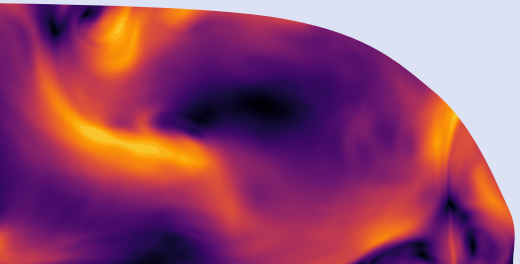
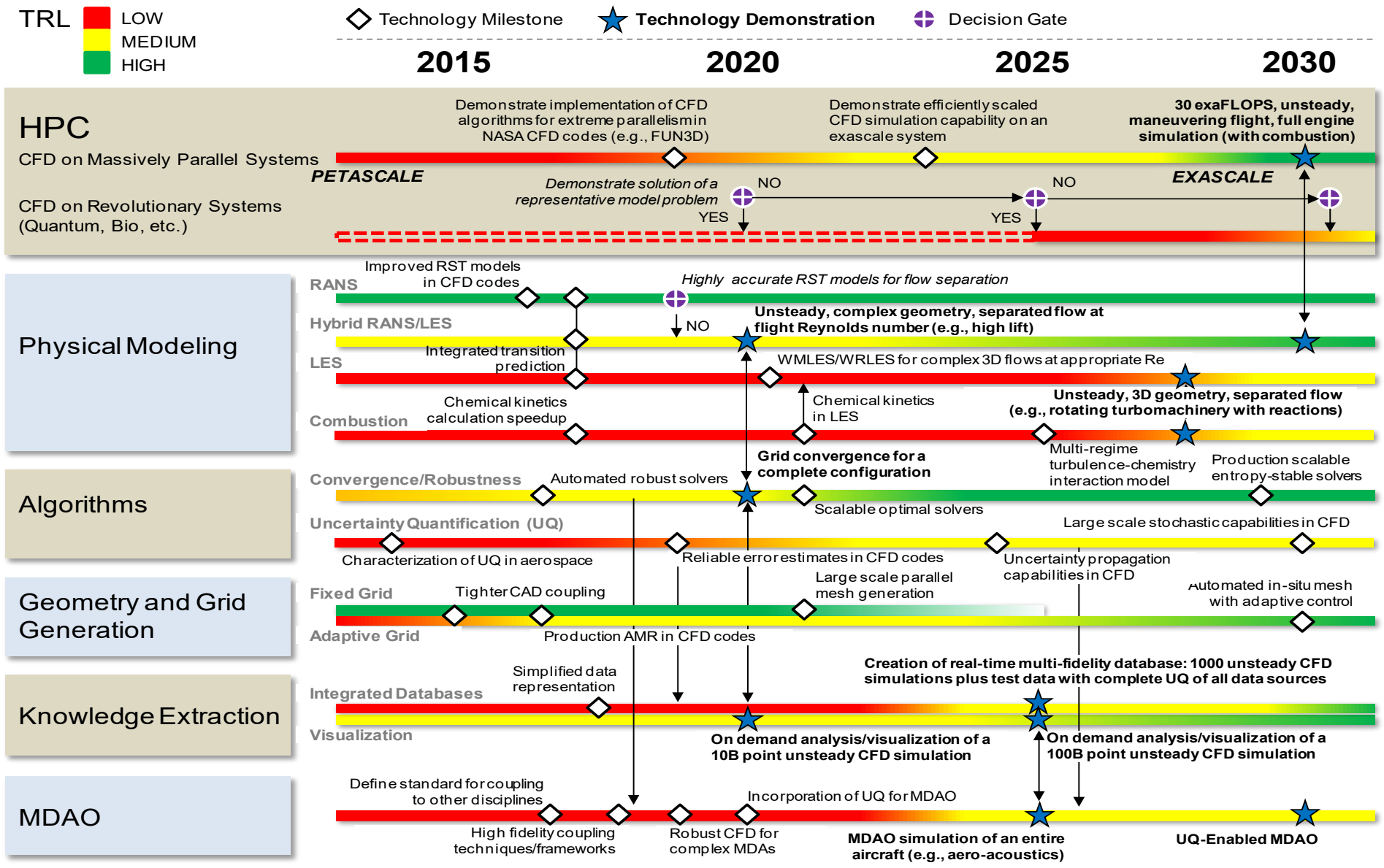
Variational Tensor Network Methods for Nonlinear Partial Differential Equations

Dieter Jaksch, University of Hamburg
and University of Oxford



Göttingen 21. November 2025

CFD Vision 2030 Research Roadmap



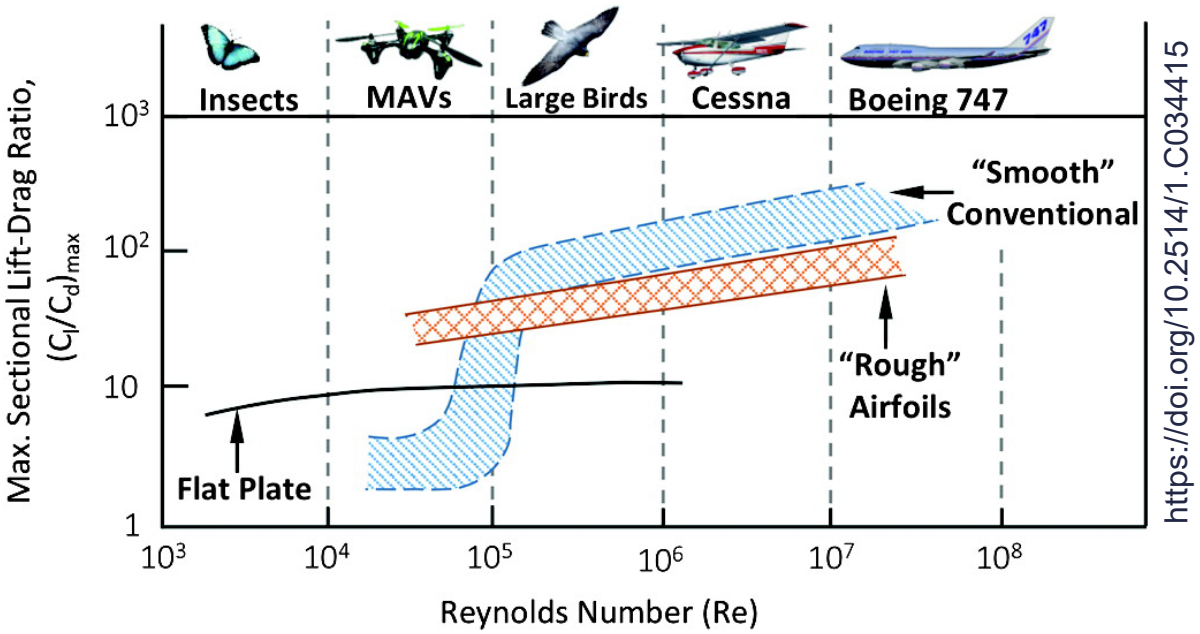
"There is steady progress on fabricating practical quantum computers, and such systems may be available in 2030. However, while a quantum computer can be used for some linear algebra calculations ..., a quantum computer is not necessarily a faster computer for CFD calculations." (CFD Vision 2030 Study Report)

Kolmogorov microscale and Reynolds number

- A simulation of fluid flow needs to cover a wide range of length scales
  - L the size of the largest eddies in the flow
  - η the Kolmogorov length scale at which eddies are dissipated into heat
- The ratio of these two length scales is the Reynolds number defined as

$$\mathrm{Re} = \left(\frac{L}{\eta}\right)^{4/3} = \frac{vL}{\nu}$$

- Here v is the speed of the flow and ν the kinematic viscosity
- Flows become turbulent when Re is greater than a couple of thousands
- Grid based methods typically scale with $\mathrm{Re}^{3K/4}$



https://doi.org/10.2514/1.C034415

wikipedia

$\mathrm{Re} \approx 10^9$

newscientist.com

$\mathrm{Re} \approx$ very large

The challenge – Resolving a wide range of length-scales

- Typically, the largest eddy size L is

$$10\text{m} \leq L \leq 100\text{m}$$

- Resolution up to the Kolmogorov length scale η, typically

$$0.1\text{mm} \leq \eta \leq 10\text{mm}$$

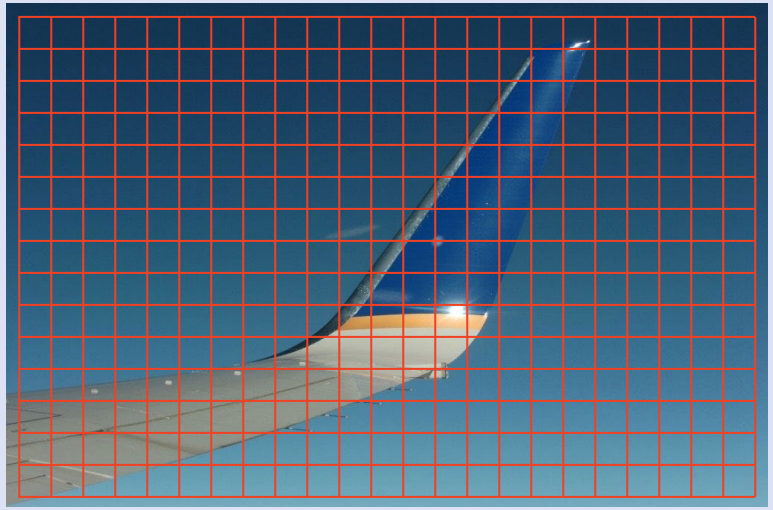
- Necessary Grid Points N

$$1\,000\,000\,000 \leq N \leq 1\,000\,000\,000\,000\,000\,000$$

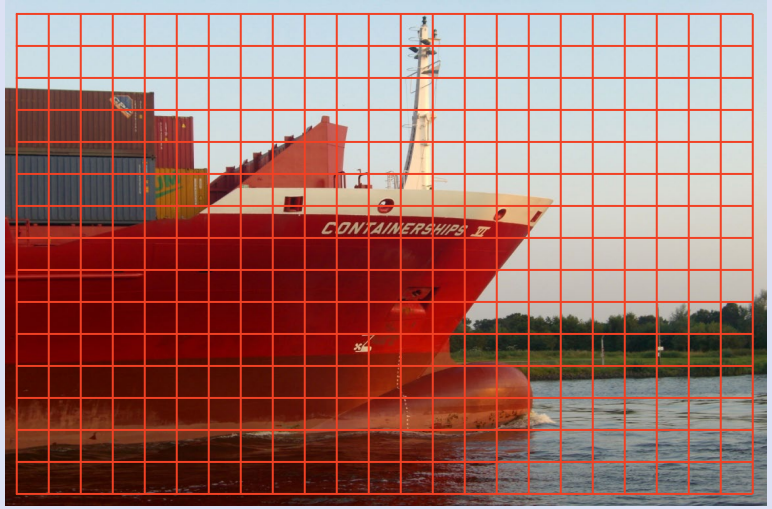
- Necessary memory M for storing a single velocity component

$$8\ \text{Mbyte} \leq M \leq 8\,000\ \text{Pbyte} = 8\,000\,000\ \text{Tbyte}$$

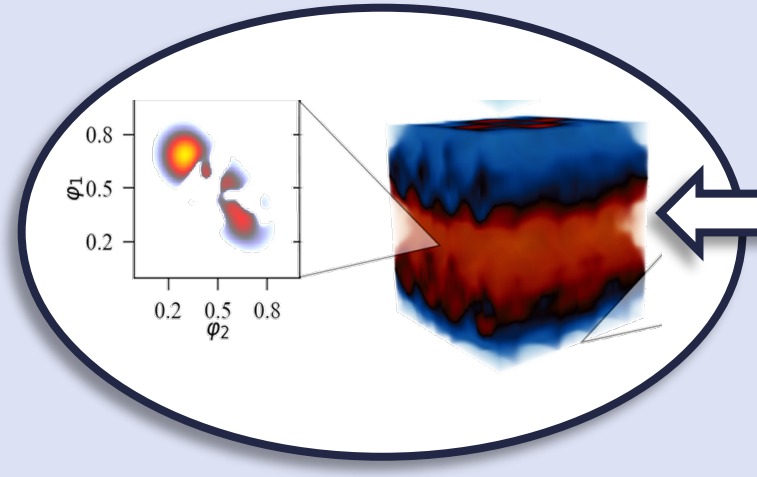
- Additionally, very small time steps are often necessary
- one usually works with approximations
  - Can work very well – especially with known configurations
  - Unfortunately, they also deliver qualitatively incorrect results



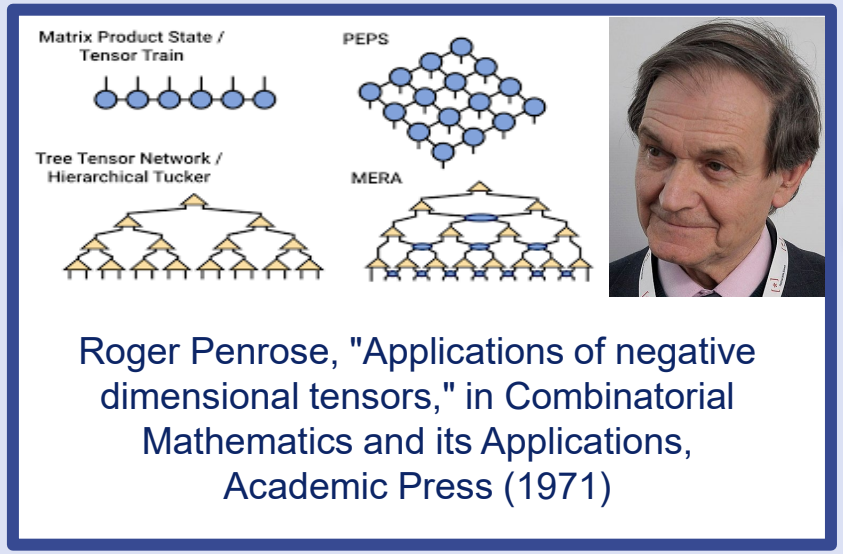
Winglets

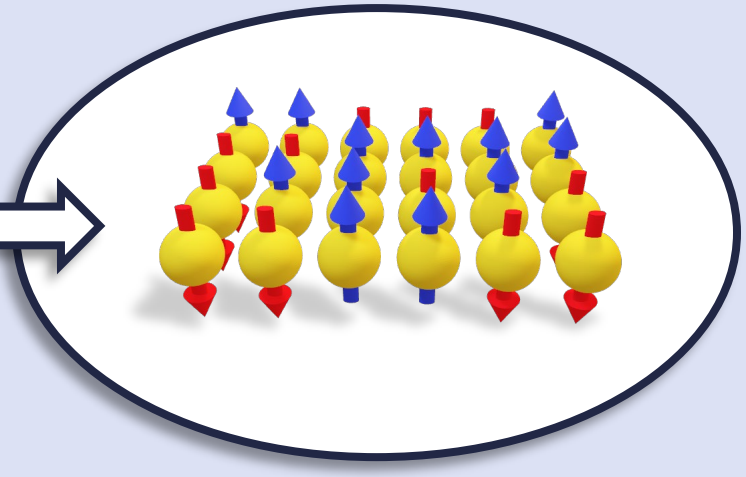


Bulbous bows



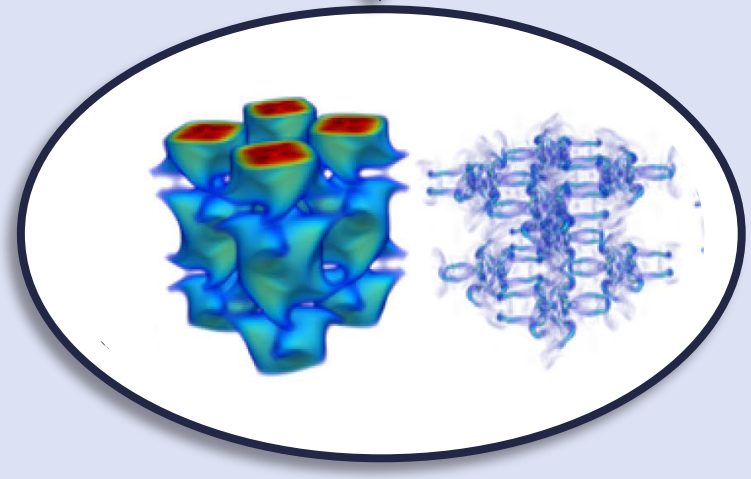
Combustion physics



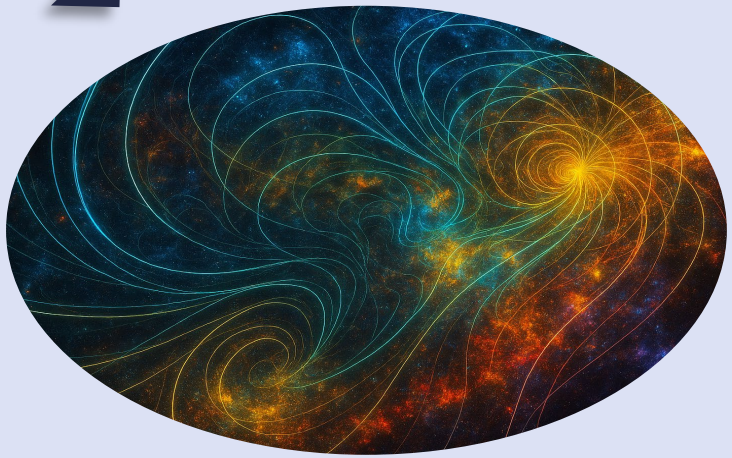


Quantum Simulation

Quantum industrial design



Fluid Flows



Plasma physics

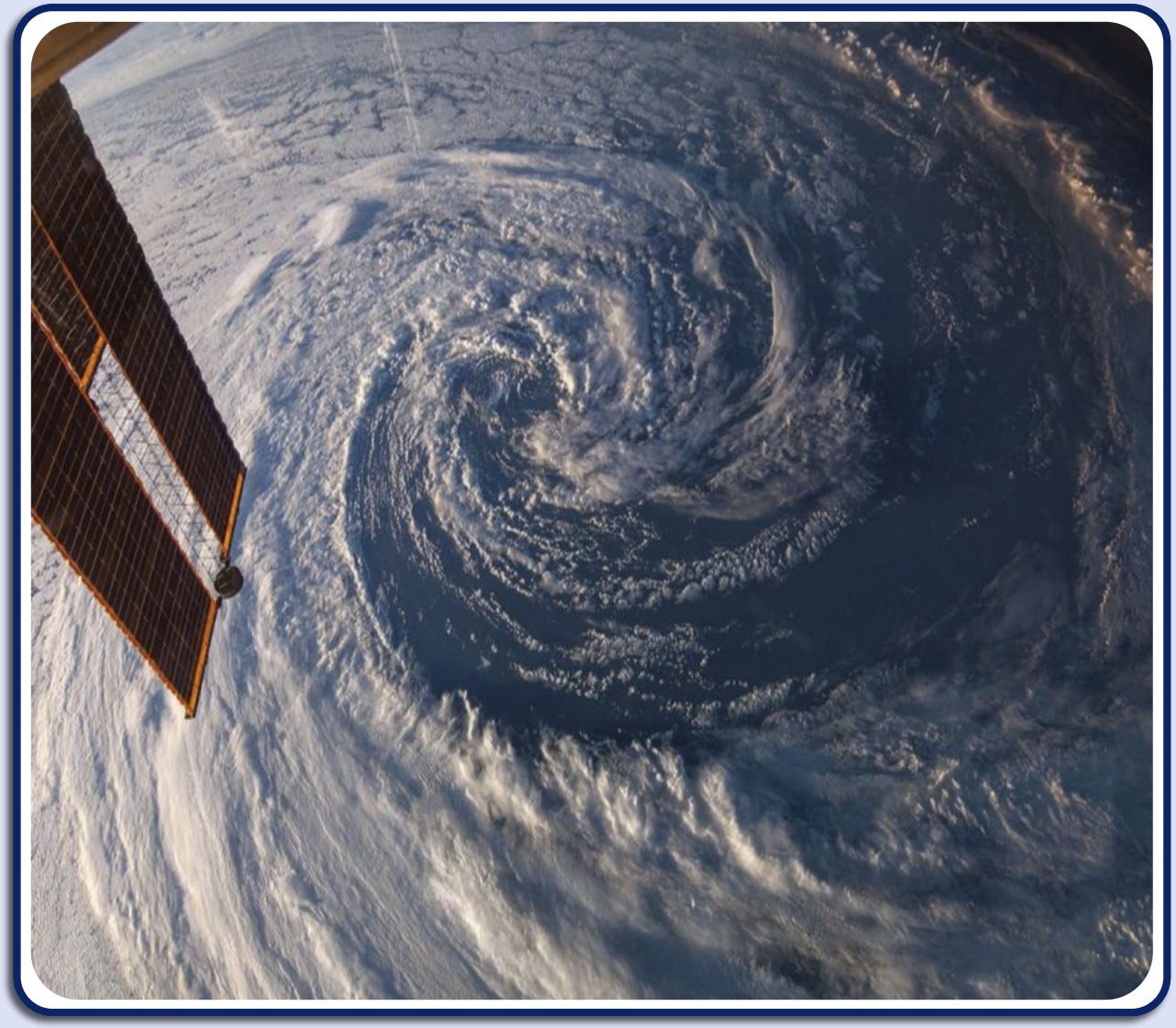
Incompressible Navier-Stokes equation

- We solve the 2D and 3D equations for simple fluid flows

$$\frac{\partial \vec{v}}{\partial t} = -(\vec{v} \cdot \nabla)\vec{v} - \nabla p + \frac{1}{\mathrm{Re}} \nabla^2 \vec{v}$$

$$\nabla \cdot \vec{v} = 0$$

- 2D → weather forecast
- 3D → aerodynamics, combustion physics, …



Outline

MPS

- Encoding grid functions
- Classical entanglement spectrum
- Turbulence correlations

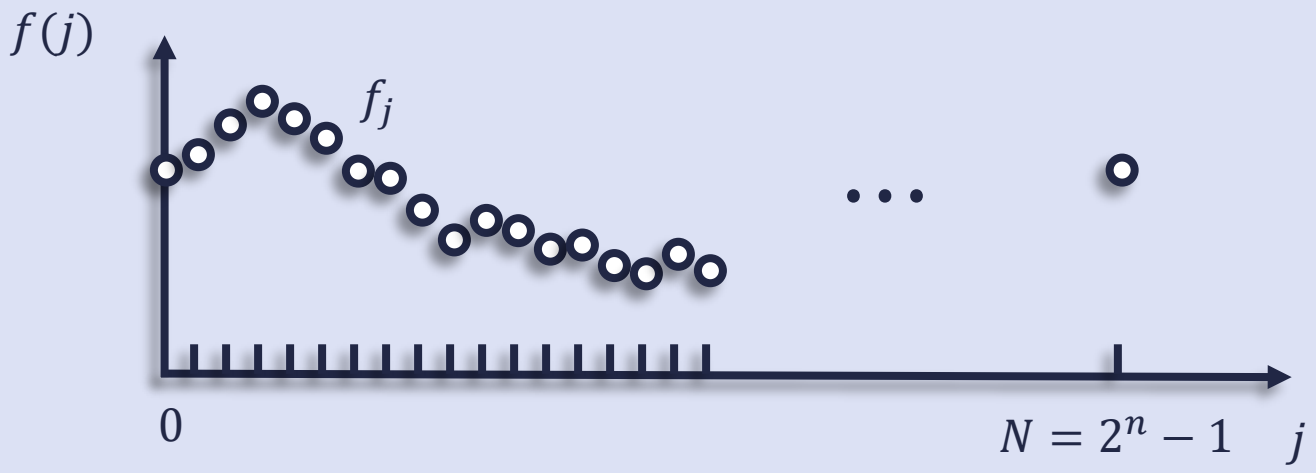
CFD Examples

- MPS algorithm
- Jet formation and Taylor Green vortex
- Driven cavity
- Magnus effect

Hybrid Optimization

- Hardware architecture
- Quantum network
- Generic cost function
- Proof of Principle Example

