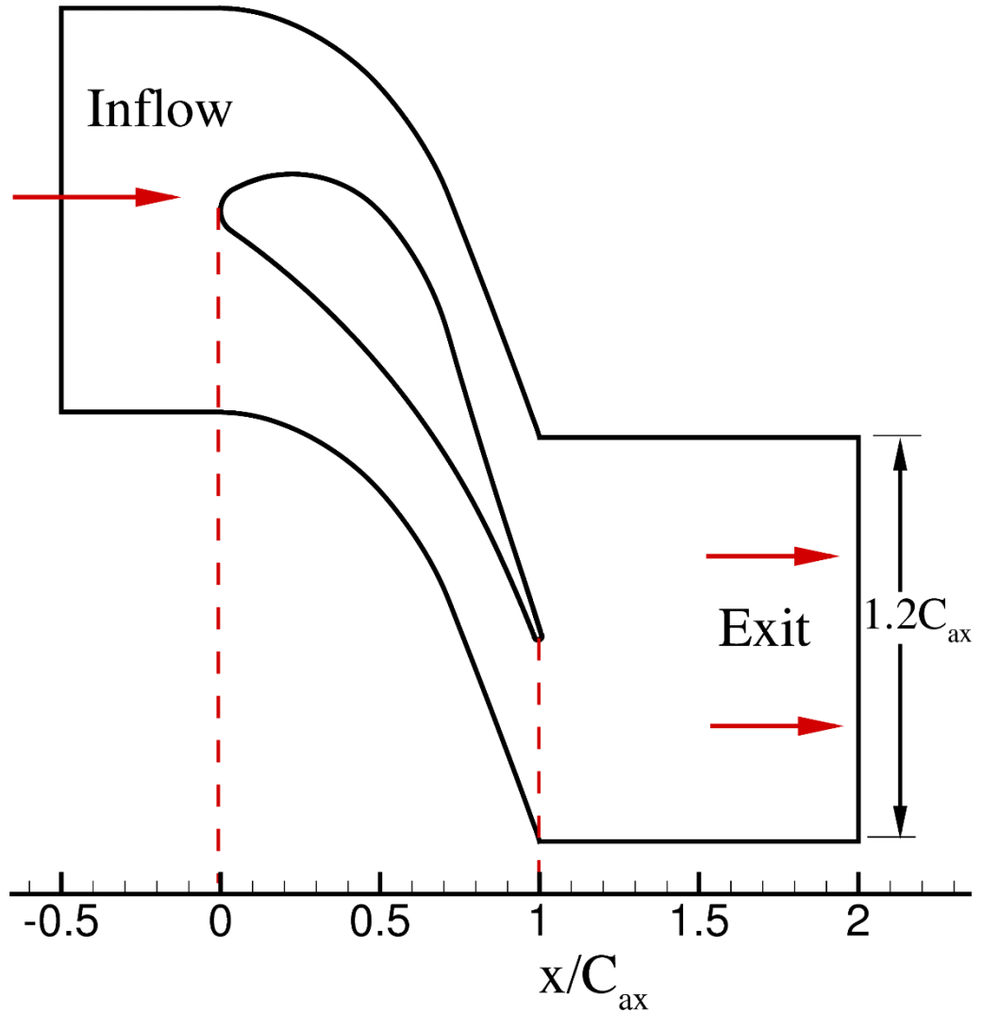


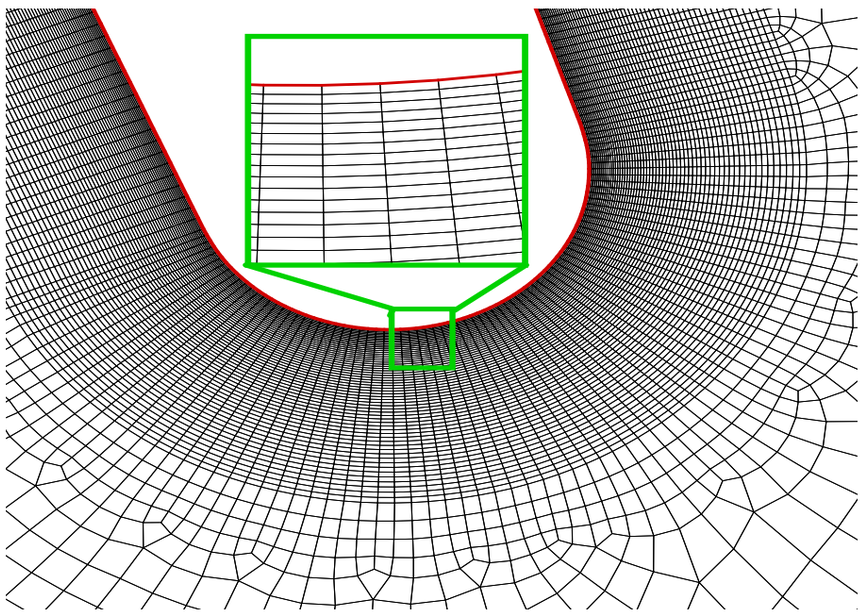

DHRUVA-3: K20 GPU; IBM Minsky: P100;
DGX: V100; SahasraT, IISc: CPUs

# DNS on High Pressure Turbine (HPT) Blade

# Computational Domain for HPT[1]

- 3D stacked mesh
- Flow is 2D in the mean
- Stator blades for STFE
- Compressible NSE
- Unstructured grids
- Boundary layer resolution near wall $Y^+ < 1$
- Used ANUROOP2 for DNS
- Used CFD++, version 16.04, Metacomp Technologies for RANS and LNS



[1]Murthy et al. (2013) ASME Gas Turbine India Conference

[1]Murthy et al. (2013) NPC

[1]Kishore et al. (2017) AeSI CFD Symp.

# Grid Details

| Grid | $N_{2D}$ | $N_z$ | $N_{total}$ ($10^6$) | $\Delta y_n/C_{ax}$ | $\Delta z/C_{ax}$ | $\Delta s^+$ | $\Delta \eta^+$ | $\Delta z^+$ |
|---|---|---|---|---|---|---|---|---|
| SST, RKE | 161169 | -- | 0.16 | 5.86E-5 | 9.04E-3 | 6.2 | 0.17 | -- |
| LNS | 161169 | 22 | 3.54 | 5.86E-5 | 9.04E-3 | 6.2 | 0.15 | 24.11 |
| DNS (94M) | 733841 | 128 | 93.9 | 9.02E-5 | 1.55E-3 | 9.2 | 0.20 | 3.36 |
| DNS (129M) | 2017152 | 64 | 129 | 6.76E-5 | 3.109E-3 | 2.1 | 0.15 | 6.72 |
| DNS (258M) | 2017152 | 128 | 258 | 6.76E-5 | 1.55E-3 | 2.1 | 0.15 | 3.36 |
| DNS (516M) | 2017152 | 256 | 516 | 6.76E-5 | 0.77E-3 | 2.1 | 0.15 | 1.68 |

Here, $N_{2D}$ = Total number of elements on 2D face; $N_z$ = no. of intervals in span-wise direction; $\Delta y_n/C_{ax}$ = height of the first cell normal to the blade surface; $\Delta z/C_{ax}$ = width of the cells along the span-wise direction; $\Delta s^+$, $\Delta \eta^+$ and $\Delta z^+$ are the maximum distances on suction side of blade surfaces measured in wall units along the stream-wise, normal and span-wise directions respectively.

