Chiral Mean Field module - 2023 report

Nikolas Cruz Camacho, J. Noronha-Hostler, R. Hass (NCSA), T. A. Manning (NCSA)
UIUC

R. Kumar , J. Peterson, V. Dexheimer
Kent State University

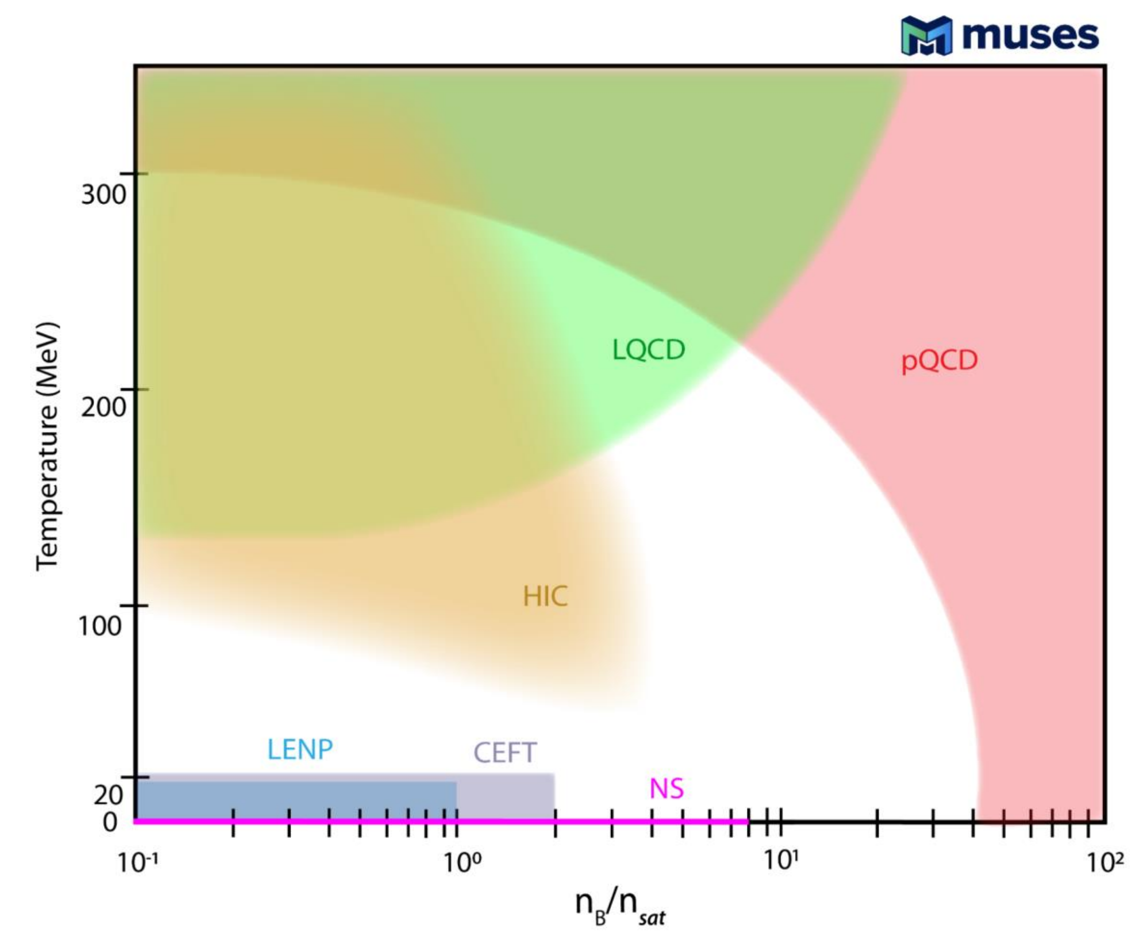
muses