Amplitude encoding of discrete functions



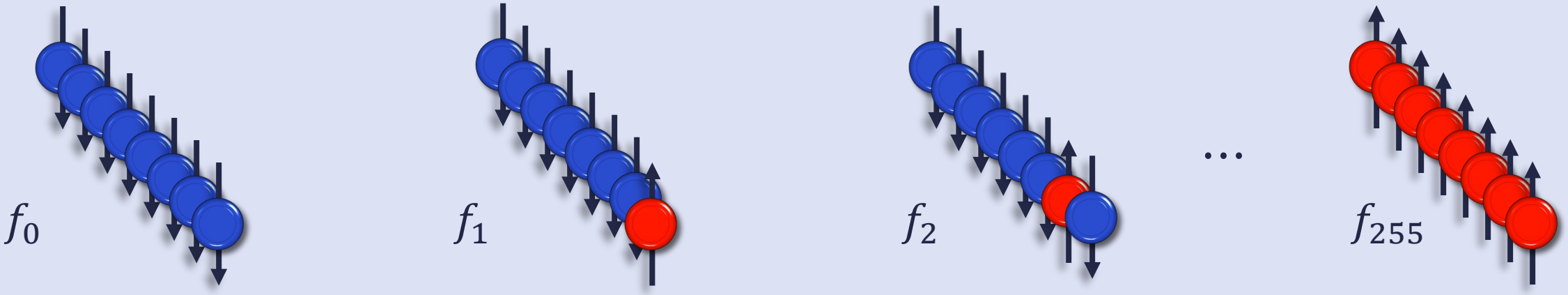
Quantum superposition

$$|\psi\rangle = \sum_i f_i |\vec{\imath}\rangle$$

n-qubit register stores 2^n function values f_j

Map $j \to \vec{j} = \mathrm{binary(j)}$ for $n = 8$:

$j = 0 \to \vec{j} = 00000000$ $\quad$ $j = 1 \to \vec{j} = 00000001$ $\quad$ $j = 2 \to \vec{j} = 00000010$ $\quad$ ... $\quad$ $j = 255 \to \vec{j} = 11111111$



MPS – Classical Entanglement Spectra

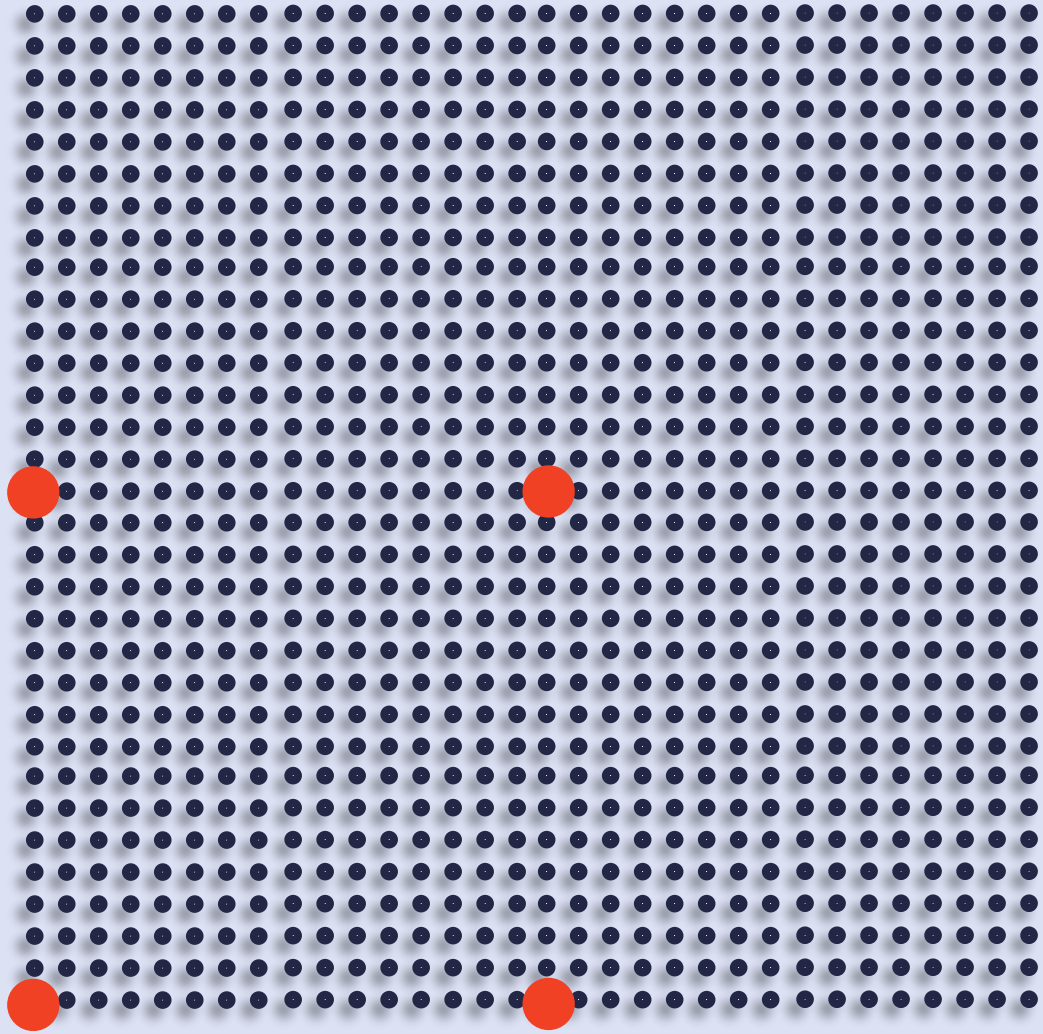
- Consider a scalar field $u(\boldsymbol{r}_q)$ on a $2^N \times 2^N$ of size $L \times L$.
- We decompose the field into functions on a coarse $L/2$ grid (red dots $\boldsymbol{X}_k$) and a fine grid (black dots) as

$$u(\boldsymbol{r}_q) = \sum_{\alpha=1}^{\chi(1)} \lambda_\alpha R_\alpha(\boldsymbol{X}_k) f_\alpha(\boldsymbol{x}_l) \quad \text{where} \quad \mathbf{r}_\mathrm{q} = \mathbf{X}_\mathrm{k} + \mathbf{x}_\mathrm{l}$$

- The maximum number of terms in this sum is

$$\chi_{\max}(1) = 4$$

- The actually required number of terms $\chi(1)$ in this sum is the so-called Schmidt number. It is a measure of correlations between $L/2$ and other length scales.
- The terms in the sum are weighted by λ_α which is the entanglement spectrum.



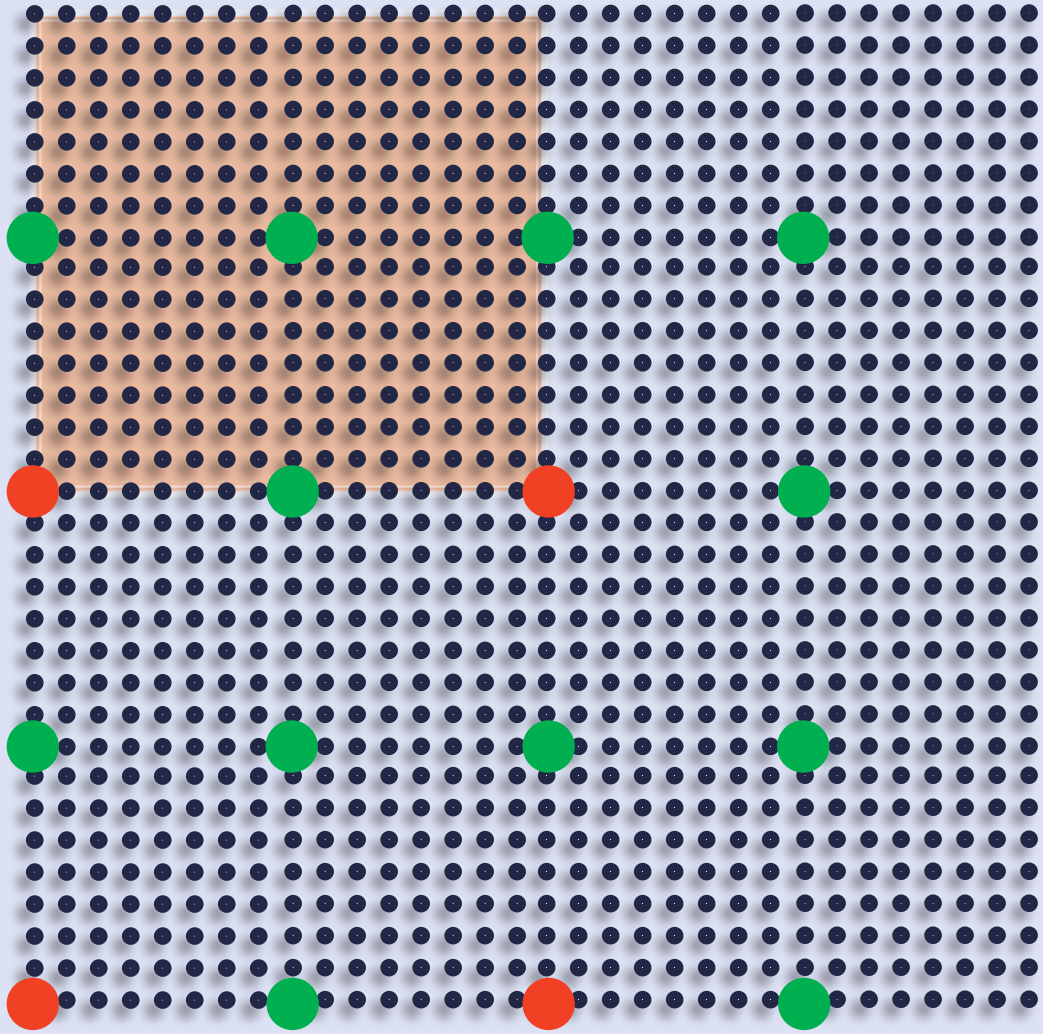
MPS – Classical Entanglement Spectra

- We repeat this decomposition to get correlations between neighbouring length scales.
- For instance, for correlations between the length scale $L/4$ and lengths scale $L/8$ we decompose each of the functions $f_\alpha(x_l)$ from before.
- This gives a representation of the field as

$$u = \sum_{\alpha=1}^{\chi(1)} \lambda_\alpha R_\alpha \sum_{\beta=1}^{\nu} \lambda_\beta R_{\alpha\beta} f_{\alpha\beta}$$

- The maximum Schmidt number $\chi(2)$ is the total number of terms in these sums

$$\chi_{\max}(2) = 4^2 = 16$$



MPS – Classical Entanglement Spectra

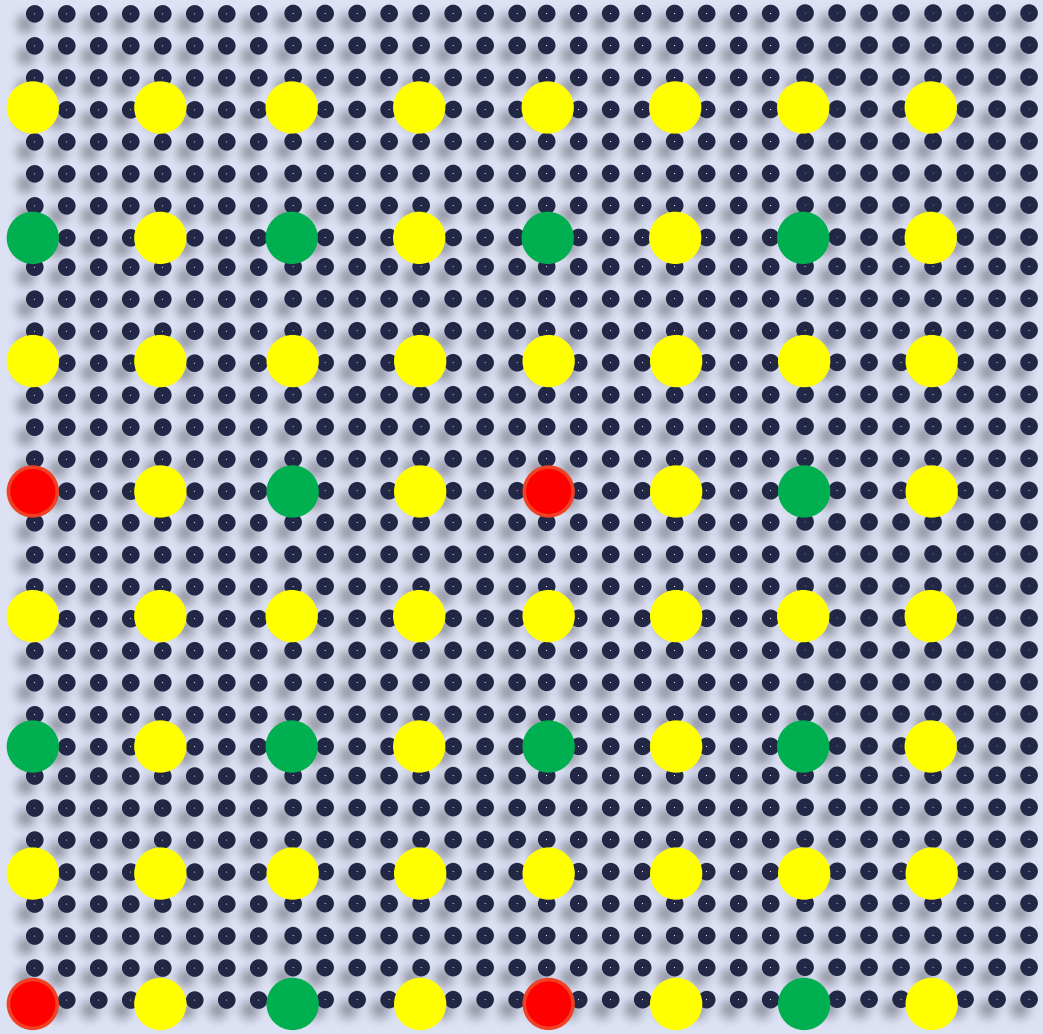
- In general, Schmidt numbers $\chi(n)$ and $\lambda_\alpha^{(n)}$ characterize the amount of correlations between length scales

$$L \times 2^{-n} \quad \text{and} \quad L \times 2^{-n-1}$$

- The repeated application of Schmidt decompositions with increasing n gives a compact Matrix-Product-State (MPS) representation of the scalar field

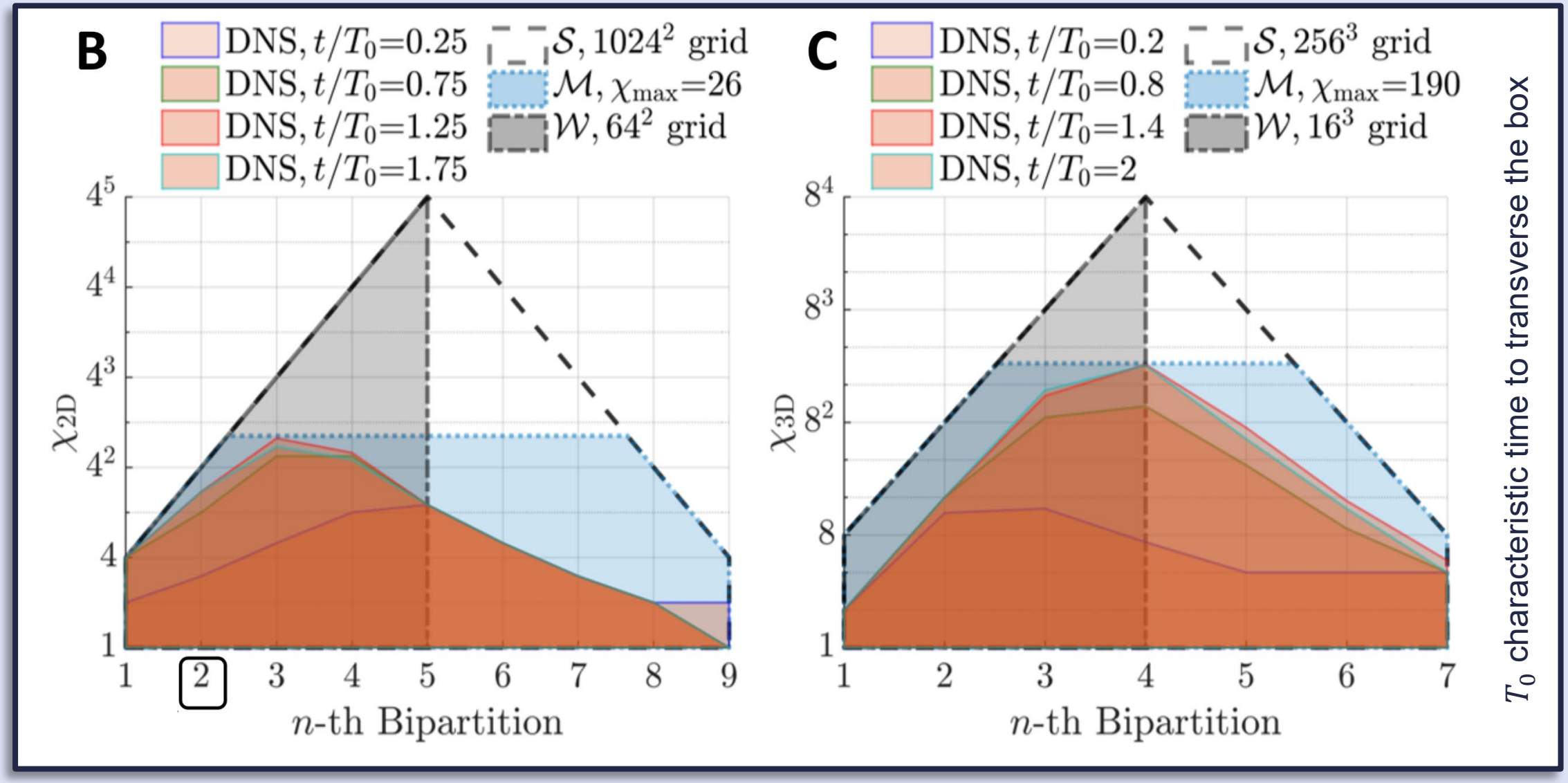
$$u(\boldsymbol{r_q}) = A^{q_1} A^{q_2} A^{q_3} A^{q_4} \cdots A^{q_N}$$

- Here A^{q_i} is a $\chi(i-1) \times \chi(i)$ matrix.
- The index q_i labels the position in the i-th 2×2 sub-grid $00, 01, 10, 11$.
- In principle the maximum χ can grow exponentially with the fineness of the grid.



MPS – Turbulence Correlations

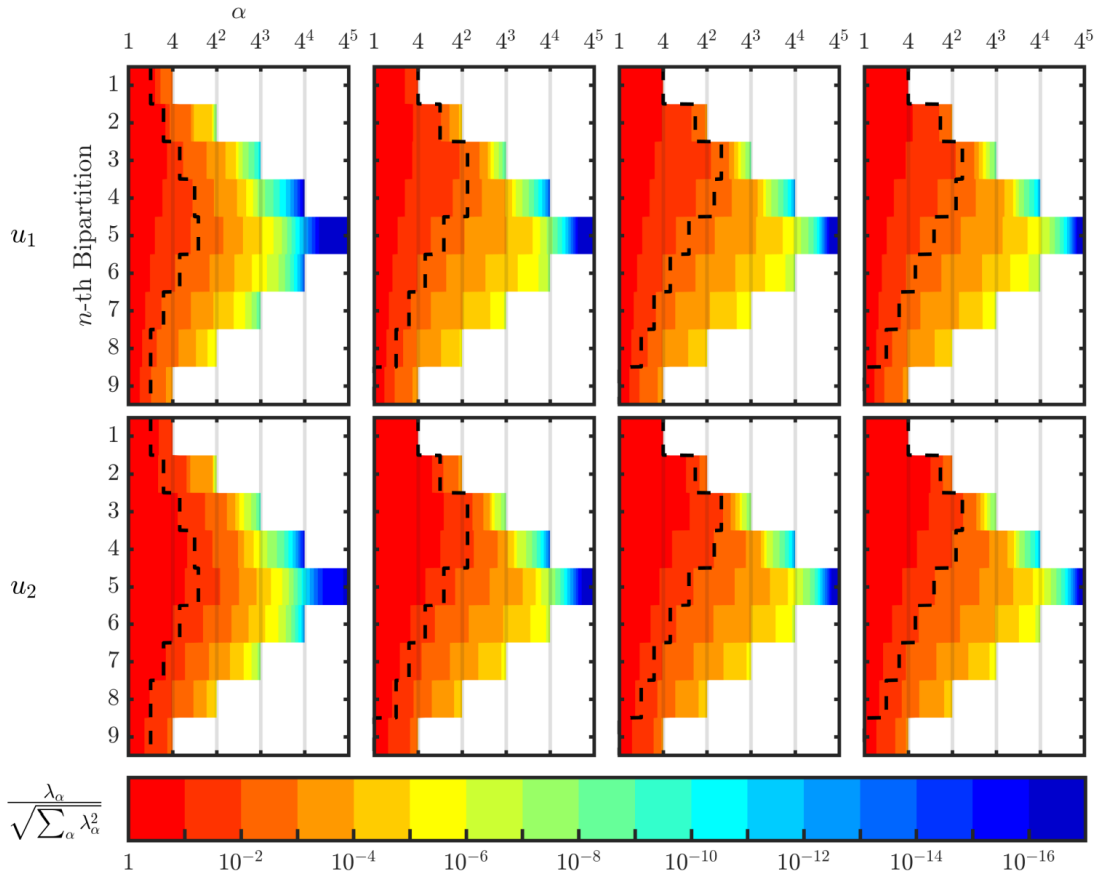
N. Gourianov, et al., Nature Computational Science **2**, 30 (2022).



