High Energy Physics

Outlook

Introduction

Lattice Gauge Theory (LGT)

The codes

Benchmarking results

BKeeper results

SOMBRERO results

Summary

Benchmarking lattice quantum field theory codes across EuroHPC platforms

Frédéric Bonnet[1], Ryan Hill[2] and Ed Bennett[1]

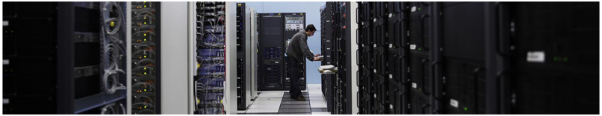
[1]School of Mathematical and Computer Science
Swansea University Bay Campus, Fabian Way, Swansea, SA1 8EN, Wales, UK.

[2]School of Physics and Astronomy
University of Edinburgh, James Clerk Maxwell Building, Peter Guthrie Tait Road, Edinburgh, EH9 3FD, UK.

September 17, 2025

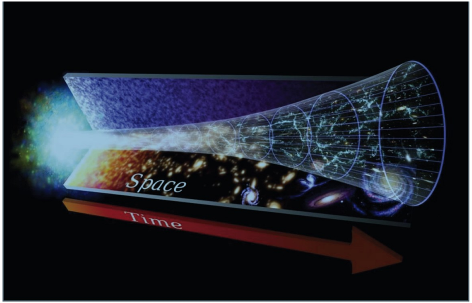
Content

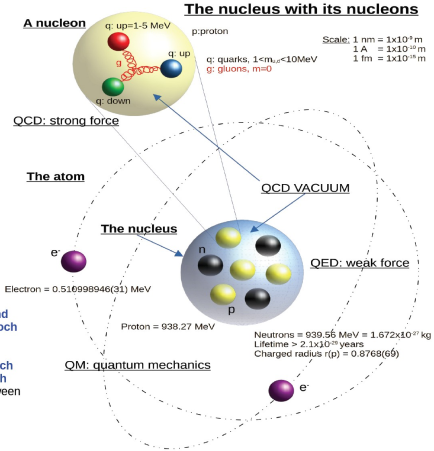
- Introduction.
- Lattice Gauge Theory (LGT).
- The codes.
- The benchmarking (BKeeper,SOMBRERO) results
- Summary.



Introduction.

- From Mega $\longrightarrow$ Tera $\longrightarrow$ PetaFlops/s $\longrightarrow$ ExaScale computing.
- As computing power increases we can simulate larger systems and more realistic.
- The code complexity increases with the both the scale and size of the problems.
- Lattice Gauge Theory (LGT) require a large amount of computing power resources.
- It requires a large code development resources gathered from computer scientists, physicists and mathematicians.
- LGT is a very powerful ab-initio numerical tool to study non–perturbative physics when analytical methods are simply not possible.
- On the lattice we can study particles, light Hadron spectrum, gauge dependent quantities (propagators), energy transfers between particles, form factors, QCD vacuum, topology of the gauge groups, composite models (physics Beyond the Standard Model) ....
- We then compare the results with experimental data coming from CERN-LHC, JLab and/or other particle accelerator, or large scale experiments.
- LGT works in both ways:
  1. Provides insights to experimental for energy scale probes.
  2. Experimental data is used to create more accurate models and confirm theory.





- **Very Early Universe** (speculative): **Planck epoch** (up to 10^{-43} sec after big bang) → **Grand unification epoch** (between $[10^{-43} \rightarrow 10^{-36}]$ secs) → **Electroweak epoch**, **Inflationary epoch** (between $[10^{-36} \rightarrow 10^{-12}]$ secs).

- **Early Universe**: **Supersymmetry breaking** (energy not lower than 1 TeV) → **Quark epoch** (between $[10^{-12} \rightarrow 10^{-6}]$ secs) → **Hadron epoch** (between $[10^{-6} \rightarrow 1]$ secs) → **Lepton epoch** (between $[1 \rightarrow 10]$ secs) → **Photon epoch**: **neucleosynthesis**, **matter domination** (between $[10 \rightarrow 380{,}000]$ secs).

- Scale is important and can determine what kind of mathematical framework one can use.

- Quark scale (strong forces) → Lagrangian non-abelian field theory: Quantum Chromodynamics (QCD) → 10×10^{-18}m
- Nuclear scale (Weak force) → Lagrangian abelian field theory: Quantum electrodynamics (QED) → 10×10^{-15}m
- Atomic scale (Electromagnetic force) → Hamiltonian theory: Quantum Mechanics (QM) → 10×10^{-10}m
- Nano scale (Electrostastic, magnetic, molecular, interparticle forces) → Hamiltonian theory (QM), statistical physics (Stat. Phys.) → 10×10^{-9}m

Proton size: 0.8768(69)(**fm**), **Protein volume:** $V(nm^3) = 1.212 \, x \, 10^{-3} (nm^3/Da) \times M(Da) \rightarrow R_{min} = \left(\frac{3V}{4\pi}\right)^{1/3} (nm)$

The space between is mostly particle free, example nucleus = 1m → first orbiting electron 50 Km away!

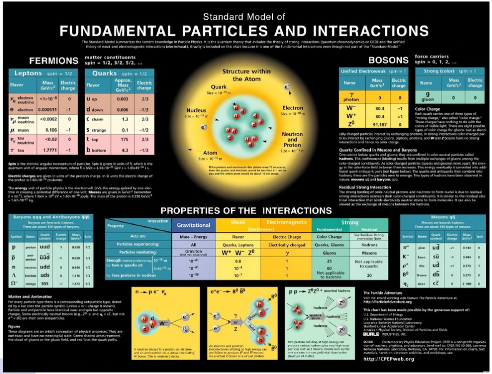
LQCD(2003):$V=24^3 x48$ @ a=0.100(fm) → Phys Vol 2.4^3 x 4.8 (fm) 10^2 configs max!! **Pure gauge and full QCD (2+1) O(a) mostly**
- LQCD(2019):$V=64^3 x192$ @ a=0.068(1) < 0.1(fm) → Phys Vol 4.352^3x 13.056 (fm) >> 10^3 configs… **Full QCD (2+1+1, u,d,s and c) O(a^{2+})**

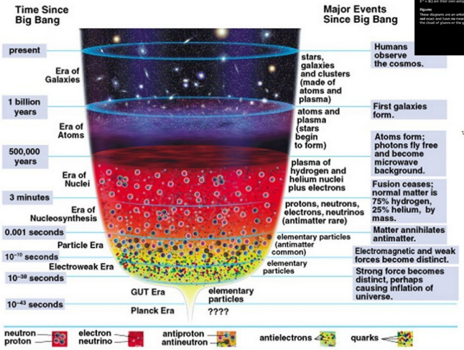
The entire universe is made up of these fundamental particles.
Physics model the world differently to mathematics (boundaries, infinities, object contacts etc…)

Theory used and framework needed usually determines what kind computing resources one will require, it is governed:
  - disk, CPU, memory, network intensive or all of these.

$$\begin{aligned}
\mathcal{L} = & -\frac{1}{4}B_{\mu\nu}B^{\mu\nu} - \frac{1}{8}tr(\mathbf{W}_{\mu\nu}\mathbf{W}^{\mu\nu}) - \frac{1}{2}tr(\mathbf{G}_{\mu\nu}\mathbf{G}^{\mu\nu}) && \text{(U(1), SU(2) and SU(3) gauge terms)} \\
& +(\bar{\nu}_L, \bar{e}_L)\,\tilde{\sigma}^\mu iD_\mu \begin{pmatrix} \nu_L \\ e_L \end{pmatrix} + \bar{e}_R\sigma^\mu iD_\mu e_R + \bar{\nu}_R\sigma^\mu iD_\mu \nu_R + \text{(h.c.)} && \text{(lepton dynamical term)} \\
& -\frac{\sqrt{2}}{v}\left[(\bar{\nu}_L, \bar{e}_L)\,\phi M^e e_R + \bar{e}_R \bar{M}^e \bar{\phi}\begin{pmatrix} \nu_L \\ e_L \end{pmatrix}\right] && \text{(electron, muon, tauon mass term)} \\
& -\frac{\sqrt{2}}{v}\left[(-\bar{e}_L, \bar{\nu}_L)\,\phi^* M^\nu \nu_R + \bar{\nu}_R \bar{M}^\nu \phi^T\begin{pmatrix} -e_L \\ \nu_L \end{pmatrix}\right] && \text{(neutrino mass term)} \\
& +(\bar{u}_L, \bar{d}_L)\,\tilde{\sigma}^\mu iD_\mu \begin{pmatrix} u_L \\ d_L \end{pmatrix} + \bar{u}_R\sigma^\mu iD_\mu u_R + \bar{d}_R\sigma^\mu iD_\mu d_R + \text{(h.c.)} && \text{(quark dynamical term)} \\
& -\frac{\sqrt{2}}{v}\left[(\bar{u}_L, \bar{d}_L)\,\phi M^d d_R + \bar{d}_R \bar{M}^d \bar{\phi}\begin{pmatrix} u_L \\ d_L \end{pmatrix}\right] && \text{(down, strange, bottom mass term)} \\
& -\frac{\sqrt{2}}{v}\left[(-\bar{d}_L, \bar{u}_L)\,\phi^* M^u u_R + \bar{u}_R \bar{M}^u \phi^T\begin{pmatrix} -d_L \\ u_L \end{pmatrix}\right] && \text{(up, charmed, top mass term)} \\
& +\overline{(D_\mu\phi)}D^\mu\phi - m_h^2[\bar{\phi}\phi - v^2/2]^2/2v^2. && \text{(Higgs dynamical and mass term)}
\end{aligned}$$

Fig: Standard Model Lagrangian





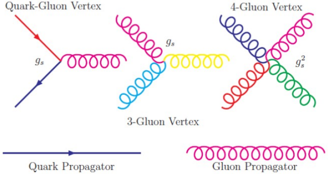
$$\begin{aligned}
S_{\mathrm{QCD}} &= \int d^4x \mathcal{L} \\
&= \int d^4x \left(-\frac{1}{4}F^a_{\mu\nu}(x)F^{a\mu\nu}(x)\sum_q \bar{\psi}^i_q(x)(i\gamma_\mu D_\mu(x) - m_q)_{ij}\psi^j_q(x)\right), \qquad (1.1)
\end{aligned}$$

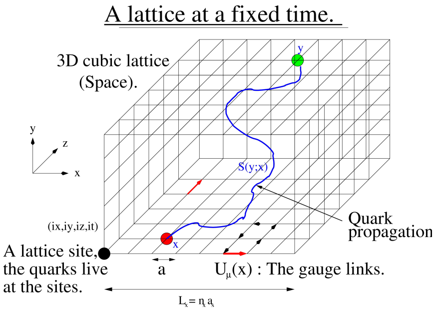
with the non–Abelian electromagnetic and covariant derivative expressed as:

$$F^a_{\mu\nu}(x) = \partial_\mu A^a_\nu(x) - \partial_\nu A^a_\mu(x) + g_s f_{abc} A^b_\mu(x) A^c_\nu(x), \qquad (1.2)$$

$$(D_\mu(x))_{ij} = \delta_{ij}\partial_\mu - ig_s \sum \frac{\lambda^a_{ij}}{2} A^a_\mu(x). \qquad (1.3)$$

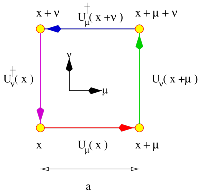
Fig: QCD





Lattice Gauge Theory (LGT)

- Is a discretisation of the continuum on a hypercubic volume with periodic boundary conditions.
- At each lattice site we attach a gauge group with a tunable lattice spacing.
- All of the dynamics are contained in the action functional S[A(x), q] constructed from plaquettes



- We then compute a path integral using Monte Carlo techniques, a heat–bath and/or Hybrid-MonteCarlo (Hamiltonian).
- Sample configurations are then collected to compute an observable(s).

The codes

In this study we have used 2 underling main codes[1]:

- `Grid`: software used to do computation in QFT. (GPU-CPU).
- `HiRep`: LGT software used for physics beyond the standard model (BSM). (CPU)

The benchmark codes used are:

- `SOMBRERO`: Sits on top of `HiRep` and is a benchmarking mini-app (CPU) and probes:
  - case-1: $SU(2)$ with adjoint matter
  - case-2: $SU(2)$ with fundamental matter
  - case-3: $SU(3)$ with fundamental matter (QCD)
  - case-4: $Sp(4)$ with fundamental matter
  - case-5: $SU(3)$ with two-index symmetric matter
  - case-6: $Sp(4)$ with adjoint matter
- `BKeeper`: Instantiates classes from `Grid` framework on an extended range of Gauge groups. XML driven to match `SOMBRERO` cases.

The frameworks for the benchmark:

- `Bench_Grid_HiRep` framework: Software used to deploy over a set of HPC cluster. It provides an automated way to deploy and launch simulation from a remote host.

[1] https://github.com/paboyle/Grid, https://github.com/claudiopica/HiRep, https://github.com/sa2c/sombrero, https://github.com/RChrHill/BKeeper, https://github.com/fbonnet08/Bench_Grid_HiRep

Outlook

Introduction

Lattice Gauge Theory (LGT)

The codes

Benchmarking results

BKeeper results

SOMBRERO results

Summary

The benchmarking results

We ran the codes on a benchmark call EHPC-BEN-2024B10-015 then EHPC-BEN-2025B03-046:

- Vega(Slovenia):
  - [Vega-CPU: 2x(64c AMD EPYC 7H12-2.6-3.3GHz), 256GB-1TB RAM, local 1.91TB M.2 SSD]
  - [Vega-GPU: 4x(NVIDIA-A100), 2x(64c AMD EPYC 7H12), 512GB RAM, local 1.91TB M.2 SSD]
- Leonardo(CINECA-Italy):
  - [Leonardo-DCGP: 2x(56c Saphire 8358-2.0GHz), 512GB RAM]
  - [Leonardo-Booster: 4x(NVIDIA-A100), 1x(32c Xeon 8358-2.6GHz), 512GB RAM]
- Lumi(Finland):
  - [Lumi-C: 2x(64c AMD EPYC 7763-2.45-3.5GHz), 256-512-1024GB RAM]
  - [Lumi-G: 8x(AMD-MI250), 1x(64c AMD EPYC 7A54), 8x(64GB DDR4) RAM]
- MareNostrum-5(Spain):
  - [MeraNostrum-5-GPP: 2x(56c Xeon 9480-1.9GHz), 8x(16GB HBM2), local 960GB NVMe]
  - [MareNostrum-5-ACC: 4x(NVIDIA-H100), 2x(40c Xeon 8460Y-2GHz), 16x(32GB DDR5)]
- EHPC-BEN-2024B10-015: $\longrightarrow$ {BKeeper, SOMBRERO} $\forall$ systems:{Vega, Leonardo, Lumi}.
- EHPC-BEN-2025B03-046: $\longrightarrow$ {BKeeper, LLR} $\forall$ systems:{Vega, Leonardo, Lumi, MareNostrum-5}.
- Cases: $\longrightarrow$ {small, large, weak, strong} $\forall$ systems: in all calls.
- LLR $\longrightarrow$ {weak, strong, volume} $\forall$ systems on EHPC-BEN-2025B03-046 only.

The figures can be found in https://doi.org/10.5281/zenodo.15782266.

BKeeper[2] results

We screened a set of MPI combinations over the nodes with a given number of GPU available on that node over space–time:

$$\begin{aligned} M &= \left\{ \Gamma = \prod_{i,j,k,j=0}^{n_{\rm gpu}} x_i y_j z_k t_l \;\middle|\; \Gamma = n_{\rm gpu}, \quad \text{and} \quad i,j,k,l = 1,..,n_{\rm gpu} \in \mathbb{Z}^+ \right\} \\ &\longrightarrow \quad M = \{x_i, y_j, z_k, t_l\}_{i,j,k,l=1,..,n_{\rm gpu}} = [1, n_{\rm gpu}] \in \mathbb{Z}^+. \end{aligned} \tag{1}$$

- Small lattice: $24^3 x32 \Longrightarrow n_{\rm nodes} = \{1, 2\}$.
- Large lattice: $32^3 x64$ and $64^3 x96 \Longrightarrow n_{\rm nodes} = \{4, 8, 12, 16, 24, 32\}$.
- Lattice spacing (a) for the conjugate gradient was set at its default size: $a = 0.1(fm)$
- EHPC-BEN-2024B10-015: ⟶Vega, Leonardo and Lumi.
- EHPC-BEN-2025B03-046: ⟶Vega, Leonardo, Lumi and MareNostrum-5.

[2] https://doi.org/10.5281/zenodo.15782266

1 and 2 Nodes (separated) `Vega-GPU` and `Leonardo-Booster`

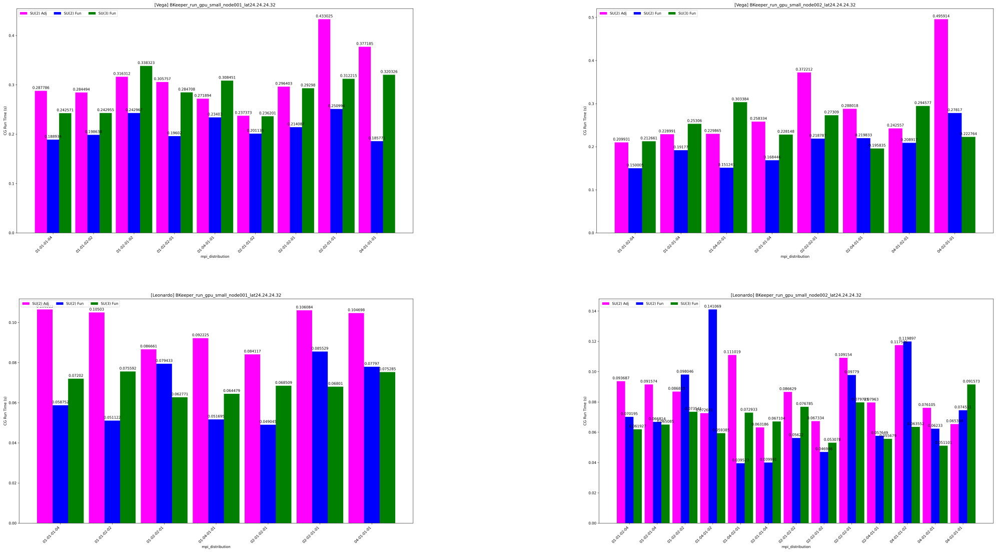


Figure: $SU(2)$–Adjoint & $SU(2,3)$–Fundamental using BKeeper (24^3x32). a). Vega-GPU node001. b). Vega-GPU node002. c). Leonardo-Booster node001. d). Leonardo-Booster node002. x-axis: mpi_distribution ⟷ y-axis: CG Run Time (s).

Outlook
Introduction
Lattice Gauge Theory (LGT)
The codes
Benchmarking results
BKeeper results
SOMBRERO results
Summary

1 and 2 Nodes (separated) `Lumi-G`

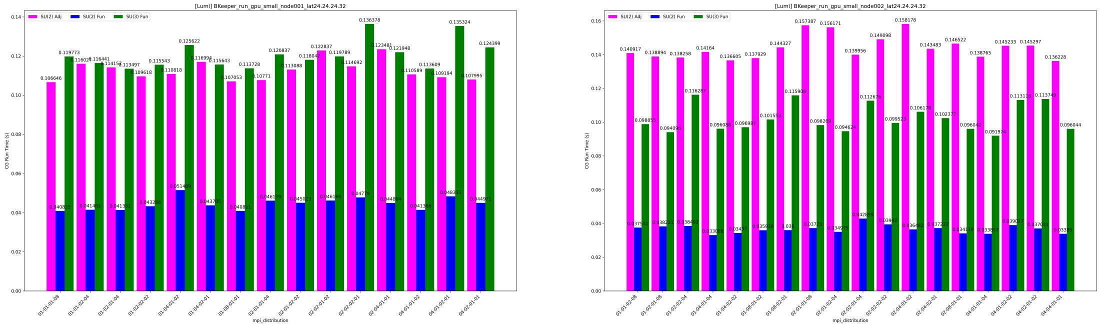


Figure: $SU(2)$–Adjoint and $SU(2,3)$–Fundamental using `BKeeper` ($24^3 x32$). a). `Lumi-G node001`. b). `Lumi-G node002`. x-axis: `mpi_distribution` $\longleftrightarrow$ y-axis: `CG Run Time (s)`.

- Large disparity in the run times, but $\exists : \{\min, \max\} \; \forall$ Representations.
- The $\{\min\} \implies$ best MPI distribution on that n_{nodes} for a given system.