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

NSF

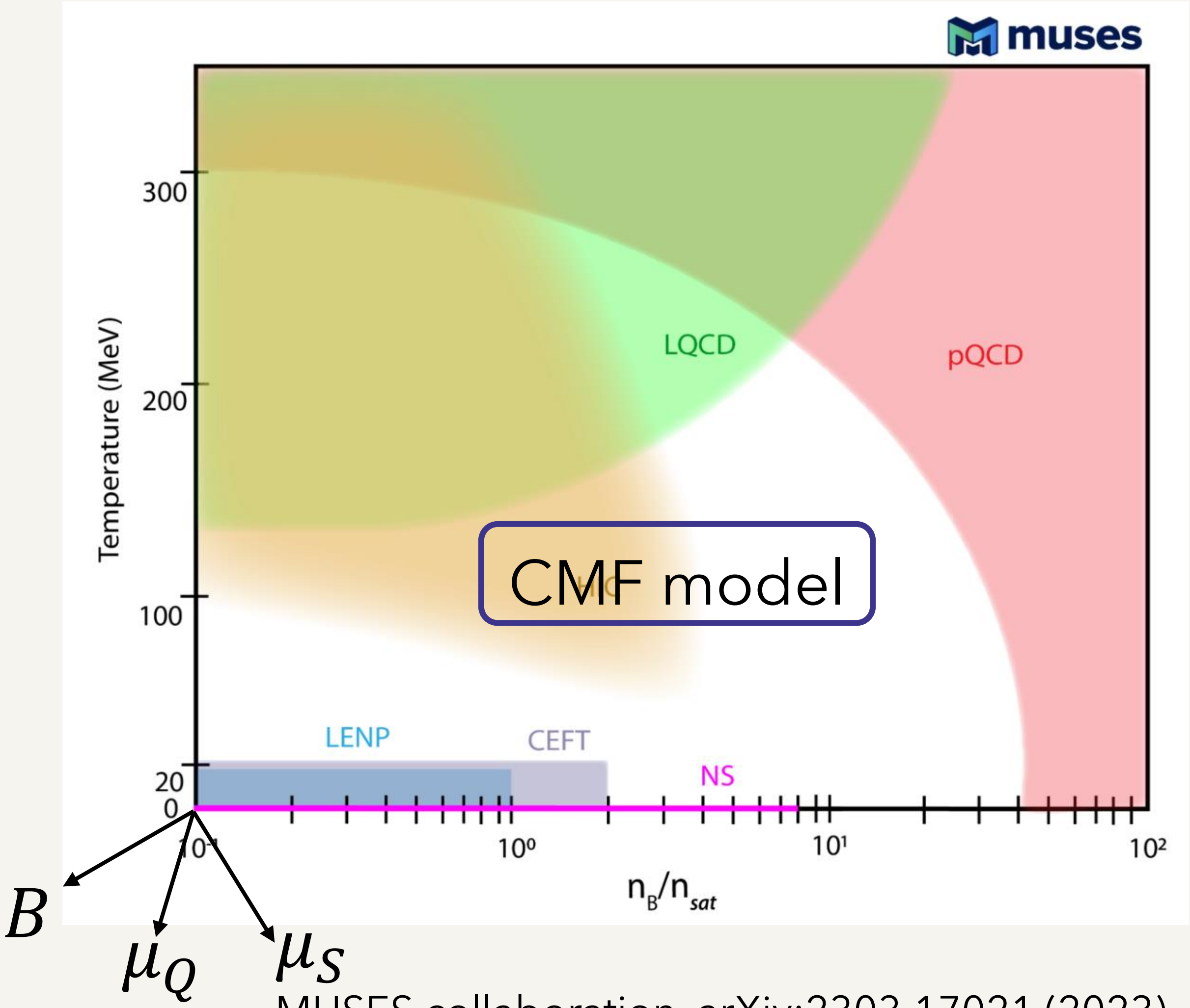
Outline

- Physical motivation / brief model description
- Last year improvements
- Results: Fortran77 vs C++20
- Next steps
- Summary



MUSES collaboration, arXiv:2303.17021 (2023).