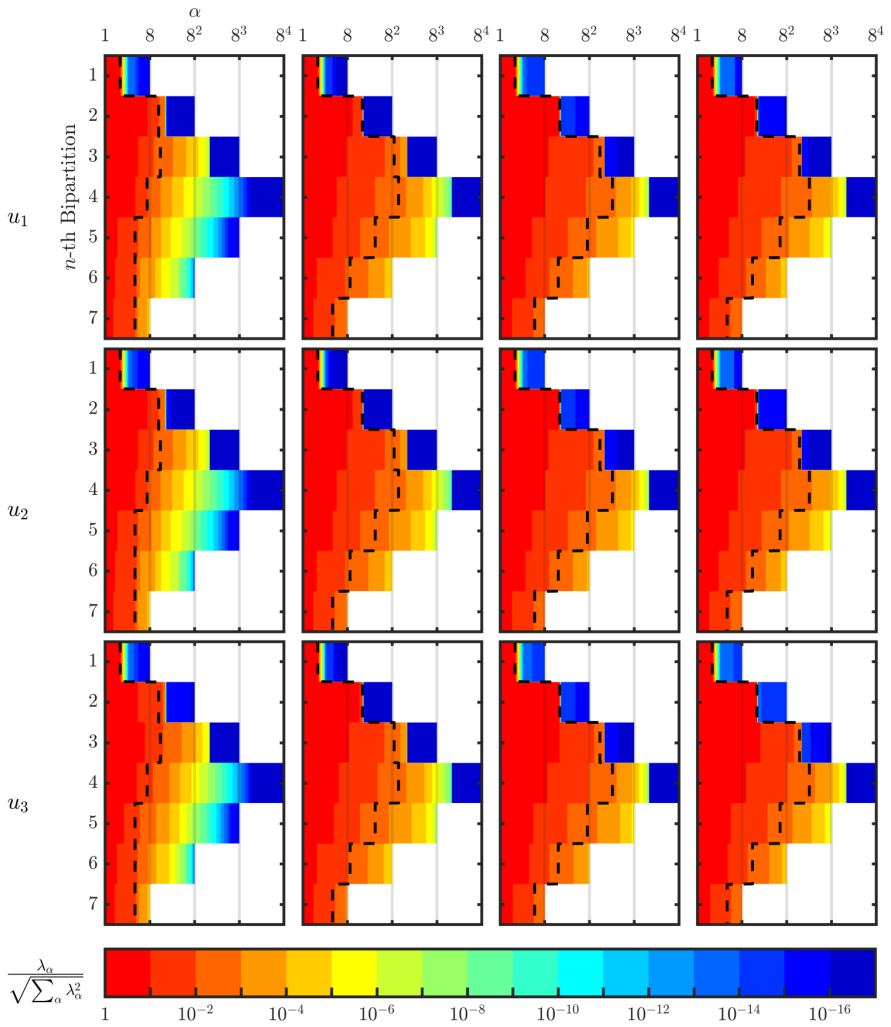
2D developing jet, Re=1000

3D Taylor-Green vortex, Re=800

Entanglement spectrum



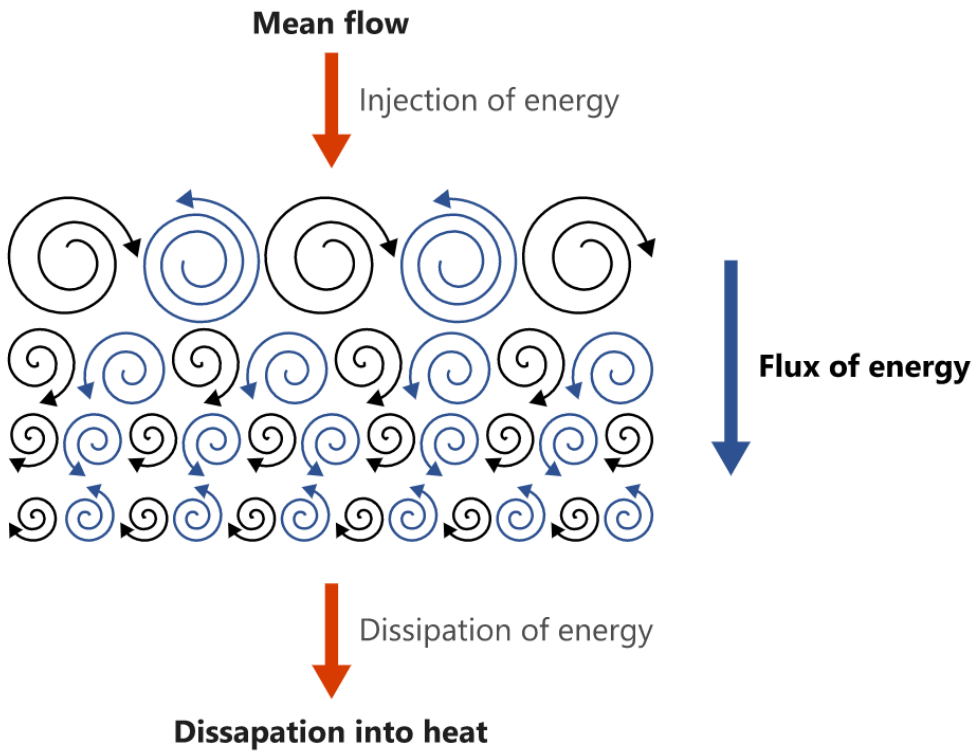
2D developing jet, Re=1000



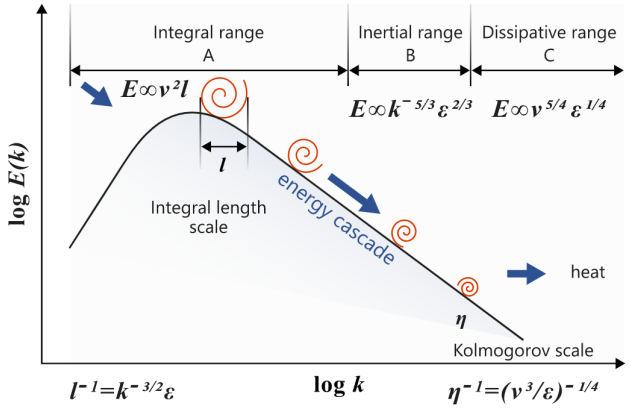
3D Taylor-Green vortex, Re=800

The energy cascade – three spatial dimensions

- Richardson in 1922 on 3D turbulence:

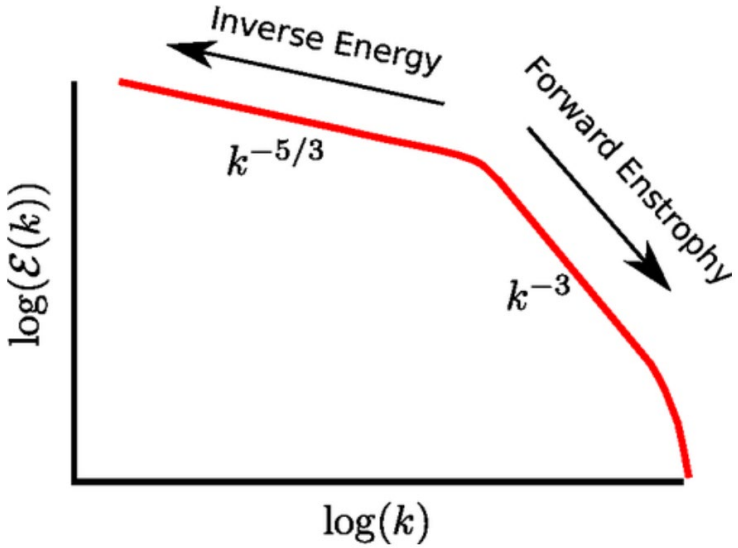


3D



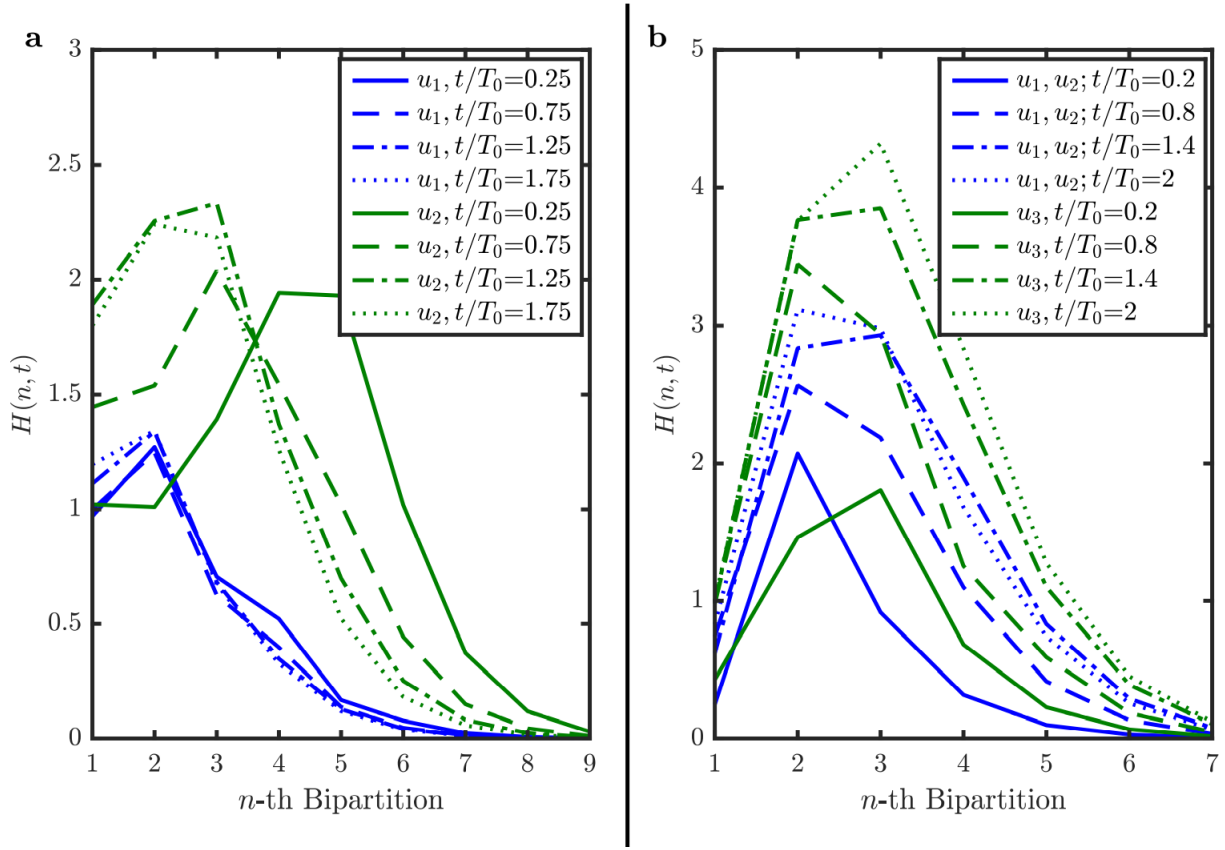
altair.com

2D



www-vortex.mcs.st-and.ac.uk/

Entanglement entropy



2D developing jet, Re=1000 3D Taylor-Green vortex, Re=800

- The entanglement entropy is different for 2D and 3D flows
- For 2D the entropy shifts to coarser length scales with increasing time, consistent with the inverse energy cascade
- In 3D the opposite happens consistent with energy cascade energising small length scales.

Outline

MPS

- Encoding grid functions
- Classical entanglement spectrum
- Turbulence correlations

CFD Examples

- MPS algorithm
- Jet formation and Taylor Green vortex
- Driven cavity
- Magnus effect

Hybrid Optimization

- Hardware architecture
- Quantum network
- Generic cost function
- Proof of Principle Example

MPS algorithm for the incompressible Navier-Stokes equation

- We need to solve the pair of equations

$$\frac{\partial \vec{v}}{\partial t} = -(\vec{v} \cdot \nabla)\vec{v} - \nabla p + \frac{1}{\mathrm{Re}} \nabla^2 \vec{v}$$

$$\nabla \cdot \vec{v} = 0$$

- We solve this set of equations using Runge-Kutta methods but illustrate the method here for a simple Euler step.
- Starting from the solution $\vec{v}$ at time t we want to obtain the velocity $\vec{v}^*$ at time $t + \Delta t$, encoded as MPS.
- We use a variational method and define the cost function (using the $\mathcal{L}_2$ norm)

$$\Theta(\vec{v}^*) = \mu |\overline{\nabla} \cdot \vec{v}^*|^2 + \left| \frac{\vec{v}^* - \vec{v}}{\Delta t} + (\vec{v} \cdot \overline{\nabla})\vec{v} - \nu \overline{\nabla}^2 \vec{v} \right|^2$$

Penalty term that ensures that the second equation is fulfilled

$\vec{v}^*$ only appears in the time derivative

MPS algorithm for evolving a 2D fluid flow in time

- We explicitly work out $\vec{v} = u_1\vec{e}_1 + u_2\vec{e}_2$ and $\vec{v}^* = u_1^*\vec{e}_1 + u_2^*\vec{e}_2$ and write the components as bold vectors on the grid, e.g. $\boldsymbol{u}_1 = \{u_1(r_1), u_1(r_1)\cdots\}$ and express the cost function as

$$\Theta(\boldsymbol{V}^*) = \sum_{i,j=1}^{2}\left\{\mu\left(\frac{\Delta\boldsymbol{u}_i^*}{\Delta x_i}\right)^t\frac{\Delta\boldsymbol{u}_j^*}{\Delta x_j}\right\} + \sum_{i=1}^{2}\left\{\frac{(\boldsymbol{u}_i^*)^t\boldsymbol{u}_i^*}{\Delta t^2} + \frac{(\boldsymbol{u}_i^*)^t}{\Delta t}\left(\frac{-\boldsymbol{u}_i}{\Delta t} + \sum_{j=1}^{2}\left\{\boldsymbol{u}_j\frac{\Delta\boldsymbol{u}_i}{\Delta x_j} - \nu\frac{\Delta^2\boldsymbol{u}_i}{\Delta x_j^2}\right\}\right)\right.$$
$$\left. + \left(\frac{-\boldsymbol{u}_i}{\Delta t} + \sum_{j=1}^{2}\left\{\boldsymbol{u}_j\frac{\Delta\boldsymbol{u}_i}{\Delta x_j} - \nu\frac{\Delta^2\boldsymbol{u}_i}{\Delta x_j^2}\right\}\right)^t\frac{\boldsymbol{u}_i^*}{\Delta t}\right\} + \left[\ldots\right],$$

- Here, $\Delta/\Delta x_j$ and $\Delta^2/\Delta x_j^2$ are the discretized first and second order derivatives in direction j.
- We actually use eighth-order central finite difference stencils and represent the Laplace operator as an MPO.
- We give a low order expression on the next slide.
- The terms in $[\cdots]$ are constant and thus irrelevant for the optimization, a step scales like $\mathcal{O}(N\chi^4)$ or $\mathcal{O}(\chi^4\log L)$.

Simple example MPO: Forward differencing in x-direction

- For this example we assume that the grid has been re-ordered, going along the lines of the grid in x-direction, with fixed boundaries

$$\left[\frac{\Delta}{\Delta x} f\right]_{p,q} = \frac{f_{p+1,q} - f_{p,q}}{\Delta x}$$

- This operator can be written as an MPS of the form

$$\frac{\Delta}{\Delta x} = A(\otimes_k B[k])C$$

- where

$$A = \frac{(1,0)}{\Delta x}, \qquad B[1 \le k \le N] = \mathbb{I}, \qquad B[N < k \le 2N] = \begin{pmatrix} \mathbb{I} & \sigma_{01} \\ 0 & \sigma_{10} \end{pmatrix}, \qquad C = (-1,1)^T$$

- For a 4×4 grid we have

$$\frac{\Delta}{\Delta x} = \frac{1}{\Delta x}(1,0)\left(\mathbb{I} \otimes \mathbb{I} \otimes \begin{pmatrix} \mathbb{I} & \sigma_{01} \\ 0 & \sigma_{10} \end{pmatrix} \otimes \begin{pmatrix} \mathbb{I} & \sigma_{01} \\ 0 & \sigma_{10} \end{pmatrix}\right)\begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

$$= \frac{1}{\Delta x}(1,0)\begin{pmatrix} \mathbb{I} \otimes \mathbb{I} \otimes \mathbb{I} \otimes \mathbb{I} & \mathbb{I} \otimes \mathbb{I} \otimes \mathbb{I} \otimes \sigma_{01} + \mathbb{I} \otimes \mathbb{I} \otimes \sigma_{01} \otimes \sigma_{10} \\ 0 & \sigma_{10} \otimes \sigma_{10} \end{pmatrix}\begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

$$= \frac{1}{\Delta x}(-\mathbb{I} \otimes \mathbb{I} \otimes \mathbb{I} \otimes \mathbb{I} + \mathbb{I} \otimes \mathbb{I} \otimes \mathbb{I} \otimes \sigma_{01} + \mathbb{I} \otimes \mathbb{I} \otimes \sigma_{01} \otimes \sigma_{10})$$

$$\sigma_{01} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

$$\sigma_{10} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$$

$$\sigma_{00} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\sigma_{11} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

Minimization of the cost function

- We put the orthogonality centre of the MPS at site n and write

$$\boldsymbol{u}_i^* = U_i^{*n} \boldsymbol{c}_i^{*n}, \quad \text{where} \quad U_i^{*n} = \Phi_i^{*n} \otimes \mathbb{1} \otimes (\Psi_i^{*n})^t$$

- This leads to the cost function

$$\begin{aligned}
\Theta(\boldsymbol{V}^*) = \sum_{i,j=1}^{2} & \left\{ -\mu (\boldsymbol{c}_i^{*n})^t \, (U_i^{*n})^t \, \frac{\Delta}{\Delta x_i} \frac{\Delta}{\Delta x_j} U_j^{*n} \boldsymbol{c}_j^{*n} \right\} \\
& + \sum_{i=1}^{2} \left\{ \frac{(\boldsymbol{c}_i^{*n})^t \, (U_i^{*n})^t \, U_i^{*n} \boldsymbol{c}_i^{*n}}{\Delta t} + \frac{(\boldsymbol{c}_i^{*n})^t \, (U_i^{*n})^t}{\Delta t} \left(\frac{-\boldsymbol{u}_i}{\Delta t} + \sum_{j=1}^{2} \left\{ \boldsymbol{u}_j \frac{\Delta \boldsymbol{u}_i}{\Delta x_j} - \nu \frac{\Delta^2 \boldsymbol{u}_i}{\Delta x_j^2} \right\} \right) \right. \\
& \left. \left(\frac{-\boldsymbol{u}_i}{\Delta t} + \sum_{j=1}^{2} \left\{ \boldsymbol{u}_j \frac{\Delta \boldsymbol{u}_i}{\Delta x_j} - \nu \frac{\Delta^2 \boldsymbol{u}_i}{\Delta x_j^2} \right\} \right)^t \frac{U_i^{*n} \boldsymbol{c}_i^{*n}}{\Delta t} \right\} + \left[\ldots \right],
\end{aligned}$$

Minimizing the cost function

- Minimizing this cost function is equivalent to solving the equation

$$\boldsymbol{c}_k^{*n} - (U_k^{*n})^t \sum_{j=1}^{2} \left\{ \mu \Delta t^2 \frac{\Delta}{\Delta x_k} \frac{\Delta}{\Delta x_j} U_j^{*n} \boldsymbol{c}_j^{*n} \right\} = (U_k^{*n})^t \left(\boldsymbol{u}_k - \Delta t \sum_{j=1}^{2} \left\{ \boldsymbol{u}_j \frac{\Delta \boldsymbol{u}_k}{\Delta x_j} - \nu \frac{\Delta^2 \boldsymbol{u}_k}{\Delta x_j^2} \right\} \right)$$

- and this can be rewritten as finding the ground state of a positive definite operator $(\mathbb{I} - \mu \Delta t^2 H)\boldsymbol{\alpha} = \boldsymbol{\beta}$ where

$$H_{kj} = (U_k^{*n})^t \frac{\Delta}{\Delta x_k} \frac{\Delta}{\Delta x_j} U_j^{*n},$$

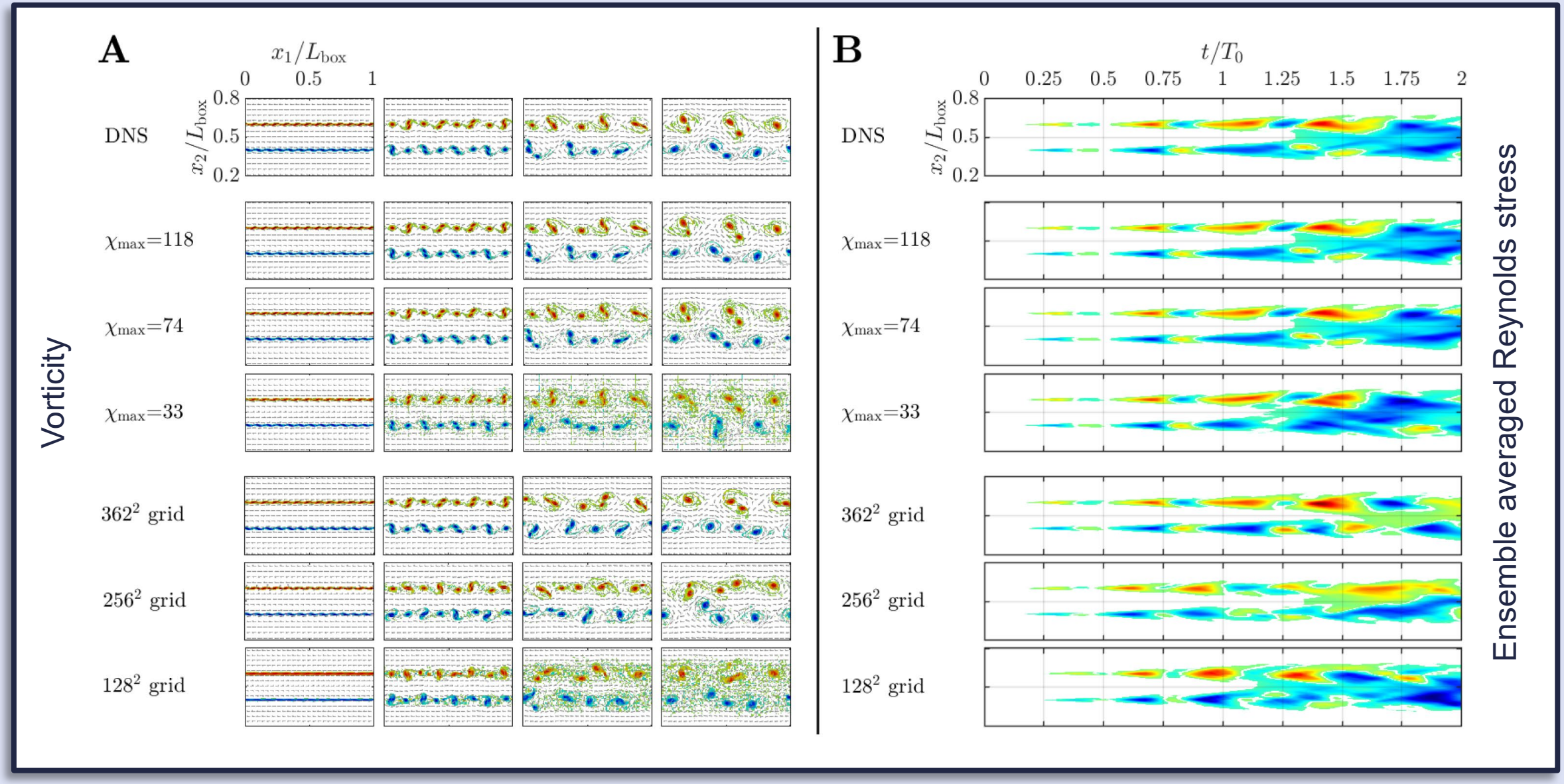
$$\boldsymbol{\alpha}_k = \boldsymbol{c}_k^{*n},$$

$$\boldsymbol{\beta}_k = (U_k^{*n})^t \left(\boldsymbol{u}_k - \Delta t \sum_{j=1}^{2} \left\{ \boldsymbol{u}_j \frac{\Delta \boldsymbol{u}_k}{\Delta x_j} - \nu \frac{\Delta^2 \boldsymbol{u}_k}{\Delta x_j^2} \right\} \right)$$

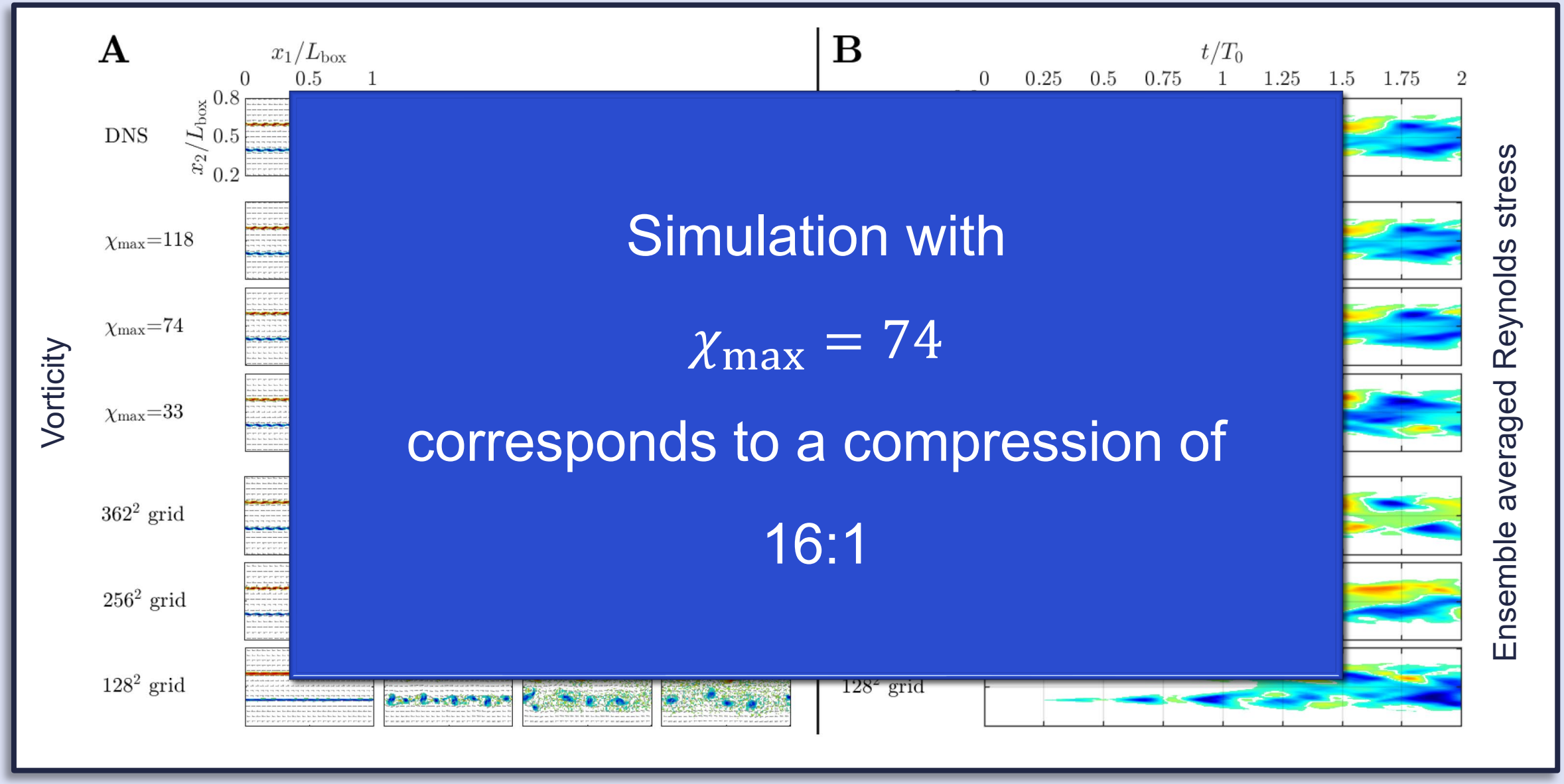
- This can be solved by gradient descent method. Global optimization is then done via the DMRG algorithm.