1 and 2 Nodes (combined) `Vega-GPU` and `Leonardo-Booster`

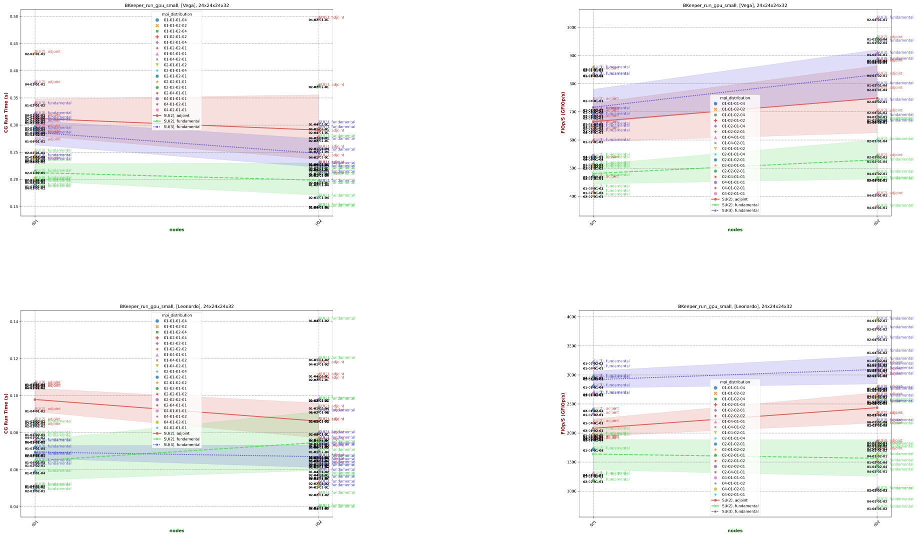
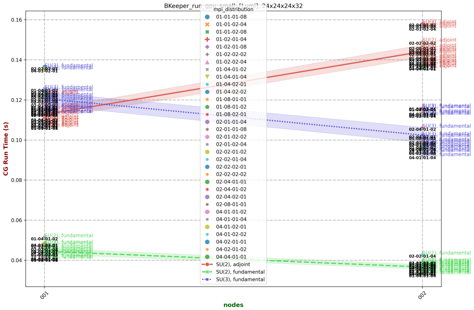


Figure: $SU(2)$–Adjoint, $SU(2{,}3)$–Fundamental [node001, node002], Small lattice $24^3 \times 32$. a). `Vega-GPU`, (`CG Run Time (s)`). b). `Vega-GPU`, (`Flops/s (GFlop/s)`). c). `Leonardo-Booster`, (`CG Run Time (s)`). d). `Leonardo-Booster`, (`Flops/s (GFlop/s)`).

1 and 2 Nodes (combined) `Lumi-G`



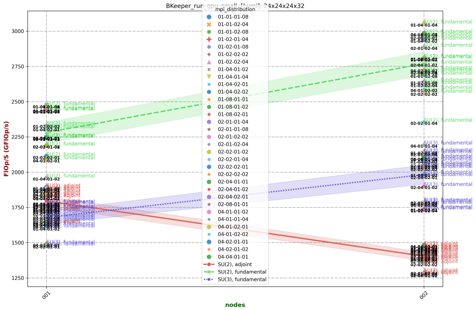


Figure: $SU(2)$–Adjoint & $SU(2,3)$–Fundamental on nodes [node001, node002], Small lattice $24^3 \times 32$. a). Lumi-G, (CG Run Time (s)). b). Lumi-G, (Flops/s (GFlop/s)).

4, 8, 12 Nodes (combined) Vega-GPU

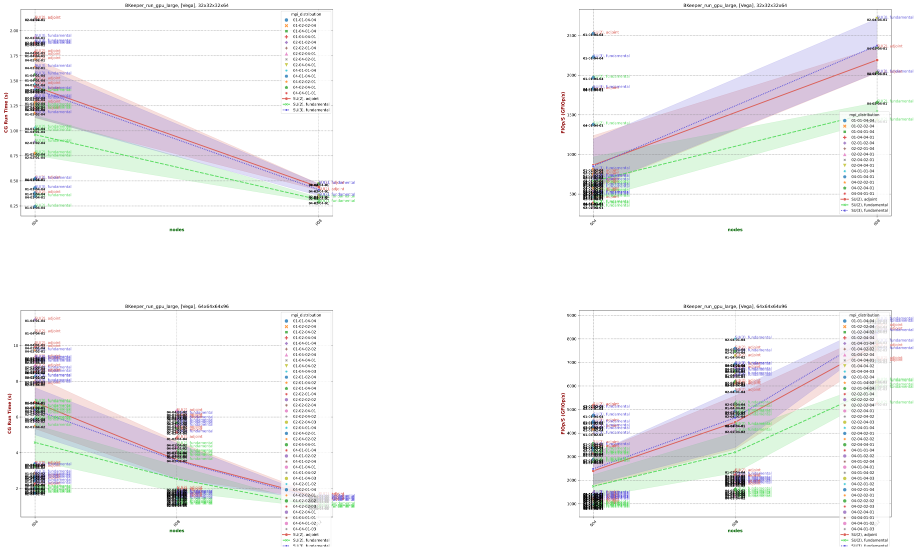


Figure: $SU(2)$–Adjoint & $SU(2,3)$–Fundamental on $n_{\text{nodes}} = \{4, 8, 12\}$. a). Vega-GPU, (CG Run Time (s)) ($32^3 \times 64$). b). Vega-GPU, (Flops/s (GFlop/s)) ($32^3 \times 64$). c). Vega-GPU, (CG Run Time (s)) ($64^3 \times 96$). d). Vega-GPU, (Flops/s (GFlop/s)) ($64^3 \times 96$).

4, 8, 12, 16, 24 and 32 Nodes (combined) `Lumi-G`

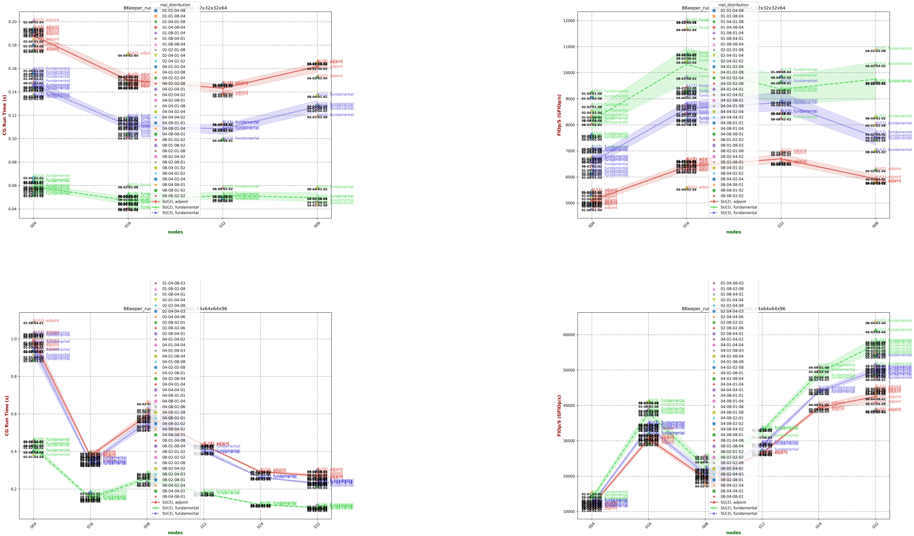


Figure: $SU(2)$–Adjoint & $SU(2,3)$–Fundamental on $n_{\text{nodes}} = \{4, 8, 12, 16, 24, 32\}$. a). `Lumi-G`, (`CG Run Time (s)`) $(32^3 \times 64)$. b). `Lumi-G`, (`Flops/s (GFlop/s)`) $(32^3 \times 64)$. c). `Lumi-G`, (`CG Run Time (s)`) $(64^3 \times 96)$. d). `Lumi-G`, (`Flops/s (GFlop/s)`) $(64^3 \times 96)$.

SOMBRERO[3] results

Sombrero is a CPU only code, we screened by varying $n_{\rm nodes} = \{1, 2, 4\}$ and number of tasks per node as $n_{\rm ntpns} = \{1, 2, 3, 6, 12, 24, 48, 96\}$.

- We are looking for communication.
- Looking at scaling: [Weak, Strong] for all case-1:-case-6:

Case	Gauge Group 2	Representation	Matrix	n–elements
case-1:	$SU(2)$	with adjoint matter	2×2	4
case-2:	$SU(2)$	with fundamental matter	2×2	4
case-3:	$SU(3)$	with fundamental matter (QCD)	3×3	9
case-4:	$Sp(4)$	with fundamental matter	4×4	16
case-5:	$SU(3)$	with two-index symmetric matter	3×3	9
case-6:	$Sp(4)$	with adjoint matter	4×4	16

[3] https://doi.org/10.5281/zenodo.15782266

[case-1:-case-4: Strong, mpi_distribution] Leonardo-DCGP

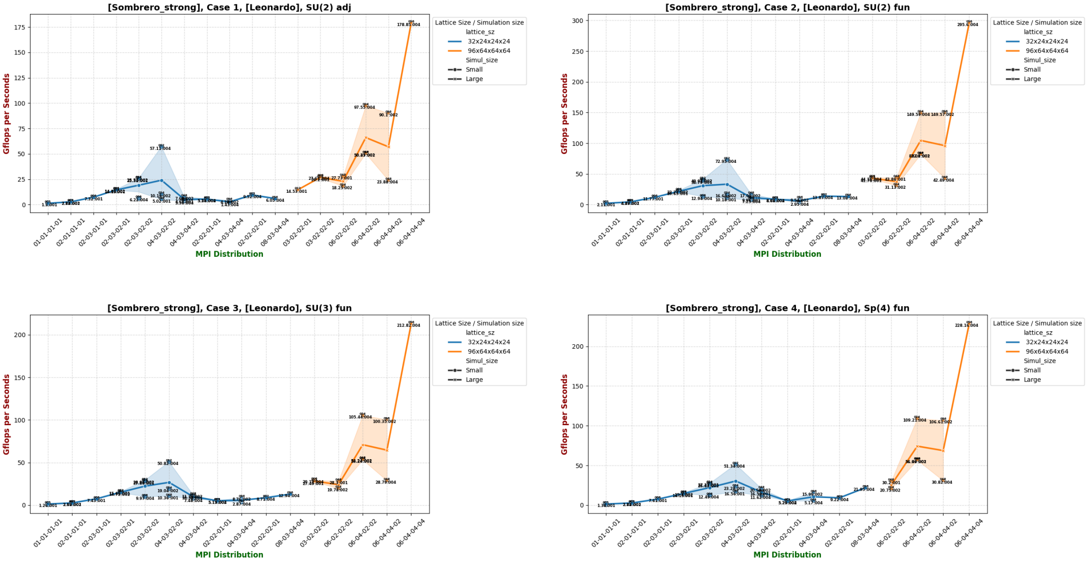


Figure: SOMBRERO on $n_{\mathrm{nodes}} = \{1, 2, 4\}$, Leonardo-DCGP, Strong cases. a). (case-1: $SU(2)$ adj). b). (case-2: $SU(2)$ fun). c). (case-3: $SU(3)$ fun). d). (case-4: $Sp(4)$ fun).

[case-5:-case-6: Strong, mpi_distribution] Leonardo-DCGP

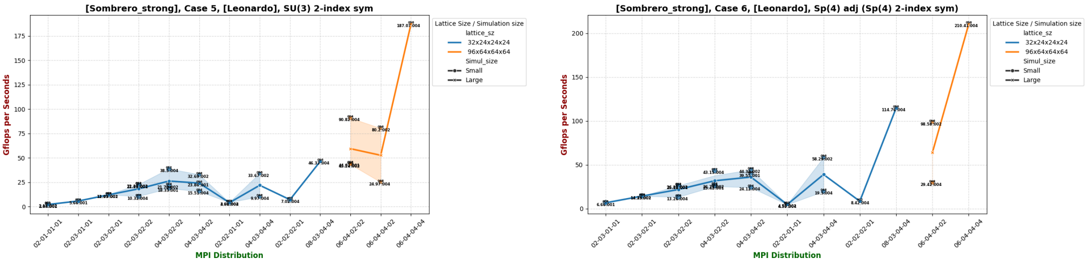


Figure: SOMBRERO on $n_{\text{nodes}} = \{1, 2, 4\}$, Leonardo-DCGP, Strong cases. a). (case-5: $SU(3)$ adj). b). (case-6: $Sp(4)$ adj).

Outlook

Introduction

Lattice Gauge Theory (LGT)

The codes

Benchmarking results

BKeeper results

SOMBRERO results

Summary

[case-1:-case-4: Strong, nodes] Leonardo-DCGP

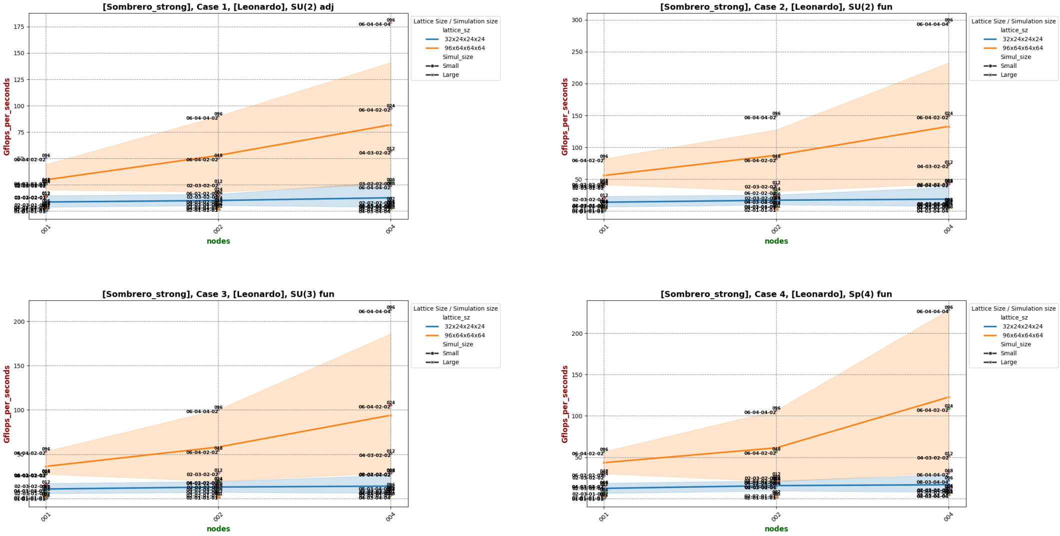


Figure: SOMBRERO on $n_{\text{nodes}} = \{1, 2, 4\}$ on Leonardo-DCGP for the Strong cases. a). (case-1: $SU(2)$ adj). b). (case-2: $SU(2)$ fun). c). (case-3: $SU(3)$ fun). d). (case-4: $Sp(4)$ fun).

[case-5:-case-6: Strong, nodes] Leonardo-DCGP

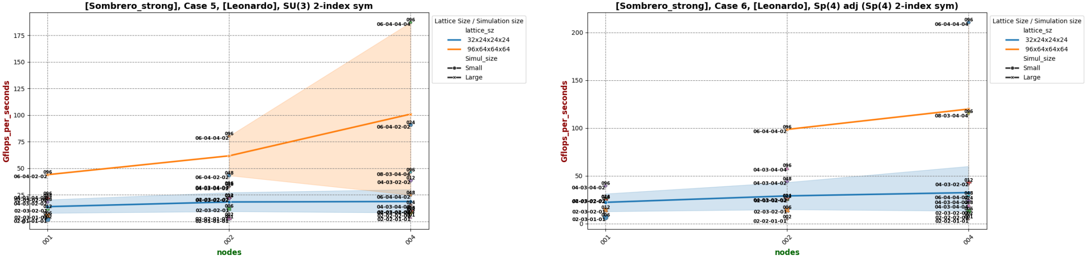


Figure: SOMBRERO on $n_{\text{nodes}} = \{1, 2, 4\}$ on Leonardo-DCGP for the Strong cases. a). (case-5: $SU(3)$ adj). b). (case-6: $Sp(4)$ adj).

Outlook

Introduction

Lattice Gauge Theory (LGT)

The codes

Benchmarking results

BKeeper results

SOMBRERO results

Summary

Summary

- We have tested our codes on EuroHPC machines: Vega, Lumi, Leonardo & MareNostrum-5.
- The test were done on a 3–month benchmark EuroHPC cycles: call EHPC-BEN-2024B10-015 mostly, but also on EHPC-BEN-2025B03-046.
- In all cases, we observed that the mpi_distribution has large impact on the overall performances.
- It is then possible to work out using warm up runs on $n_{\mathrm{nodes}} = \{1, 2\}$ for best set of mpi_distribution on a particular machine.
- We can also see which machine provides best performances for a particular problem at hand.
- The framework Bench_Grid_HiRep can deploy, launch and analyse very fast a large set of cases, and remotely from a simple host laptop to any machine with given access.

MULTI-GPU PORTING OF A PHASE-CHANGE CASCADED LATTICE

Xander de Wit, Alessandro Ga

Thrid EuroHP

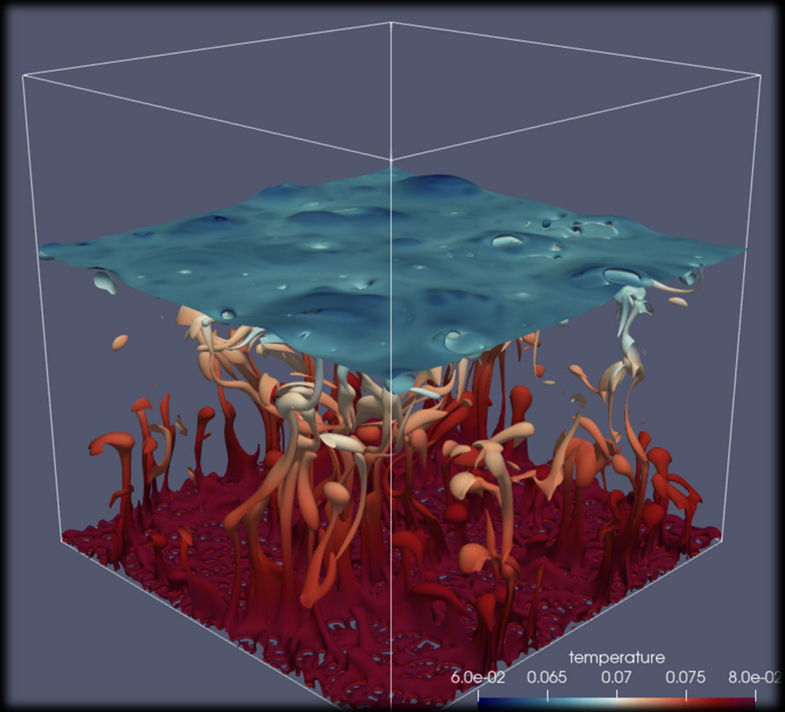
EuroHPC Joint Undertaking

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Los A NATIONAL

KEY CHALLENGES FOR *IN SILICIO* BOILING

- Vast **dynamic range** of active **space** and **time** scales
- Handle large **density** ratios
- Resolve **interfaces**
- Collective dynamics is **3D by nature**



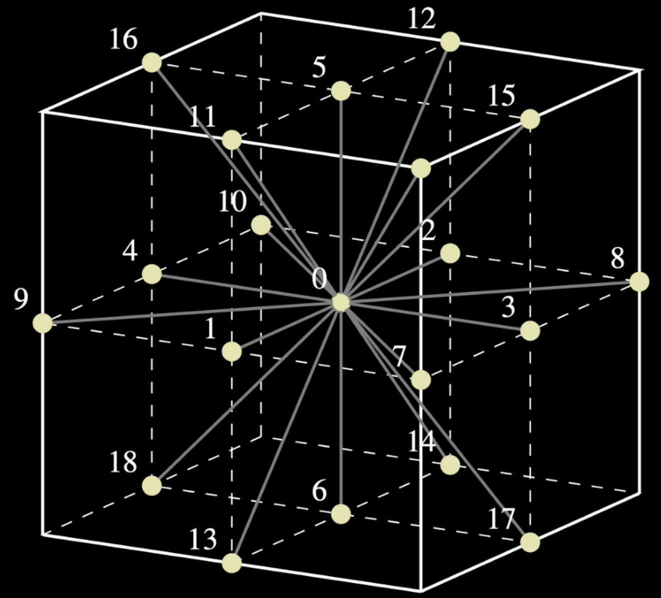
- So far mostly 2D or reduced models
- ***Recent HPC advances enable fully resolved 3D boiling***

EuroHPC Joint Undertaking

ESSENTIALS OF THE *LATTICE BOLTZMAN METHOD*

Solves Boltzmann eq. for **quasi-particle** distribution in **position-momentum space** $f(\mathbf{r}, \mathbf{p}, t)$

$$\frac{\partial f}{\partial t} + \frac{\mathbf{p}}{m} \cdot \nabla f + \mathbf{F} \cdot \frac{\partial f}{\partial \mathbf{p}} = \left(\frac{\partial f}{\partial t}\right)_{\mathrm{coll}}$$