Physical Motivation – CMF model



MUSES collaboration, arXiv:2303.17021 (2023).

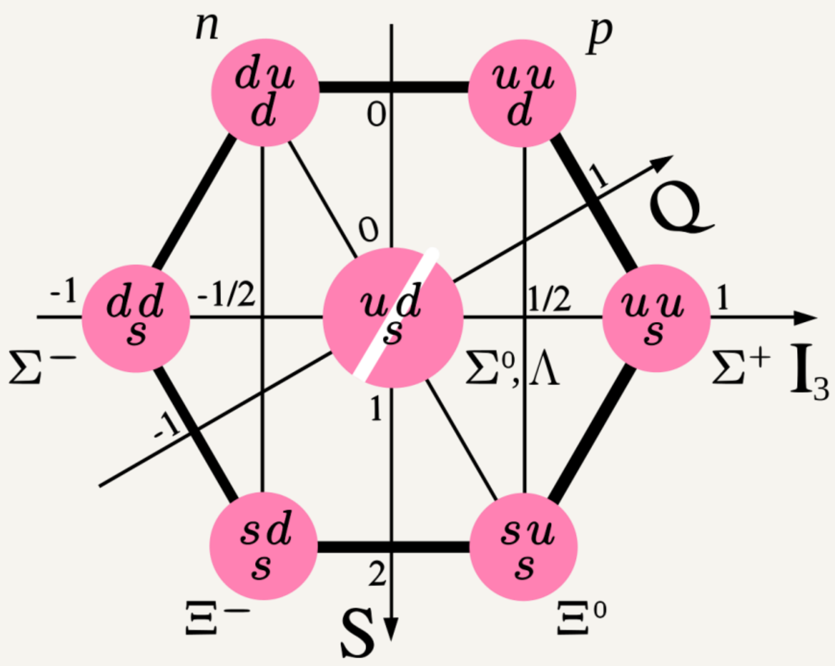
- How to complete the QCD phase diagram?
- How to merge lattice QCD with effective field theories?