Example: 2D Jet Formation

CFD Examples – Jet Formation



CFD Examples – Jet Formation



Simulation with

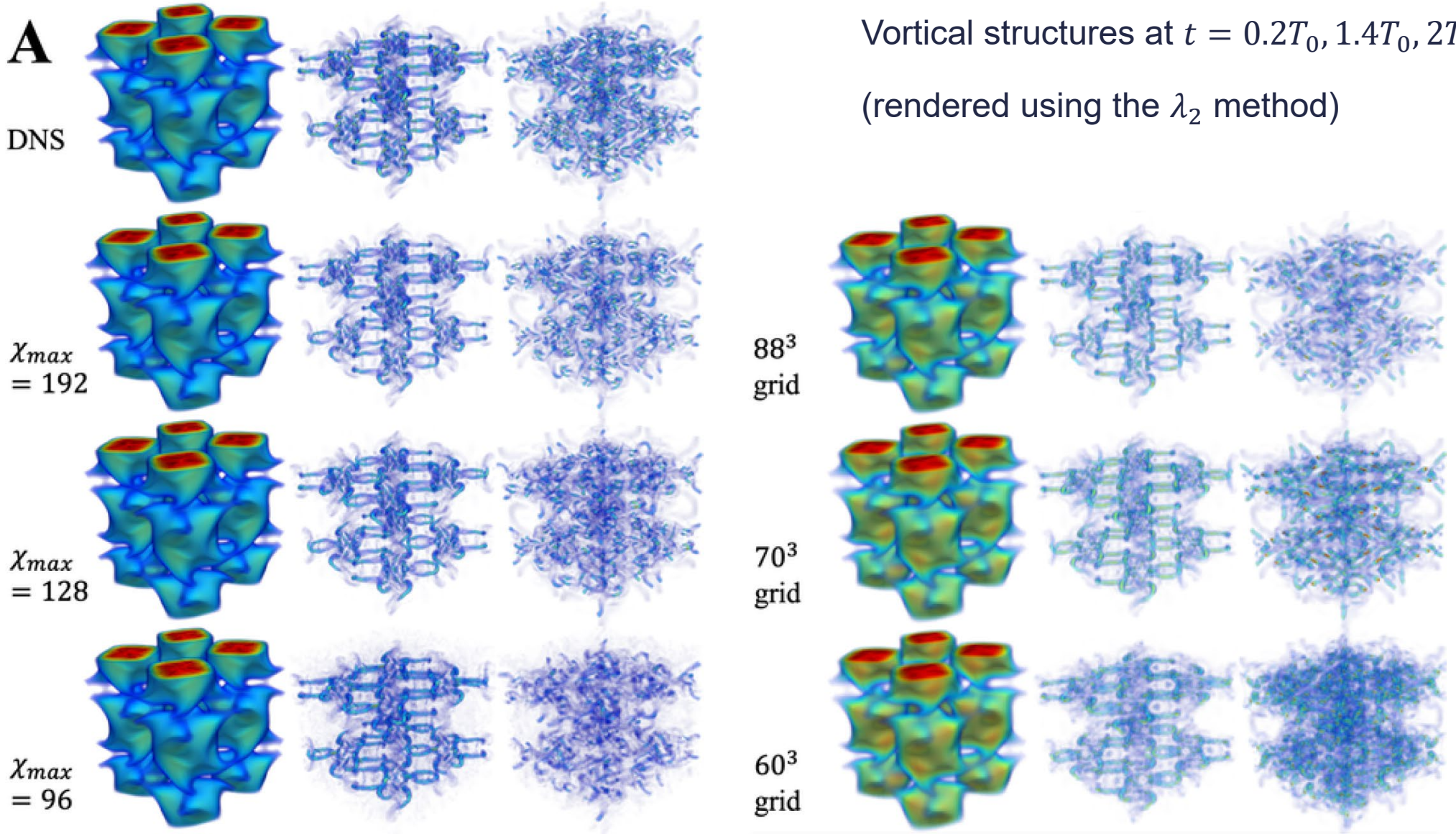
$\chi_{\max} = 74$

corresponds to a compression of

16:1

Example: Taylor Green Vortex

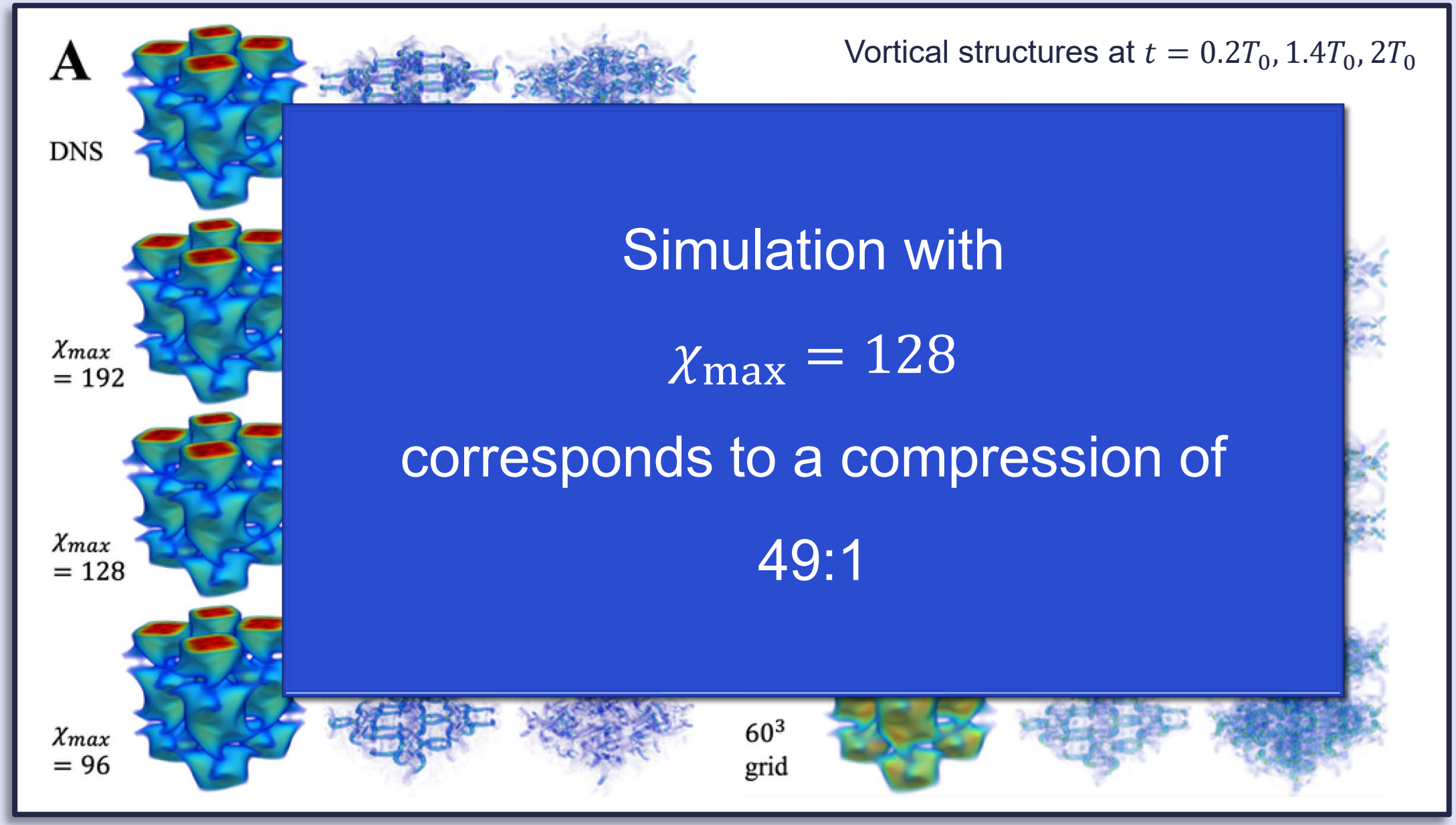
CFD Examples – Taylor Green Vortex



Vortical structures at $t = 0.2T_0, 1.4T_0, 2T_0$

(rendered using the λ_2 method)

CFD Examples – Taylor Green Vortex



Vortical structures at $t = 0.2T_0, 1.4T_0, 2T_0$

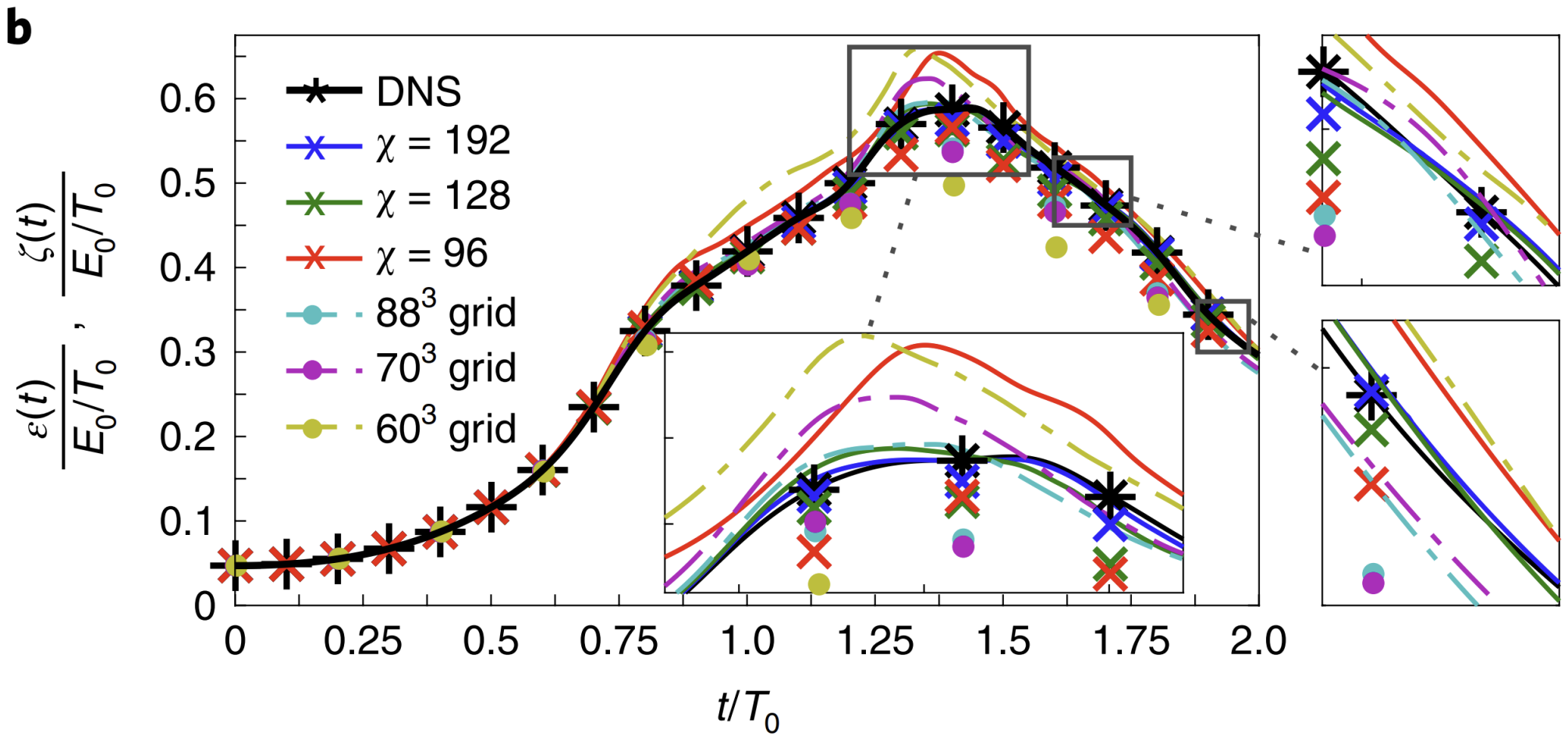
Simulation with

$\chi_{\max} = 128$

corresponds to a compression of

49:1

CFD Examples – Taylor Green Vortex



Enstrophy $\xi(t)$ and energy decay $\epsilon(t)$

The enstrophy

- The enstrophy of a flow in a volume Ω is defined as

$$\zeta(\vec{v}) = \int_{\Omega} |\nabla \vec{v}|^2 \, \mathrm{d}\Omega$$

- Here $|\nabla \vec{v}|^2 = \sum_{i,j} \left|\partial_i v_j\right|^2$
- For an incompressible flow with $\nabla \cdot \vec{v} = 0$ this is the same as the integral over the squared vorticity $\vec{\omega} = \nabla \times \vec{v}$

$$\zeta(\vec{v}) = \int_{\Omega} |\nabla \times \vec{v}|^2 \, \mathrm{d}\Omega = \int_{\Omega} |\vec{\omega}|^2 \, \mathrm{d}\Omega$$

- Also, for the incompressible Navier Stokes equation the enstrophy describes the dissipation of energy

$$-\frac{d}{dt}\left(\frac{1}{2}\int_{\Omega} |\vec{v}|^2 \, \mathrm{d}\Omega\right) = \nu \zeta(\vec{v})$$

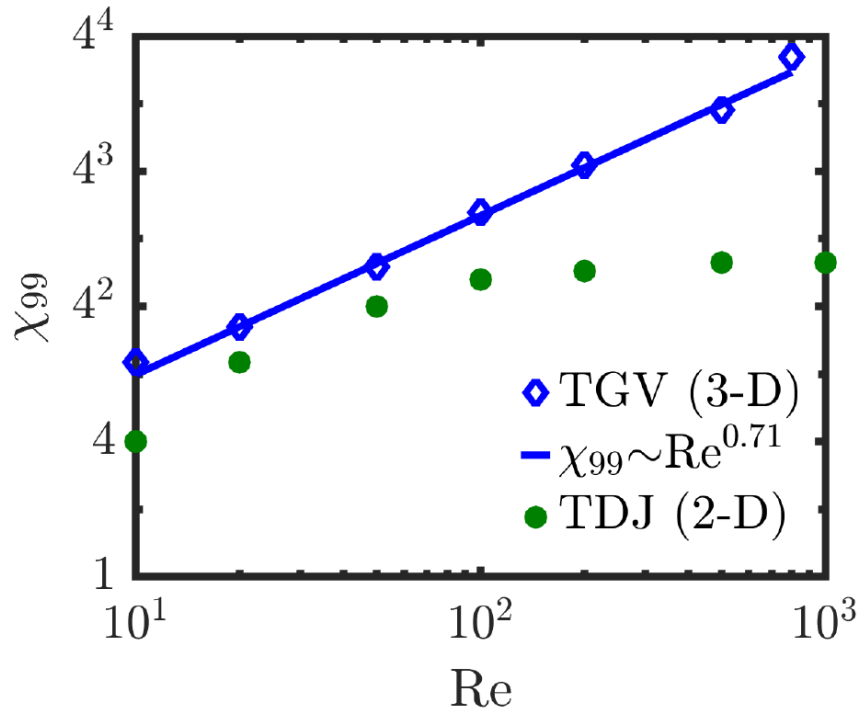
- Comparing these two quantities can thus be used to test the accuracy of a simulation.

Gaining a computational advantage in runtime?

- Scaling is often assessed as a function of characteristic numbers like the Reynolds number

$$Re = \left(\frac{L}{\eta}\right)^{4/3}$$

- Here L is the largest size of the energy containing eddies and η is the Kolmogorov microscale.
- Typically, numerically exact methods are expected to scale like $Re^{3K/4}$ where K is the number of spatial dimensions.
- The runtime scales as $\mathrm{Re}^{4\chi_{99}}$ which means favourable scaling for TDJ with $K = 2$ where $\chi_{99} \approx const.$ but not for the TGV with $K = 3$ where it grows with an exponent of ≈ 0.71 for the two examples studied above.
- **Note:** we do not know the general scaling.

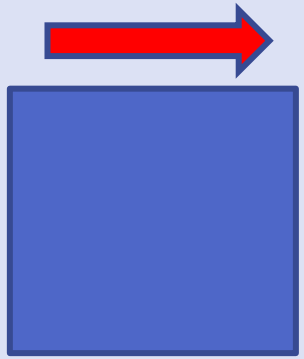


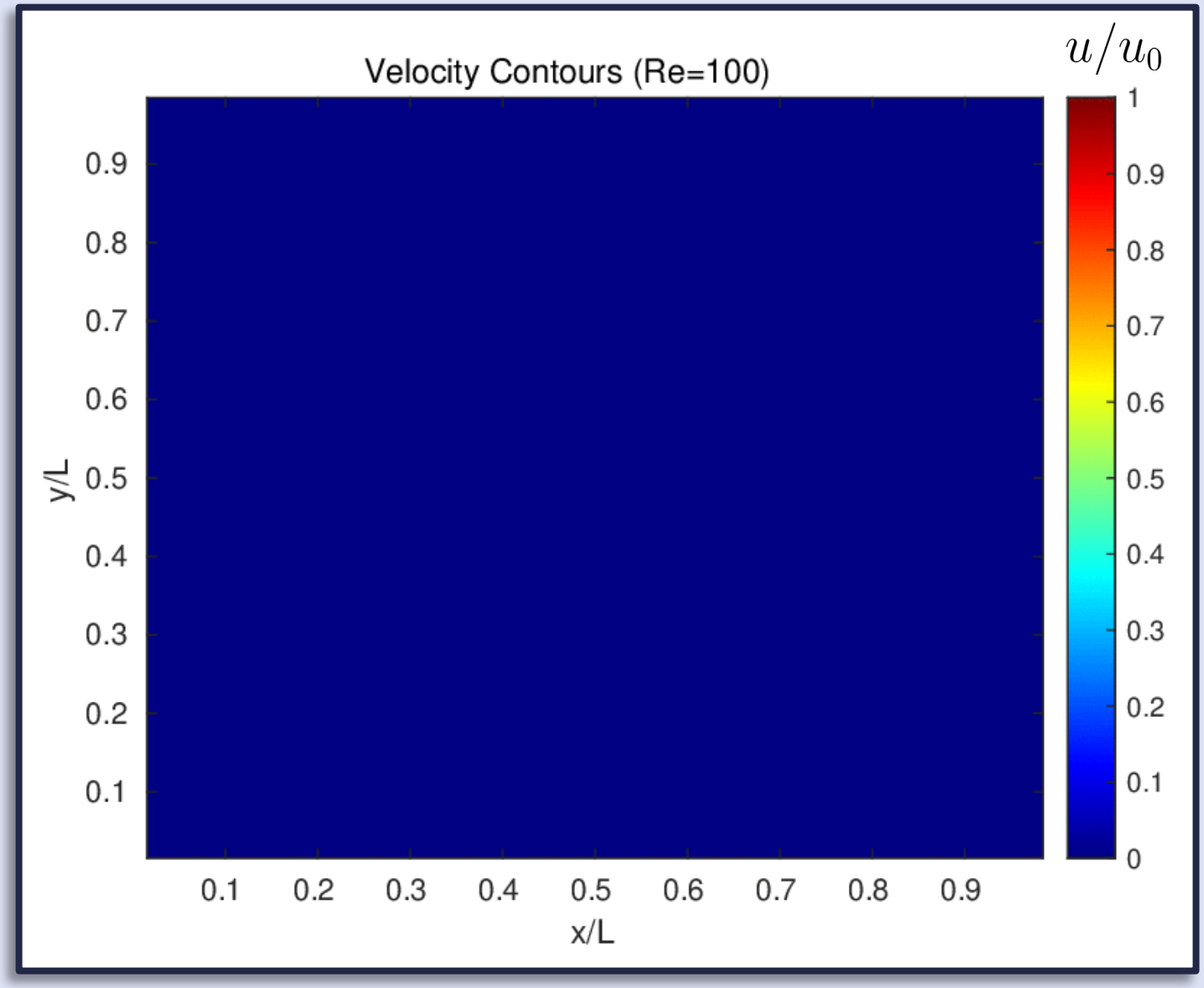
Schmidt number χ_{99} required for a 99% accurate representation of a flow field. The blue diamonds are for a decaying TGV in 3D where $\chi_{99} \propto Re^{0.71}$ for sufficiently large Re. The green dots arise from a 2D TDJ. Here $\chi_{99} \propto Re^{0}$ for sufficiently large Re.

Example: Lid driven cavity

CFD Examples – 2D driven cavity

- Time evolution starting from the fluid and the lid at rest.





W. Y. Soh and J. W. Goodrich, Journal of Computational Physics **79**, 113 (1988).

MPS algorithm for problems with boundaries in 2D

- The Navier Stokes equation is rewritten as

$$\partial_t w = -\left[\partial_x(uw) + \partial_y(vw)\right] + \nu\Delta w$$
$$\Delta\psi = -w\,.$$

- With the stream function ψ related to the velocity components as $u = \partial_y\psi\,,\ v = -\partial_x\psi$
- The vorticity w is evolved according to

$$\partial_t w = \partial_x F + \partial_y G$$

- where $F = -uw + \nu(\partial_x w)$ and $G = -vw + \nu(\partial_y w)$
- in a 2-step predictor-corrector McCormack scheme.
- Finally, the stream function is updated by solving the Poisson equation $\nabla\psi = -w$.