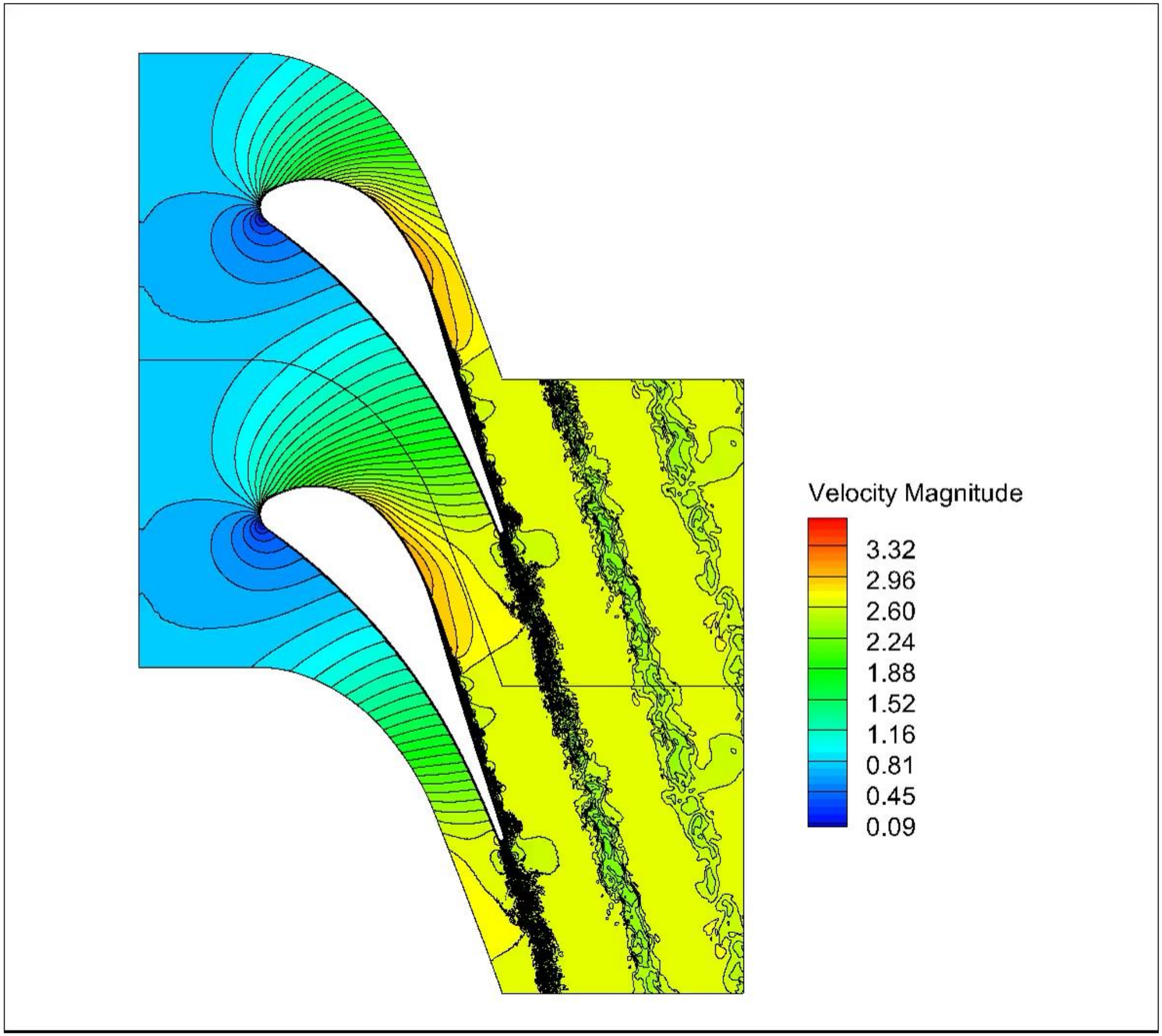
# Test Case: Experimental Flow Condition[1]

| Parameter | | Inlet (1) | Exit (2) |
|---|---|---|---|
| Reynolds number | *Re* | 152,000 | 486,000 |
| Mach number | *M* | 0.16 | 0.593 |
| Total pressure (kPa) | $P_0$ | 117.750 | 116.777 |
| Static pressure (kPa) | P | 115.657 | 92.072 |
| Static temperature (K) | *T* | 298 | - |
| Angle of Attack | AoA | 0 | 72.1* |