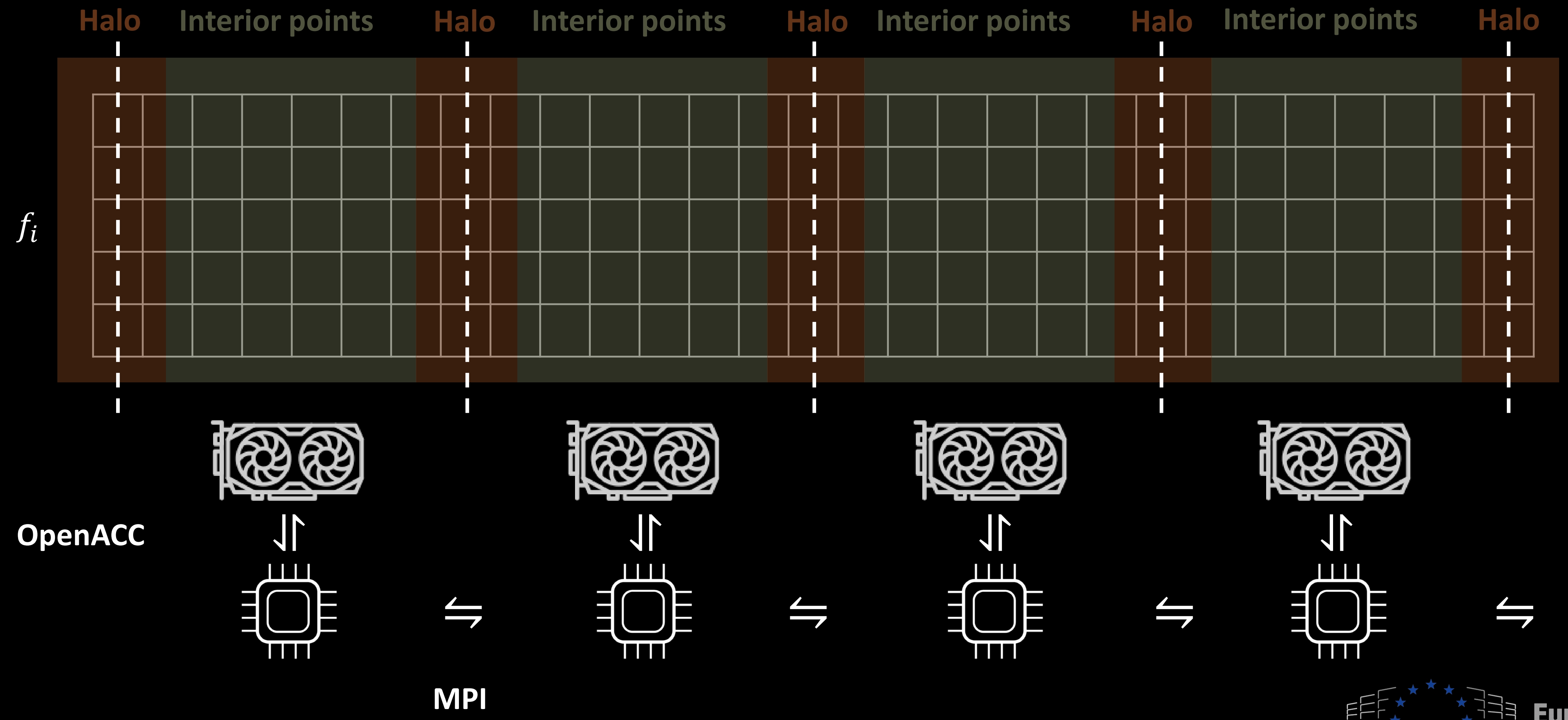


Approaches Navier-Stokes equation in continuum limit

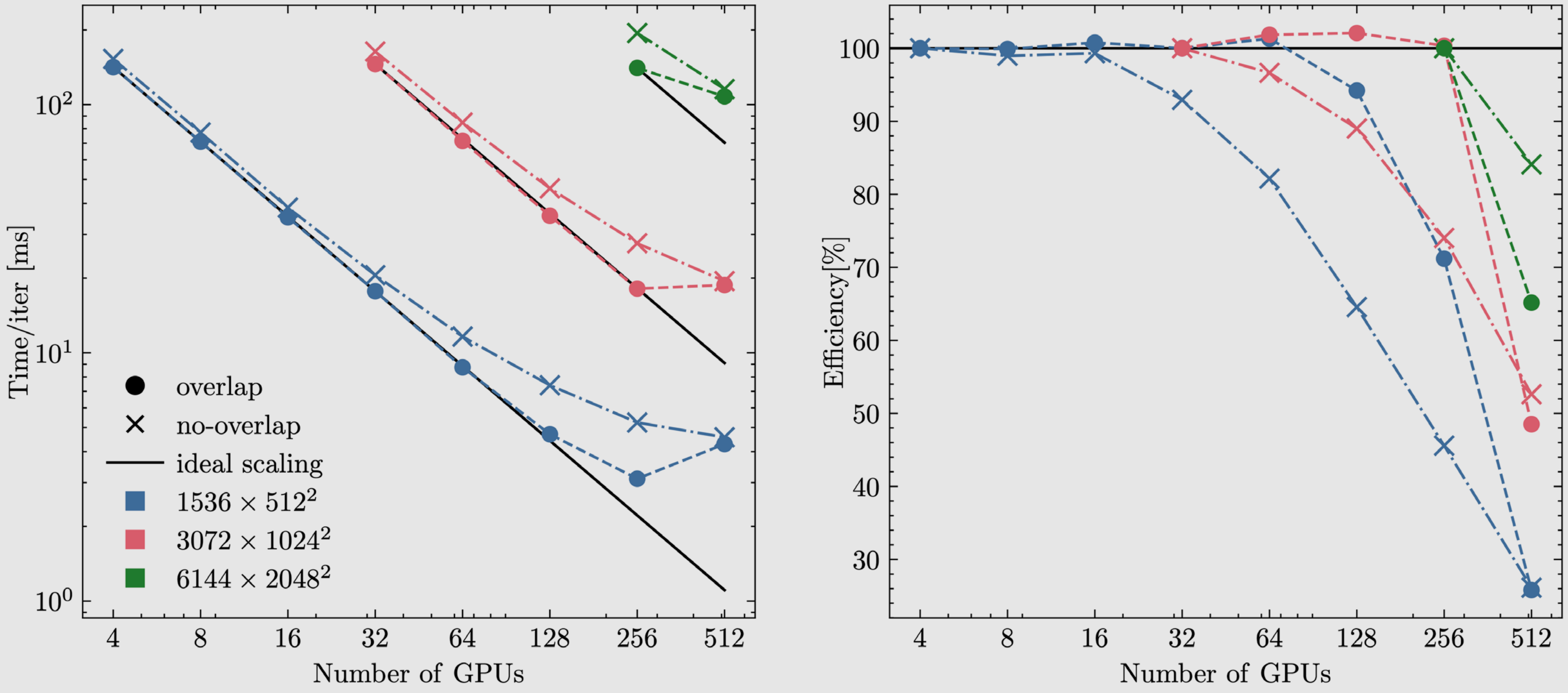
Position-momentum space is discretized on a lattice:

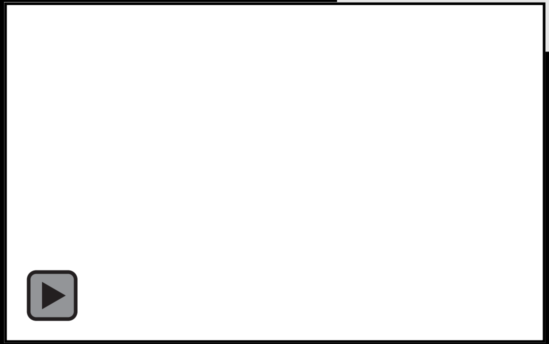
1. **Streaming step (*communication* between lattice points)**
2. **Collision step (expensive, but fully *local*)**

NUMERICAL IMPLEMENTATION



PERFORMANCE

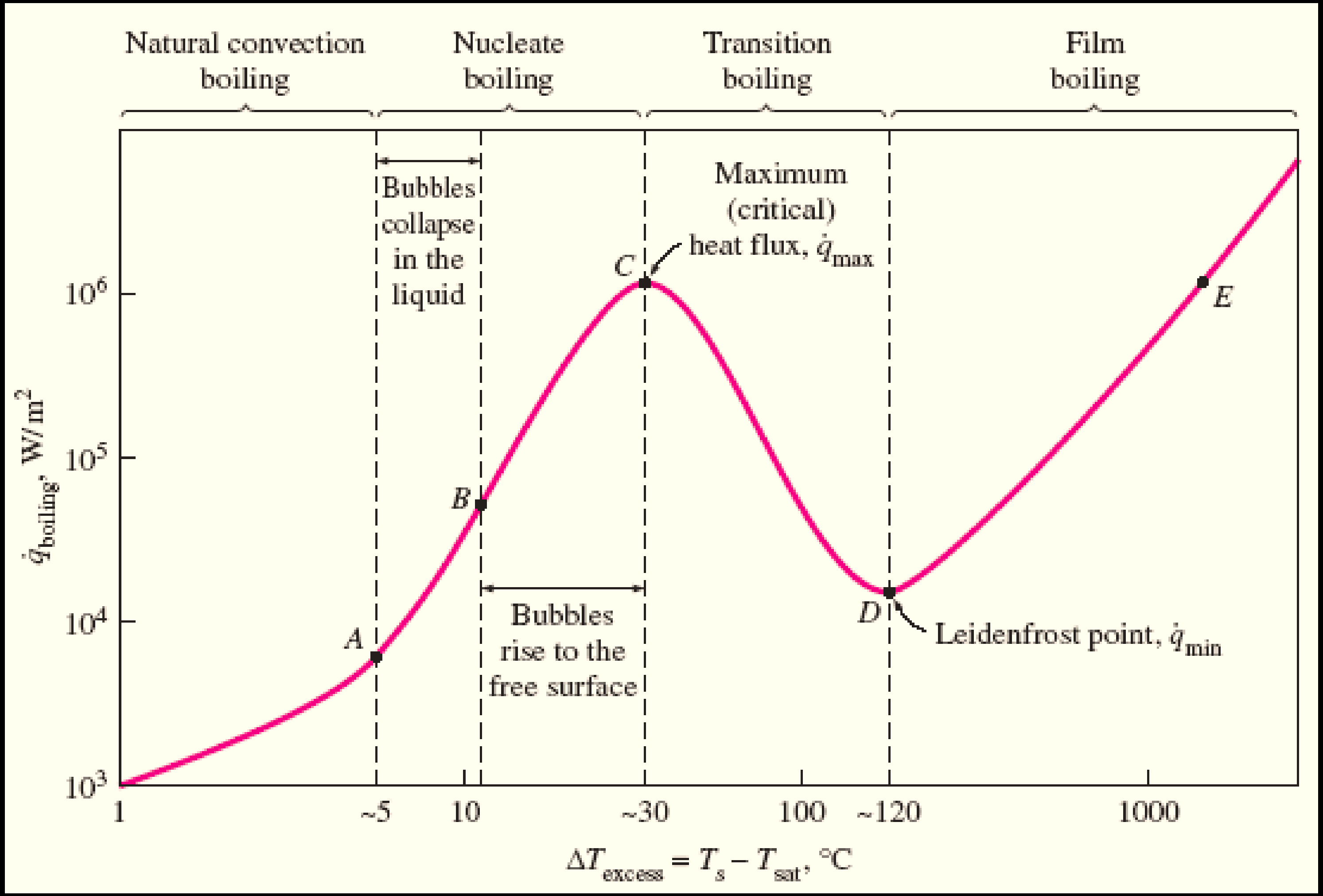




Results optained on EuroHPC
MareNostrum 5

EuroHPC
Joint Undertaking

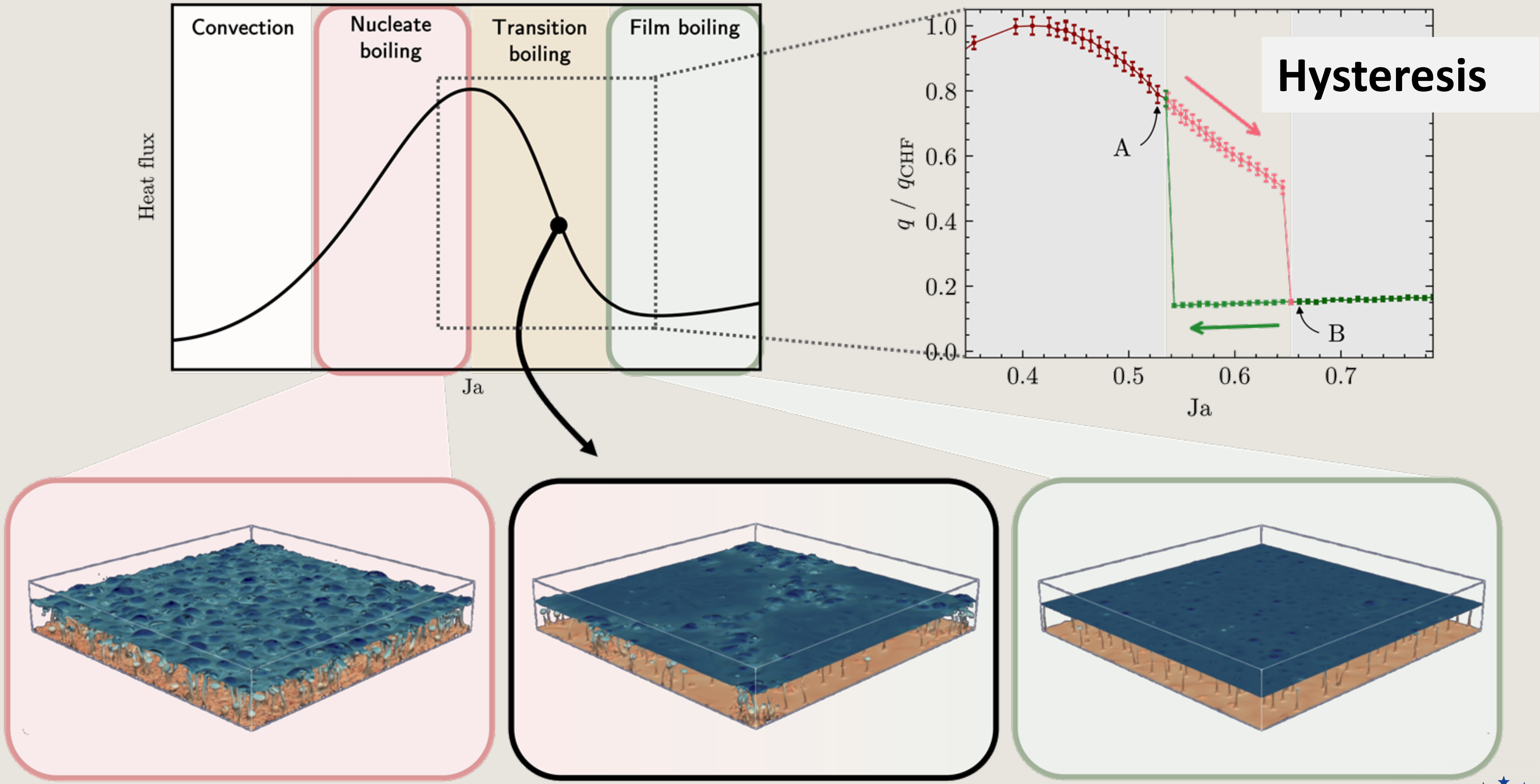
PHYSICS OF BOILING



Faghri & Zhang (2006)

TRANSITION BOILING

TRANSITION BOILING

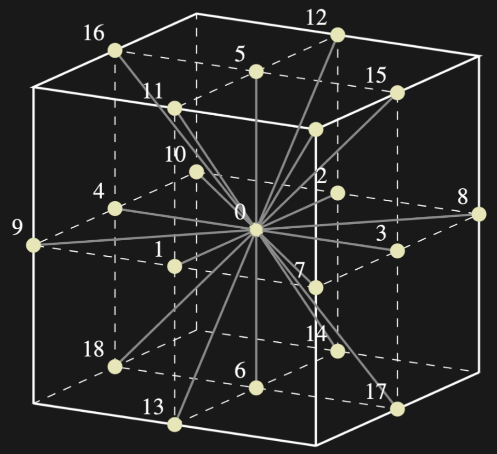


EuroHPC Joint Undertaking

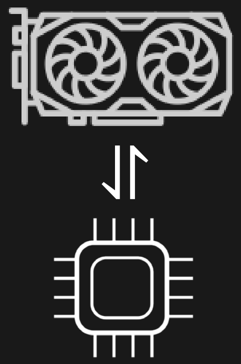
CONCLUSIONS

Advances in HPC enable investigation of **fully resolved boiling in 3D**

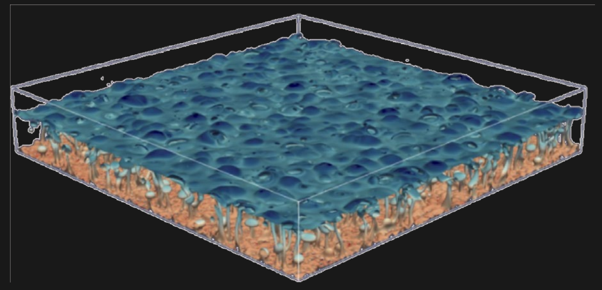
Lattice Boltzmann method with **phase change**



Interleaved GPU/CPU parallelization using **OpenACC** and **MPI**



Reveals novel insights: **hysteresis** in **transition boiling**



EuroHPC Joint Undertaking

THANK YOU!





TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Los Alamos NATIONAL LABORATORY

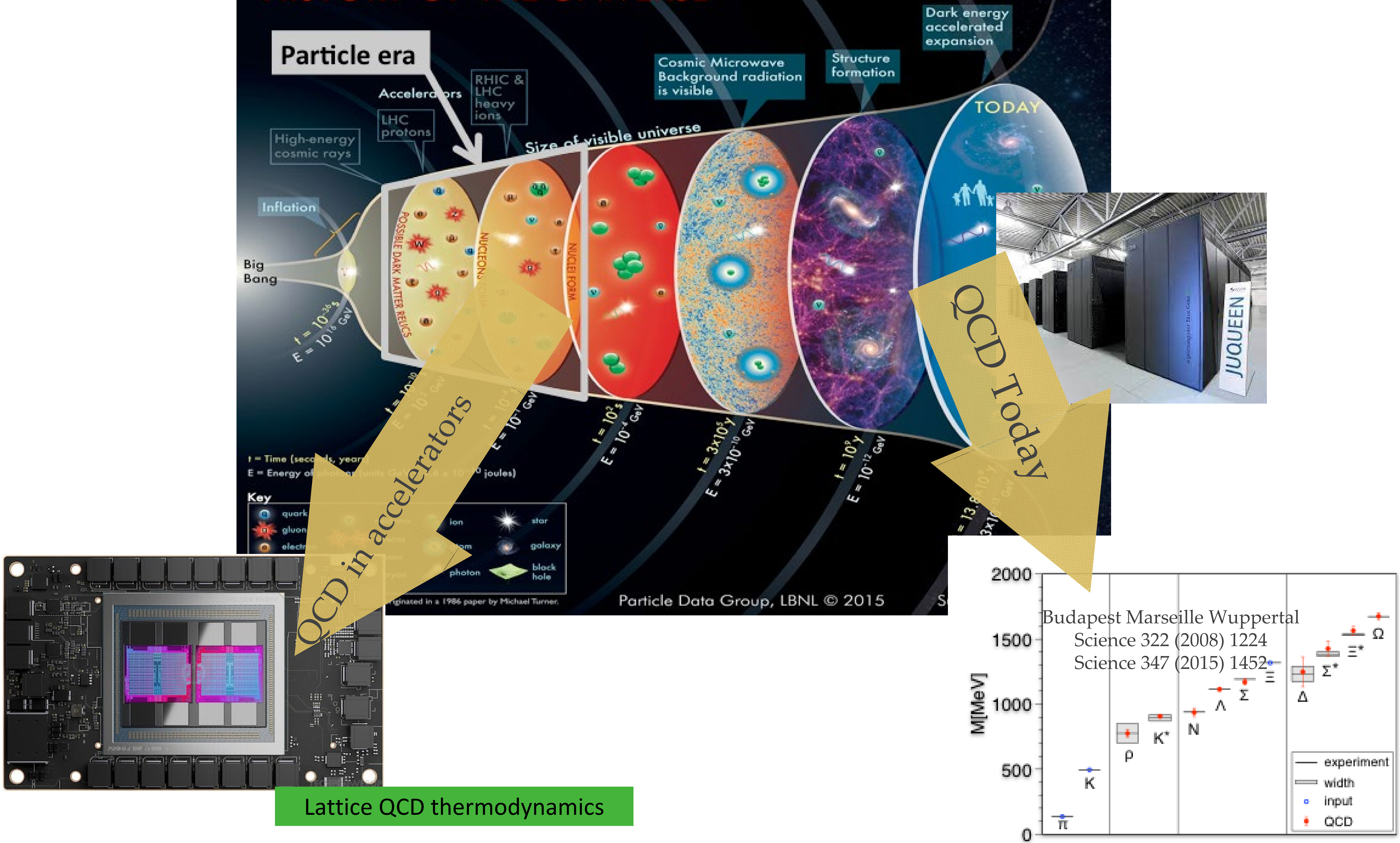
EuroHPC Joint Undertaking

Szabolcs Borsányi

The QCD crossover line in a finite volume

EuroHPC User day
Copenhagen

30.09.2025


Particle era
Accelerators
RHIC & LHC heavy ions
LHC protons
High-energy cosmic rays
Inflation
Big Bang
Cosmic Microwave Background radiation is visible
Structure formation
Dark energy accelerated expansion
TODAY
Size of visible universe
POSSIBLE DARK MATTER RELICS
NUCLEONS
NUCLEI FORM
t = Time (seconds, years)
Key
quark
gluon
ion
star
galaxy
photon
black hole
Particle Data Group, LBNL © 2015
QCD in accelerators
QCD Today
JUQUEEN
Budapest Marseille Wuppertal
Science 322 (2008) 1224
Science 347 (2015) 1452
M[MeV]
2000
1500
1000
500
0
π
K
ρ
K*
N
Λ
Σ
Ξ
Δ
Σ*
Ξ*
Ω
experiment
width
input
QCD
Lattice QCD thermodynamics

QCD's role in cosmology

Particle Data book, Chapter 22, Big Bang Cosmology

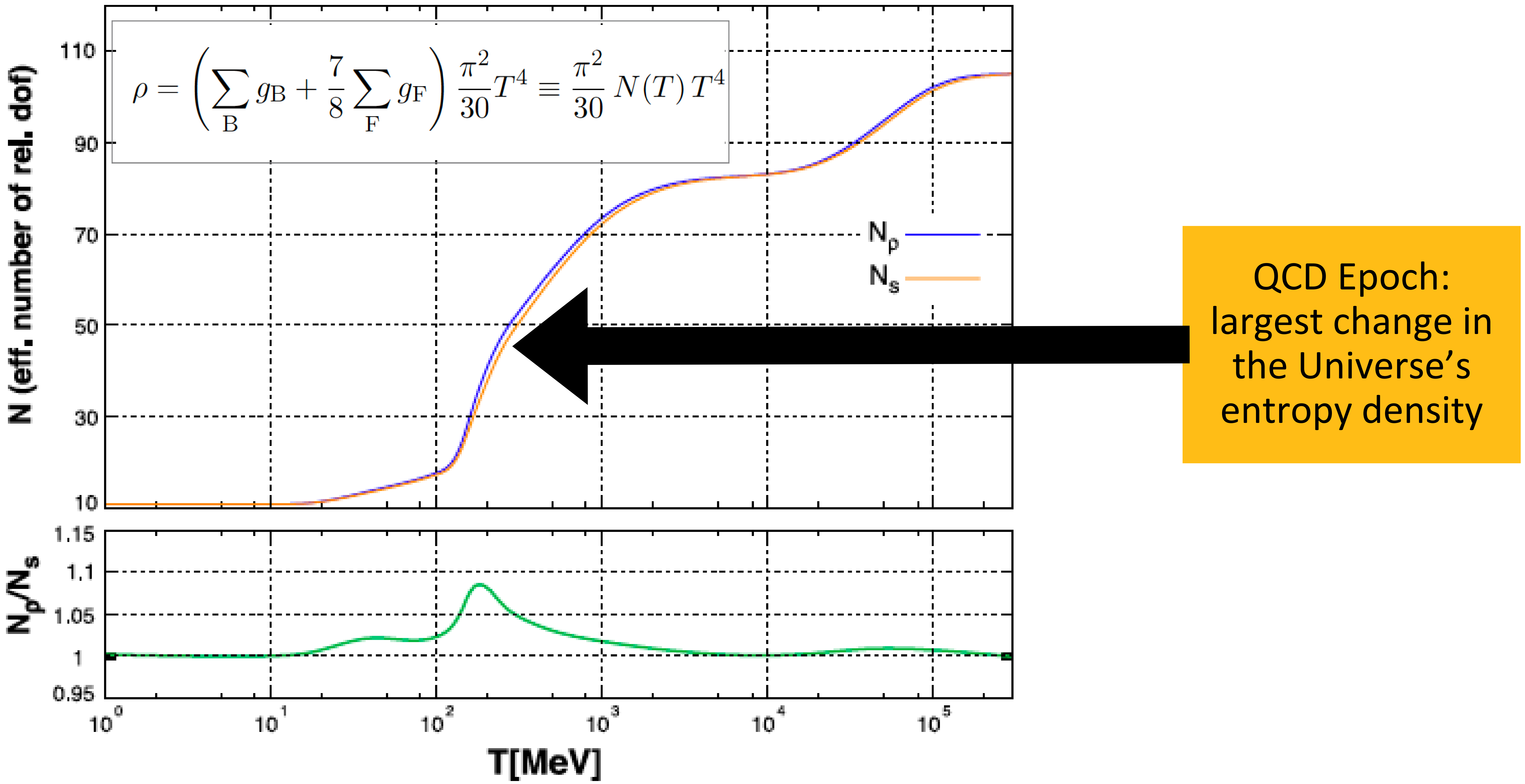
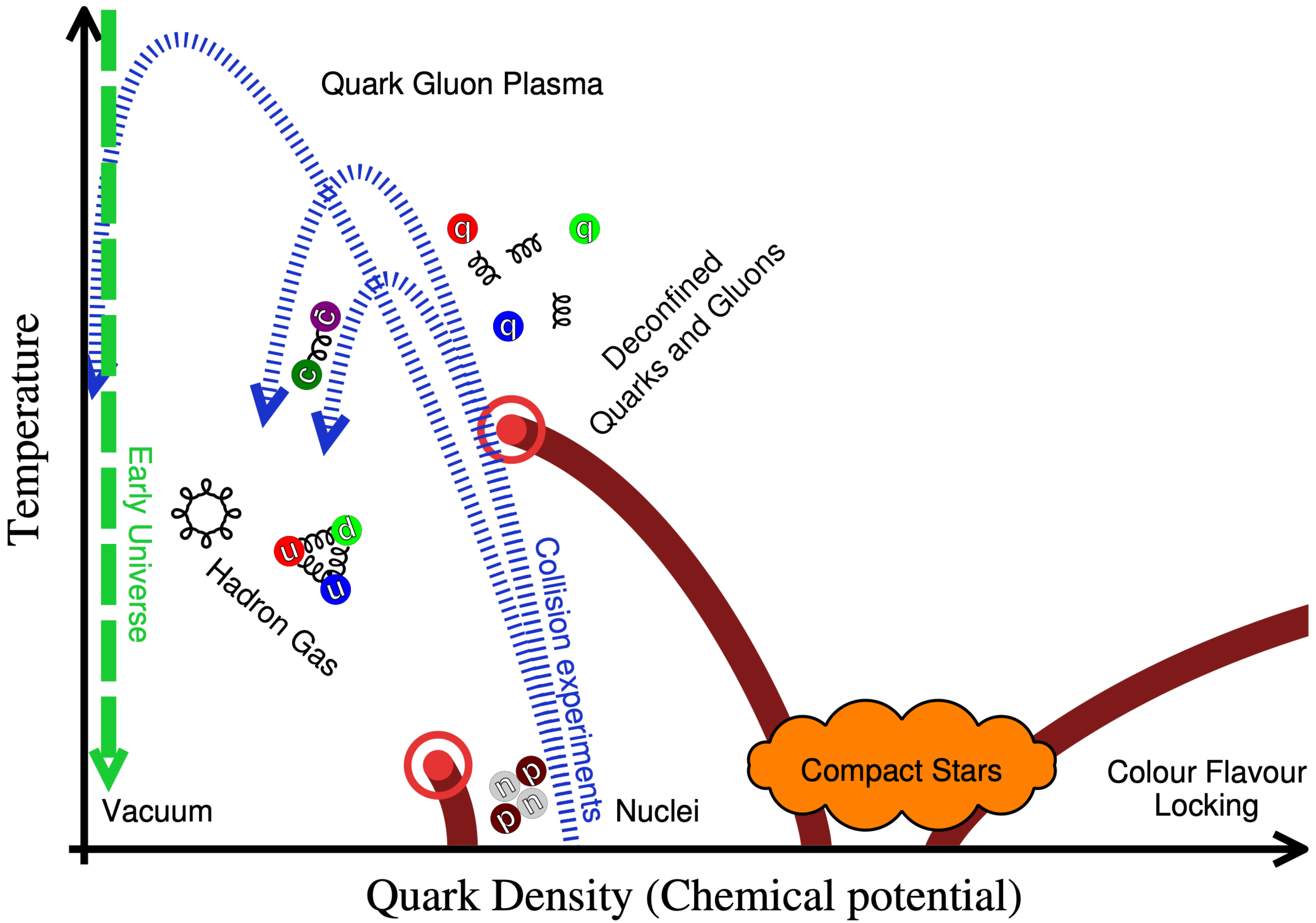


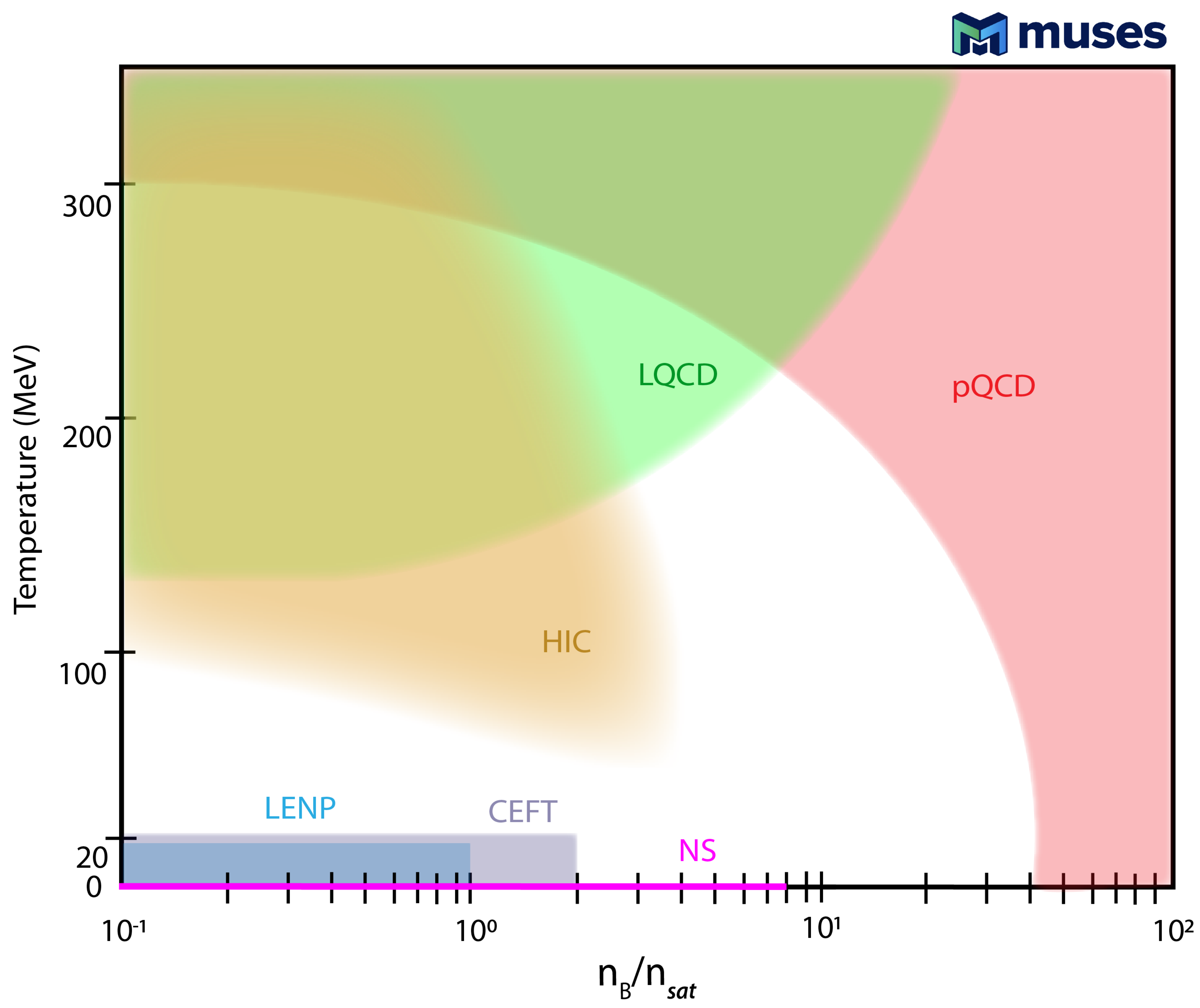
Figure 22.3: The effective numbers of relativistic degrees of freedom as a function of temperature. The sharp drop corresponds to the quark-hadron transition. The bottom panel shows the relative ratio between the number of degrees of freedom characterizing the energy density and the entropy.

[Borsanyi et al Nature 2016]

A phase diagram of strong interactions

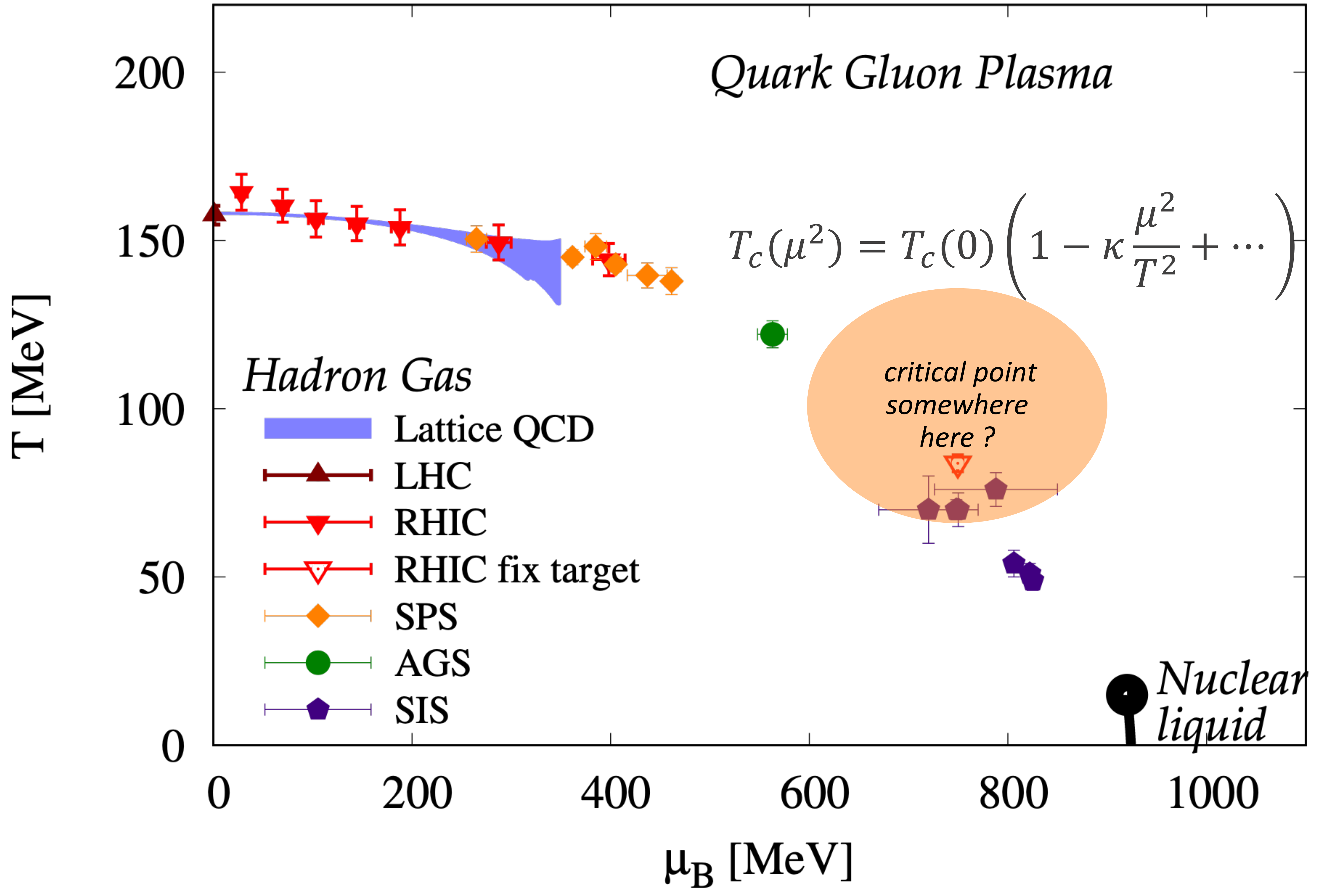


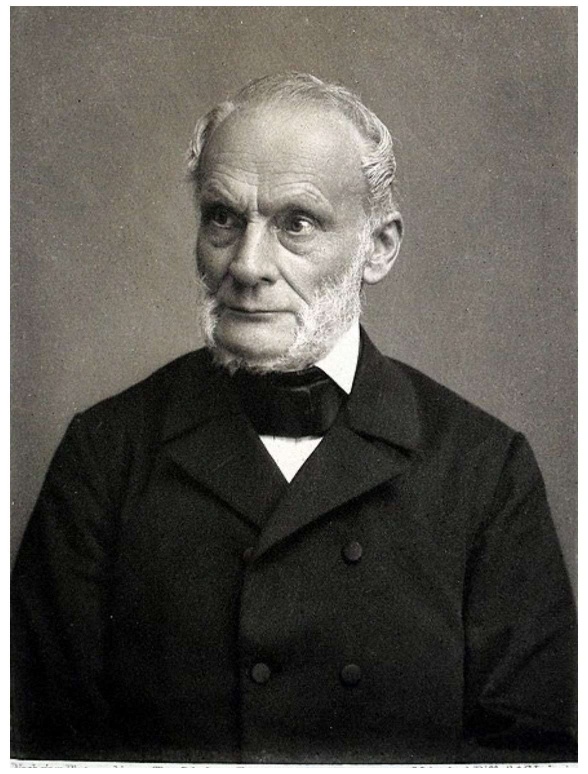
Phase diagram by source of information



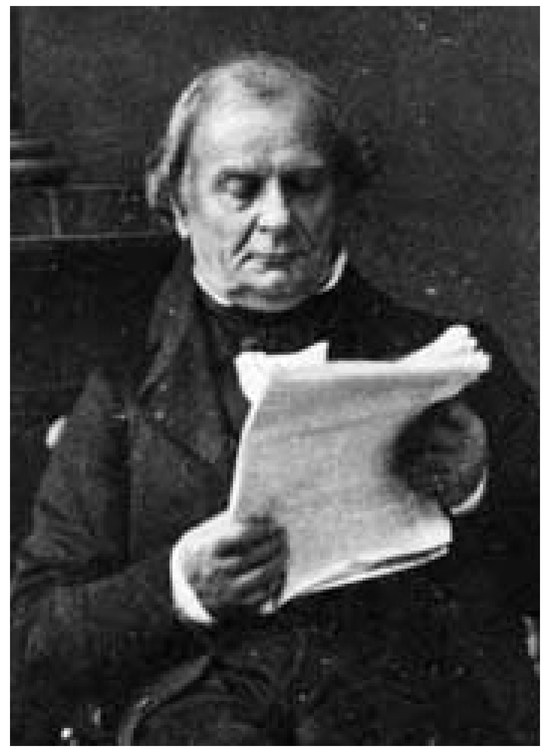
[Muses 2303.17021]

The crossover line in the phase diagram





Rudolf Clausius
(1822-1888)



Émile Clapeyron
(1799-1864)

Clausius - Clapeyron relation:
(applies to 1st order)

$$\kappa = \frac{1}{2}\cdot\frac{\Delta\text{susceptibility}}{\Delta\text{entropy}}$$

$\kappa > 0$ because of these two aspects of deconfinement

$\Delta\text{baryonsusceptibility} > 0$

Free quarks are much lighter than the hadrons they were confined into.

$\Delta\text{entropy} > 0$

Gluons are liberated and contribute as massless fields.