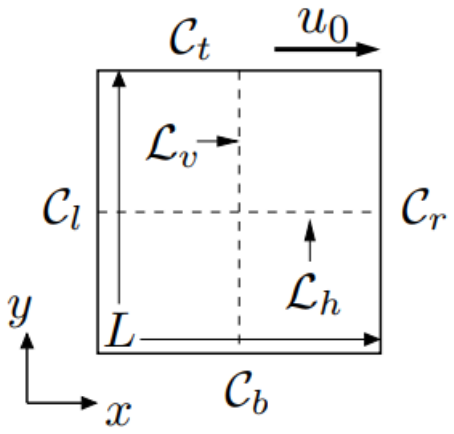
Operation	Algorithm	Scaling
Addition	Variational addition of MPS (see Sec. 4.5 in [10]).	χ^3
Multiplication	Multiplication algorithm in [22] combined with variational compression [10] of the product MPS.	χ^4
Poisson solver	MPS algorithm for solving the Poisson equation in [23].	χ^3
Matrix-vector multiplication	MPO-MPS contraction combined with variational compression (see Sec. 5 in [10]). For the system considered here, the MPO bond dimension $D \leq 6$ and thus $D \ll \chi$.	$D\chi^3$

M. Kiffner and DJ, *Tensor network reduced order models for wall-bounded flows*, Phys. Rev. Fluids **8**, 124101 (2023).

Boundary conditions enforced by walls

- We use ghost points that represent the boundaries with indices -1 and K
- The b.c. are

	$\mathcal{C}_t$	$\mathcal{C}_r$	$\mathcal{C}_b$	$\mathcal{C}_l$
u	u_0	0	0	0
v	0	0	0	0
ψ	0	0	0	0



- From this we can get the bc for the vorticity

$$
\begin{aligned}
w_{p,K} &= -\frac{3}{h}u_0 + \frac{1}{h^2}\left(-4\psi_{p,K-1} + \frac{1}{2}\psi_{p,K-2}\right) \\
w_{p,-1} &= \frac{1}{h^2}\left(-4\psi_{p,0} + \frac{1}{2}\psi_{p,1}\right), \\
w_{-1,q} &= \frac{1}{h^2}\left(-4\psi_{0,q} + \frac{1}{2}\psi_{1,q}\right), \\
w_{K,q} &= \frac{1}{h^2}\left(-4\psi_{K-1,q} + \frac{1}{2}\psi_{K-2,q}\right).
\end{aligned}
$$

- For implementing these boundary conditions we need MPOs that extract lines or rows of values of a function
- For instance Q_e extracts function values at $(K-1, q)$ from a function

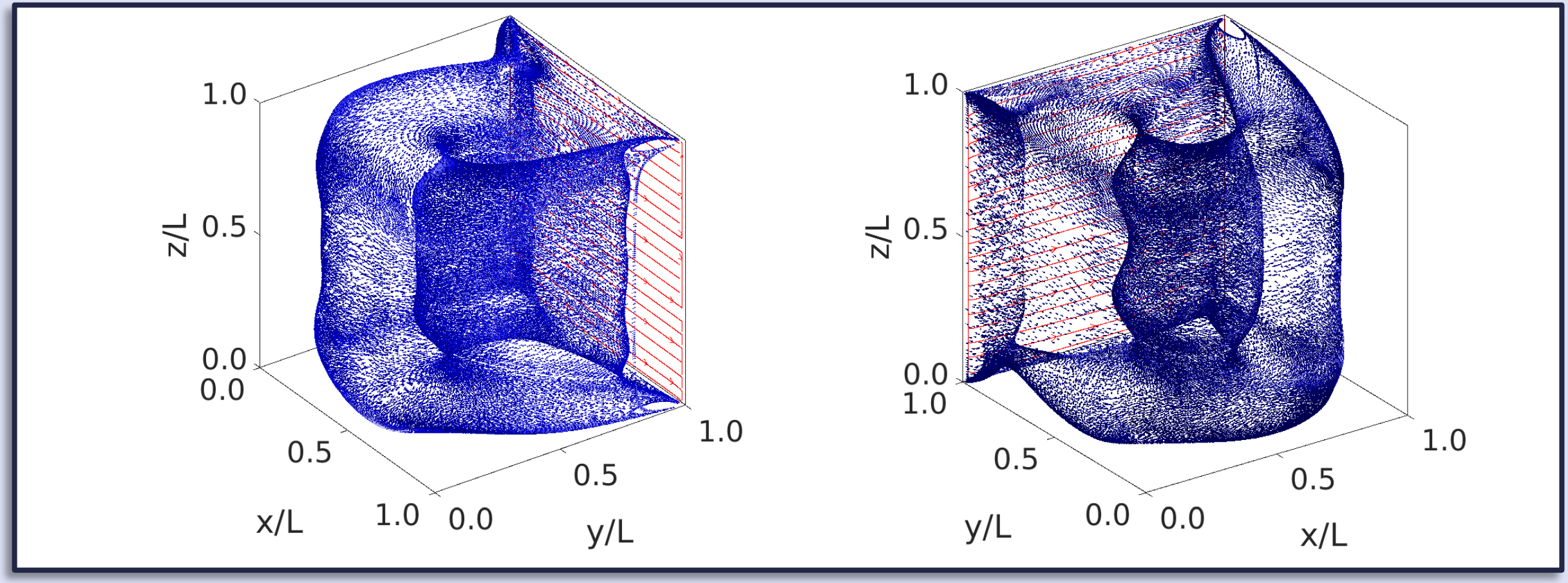
$$[Q_e f]_{p,q} = f_{p,q}\delta_{K-1,q}$$

- Q_e is build from matrices

$$
\begin{aligned}
A &= 1\,, \\
B[k] &= \mathbb{1}\,, \quad 1 \le k \le N\,, \\
B[k] &= \sigma_{11}\,, \quad N < k \le 2N \\
C &= 1\,.
\end{aligned}
$$

- And similar for all other terms that we need.

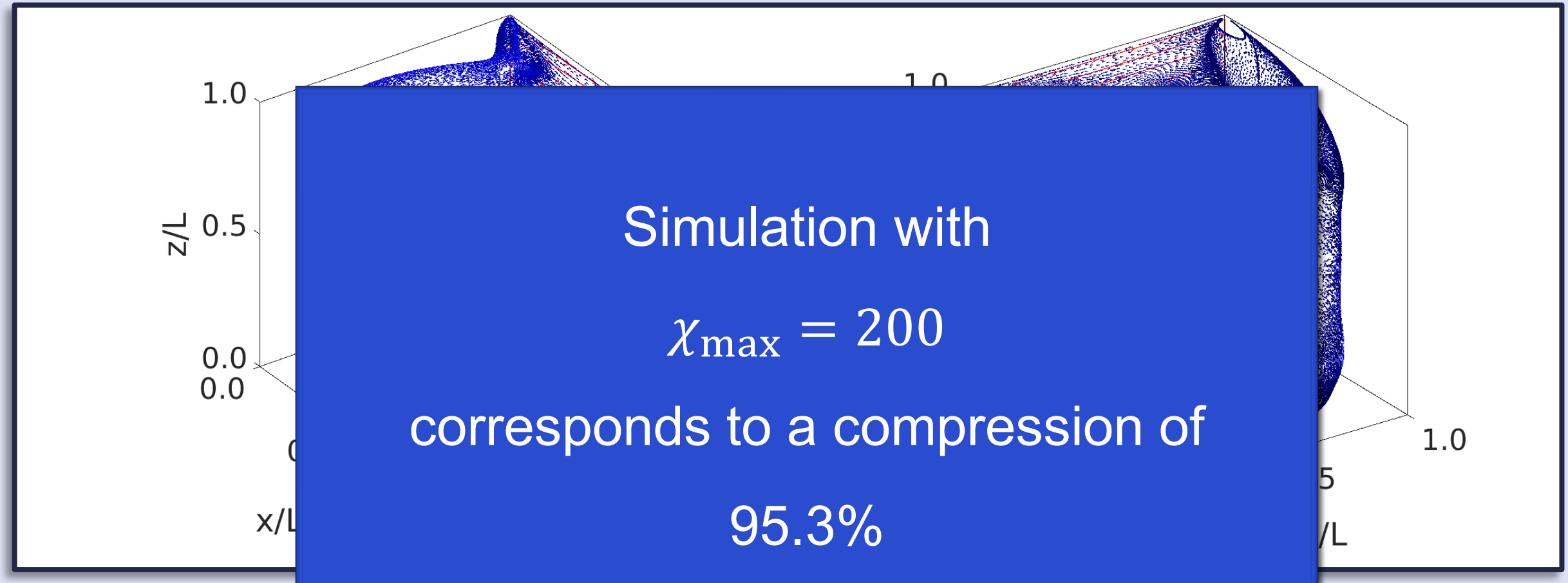
CFD Examples – Driven 3D Cavity



- Reynolds number $Re = 1000$,
- Grid size $256 \times 256 \times 256$

- Residuals after 3000 time steps
  - Momentum equations: $|\partial_t \vec{v}| \approx 3.6 \times 10^{-3}$
  - Continuity equation: $|\nabla \cdot \vec{v}| \approx 1.0 \times 10^{-2}$

CFD Examples – Driven 3D Cavity



Simulation with

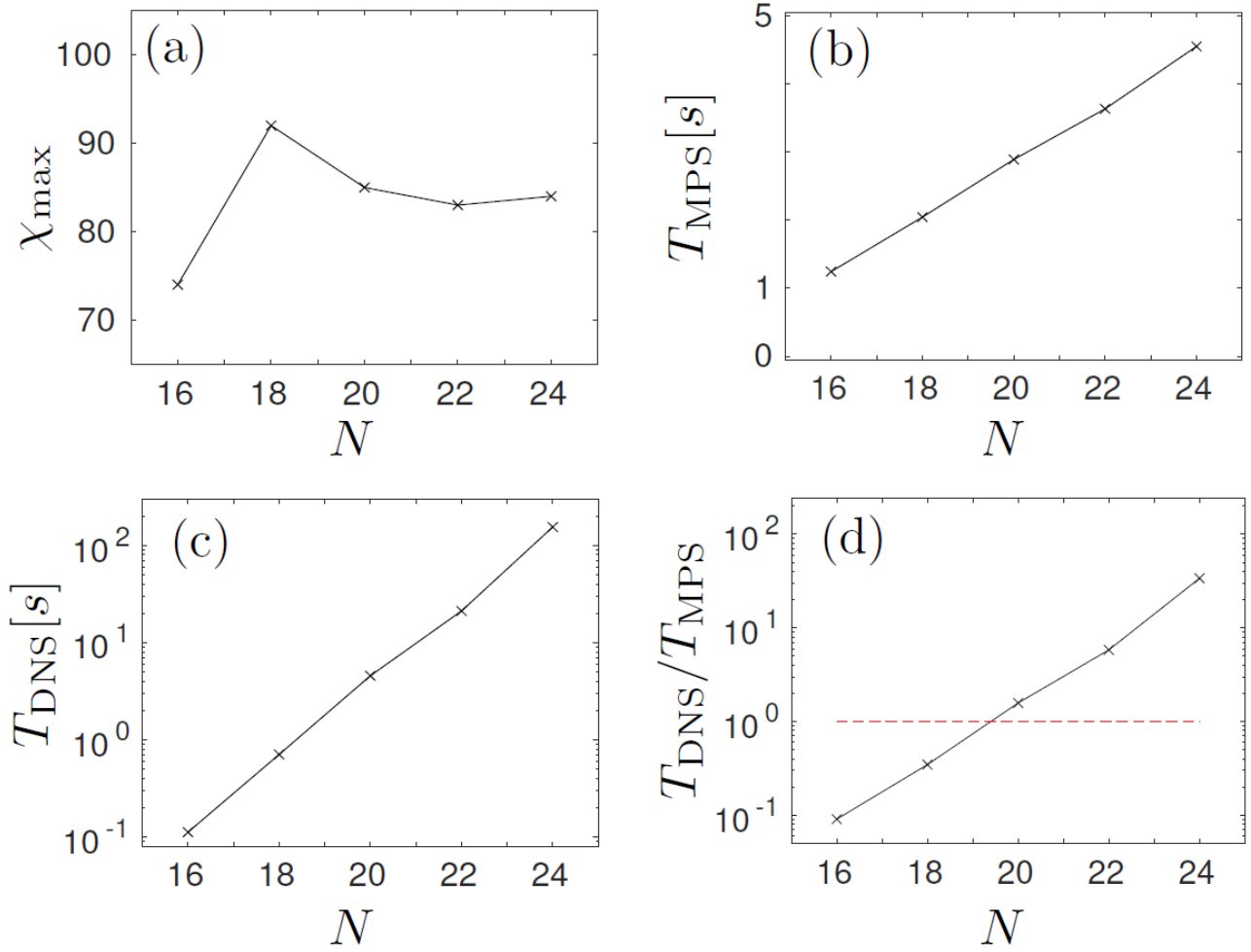
$\chi_{\max} = 200$

corresponds to a compression of

95.3%

- Reynolds numbe
- Grid size $256 \times 256 \times 256$
- Momentum equations: $|\partial_t \vec{v}| \approx 3.6 \times 10^{-3}$
- Continuity equation: $|\nabla \cdot \vec{v}| \approx 1.0 \times 10^{-2}$

Scaling compared to DNS calculations in 2D



- Simulations in 2D

$$\mathrm{Re} = 24000$$

$$\frac{\eta}{L} = 5.19 \times 10^{-4}$$

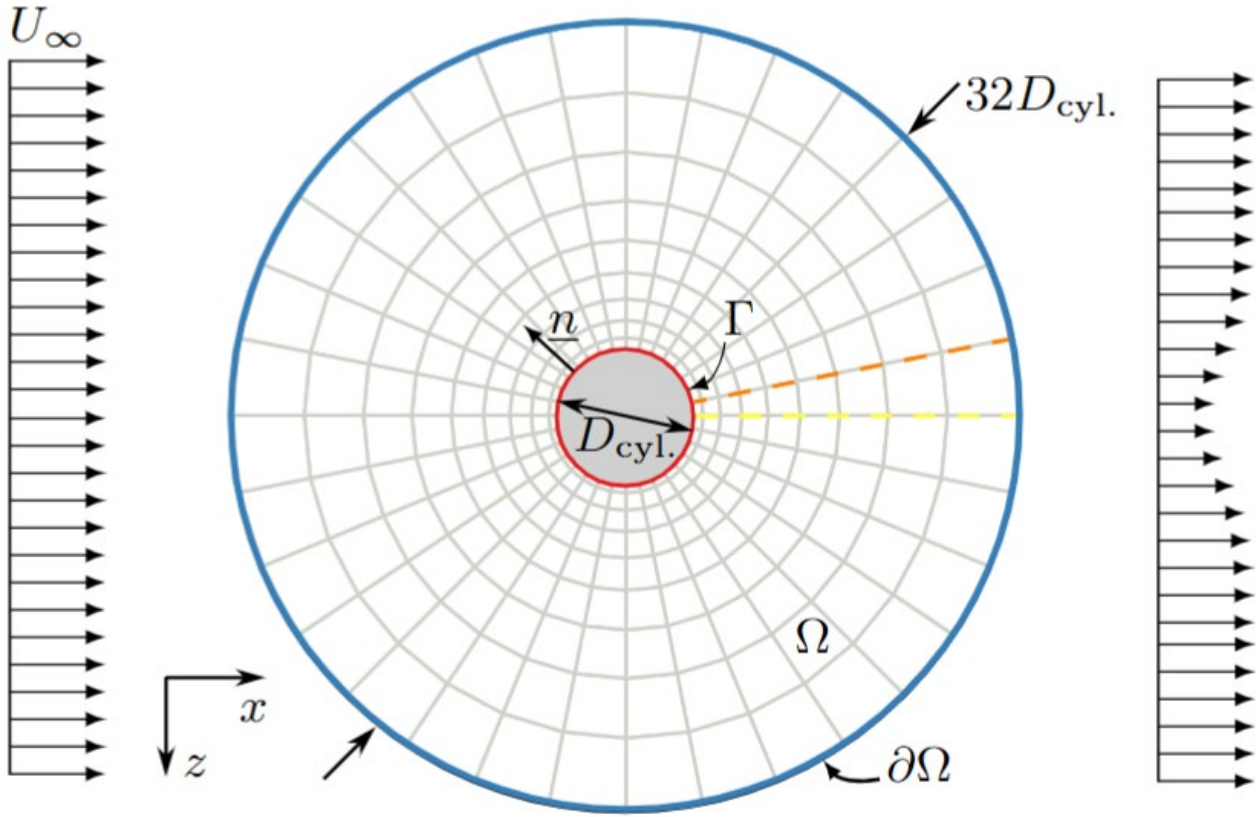
- Grid size required

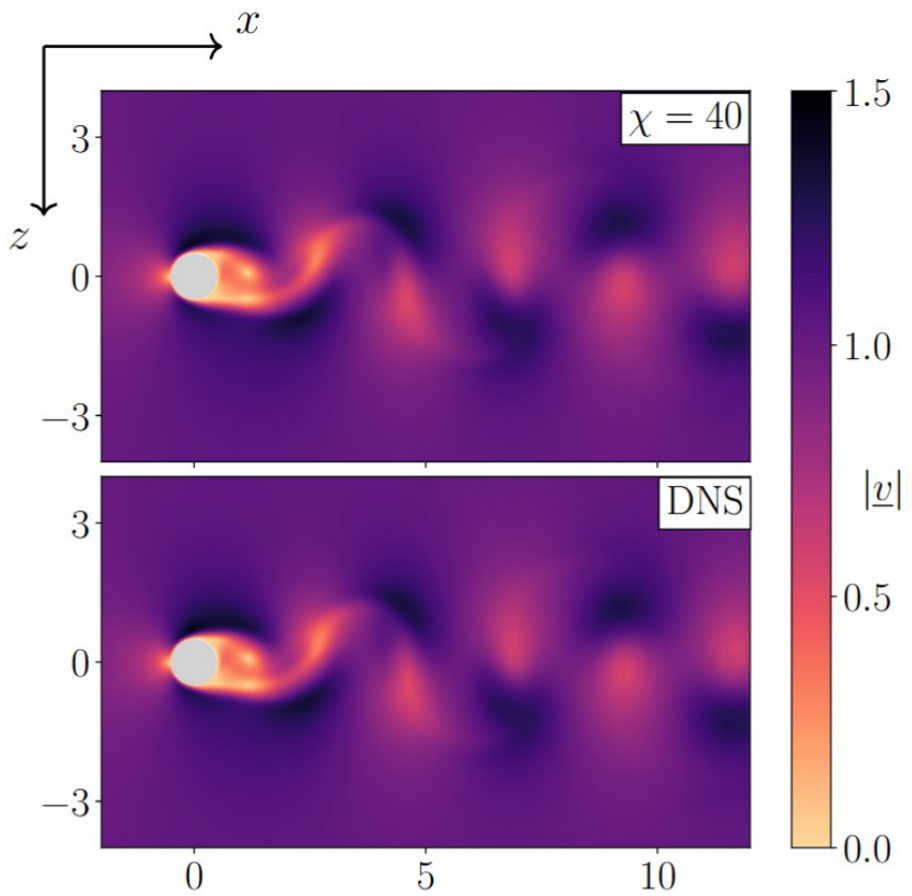
$$2^{11} \times 2^{11}$$

M. Kiffner and DJ, *Tensor network reduced order models for wall-bounded flows*, Phys. Rev. Fluids **8**, 124101 (2023).

Example: Magnus Effect

Body fitted coordinates – flow around a cylinder

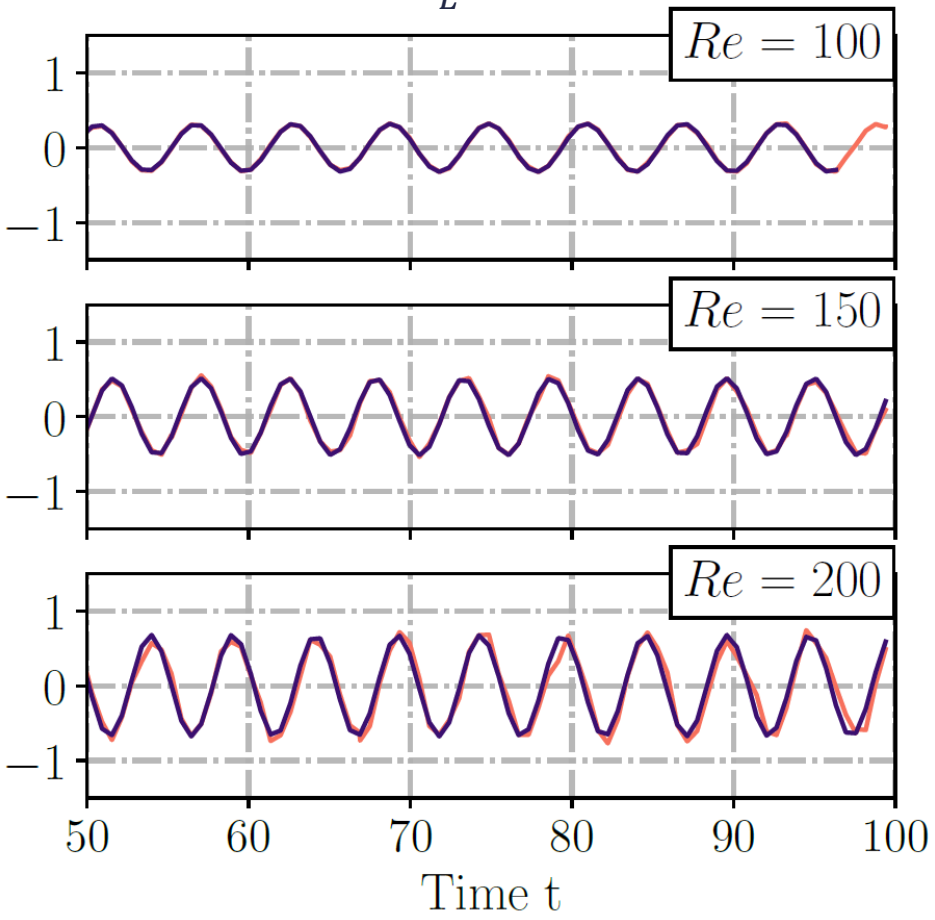


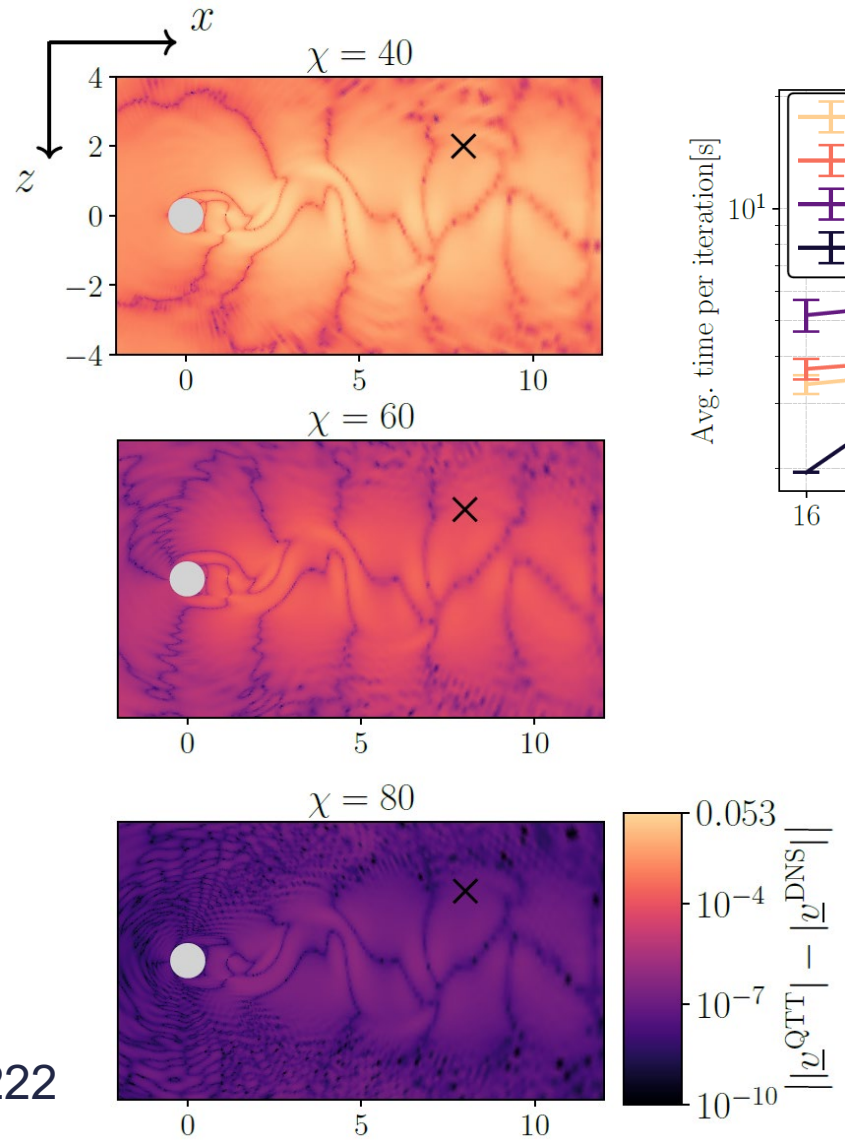


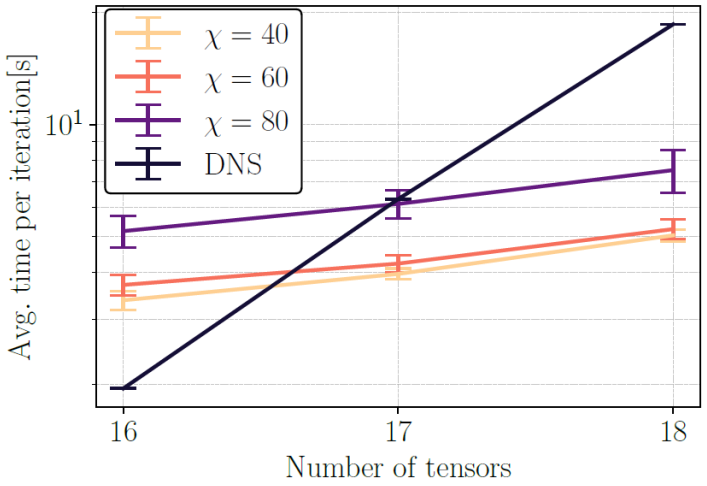
- Generate a body-fitted mesh from a one-dimensional boundary point distribution for convex objects.
- Transform cartesian differential operators into curvilinear matrix product operators, using a given set of grid coordinates.
- Compute key flow characteristics, such as aerodynamic lift and drag coefficients and use to benchmark computations