CMF model

one 3D execution takes a month

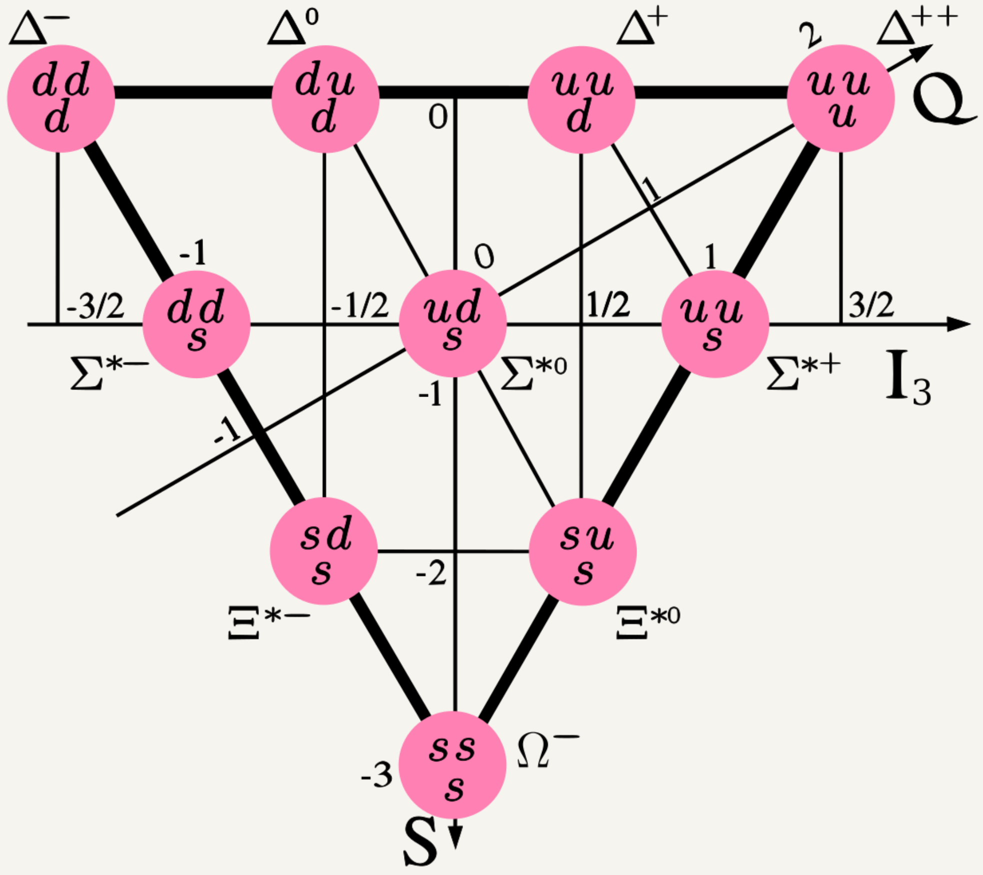
$$5D\colon\ T, \mu_B, \mu_S, \mu_Q, B$$

CMF - particles considered



Baryon octet

SU(3) quarks



Baryon decuplet

CMF Lagrangian

CMF
CHIRAL MEAN FIELD

$$\mathcal{L} = \mathcal{L}_{kin} + \mathcal{L}_{int} + \mathcal{L}_{Self} + \mathcal{L}_{SB} - U$$

Euler-Lagrange

$$\sigma: \quad 0 = \sum_i g_{i\sigma} n_S + k_0 \chi^2 \sigma - 4k_1 \left(\sigma^2 + \delta^2 + \zeta^2\right)\sigma - 2k_2\left(\sigma^2 + 3\delta^2\right)\sigma - 2k_3 \chi \sigma \zeta - \frac{2\epsilon}{3}\chi^4 \frac{\sigma}{\sigma^2 - \delta^2} + m_\pi^2 f_\pi$$

$$\zeta: \quad 0 = \sum_i g_{i\zeta} n_S + k_0 \chi^2 \zeta - 4k_1 \left(\sigma^2 + \delta^2 + \zeta^2\right)\zeta - 4k_2 \zeta^3 + k_3 \chi \left(\sigma^2 - \delta^2\right) - \frac{\epsilon}{3\zeta}\chi^4 + \sqrt{2} m_k^2 f_k - \frac{1}{\sqrt{2}} m_\pi^2 f_\pi$$

$$\delta: \quad 0 = \sum_i g_{i\delta} n_S + k_0 \chi^2 \delta - 4k_1 \left(\sigma^2 + \delta^2 + \zeta^2\right)\delta - 2k_2\left(3\sigma^2 + \delta^2\right)\delta + 2k_3 \chi \delta \zeta + \frac{2\epsilon}{3}\chi^4 \frac{\delta}{\sigma^2 - \delta^2}$$

$$\omega: \quad 0 = \sum_i g_{i\omega} n_B - m_\omega^2 \omega - 2g_4 \left[2\omega^3 + 6\beta\rho^2\omega + 3(1-\beta)\phi^2\omega + 3\sqrt{2}\alpha\omega^2\phi + \frac{\sqrt{2}}{2}\alpha\phi^3\right]$$

$$\phi: \quad 0 = \sum_i g_{i\phi} n_B - m_\phi^2 \phi - 2g_4 \left\{(3\beta + 1)\left(1 - \frac{\alpha}{2}\right)\phi^3 + 3(1-\beta)\left[(1-\alpha)\rho^2 + \omega^2\right]\phi + \sqrt{2}\alpha\omega^3 + \frac{3\alpha}{\sqrt{2}}\omega\phi^2\right\}$$