Direct result from lattice QCD

$$\kappa = 0.015(2)$$

Pisa 2015, Wuppertal 2015, Bielefeld/BNL 2018, Wuppertal 2020

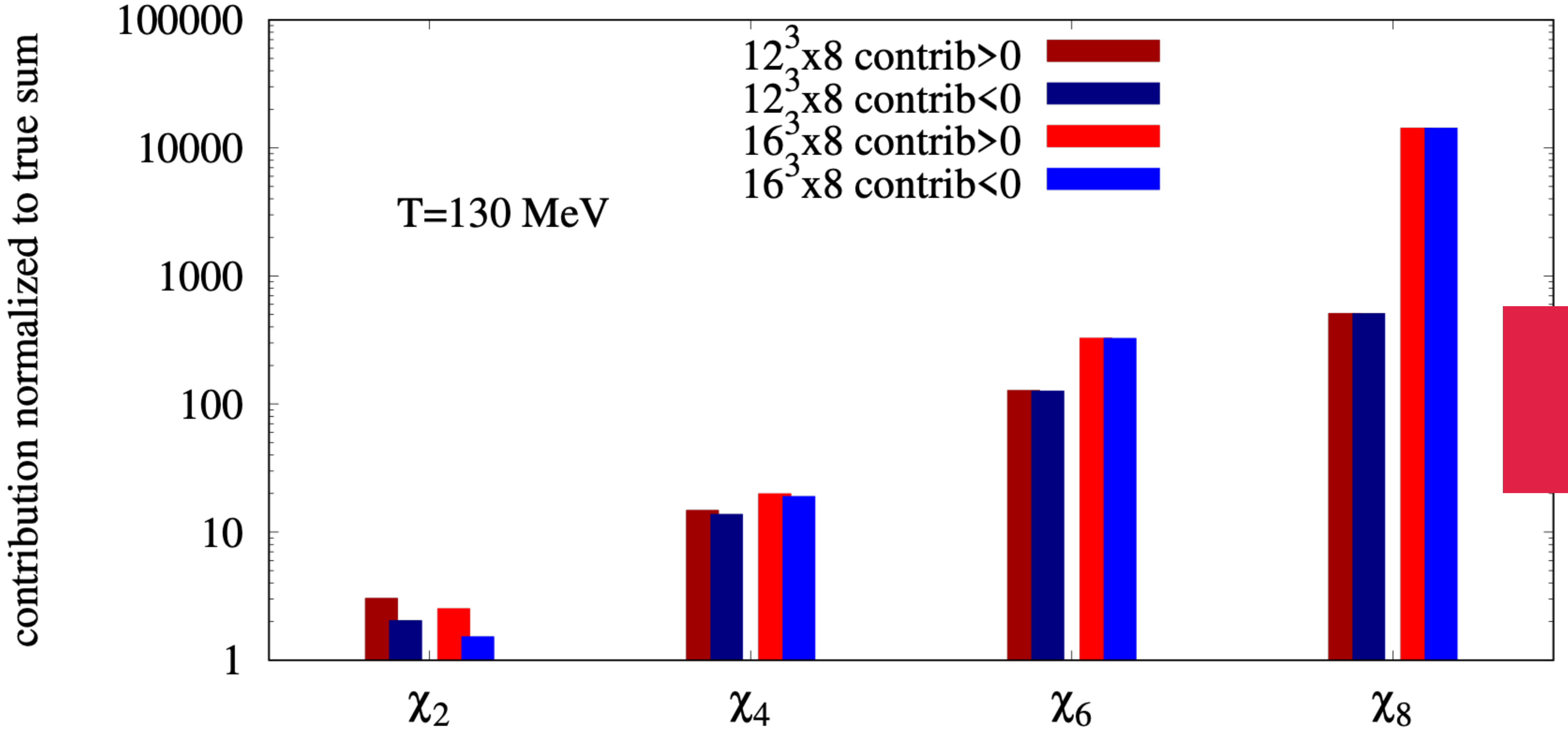
Expansion to finite density

Lattice QCD samples gluon configurations with the probability corresponding to its weight in the quantum field theory.

If we break the charge conjugation symmetry by a chemical potential:
complex probabilities

A Taylor expansion can be defined, nevertheless, the $\chi_n^B(T)$ coefficients can be simulated.

$$\frac{p(\mu_B)}{T^4} = \frac{p(0)}{T^4} + \frac{1}{2!}\frac{\mu_B^2}{T^2}\chi_2^B(T) + \frac{1}{4!}\frac{\mu_B^4}{T^4}\chi_4^B(T) + \frac{1}{6!}\frac{\mu_B^6}{T^6}\chi_6^B(T) + \ldots$$



High orders ~ Cancellation problem
difficulty ~ power of a volume

This work

1. Crossover temperature extracted from the peak of static quark entropy $S_Q(T) = \frac{\partial T \log|P|}{\partial T}$,

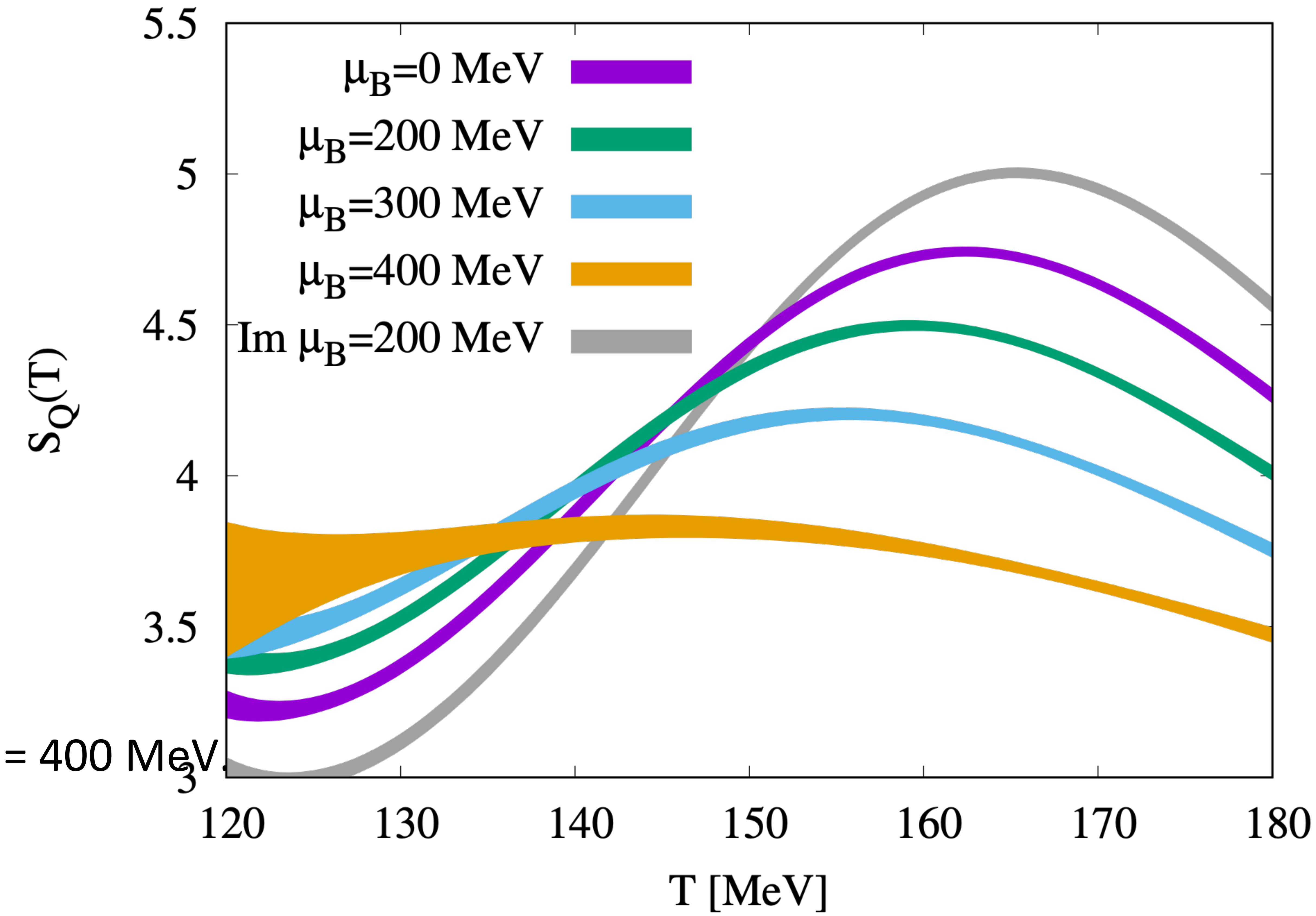
   P is the Polyakov loop [Bazavov et al (TUM), 2016]

2. We worked out a power series expansion scheme for *log |P|* (previously existing for the pressure)

3. Simulated cca. 2 million configurations at 18 temperatures on a $16^3 \times 8$ lattice.

4. Extracted the full chemical potential dependence of the configurations using the reduced matrix formalism [Hasenfratz & Toussaint 1992]

5. Extrapolated the transition line in the Taylor expansion up to μ_B = 400 MeV.

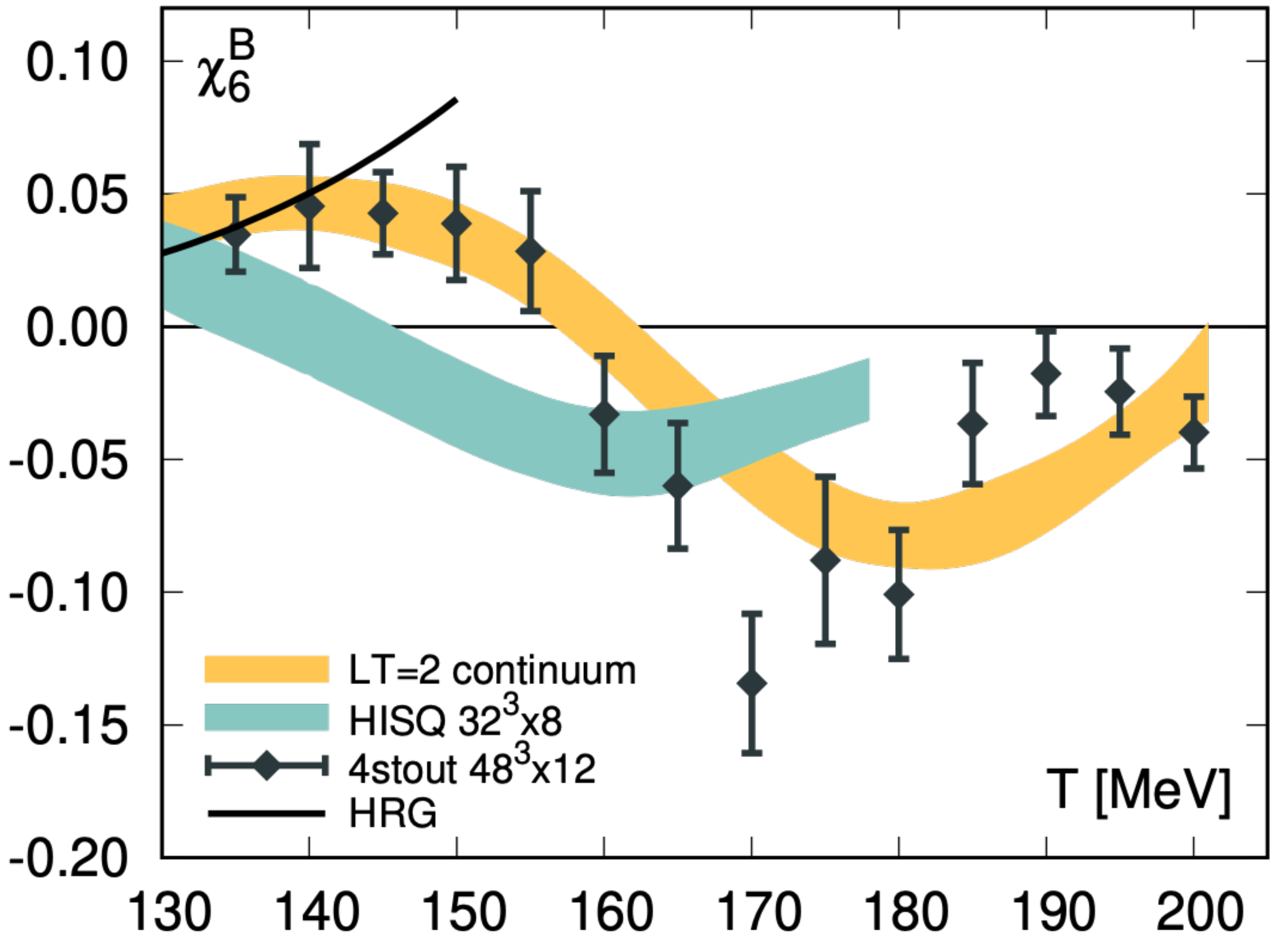


LUMI: all eigenvalues of 64 million 24576x24576 matrices

MAGMA (univ of Tenessee)

Taylor coefficients of the pressure

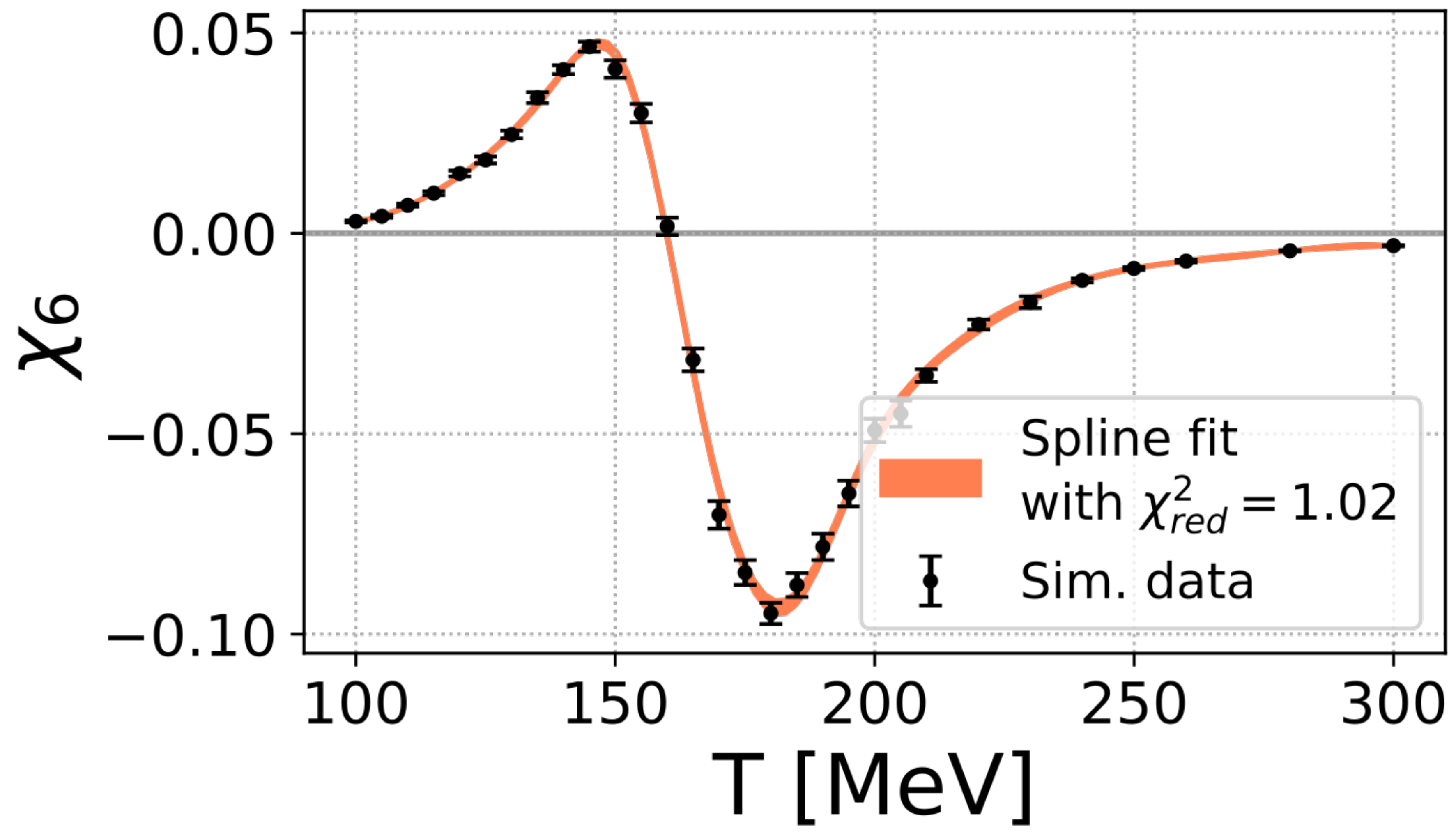
Before this project



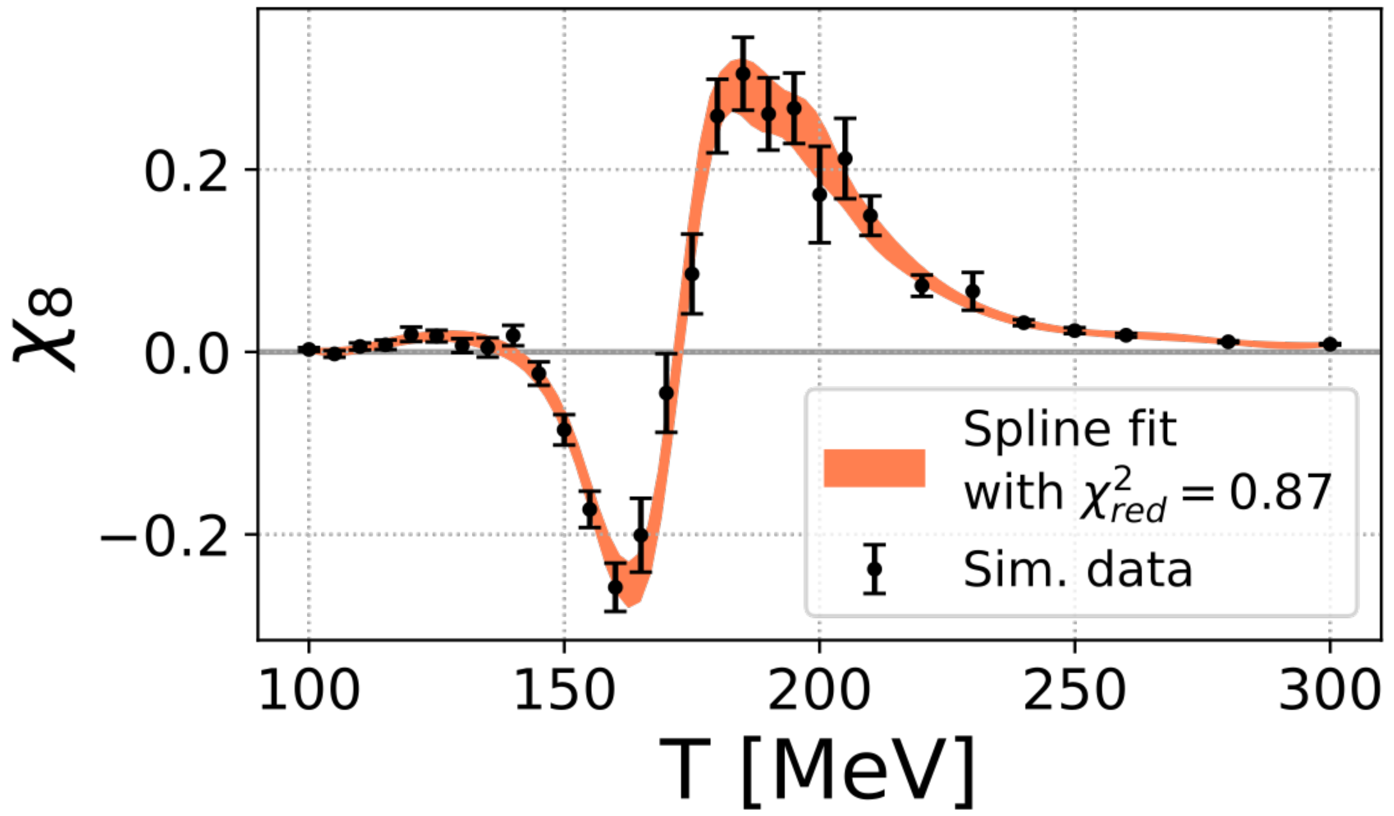
[Wuppertal 2023]