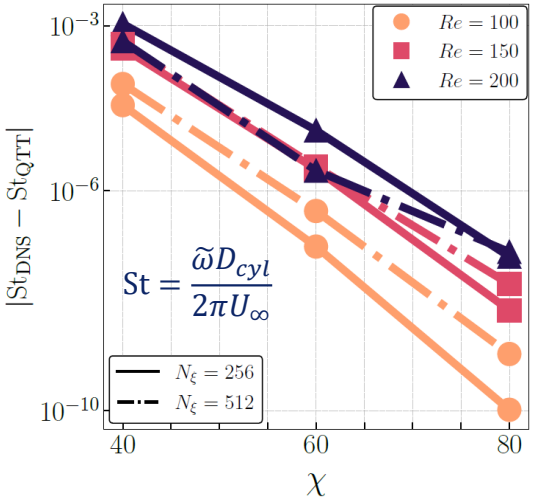
Body fitted coordinates – flow around a cylinder

- Lift coefficient C_L



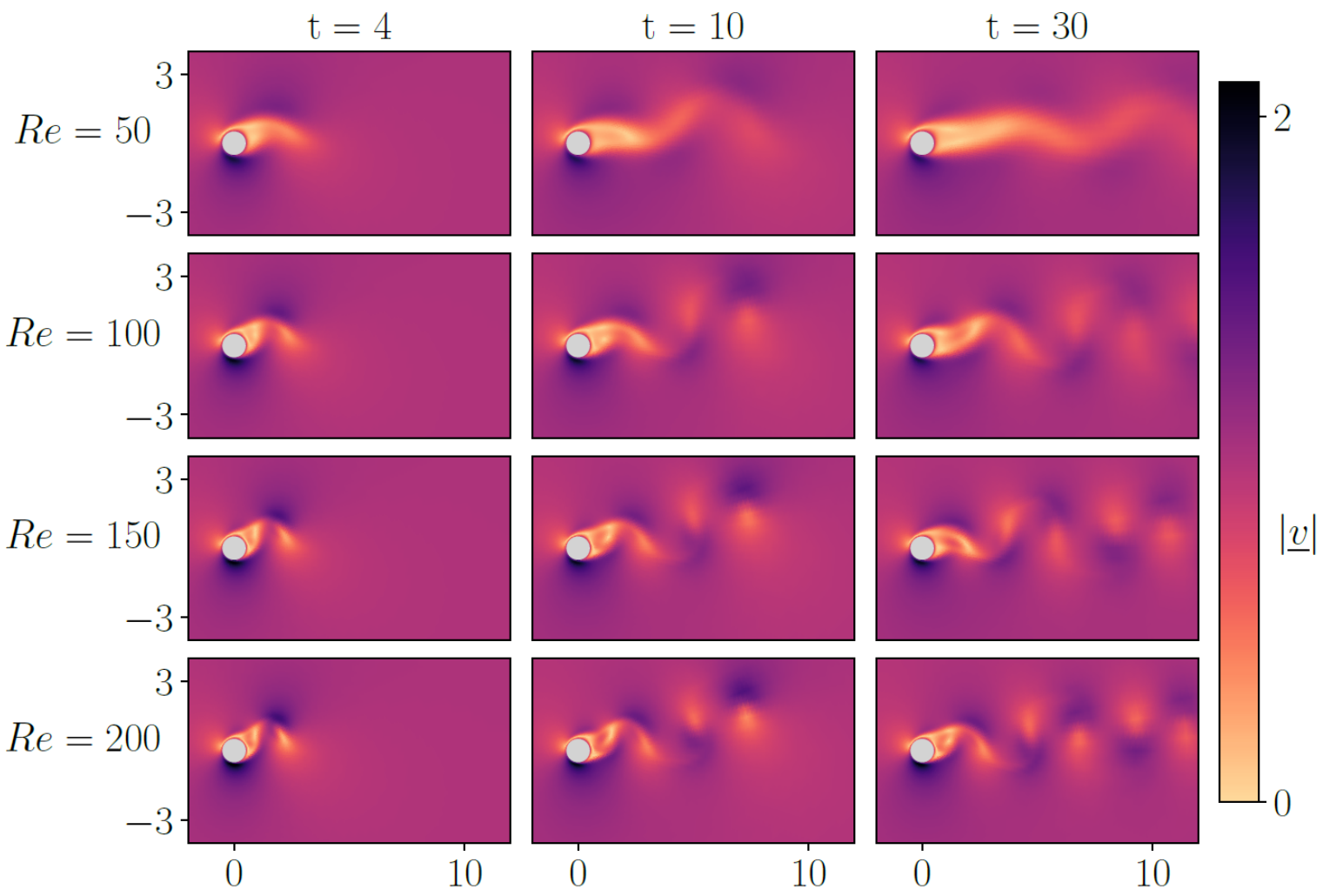




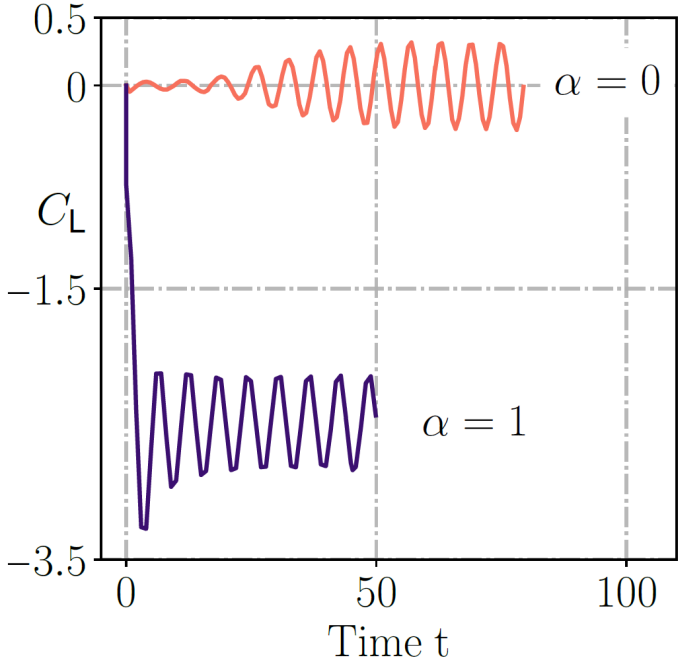


Nis-Luca van Huelst et al., arXiv:2507.05222

Body fitted coordinates – Magnus effect



Nis-Luca van Huelst et al., arXiv:2507.05222 (2025)



Tangental velocity $U_T = \alpha U\_\infty$

Discretization	χ	$\overline{C}_L$ abs	$\overline{C}_L$ %	St abs	St %
241 × 241	–	**2.4881**	-	**0.1655**	–
258 × 256	40	2.4914	0.13	0.1662	0.40
258 × 256	50	2.4899	0.07	0.1663	0.48
258 × 256	60	2.4904	0.09	0.1663	0.47

Outline

MPS

- Encoding grid functions
- Classical entanglement spectrum
- Turbulence correlations

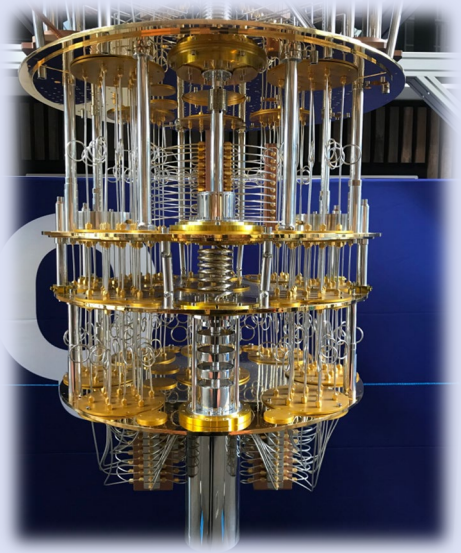
CFD Examples

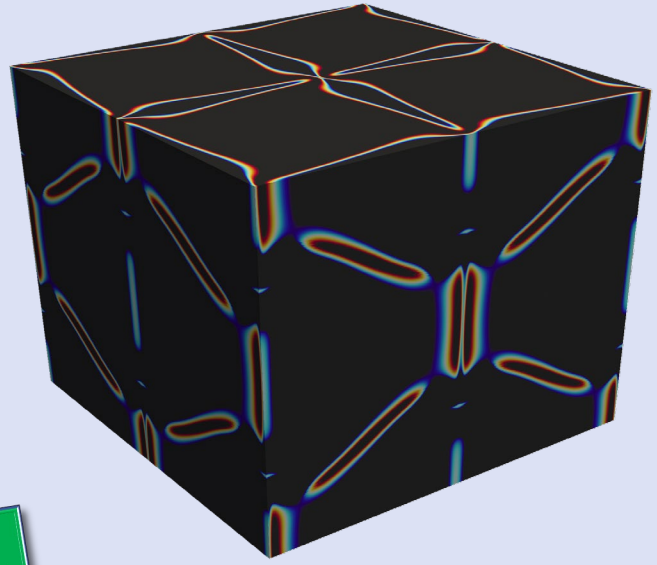
- MPS algorithm
- Jet formation and Taylor Green vortex
- Driven cavity
- Magnus effect

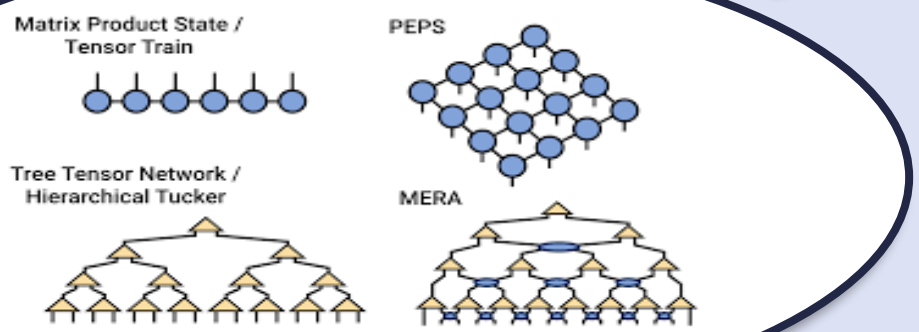
Hybrid Optimization

- Hardware architecture
- Quantum network
- Generic cost function
- Proof of Principle Example

Tensor networks as a quantum programming paradigm

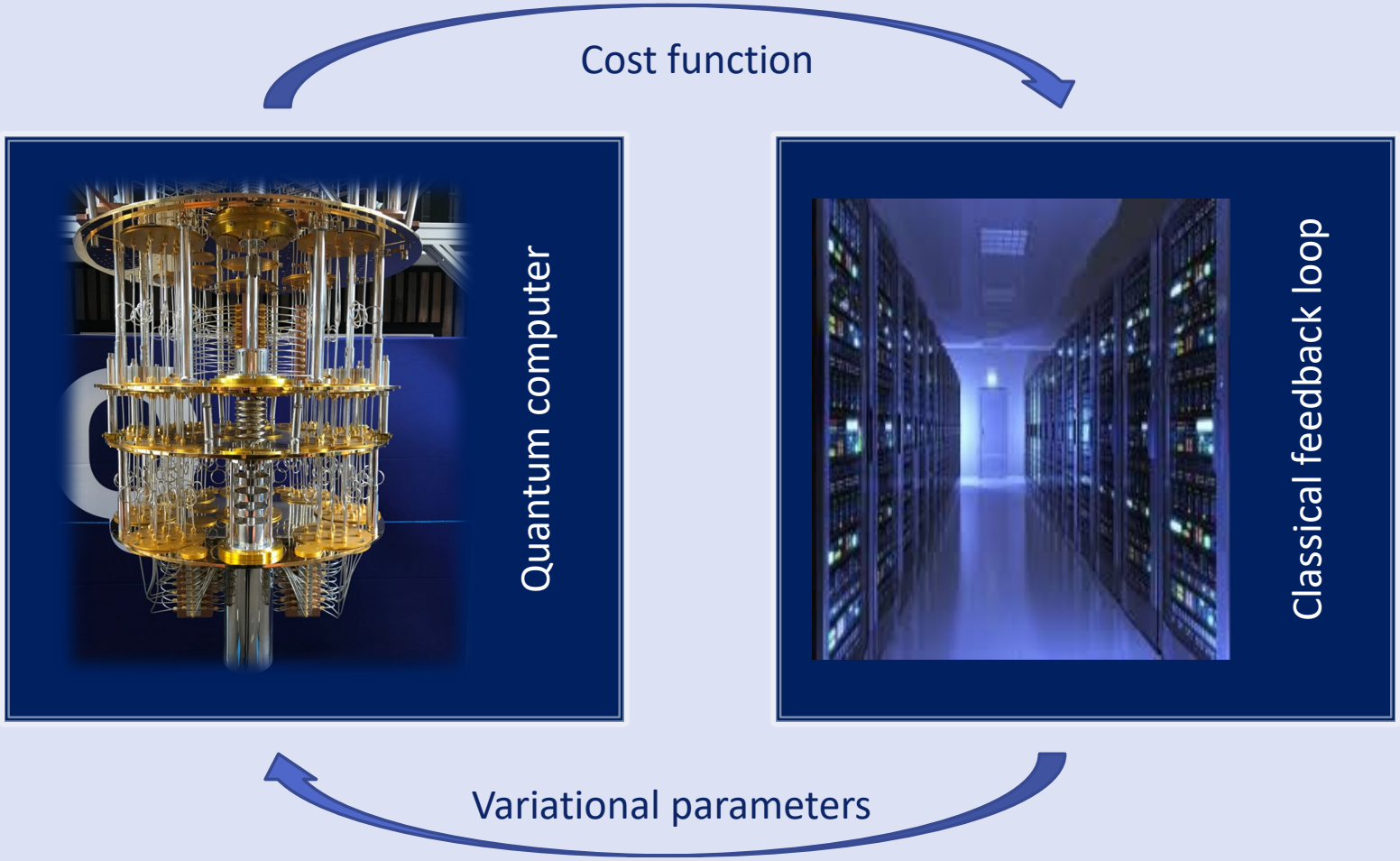






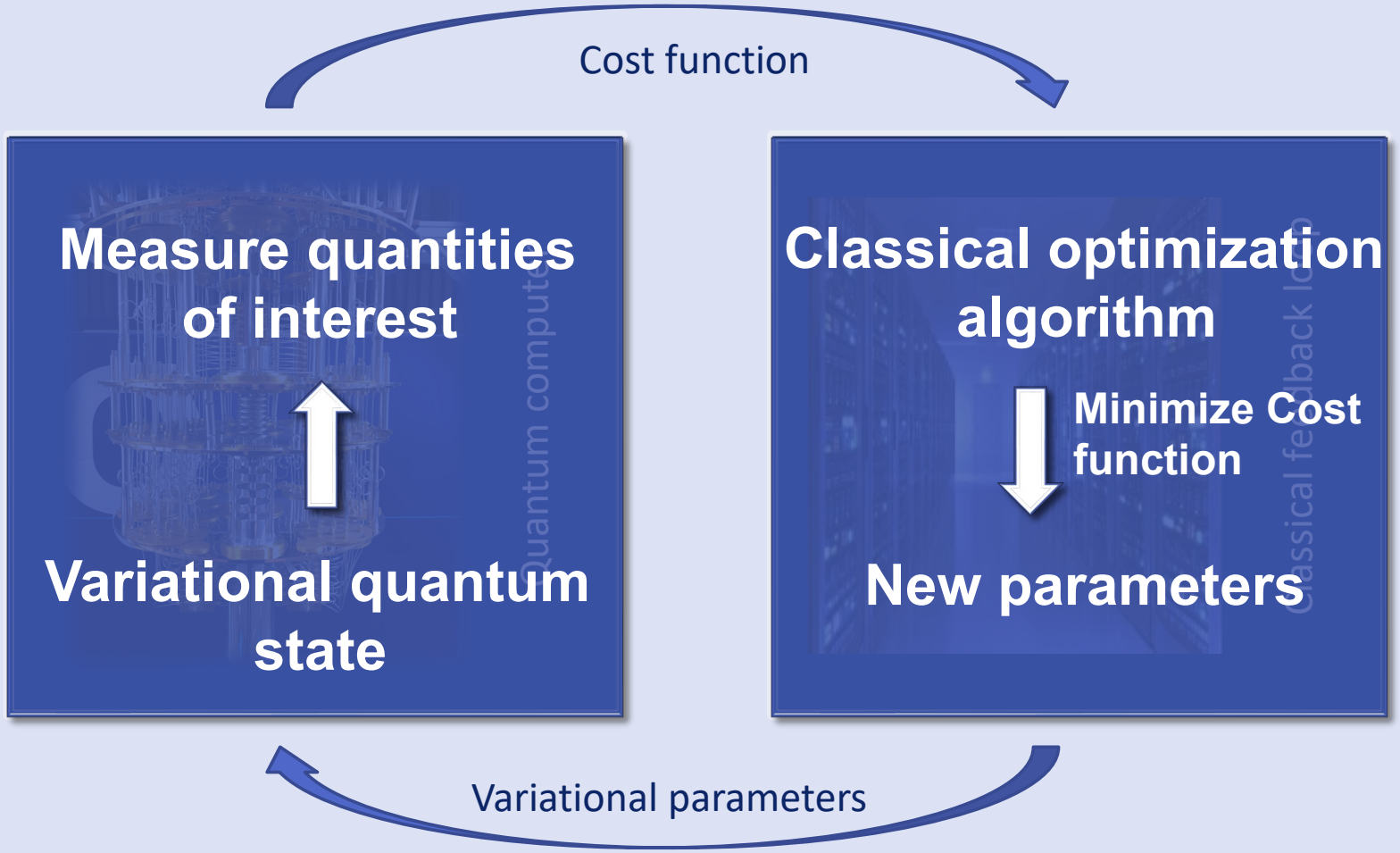
Variational quantum algorithms for computational fluid dynamics,
D Jaksch, P Givi, AJ Daley, T Rung
AIAA journal **61**, 1885 (2023)

Hybrid Classical-Quantum Optimization – Architecture



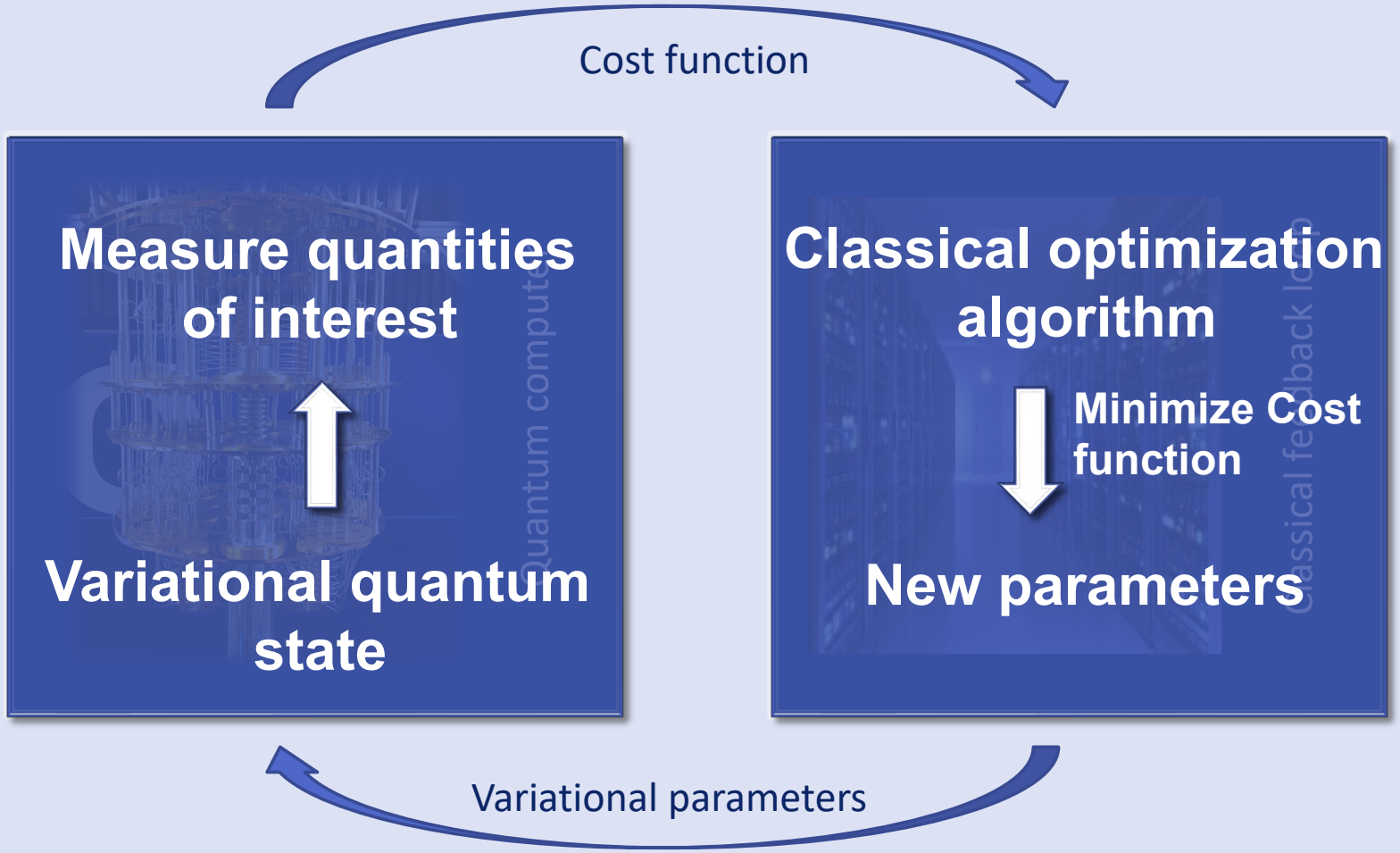
A. Peruzzo, et al., Nat. Commun. **5**, 4213 (2014).