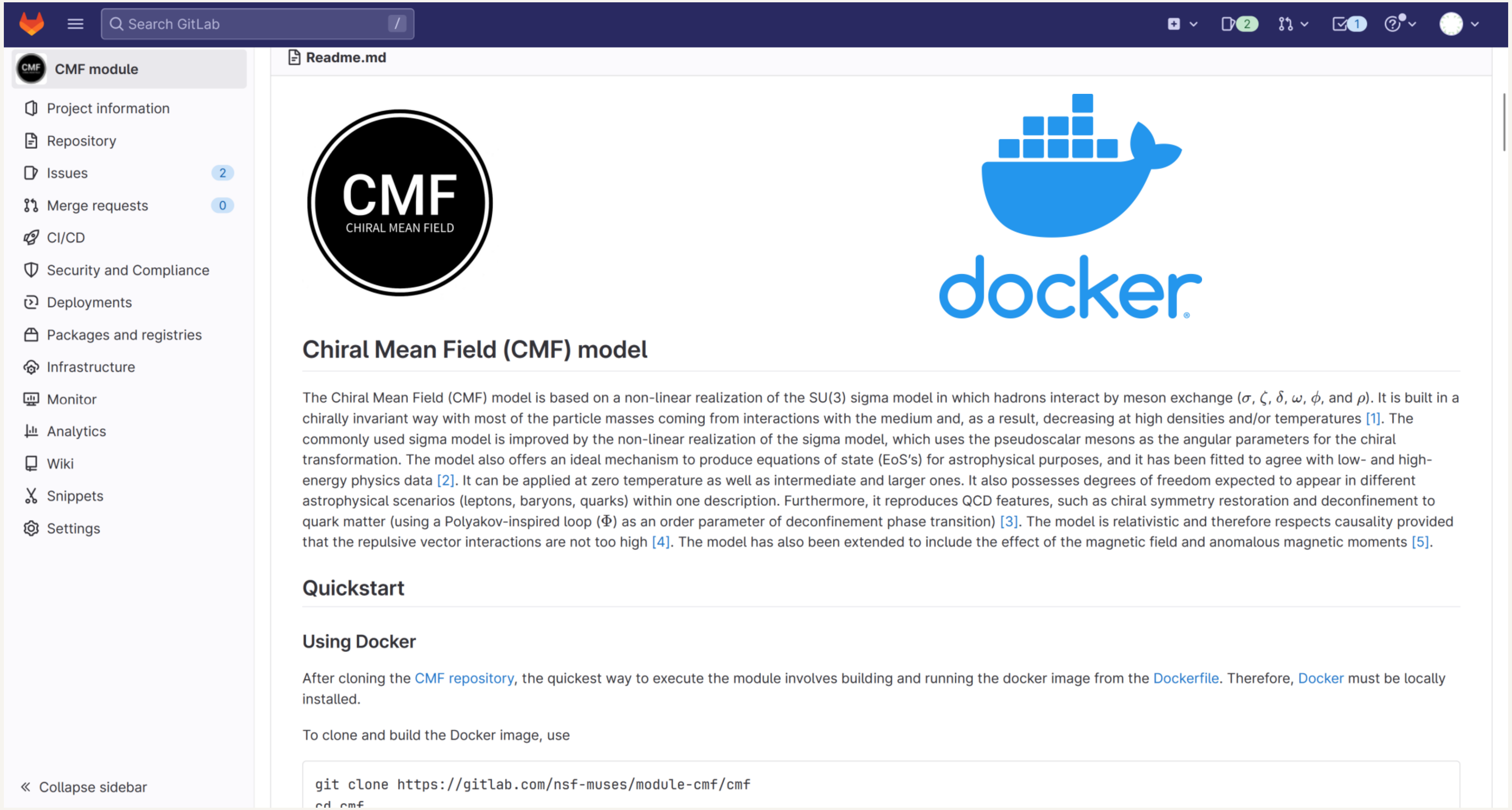
$$\rho: \quad 0 = \sum_i g_{i\rho} n_B - m_\rho^2 \rho - 2g_4 \left[2(1-\alpha)\rho^3 + 6\beta\rho\omega^2 + 3(1-\beta)(1-\alpha)\rho\phi^2\right]$$