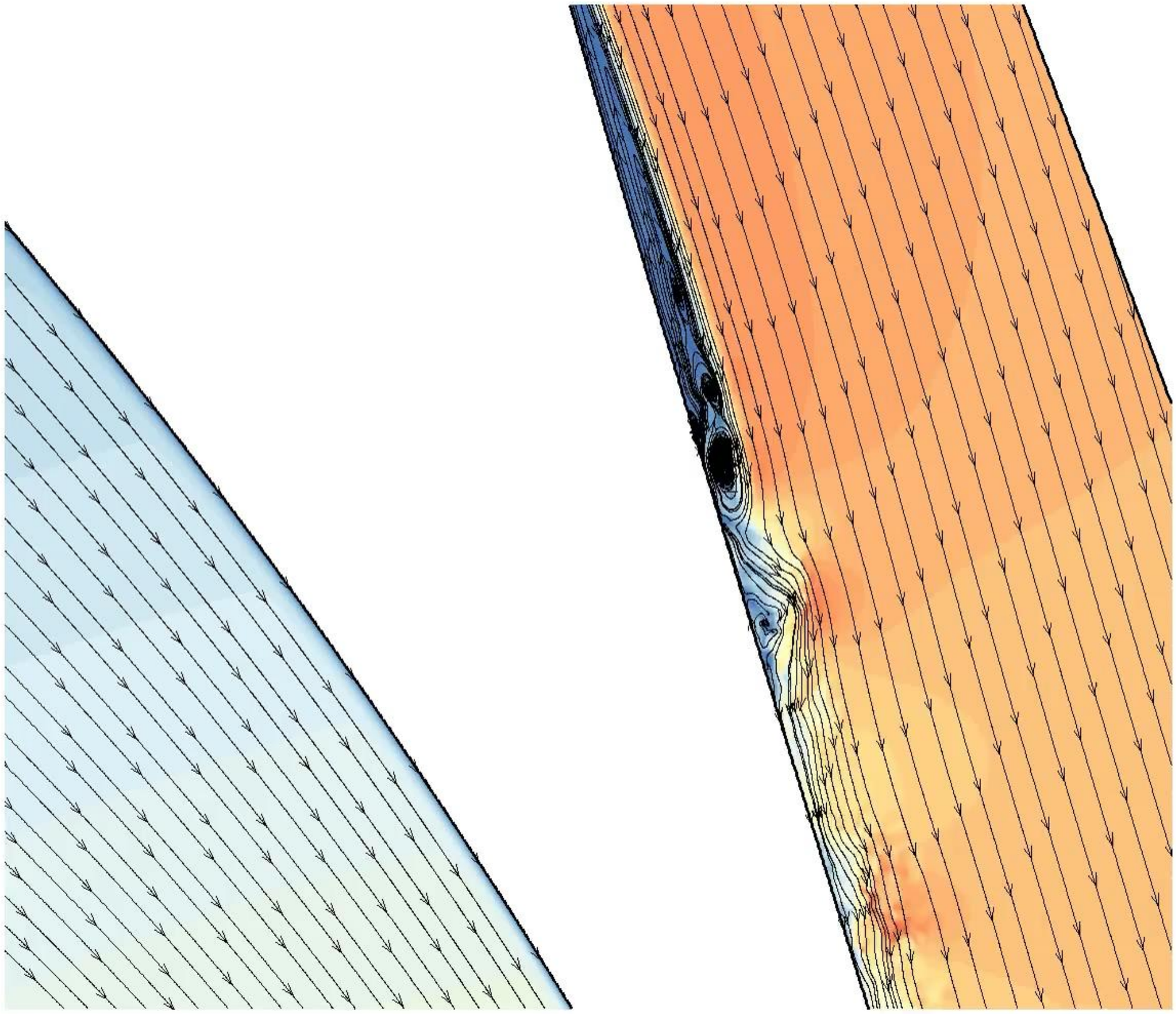
* Flow inclination angle

Data from: [1]Kumaran et al. (2014) NAL report

# Flow Past HPT Blades



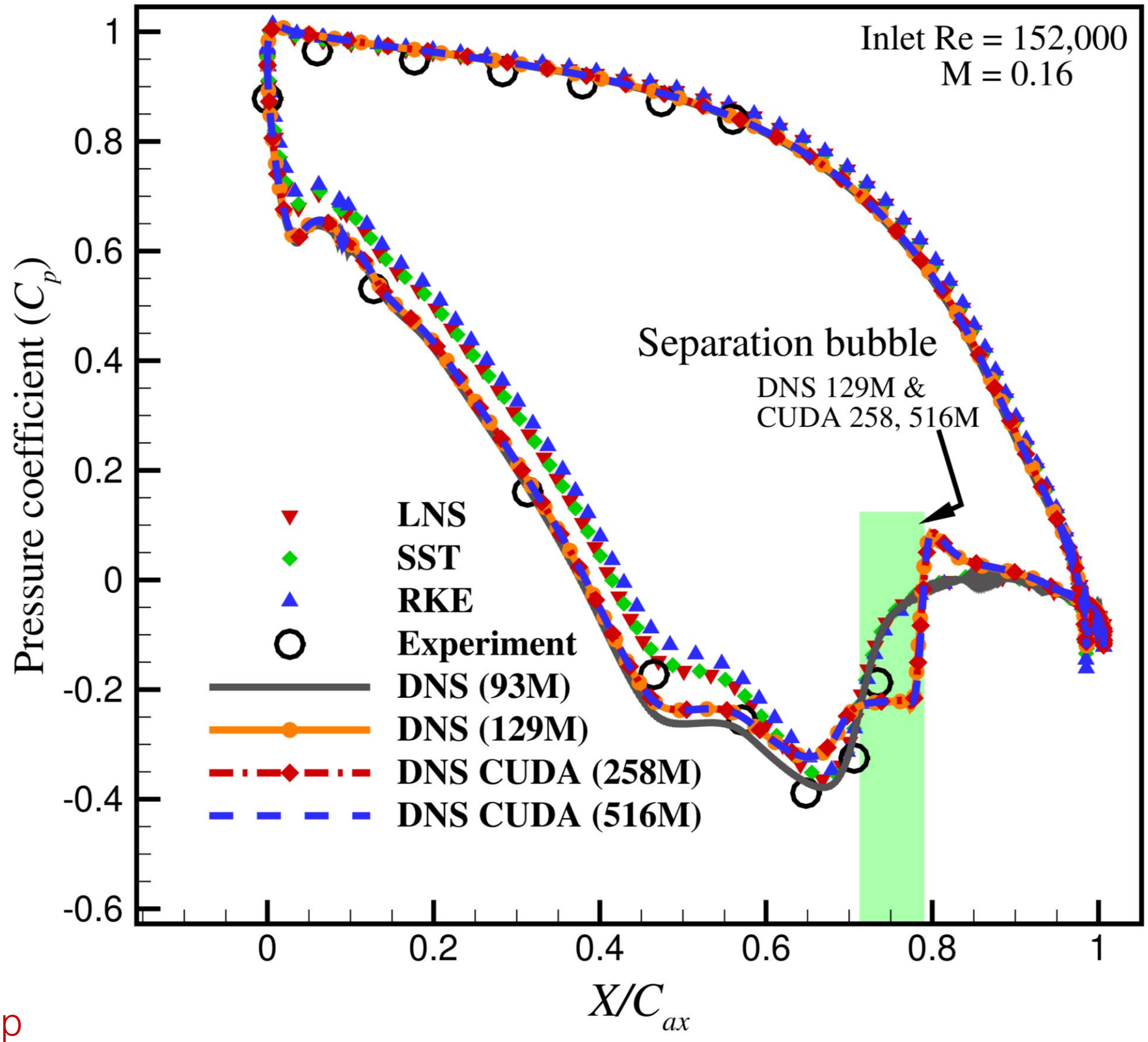
# Flow Past HPT Blades: Flow Separation



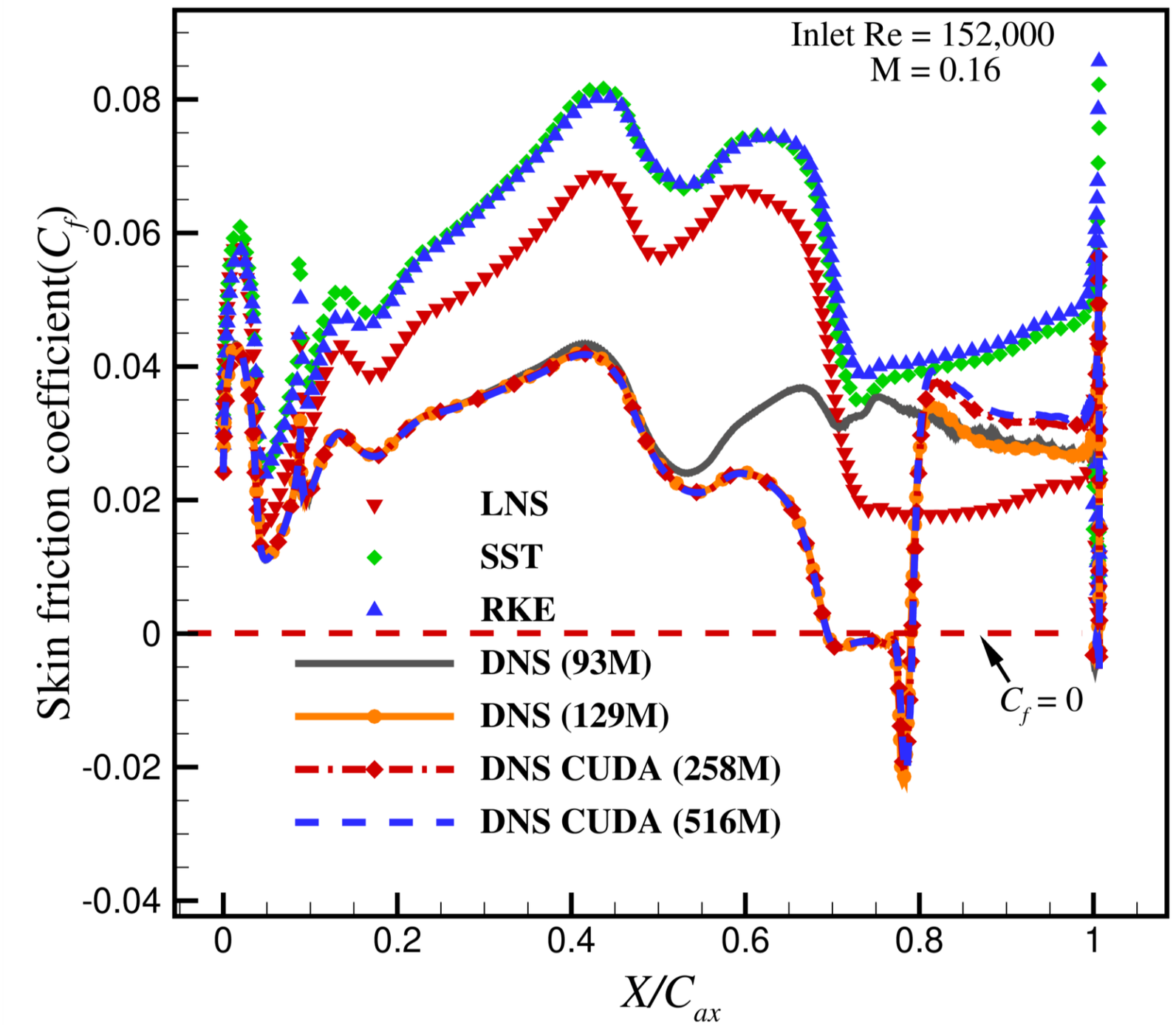
# Coefficient of Pressure, $C_p$

$$C_p = \frac{p - p_2}{p_{01} - p_2}$$

- DNS at $93.9 \times 10^6$ Mesh cells
- LNS and SST are close to each other, and to both DNS and experimental results, compared to RKE



# Skin Friction Coefficient ($C_f$, Suction Side)




LNS is closer to but higher than DNS till $0.7C_{ax}$ and lower beyond this point, while the SST is closer beyond $0.7C_{ax}$

# Assessment of GPUs in Comparison with CPUs

These numbers are (rough estimates) computed (by extrapolating the benchmark data obtained by using 240M cell simulation) for DNS of flow past STFE blade with 1B mesh cells.