Hybrid Classical-Quantum Optimization – Architecture



A. Peruzzo, et al., Nat. Commun. **5**, 4213 (2014).

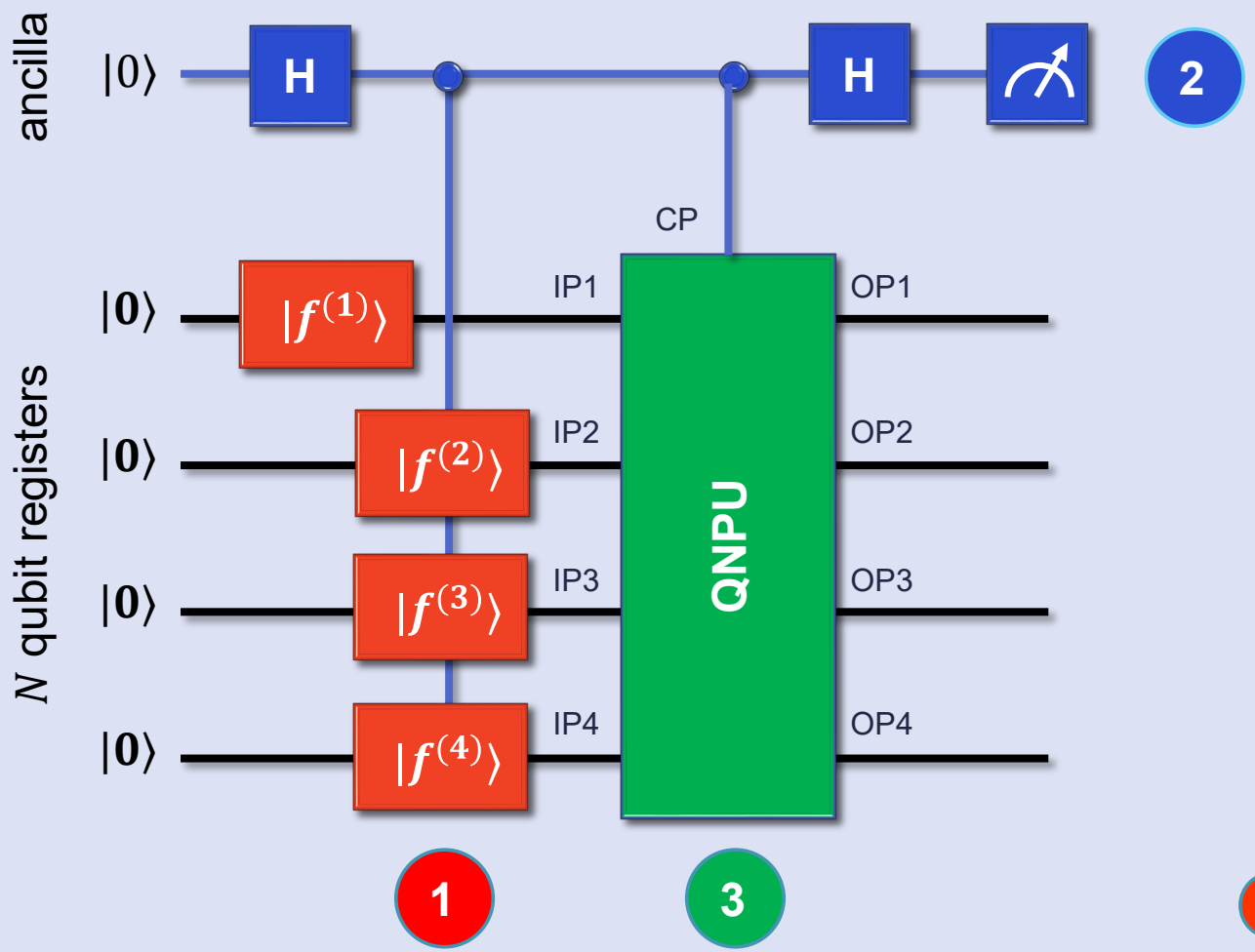
Hybrid Classical-Quantum Optimization – Architecture



Our work: Extension to **nonlinear** problems

M. Lubasch, J. Joo, P. Moinier, M. Kiffner & DJ, Phys. Rev. A **101**, 010301(R) (2020).

The QNPU Quantum Network for cost function $\mathcal{C}$

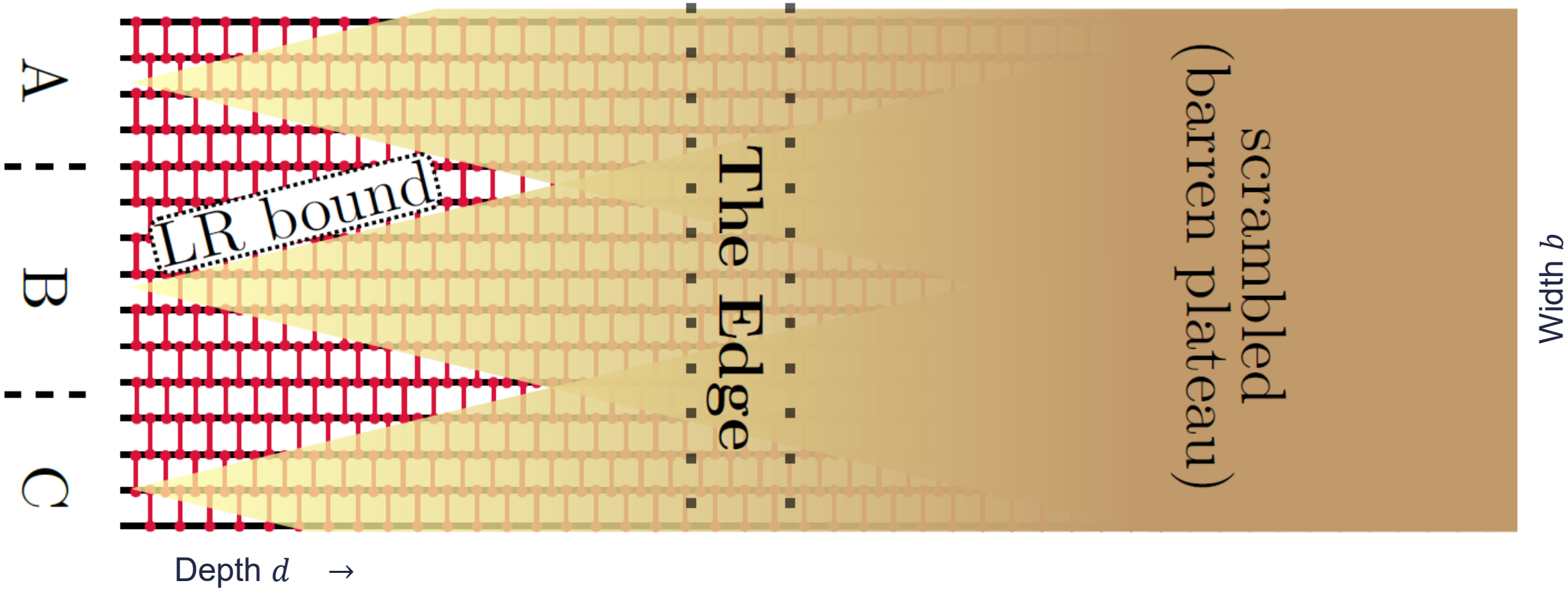


$$\mathcal{C} = f^{(1)^*} \prod_{j=1}^{r} \left(O_j f^{(j)} \right)$$

1. Prepare a variational state ($\propto n$)
2. Measure cost function via ancilla qubit
3. QNPU - evaluate the cost function ($\propto n$)

M. Lubasch, J. Joo, P. Moinier, M. Kiffner & DJ, Phys. Rev. A **101**, 010301(R) (2020).

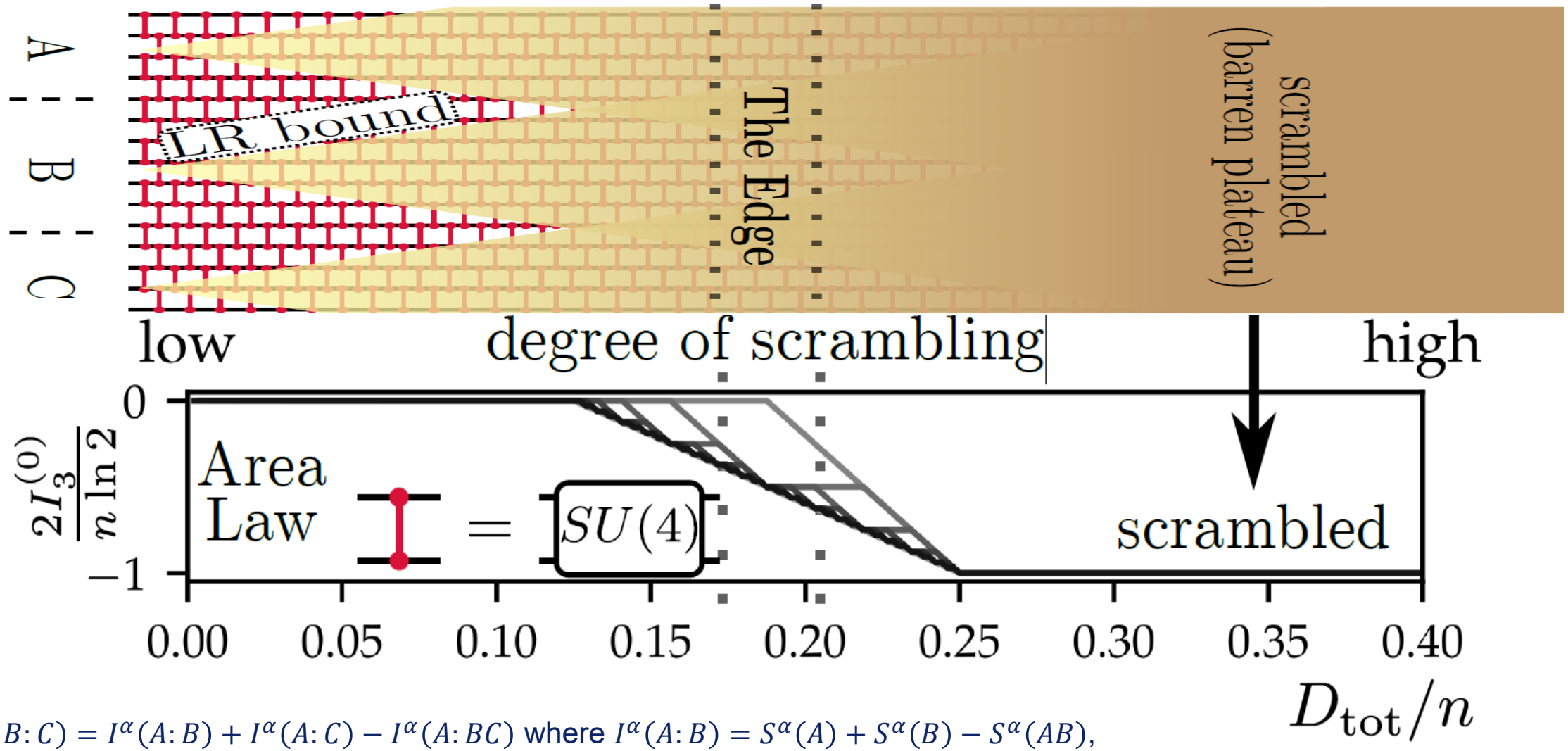
1 The edge of chaos



- The Lieb-Robinson bound gives the speed at which effects of gate parameters spread across the system
- When cones overlap the system becomes overdetermined and early gates identical to random circuit

T. Hashizume, et al., *Variational Quantum Computing at the edge of chaos*, in preparation.

1 Tripartite Mutual Information



$I_3^\alpha(A{:}B{:}C) = I^\alpha(A{:}B) + I^\alpha(A{:}C) - I^\alpha(A{:}BC)$ where $I^\alpha(A{:}B) = S^\alpha(A) + S^\alpha(B) - S^\alpha(AB)$,
$S^\alpha = \mathrm{tr}(\rho_A^\alpha)/(1-\alpha)$ and $\rho_A = \mathrm{tr}_{\bar{A}}(\rho)$

1 A quantum Nyquist-Shannon theorem

- We consider an amplitude encoded sin function in n qubits with $x \in [0,1)$ and $k = 2\pi/\lambda$

$$|\psi\rangle = \sum_i \sin(kx_i)|\sigma(i)\rangle$$

- We rewrite this using a tensor train decomposition

$$|\Psi_\lambda\rangle = \sum_{\boldsymbol{\sigma}} \sum_{\boldsymbol{\kappa}} M_{\kappa_0}^{(\sigma_0)} M_{\kappa_0,\kappa_1}^{(\sigma_1)} \dots M_{\kappa_{n-1}}^{(\sigma_{n-2})} |\boldsymbol{\sigma}\rangle$$

- With tensors

$$M_{\kappa_0}^{(0)} = \begin{pmatrix} \sin\frac{k}{2^n} & \cos\frac{k}{2^n} \end{pmatrix}, M_{\kappa_0}^{(1)} = \begin{pmatrix} 0 & 1 \end{pmatrix} \qquad M_{\kappa_0}^{(0)} = \begin{pmatrix} \cos\frac{k}{2} \\ \sin\frac{k}{2} \end{pmatrix}, M_{\kappa_0}^{(1)} = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

$$M_{\kappa_{q-1},\kappa_q}^{(0)} = \begin{pmatrix} \cos(\frac{k}{2^{n-q}}) & -\sin(\frac{k}{2^{n-q}}) \\ \sin(\frac{k}{2^{n-q}}) & \cos(\frac{k}{2^{n-q}}) \end{pmatrix}, M_{\kappa_{q-1},\kappa_q}^{(1)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- In the limit $n \to \infty$ we work out the reduced density operator of the q-th qubit

1 A quantum Nyquist-Shannon theorem

- We obtain for the reduced density operator

$$\rho_q = \lim_{n\to\infty} \sum_{\boldsymbol{\kappa},\boldsymbol{\kappa}',\boldsymbol{\sigma}\neq\sigma_q} M_{\kappa_0}^{(\sigma_0)} \dots M_{\kappa_{n-1}}^{(\sigma_{n-2})} (M_{\kappa_0'}^{(\sigma_0)})^* \dots (M_{\kappa_{n-1}'}^{(\sigma_{n-2})})^*$$

$$= \begin{pmatrix} \frac{2k - \sec(2^{-q-1}k)\sin((2+2^{-q-1})k) + \tan 2^{-q-1}k}{4k - 2\sin 2k} & \frac{2k\cos 2^{-q-1}k - \frac{\sin 2k}{\cos 2^{-q-1}k}}{(4k - 2\sin 2k)} \\ \frac{2k\cos 2^{-q-1}k - \frac{\sin 2k}{\cos 2^{-q-1}k}}{(4k - 2\sin 2k)} & \frac{2k - \sec(2^{-q-1}k)\sin((2-2^{-q-1})k) - \tan 2^{-q-1}k}{4k - 2\sin 2k} \end{pmatrix}$$

- This density operator converges to

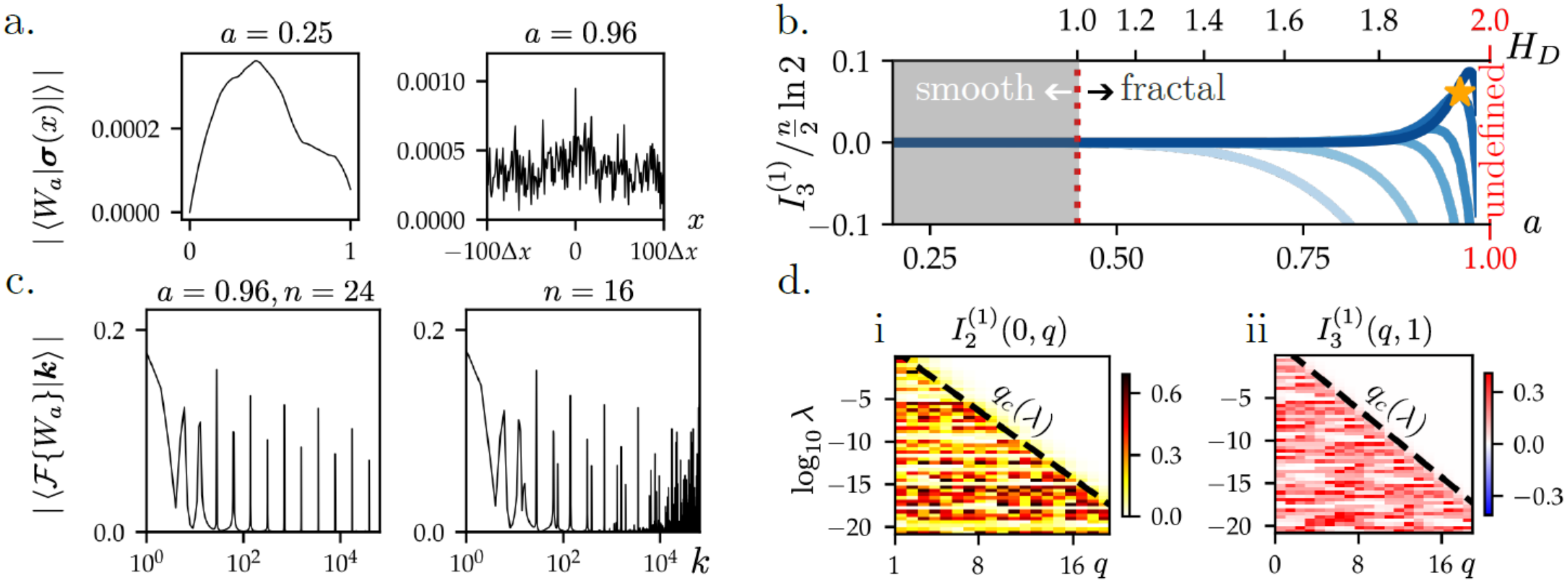
$$\lim_{q\to\infty} \rho_q = |+\rangle_q\langle+|$$

- exponentially with q for $q > q_c(\lambda)$ with

$$q_c(\lambda) = \log_2\left(\frac{\pi}{\lambda}\right)$$

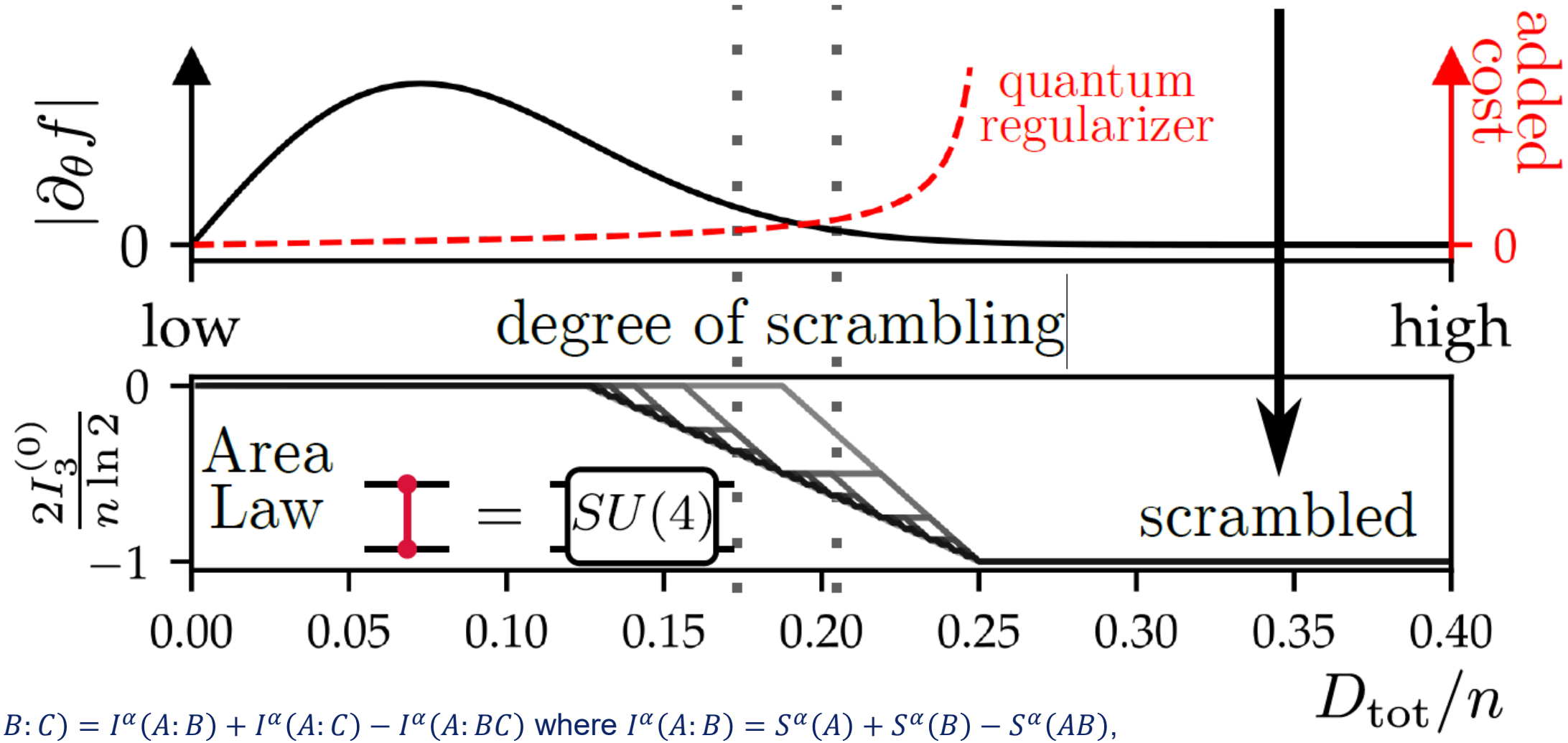
- A qubit state $|+\rangle_q\langle+|$ corresponds to linear interpolation and hence only q_c qubits are required in the TT.

1 Example: The Weierstrass function



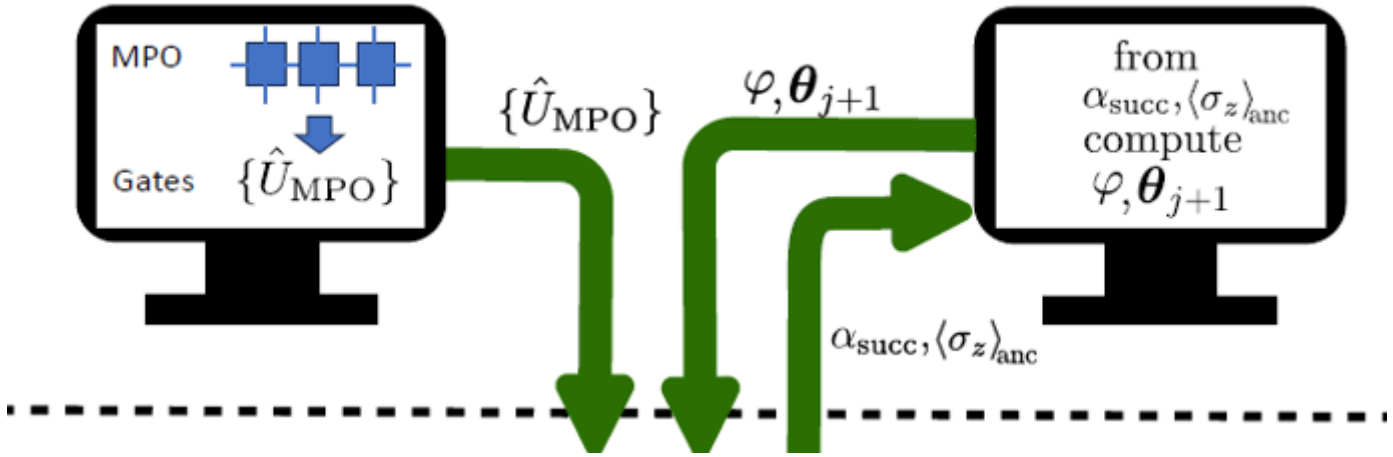
Weierstrass $W(x) = \sum_n a^n \sin(b^n \pi x)$, smooth for $a < \frac{1}{b}$ and fractal for $\frac{1}{b} < a < 1$. Here for $b = \sqrt{5}$.

1 Trainability and Tripartite Mutual Information

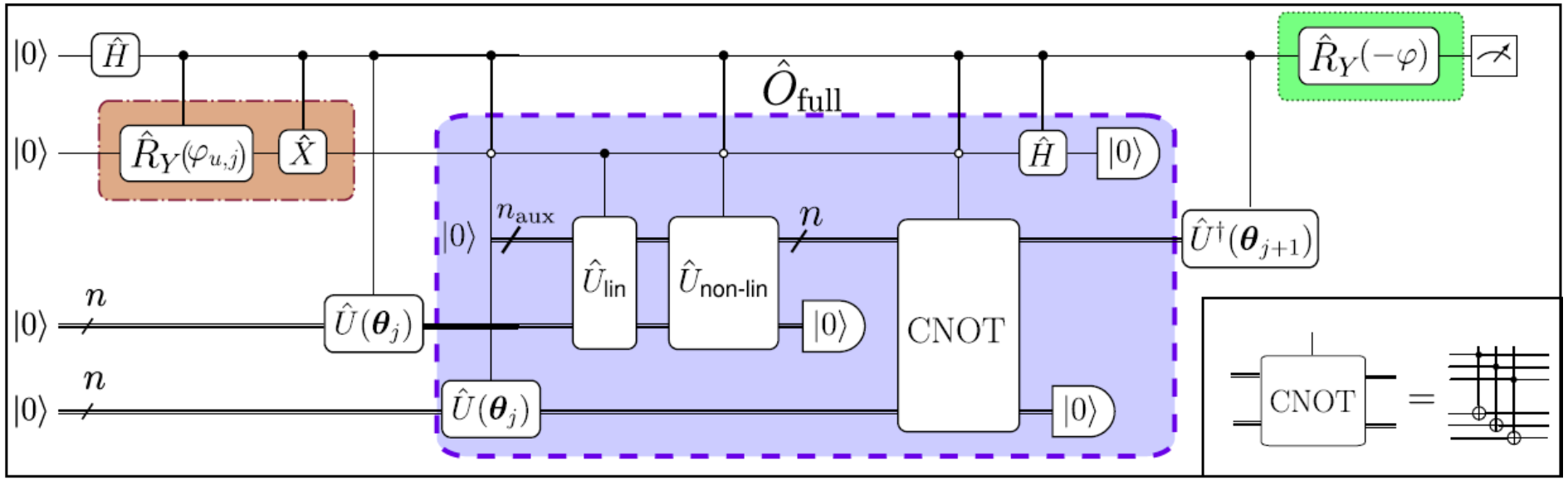


$I_3^\alpha(A{:}B{:}C) = I^\alpha(A{:}B) + I^\alpha(A{:}C) - I^\alpha(A{:}BC)$ where $I^\alpha(A{:}B) = S^\alpha(A) + S^\alpha(B) - S^\alpha(AB)$,
$S^\alpha = \mathrm{tr}(\rho_A^\alpha)/(1-\alpha)$ and $\rho_A = \mathrm{tr}_{\bar{A}}(\rho)$

2 3 Variational Quantum Algorithm



Termanova, et al., *Tensor quantum programming*, New J. Phys. **26**, 123019 (2024).