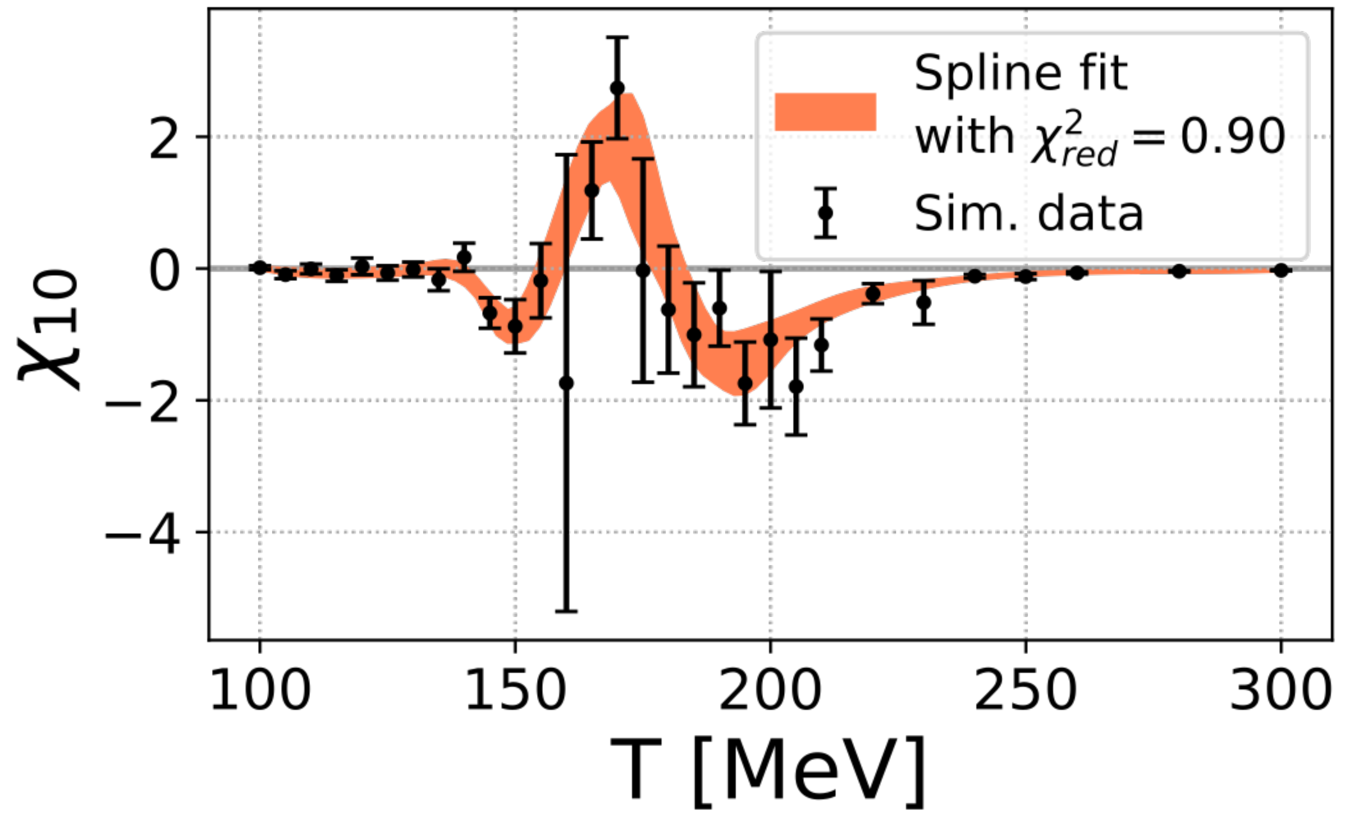
LUMI

After this project



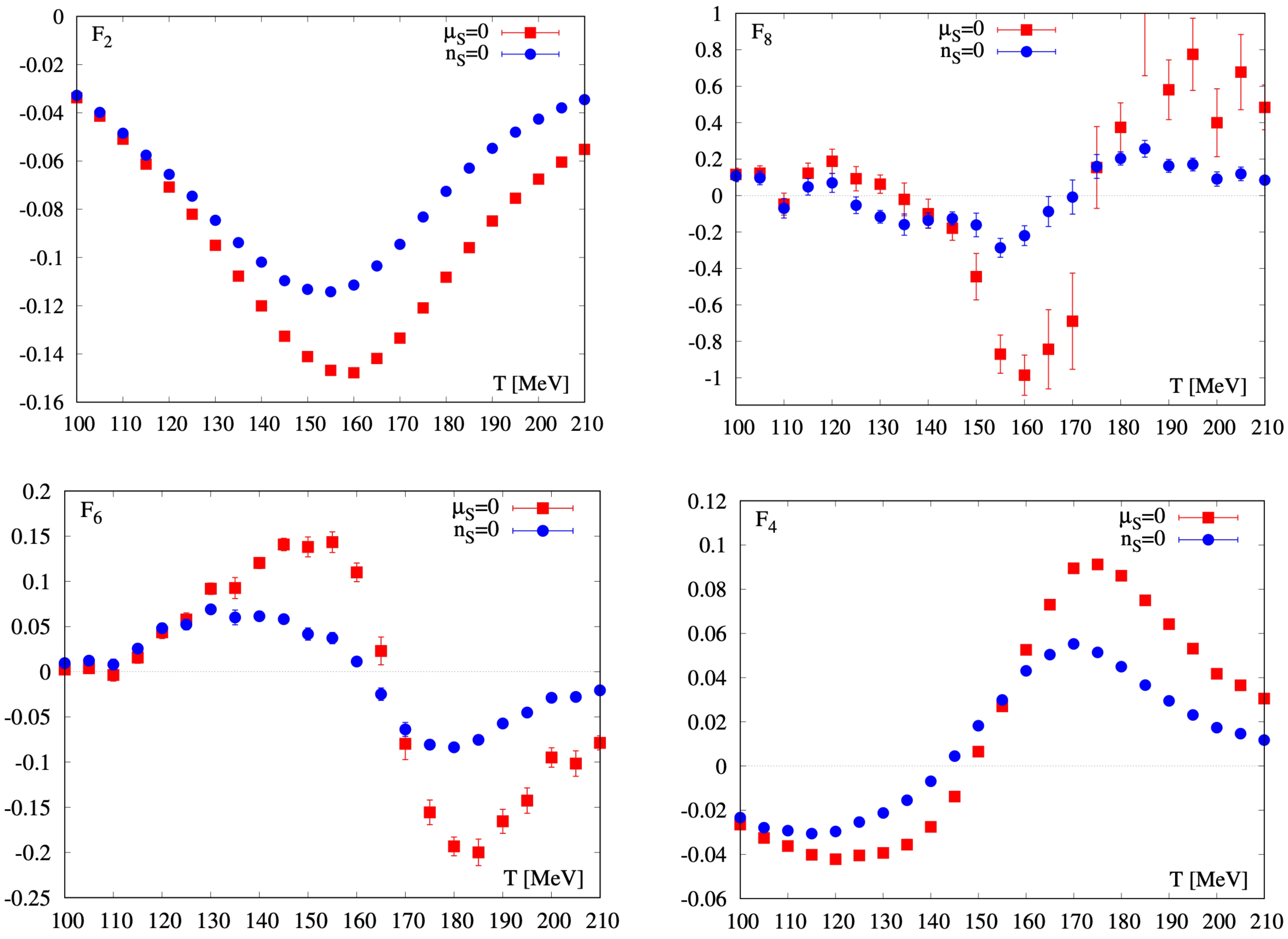
Higher orders



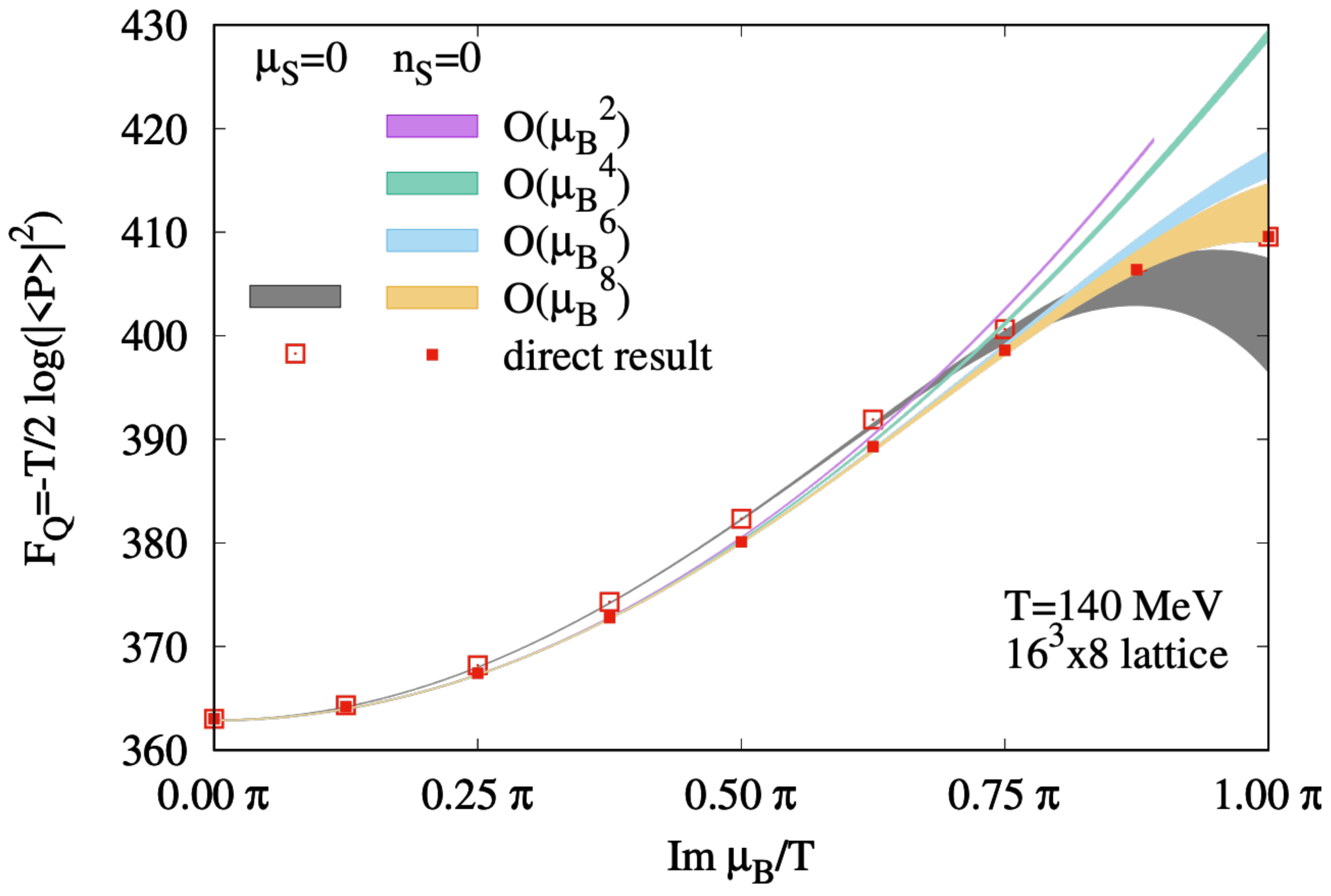


[Wuppertal 2025]

Extrapolation coefficients of $F_Q = -T \log |P|$

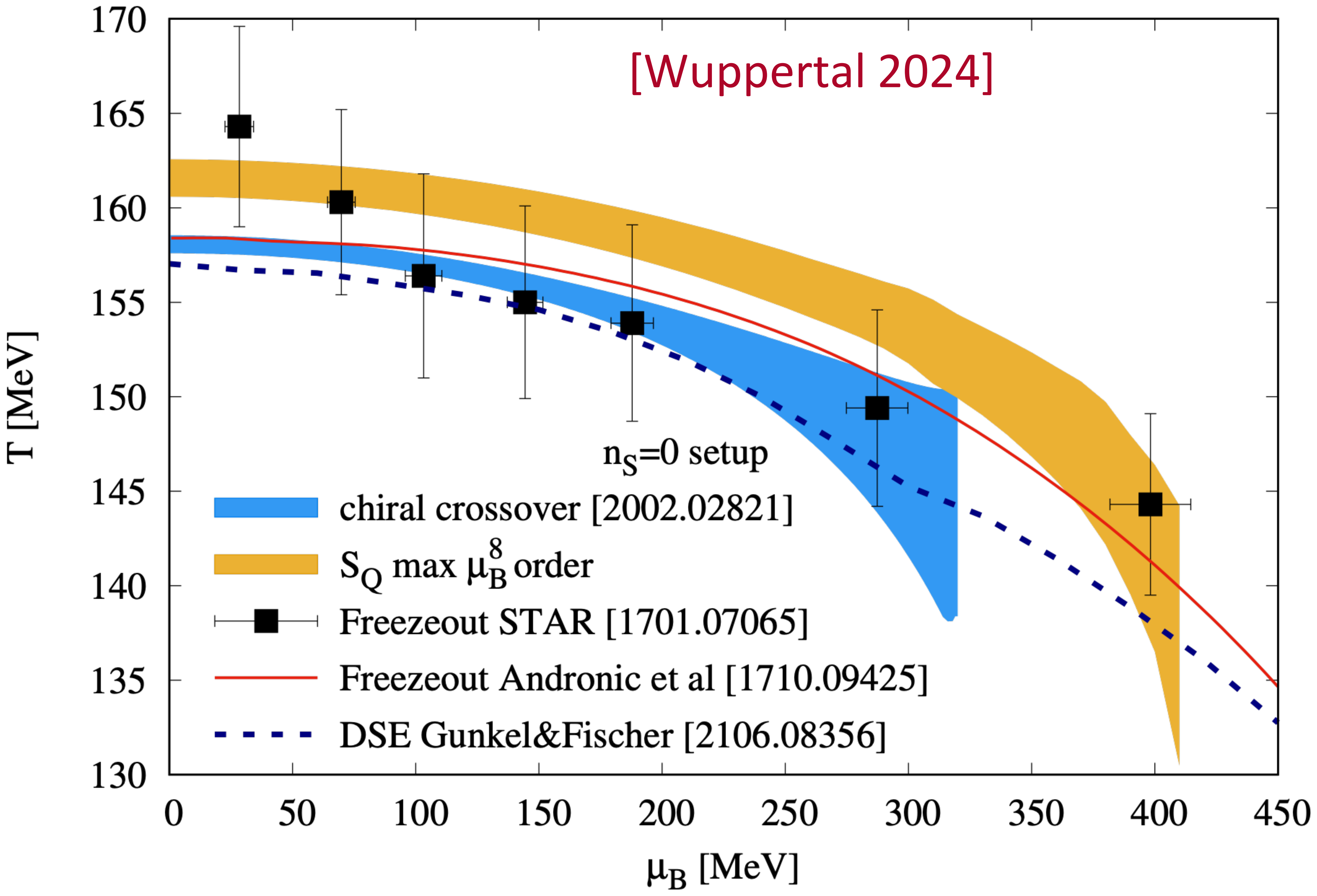


Test case: extrapolation to the imaginary direction, where direct data exists:

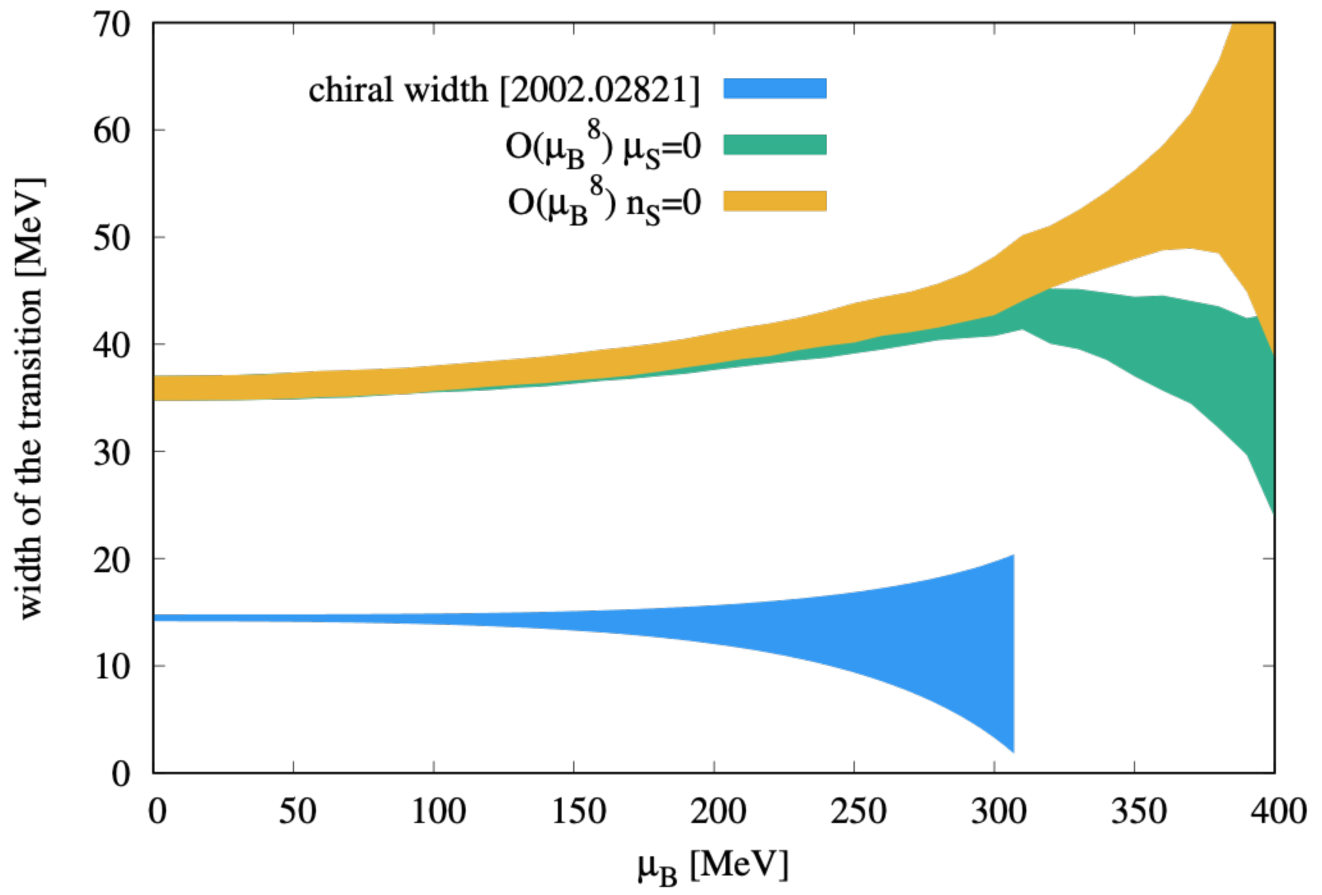


[Wuppertal 2024]

The resulting transition line



In one scenario the transition is getting sharper, potentially indicating an end of the crossover region.