| Parameters | Value |
|---|---|
| Number of cells | $10^9$ |
| Approximate number of P100 GPU nodes required | 29.0 |
| Number of P100 nodes required (roundup) | 30 |
| Approximate number of CPU nodes required | 425.92 |
| Number of CPU nodes required (roundup) | 426 |
| Ratio of CPU cluster cost to GPU cluster cost | 2.84 |
| Total power required in kW for GPU cluster | 66 |
| Total power required in kW for CPU cluster | 255.6 |
| Ratio of CPU power to GPU power | 3.87 |

# Conclusions

- ANUROOP on a single P100 GPU was found to be approximately 110x faster compared to single CPU core
- ANUROOP on single P100 is 2.7x faster compared to K40 and 3.1x faster compared to one node on Cray XC40 (SahasraT, IISc)
- Approximately 2x faster on V100 compared to P100 (Power8)
- It has been scaled to 500TF (with 95% utilization) on Minsky.
- Linear scaling has been demonstrated up to 220 K20,128 P100 and 200 V100 GPUs

# Conclusions

- For our problem, GPU-based cluster costs about 0.3 times less and requires 0.25 times less power, than a CPU-based cluster for similar performance
- Based on our experience with ANUROOP2 (on several GPU clusters), recommend use of GPU-based heterogeneous supercomputers for HPC in CFD. They can provide results for bigger problems (e.g. at higher ***Re***) in more realistic time frames. Simulation of flow past gas turbine blades at more practical ***Re*** seems to be getting closer to reality

# Acknowledgements


- Work reported here was supported by three research grants (two by GTRE and one by ANURAG)
- NVIDIA, Bangalore
- IBM, India for providing access to Minsky and V100 GPU on Power9
- SERC, IISc, Bangalore for providing access to SahasraT supercomputer
- CSIR 4PI for providing access to Ananta cluster

# Thank you