P. Siegl et al., Tensor-Programmable Quantum Circuits for Solving Differential Equations, arXiv:2502.04425.

2 Measure cost function via ancilla qubit

- Controlled application of network A

$$cA = |0\rangle\langle 0|\, 1 + |1\rangle\langle 1|\, A$$

- Hadamard applied to ancilla $H|0\rangle = |+\rangle \propto |0\rangle + |1\rangle$

$$cA|+\rangle|\psi\rangle \propto |0\rangle|\psi\rangle + |1\rangle A|\psi\rangle$$

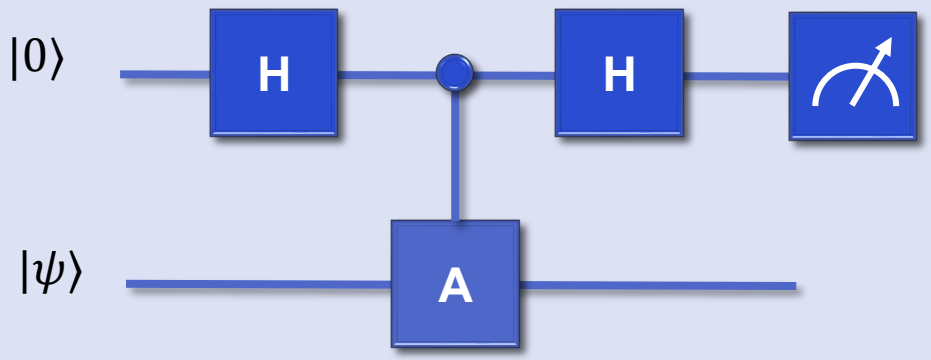
- After the second Hadamard $H|1\rangle = |-\rangle \propto |0\rangle - |1\rangle$

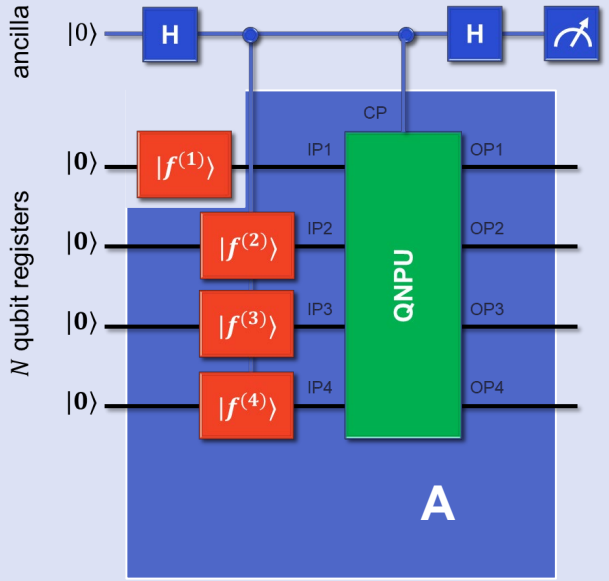
$$|+\rangle|\psi\rangle + |-\rangle A|\psi\rangle = |0\rangle(|\psi\rangle + A|\psi\rangle) + |1\rangle(|\psi\rangle - A|\psi\rangle)$$

- Measuring the ancilla qubit gives (using $(\sigma^z)^2 = 1$)

$$\langle\sigma^z\rangle \propto \langle\psi|A|\psi\rangle + \langle\psi|A^\dagger|\psi\rangle$$

- Note that A contains conditional networks of trial functions before IP2 to Ipn.





3 QNPU – create a nonlinear term

- Encode single qubit function (all real)

$$|f^1\rangle = |f^2\rangle = c_0|0\rangle + c_1|1\rangle$$

- Apply U to first qubit

$$|0\rangle_1 \rightarrow \ |\psi\rangle_1 = c_0|0\rangle_1 + c_1|1\rangle_1$$

- The controlled operation A is

$$A = CNOT_{12}\, U_2$$

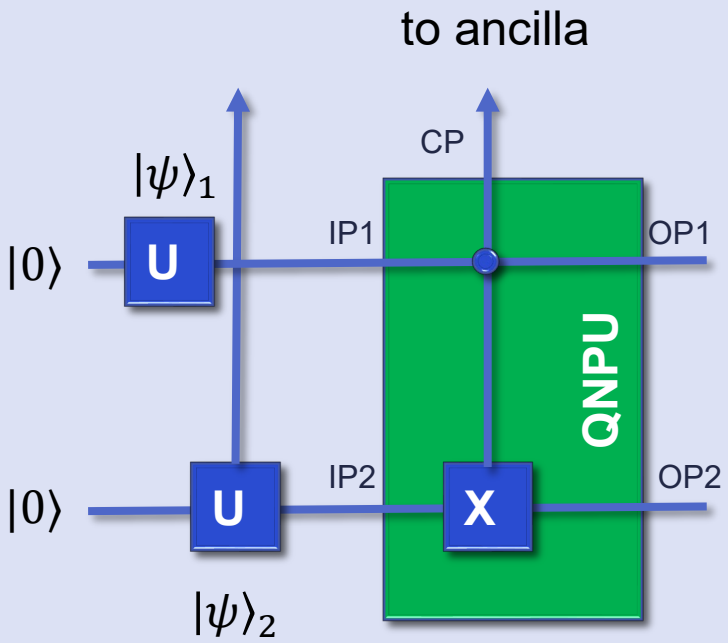
- The ancilla measurement yields

$$\langle\psi 0|A|\psi 0\rangle = \langle\psi 0|CNOT_{12}|\psi\psi\rangle$$

$$= \langle\psi 0|(c_0c_0|00\rangle + c_0c_1|01\rangle + c_1c_0|11\rangle + c_1c_1|10\rangle)$$
$$= (\langle 00|c_0 + \langle 10|c_1)(c_0c_0|00\rangle + c_1c_1|10\rangle)$$

$$= c_0^3 + c_1^3$$

- By adding more copies we get higher powers
- Exercise: create the term f^4 in the GPE.



3 QNPU – calculate a derivative

- We can use the adder network $\mathcal{A}$ to shift the function values by one This network takes
- The network input is the state

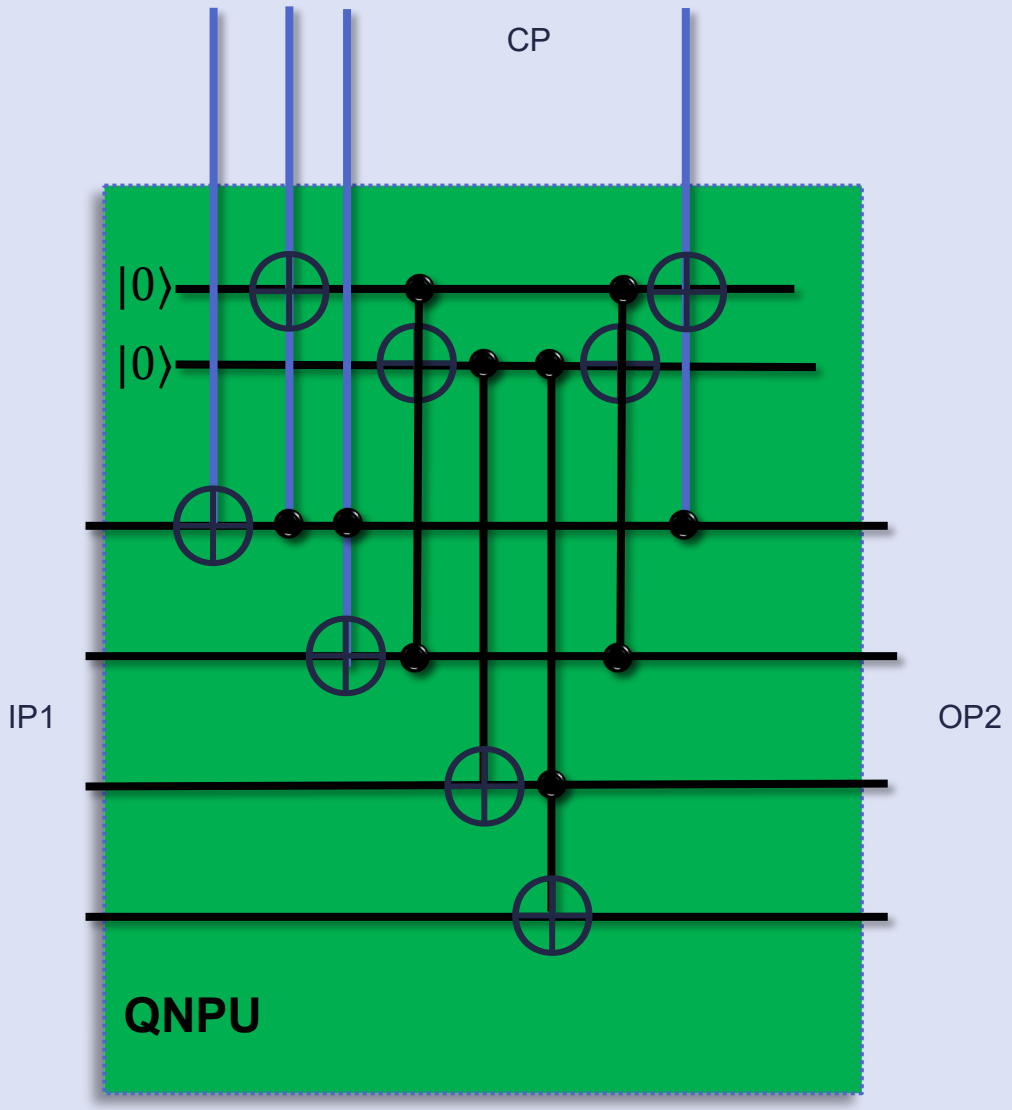
$$|f\rangle = \sum_i f_i |\vec{i}\,\rangle$$

- where $\vec{i} = i_3 i_2 i_1 i_0$ is the binary representation of i.
- IP1 and turns it into

$$|f'\rangle = \sum_i f_{i+1} |\vec{i}\,\rangle .$$

- The network thus evaluates

$$\sum_i f_i^* f_{i+1}$$



This network forms the basis for working out derivatives numerically

The adder network $\mathcal{A}$ explained

- The least significant qubit i_0 is at the top of the register and we assume the ancilla bit to be in state $|1\rangle$, i.e. all of the ancilla controlled CNOT gates can be active.
- If $i_0 = 1$ then it is flipped to $i_0 = 0$ by the first CNOT gate. It is then easy to see that all other CNOT gates will be inactive. Instead, if $i_0 = 0$ it will be flipped to $i_0 = 1$ by the first CNOT.
- The upper of the two auxiliary qubits is set to $c_1 = 1$ by the next CNOT, indicating a carry. The next CNOT then flips i_1.
- If this results in $i_1 = 0$ no further action is necessary and all other CNOT gates acting on the register will be inactive.
- Importantly, however, the last CNOT will reset the first carry qubit to $c_1 = 0$ to ensure that it is disentangled from the register.
- If $c_1 = 1$ and $b_1 = 1$ then further action is required and the second carry is set to $c_2 = 1$ by the next CNOT to indicate this. The following two CNOT gates then flip i_2 and i_3 as required for the operation $|\vec{i}\rangle \to |\overrightarrow{i-1}\rangle$.
- When the ancilla is in state $|0\rangle$ none of the CNOTs will act and hence the state of the quantum register will remain unchanged.
- The result on the previous slide then follows.

3 QNPU – multiplying the potential term

- In this case the internal QNPU network $\hat{V}$ of N auxiliary qubits creates a potential term of the form

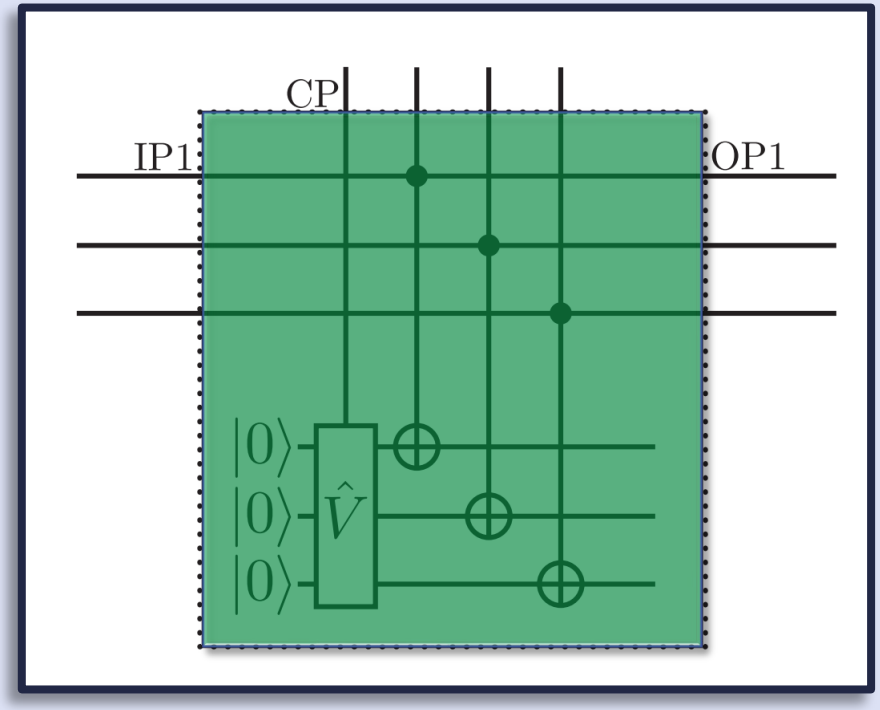
$$|V\rangle = \hat{V}|\mathbf{0}\rangle = \sum_i V_i |\vec{\imath}\rangle$$

- Here the controlled network is given by

$$A = \prod_n \mathrm{CNOT}_{nn'}\, \hat{V}$$

- where the first index labels qubits in IP1 and the second index the corresponding auxiliary qubit.
- The measurement of the ancilla qubit gives thus

$$\langle f\mathbf{0}|A|f\mathbf{0}\rangle = \left\langle f\mathbf{0} \middle| \prod_n \mathrm{CNOT}_{nn'} \middle| fV \right\rangle = \sum_i f_i^2 V_i$$



The action of the CNOT gates can be understood by considering a computational basis state:

$$\prod_n \mathrm{CNOT}_{nn'}\, |\vec{\imath}, \vec{\imath}'\rangle = |\vec{\imath}, \vec{\imath} \oplus \vec{\imath}'\rangle$$

Proof of Principle: the non-linear Schrödinger equation

- Non-linear PDE

$$\left[-\frac{1}{2}\frac{d^2}{dx^2} + V(x) + g\, f^2(x)\right] f(x) = E f(x)$$

- Cost function

$$\mathcal{C} = \langle\langle K \rangle\rangle_c + \langle\langle P \rangle\rangle_c + \langle\langle I \rangle\rangle_c$$

- We estimate $\mathcal{C}$ from grid kinetic, potential and non-linear terms which leads to a grid error

$$\langle\langle \cdot \rangle\rangle_c = \langle\langle \cdot \rangle\rangle + \mathcal{E}_{\text{grid}}$$

- The error $\epsilon_{\text{grid}} \propto 1/N^2$

- Kinetic energy (h_N is the grid spacing)

$$\langle\langle K \rangle\rangle = -\frac{1}{2h_N^2}\sum_k f_k^*(f_{k+1} - 2\, f_k + f_{k-1})$$

- Potential energy

$$\langle\langle P \rangle\rangle = \sum_k f_k^* V(k) f_k$$

- Nonlinear term

$$\langle\langle I \rangle\rangle = \frac{g}{2h_n}\sum_k f_k^4$$

Measuring the cost function

- Kinetic energy

$$\langle\langle K\rangle\rangle = \frac{1 - \langle\hat{\sigma}_z\rangle_{anc}^{K}}{h_N^2}$$

- Potential energy (α rescales the potential)

$$\langle\langle P\rangle\rangle = \alpha\langle\hat{\sigma}_z\rangle_{anc}^{P}$$

- Nonlinear term

$$\langle\langle I\rangle\rangle = \frac{g}{2h_n}\langle\hat{\sigma}_z\rangle_{anc}^{I}$$

- Note that measuring the ancilla performs summation over all grid points at no extra computational cost

- Relative sampling error

$$\epsilon_{MC}^{P} = \frac{C_P}{\sqrt{M}}$$

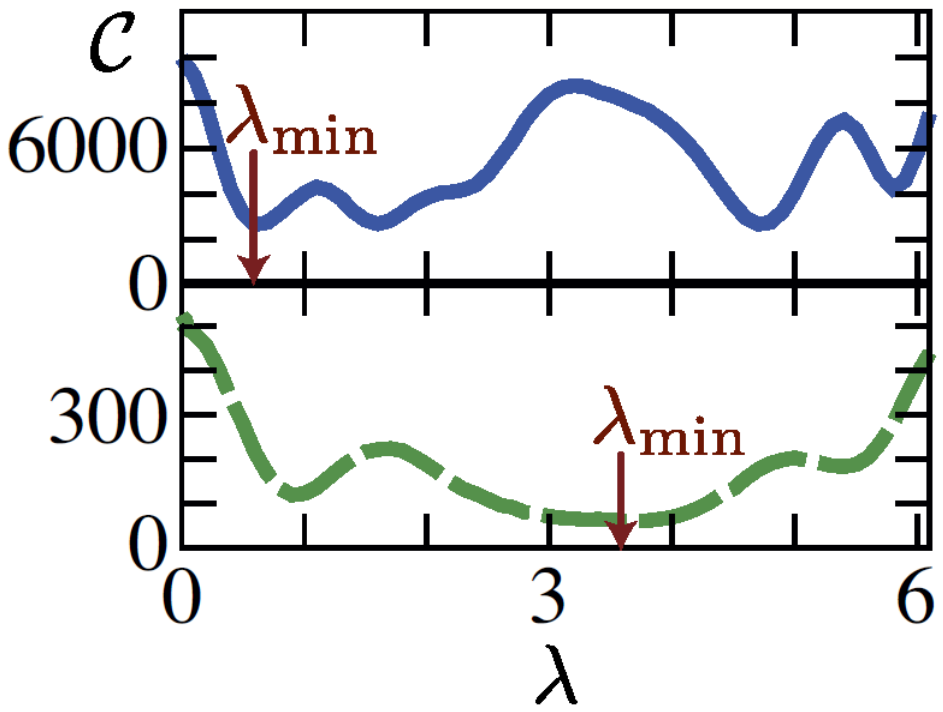
$$\epsilon_{MC}^{K} = \frac{C_K}{\sqrt{M}}\frac{N}{N_{\min}}$$

$$\epsilon_{MC}^{I} = \frac{C_I}{\sqrt{M}}\frac{N}{N_{\min}}$$

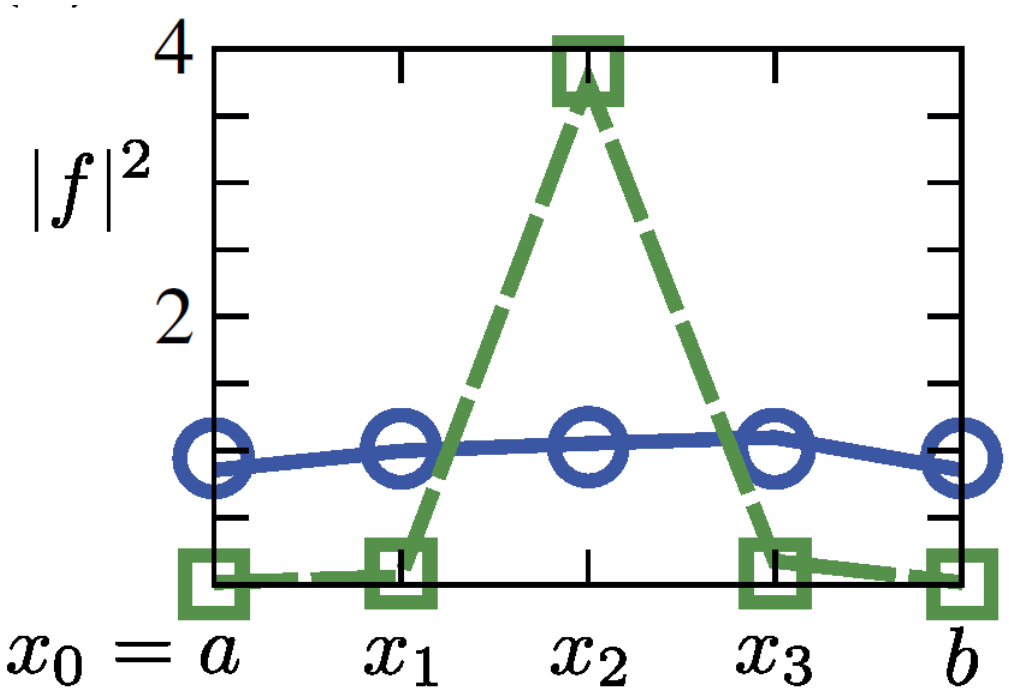
- C_X is of order 1 in all cases
- $N_{\min}$ is the minimum grid size required to obtain an accurate approximation of the solution.

Proof of Principle Example: Trapped BEC on 4 grid points

- Variational minimization of total energy



$g = 10^4$ $g = 10$



BEC in harmonic trap with a single variational parameter λ

QCFD consortium and collaborations beyond

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Dieter Jaksch@UHH
Thomas Rung@TUHH
Felix Motzoi@Juelich
Barbara Kraus@TUM
Dimitris Angelakis@TUC
Paolo Geremia@ENGYS
Martin Kiffner@PlanQC

Steven@Cornell

Nik@OXF

Peyman@PIT

Michael@Q

The QCFD project is funded under the EU's Horizon Programme (HORIZON-CL4-2021-DIGITAL-EMERGING-02-10), Grant Agreement 101080085.

QCFD (qcfd-h2020.eu)

AFOSR AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

CUI THE HAMBURG CENTRE FOR ULTRAFAST IMAGING
Advanced Imaging of Matter

Related publications

- **MPS encoding and Multigrid Renormalization**
  - M. Lubasch, P. Moinier and D. Jaksch, *Multigrid Renormalization,* J. Comp. Phys. **372**, 587 (2018)
- **Quantum Algorithms for nonlinear optimization**
  - M. Lubasch, J. Joo, P. Moinier, M. Kiffner and D. Jaksch, *Variational Quantum Algorithms for Nonlinear Problems,* Phys. Rev. A **101**, 010301(R) (2020)
- **QCFD and tensor network algorithms**
  - N. Gourianov, M. Lubasch, S. Dolgov, Q.Y. van den Berg, H. Babaee, P. Givi, M. Kiffner and D. Jaksch, *A quantum-inspired approach to exploit turbulence structures,* Nature Computational Science **2**, 30 (2022).
  - D. Jaksch, P. Givi, A.J. Daley and T. Rung, *Variational Quantum Algorithms for Computational Fluid Dynamics*, AIAA Journal **61**, 1885 (2023)
  - M. Kiffner and D. Jaksch, *Tensor network reduced order models for wall-bounded flows*, Phys. Rev. Fluids **8**, 124101 (2023)
  - P. Over, S. Bengoechea, T. Rung, F. Clerici, L. Scandurra, E. de Villiers, D. Jaksch, *Boundary treatment for variational quantum simulations of partial differential equations on quantum computers,* Computers & Fluids **288**, 106508 (2025)
  - N. Gourianov, P. Givi, D. Jaksch, and S.B. Pope, *Tensor networks enable the calculation of turbulence probability distributions*, Sci. Adv. **11**, eads5990 (2025)
  - S. Bengoechea, P. Over, D. Jaksch, and T. Rung, *Towards Variational Quantum Algorithms for generalized linear and nonlinear transport phenomena*, arXiv:2411.14931 (2024)
  - P. Siegl, G.S. Reese, T. Hashizume, N.-L. van Hülst, and D. Jaksch, *Tensor-Programmable Quantum Circuits for Solving Differential Equations*, arXiv:2502.04425 (2025*)*
  - Nis-Luca van Hülst, Pia Siegl, Paul Over, Sergio Bengoechea, Tomohiro Hashizume, Mario Guillaume Cecile, Thomas Rung, Dieter Jaksch, *Quantum-Inspired Tensor-Network Fractional-Step Method for Incompressible Flow in Curvilinear Coordinates,* arXiv:2507.05222 (2025)


QS