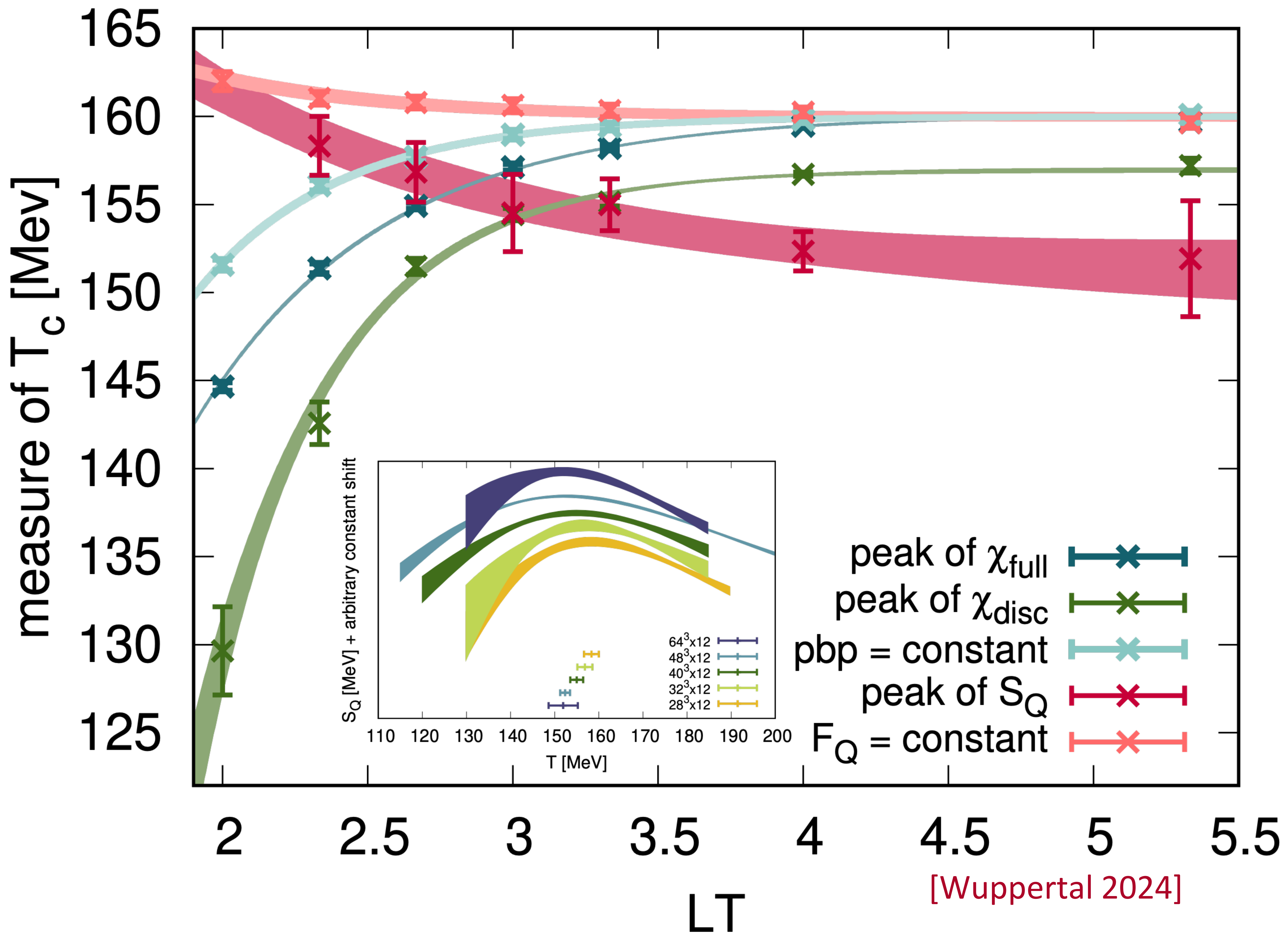


Work for the 2nd project year:
- extension to up to 500 MeV using a different proxy
- extension to up to 600 MeV and beyond using a different algorithm

Target: 600-650 MeV, where functional methods predict a critical point

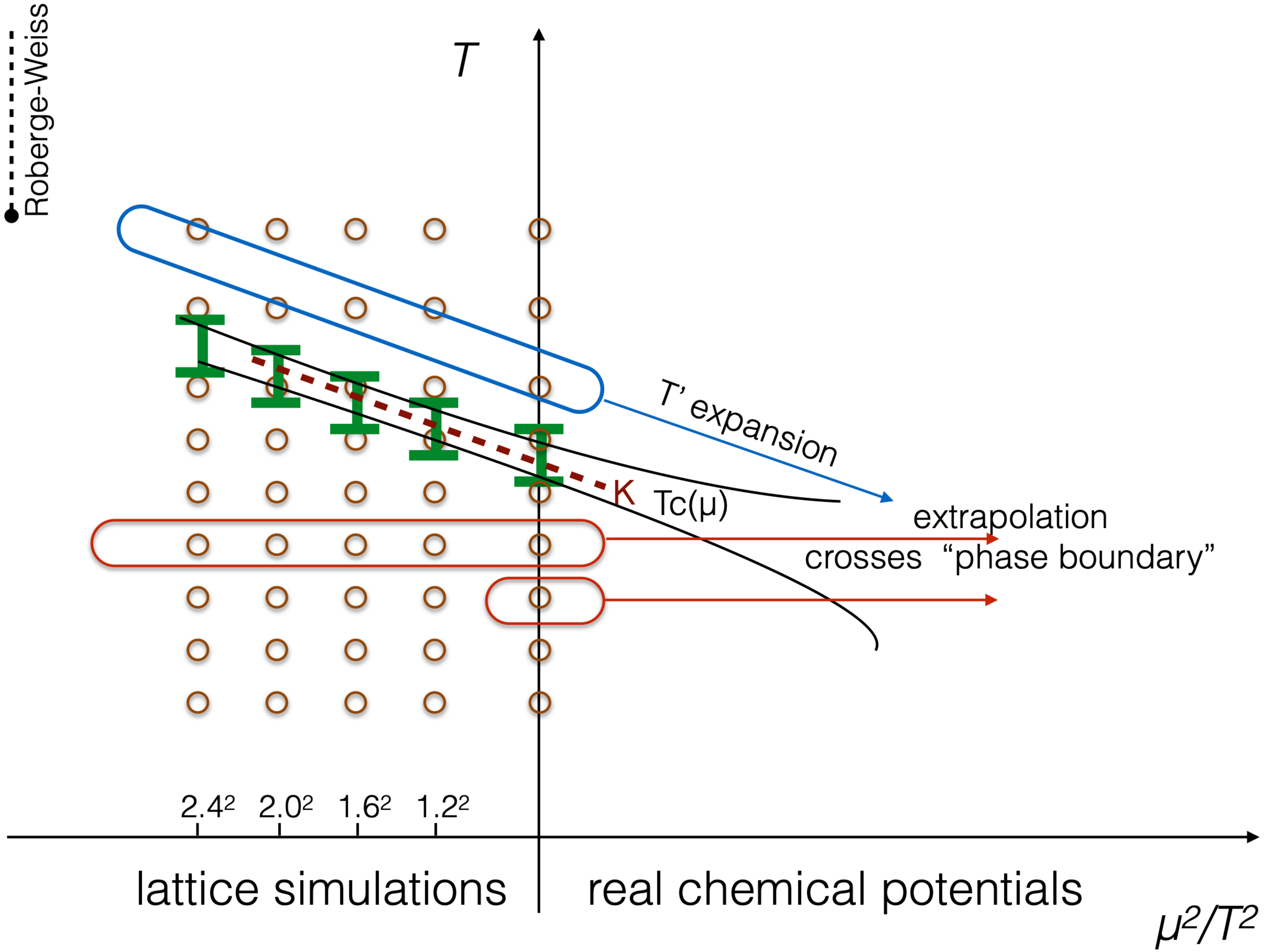
Backup slides

Volume dependence of T_c measures



[Wuppertal 2024]

Extrapolation strategies

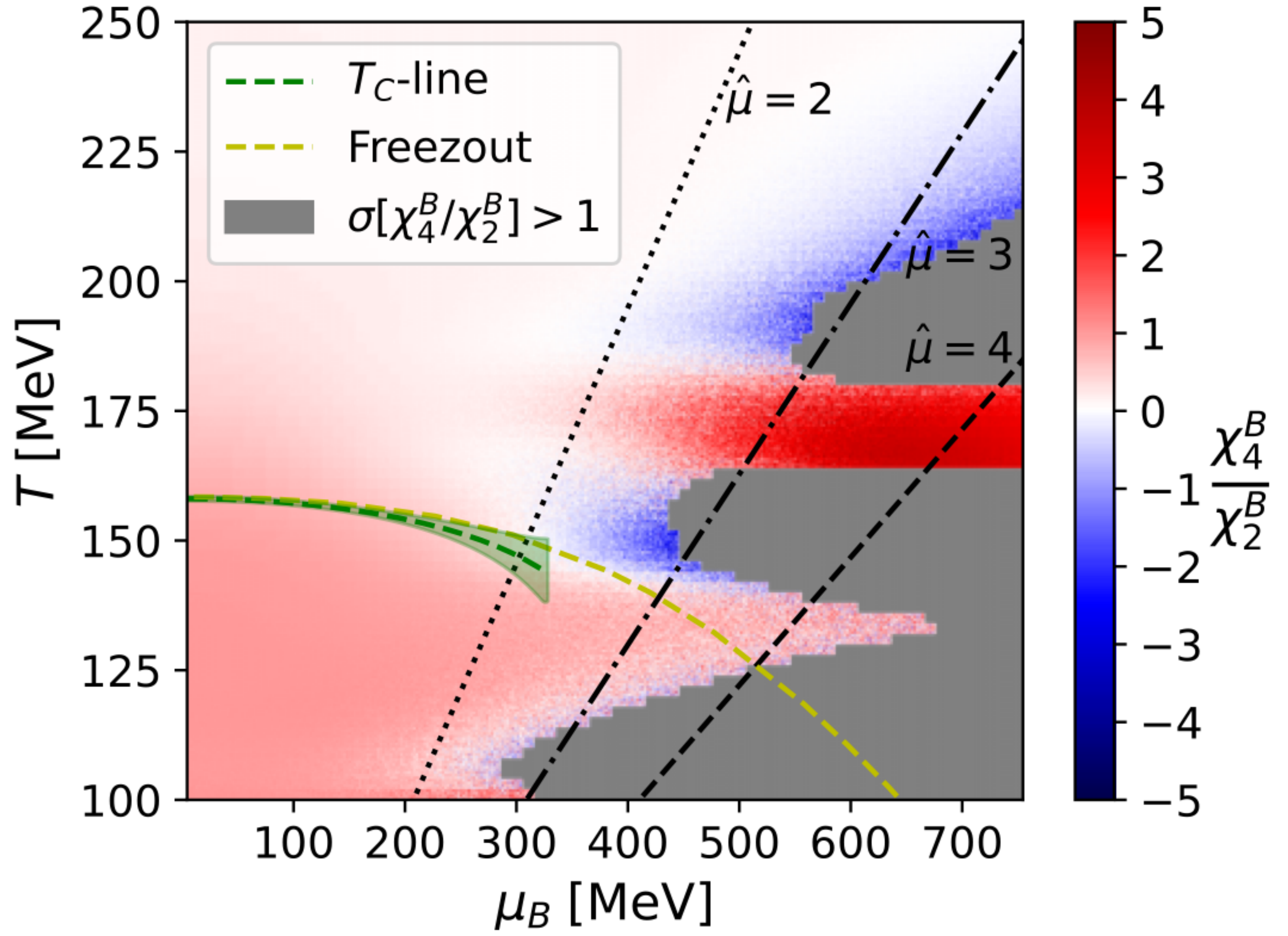


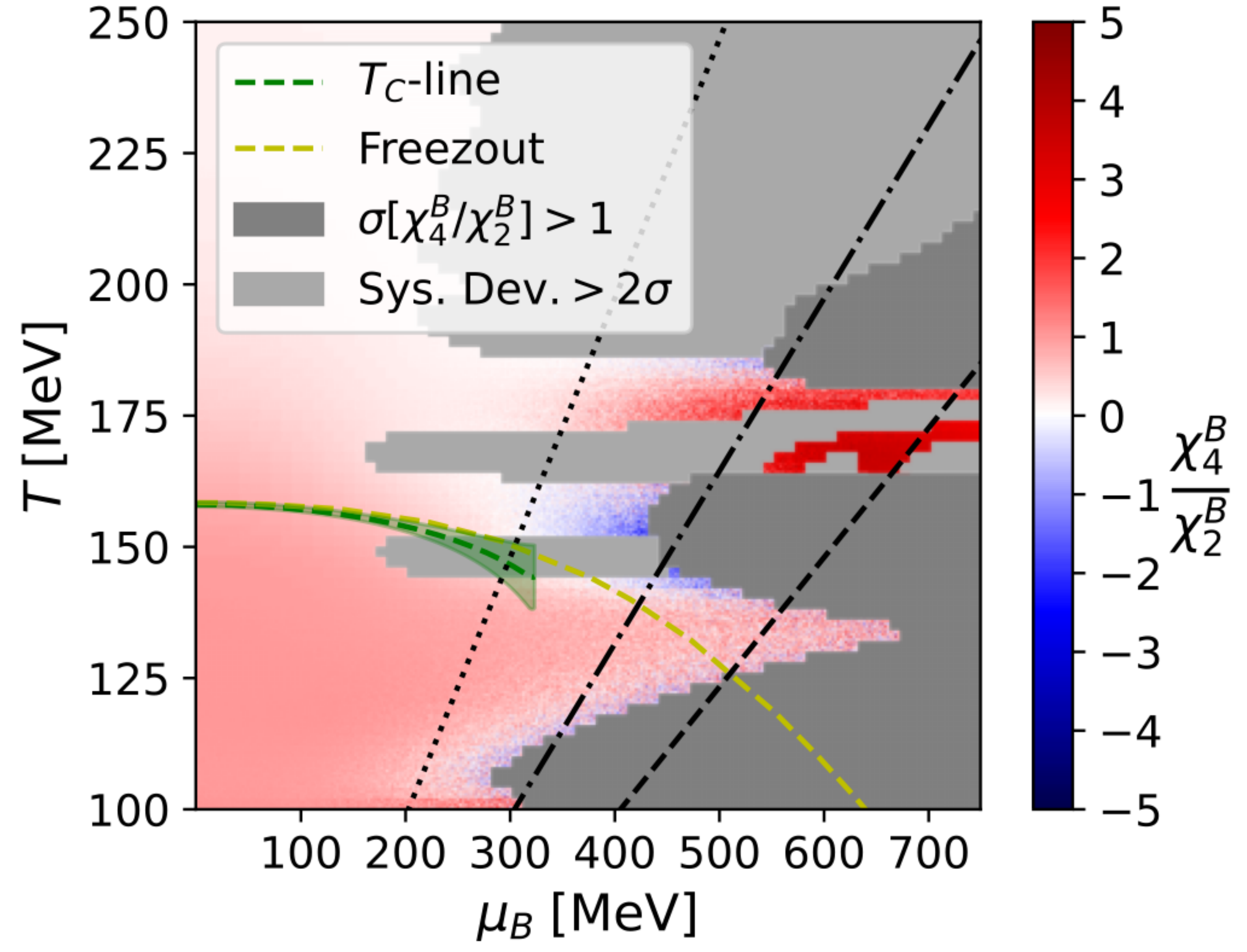
Ripples around the critical point

Extrapolate χ_4/χ_2 using Ratio of Taylor

$$\frac{\chi_4^B(T,\mu_B)}{\chi_2^B(T,\mu_B)} \approx \frac{\chi_4 + \chi_6\frac{\hat{\mu}_B^2}{2} + \chi_8\frac{\hat{\mu}_B^4}{4!} + \chi_{10}\frac{\hat{\mu}_B^6}{6!},}{\chi_2 + \chi_4\frac{\hat{\mu}_B^2}{2} + \chi_6\frac{\hat{\mu}_B^4}{4!} + \chi_8\frac{\hat{\mu}_B^6}{6!} + \chi_{10}\frac{\hat{\mu}_B^8}{8!}}$$

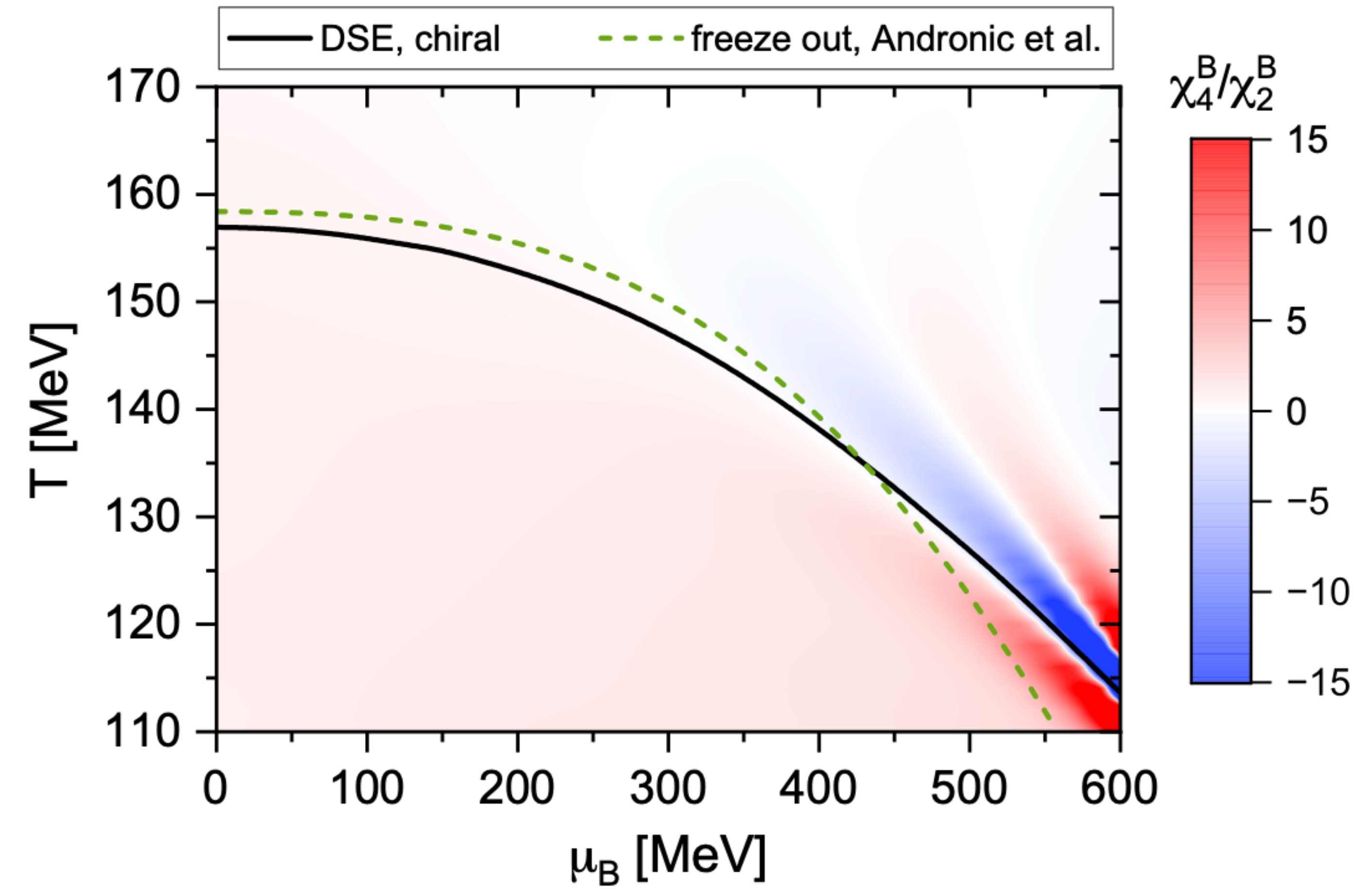
[This work: Wuppertal 2025]





using up to χ_{10}

One of the predictions based on functional methods:
[Fu, Luo, Pawlowski, Rennecke, Yin 2023]
[Lu,Gau,Liu,Pawlowski 2025]



POLITECNICO MILANO 1863

Hierarchical Dynamic Load Balancing Strategy for a p-adaptive Discontinuous Galerkin Compressible LES Solver

Paolo Valvo[a], Antonella Abbà[a]
[a]DAER Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano

3rd EuroHPC User Day, Copenhagen, Denmark
30/09 - 01/10/2025

Discussion Layout

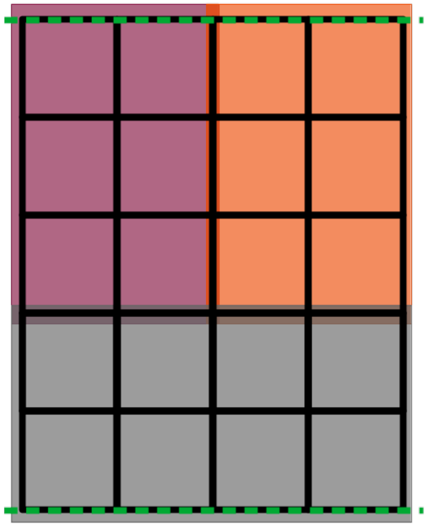
Goal: implement hardware-locality informed dynamic load balacing in the adaptive CFD LDG solver *dg-comp* (*FEMilaro* library)

Layout

- Formulation of the problem
- *Zoltan*: methods and capabilities
  - Geometric methods
  - Connectivity-based methods
- Concept of hierarchical partitioning
- Test case: flow past a square cylinder

Formulation of the problem (1/3)

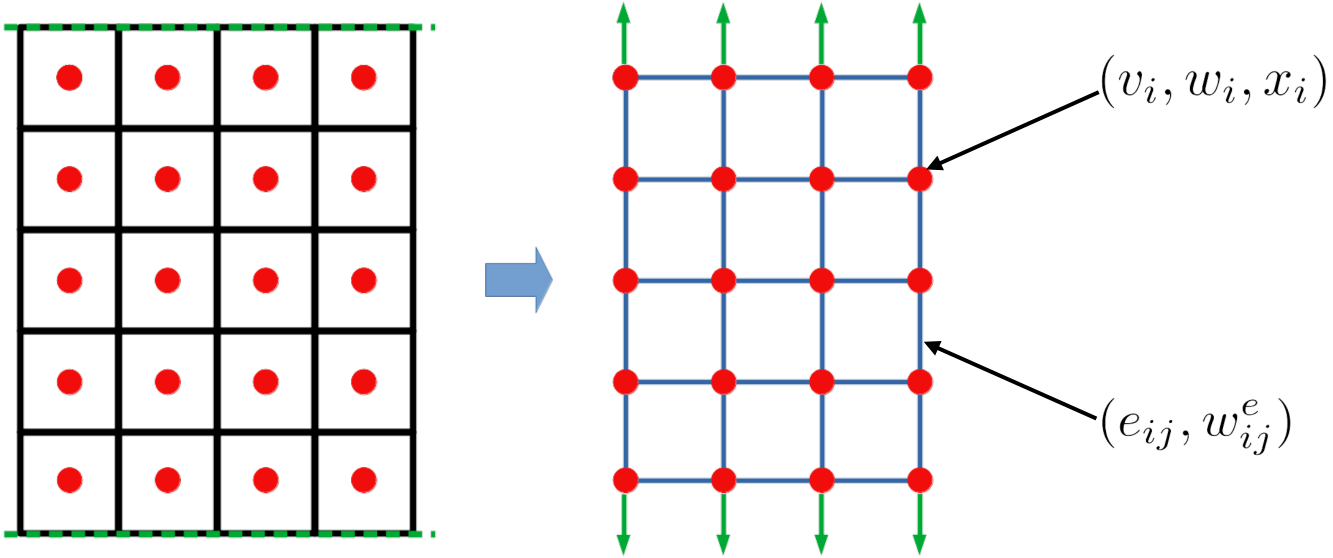
dg-comp: load repartition by means of grid decomposition



MPI Task 1
MPI Task 2
MPI Task 3
Periodic BC side

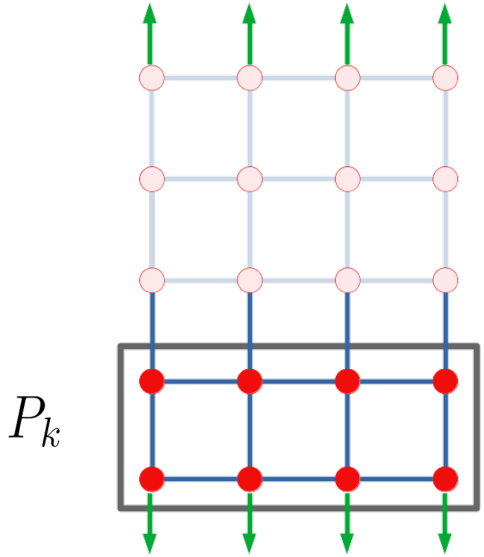
Formulation of the problem (2/3)

Grid partitioning → Graph partitioning



Formulation of the problem (3/3)

A useful metrics: the communication volume



$$V_{comm} = \sum_{e_{ij} \cap \partial P_k \neq \emptyset} w^e_{ij}$$

Zoltan: methods and capabilities (1/2)

Geometric methods
Vertices agglomeration via geometric criteria, ensuring:

$$\sum_{P_k} w_i \simeq const. \quad \forall k$$

Zoltan provides three methods:

- RCB (Recursive Coordinate Bisection)
- RIB (Recursive Inertial Bisection)
- HSFC (Hilbert Space Filling Curve)

Zoltan: methods and capabilities (2/2)

Connectivity-based methods
Partition the graph to achieve:

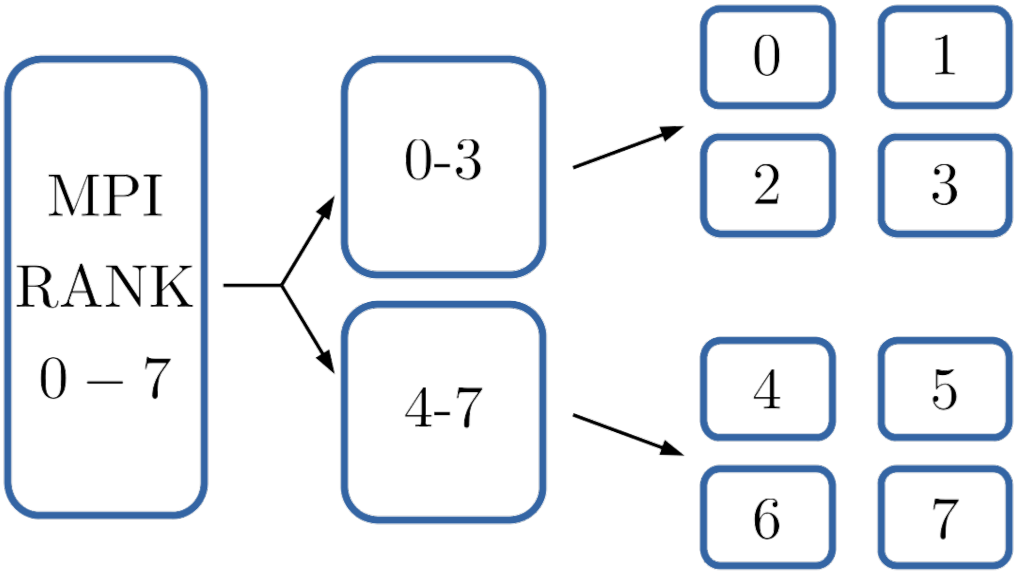
$$\sum_{P_k} w_i \simeq const. \quad \forall k$$

$$\min\{V_{comm}\} \quad \forall P_k$$

Zoltan provides a native multilevel graph partitioner

Concept of hierarchical partitioning

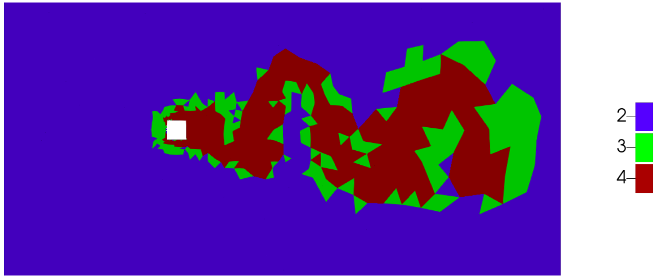
Divide the partitioning process in multiple steps.



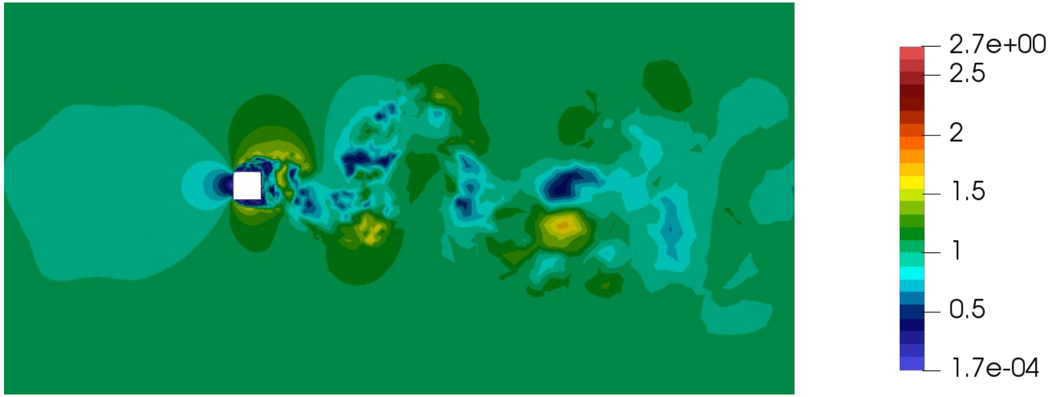
With CPU binding, hardware-locality informed partitioning is achieved

Test case (1/4)

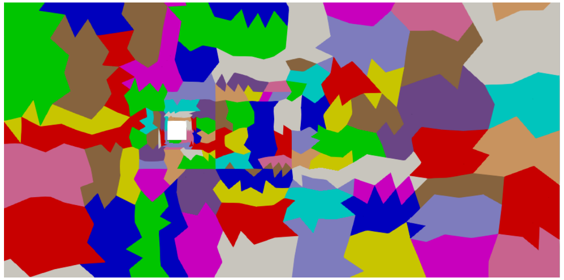
Turbulent flow past a square cylinder



Polynomial distribution



Momentum magnitude

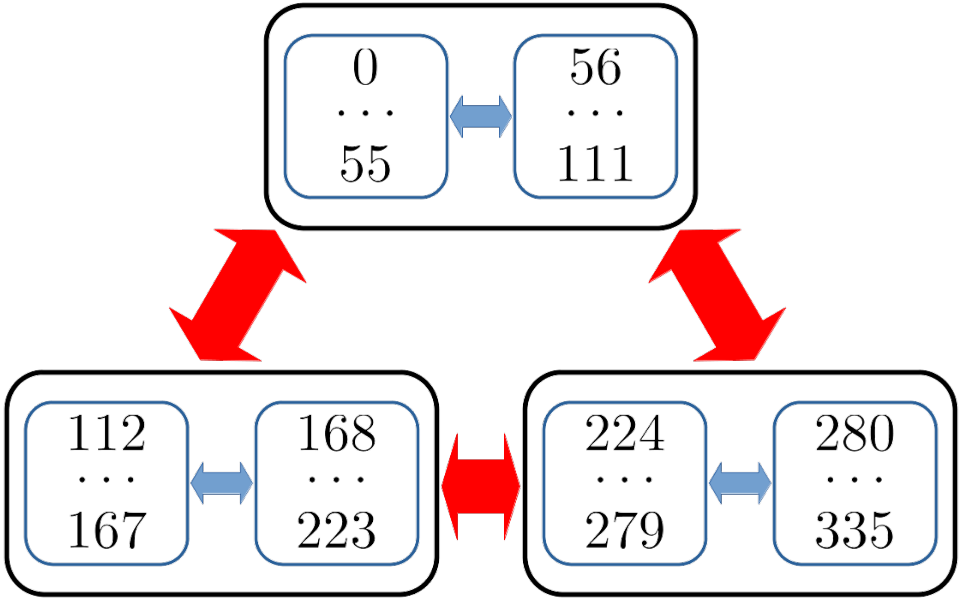


Grid partitions (RCB)

Test case (2/4)

Leonardo supercomputer:
- 3 nodes, 2 56-cores processors each
- Total: 336 MPI tasks

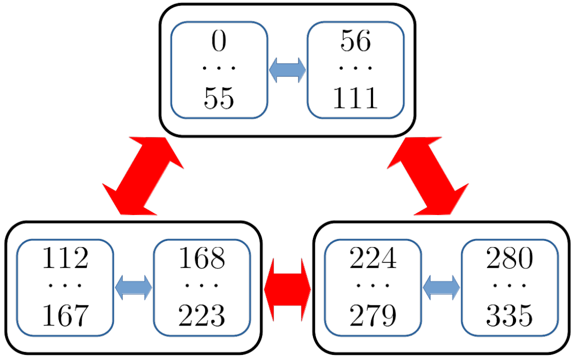
3 step hierarchical load balancing
- 3 partitions (connectivity)
- 2 partitions (connectivity)
- 56 partitions (geometric)



Test case (3/4)

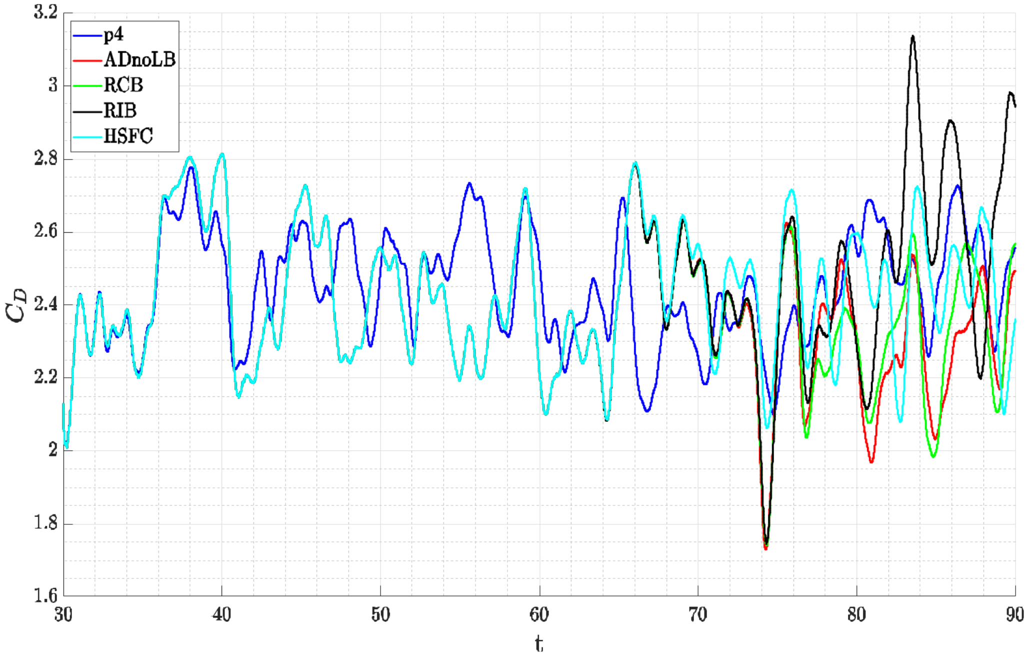
Effectiveness of hierarchical partitioning

	RCB	RIB	HSFC	HIER RCB	HIER RIB	HIER HSFC
V_{comm}^{0-111}	14142	12630	20460	4164	4626	4047
V_{comm}^{intra1}	6996	4344	1824	1848	2040	1866



Test case (4/4)

Results accuracy: load balancing introduces a small but acceptable noise



THANK YOU FOR YOUR ATTENTION!

Link to *FEMilaro* source code: https://bitbucket.org/mrestelli/femilaro/wiki/Home
Questions? mailto:paolo.valvo@polimi.it

Appendix A: adaptation strategy

$$Ind_{SF}(K) = \sqrt{\sum_{ij} [D_{ij}(K) - D_{ij}^{ISO}]^2}$$

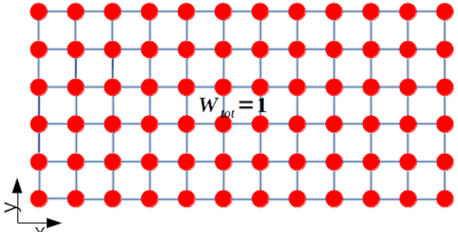
$$D_{ij}(\mathbf{x}, \mathbf{r}) = \langle [u_i(\mathbf{x} + \mathbf{r}, t) - u_i(\mathbf{x}, t)][u_j(\mathbf{x} + \mathbf{r}, t) - u_j(\mathbf{x}, t)] \rangle$$

Adaptation loop:

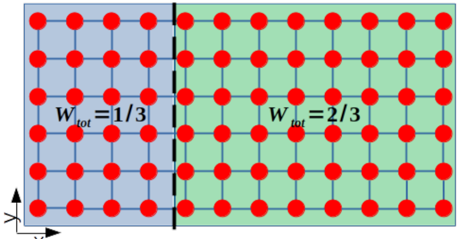
1. Average the indicator value for a time dt_{adapt}, computing it each $dt_{indicator}$
2. Each dt_{adapt}, vary the polynomial distribution according user-selected indicator thresholds

Appendix B: *Zoltan* partitioners (1/3)

RCB & RIB

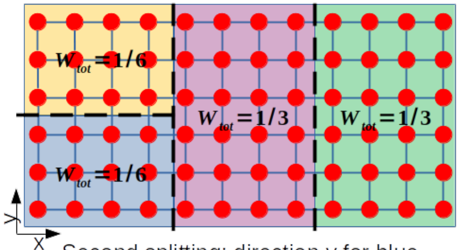


Initial graph

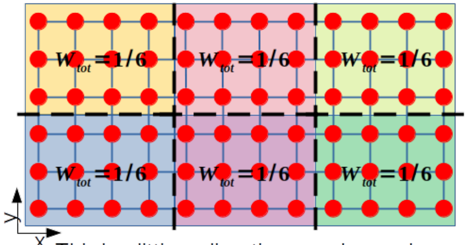


First splitting: direction x, α=1/3

Note: RIB uses the principal axes of inertia as splitting direction



Second splitting: direction y for blue partition, x for green, α=1/2

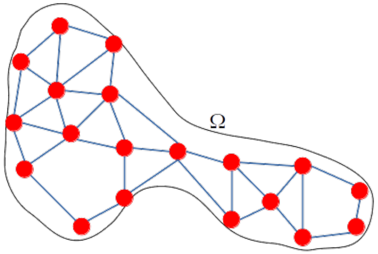


Third splitting: direction y, only purple and green partitions, α=1/2.

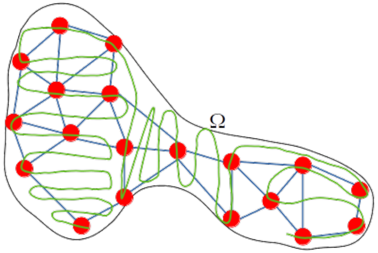
Image from *FEMILARO* documentation

Appendix B: *Zoltan* partitioners (2/3)

HSFC

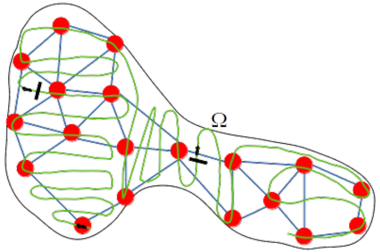


Initial graph and a surrounding geometry

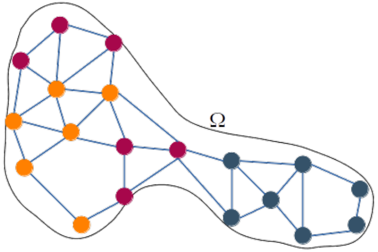


Graph and a space filling curve for the geometry Ω

Note: *Zoltan* uses Hilbert curves to fill the domain.



Partition of the grid following the line (3 partitions, uniform weight)



Final partitioned graph. Note the spatially disjoint partition

Image from *FEMILARO* documentation

Appendix B: *Zoltan* partitioners (3/3)

Connectivity-based

In *dg-comp*, this implies a multilevel graph partitioner with an agglomerative inner product matching coarsening strategy and an approximate Fiduccia-Mattheyses refinement algorithm.

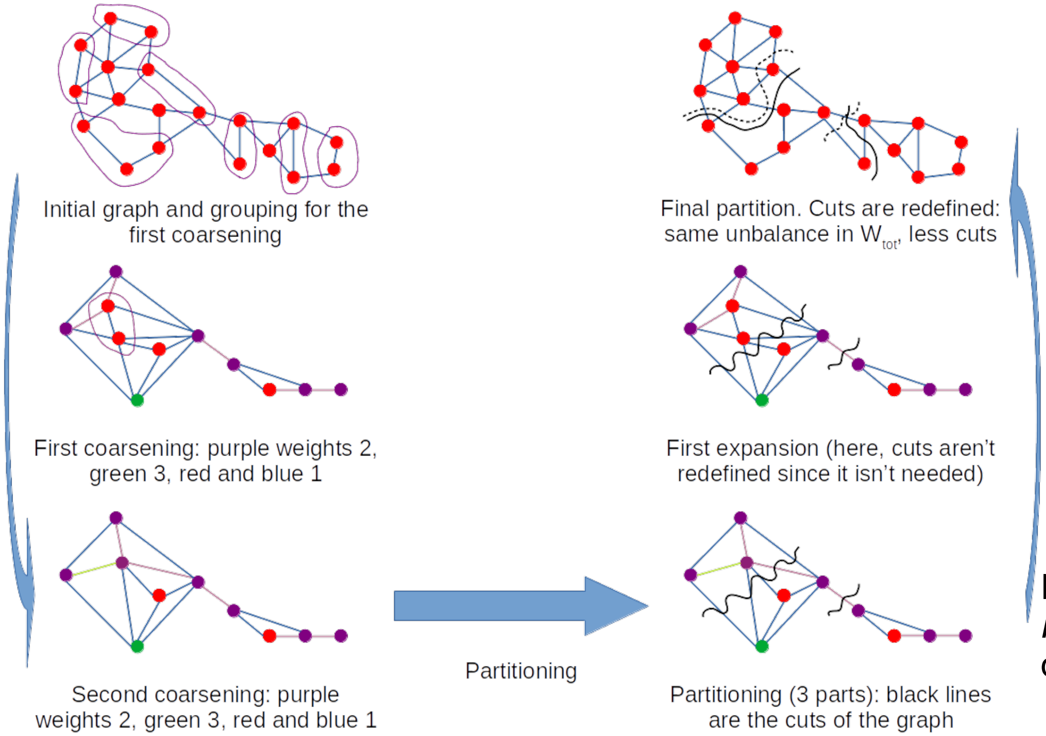


Image from *FEMILARO* documentation

Appendix C: time saving

Time [s]	p4	ADnoLB	RCB	RIB	HSFC	HIER RCB	HIER RIB	HIER HSFC
dg-comp	18568	16866	9446	9634	9924	9886	9860	9899
Zoltan	0	0	839	912	827	864	882	848
total	18568	16866	10285	10546	10751	10750	10742	10747