$$\Phi: \quad 0 = \sum_i g_{i\Phi} n_S - 2\left(a_0 T^4 + a_1 \mu_B^4 + a_2 T^2 \mu_B^2\right)\Phi - a_3 T_0^4 \frac{12\Phi}{3\Phi^2 - 2\Phi - 1}$$

i goes over all particles

α, β allows four different vector potential configuration

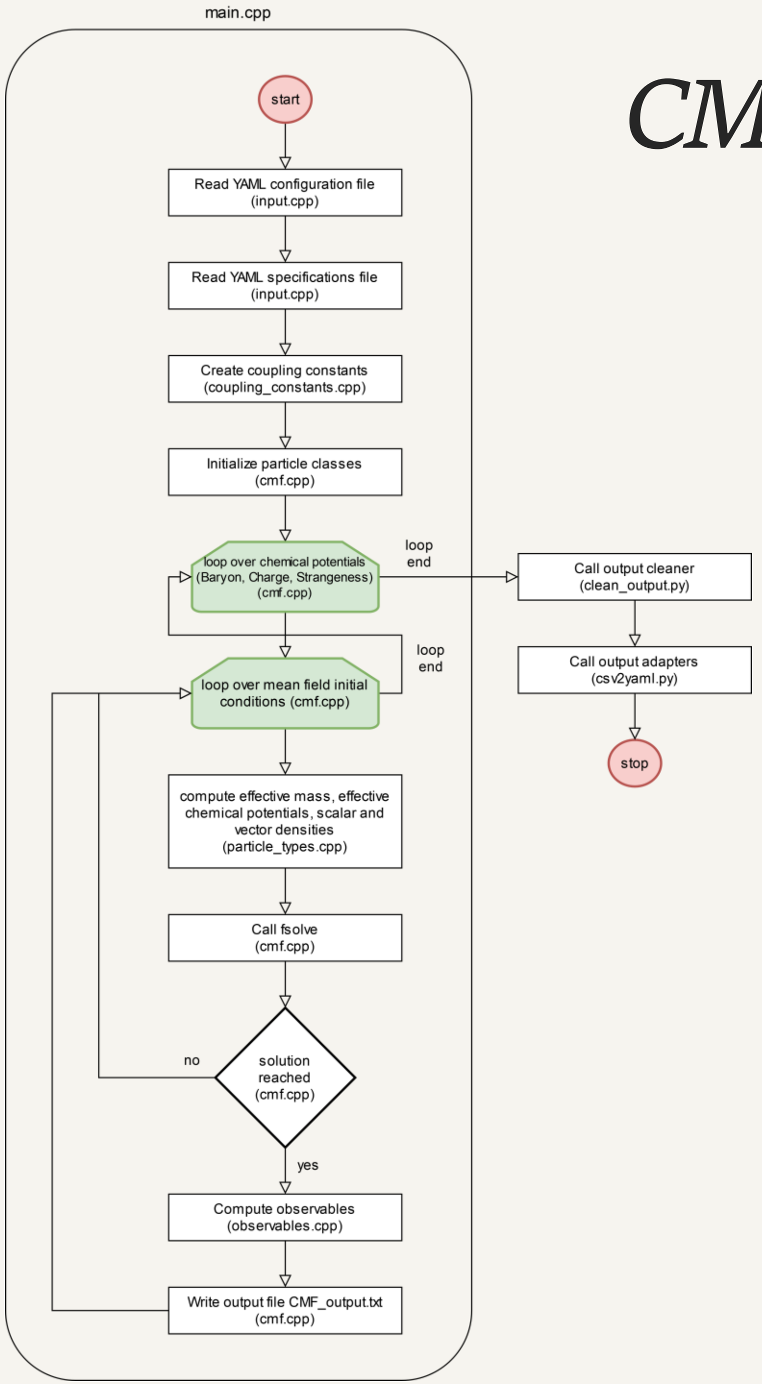
Last year improvements

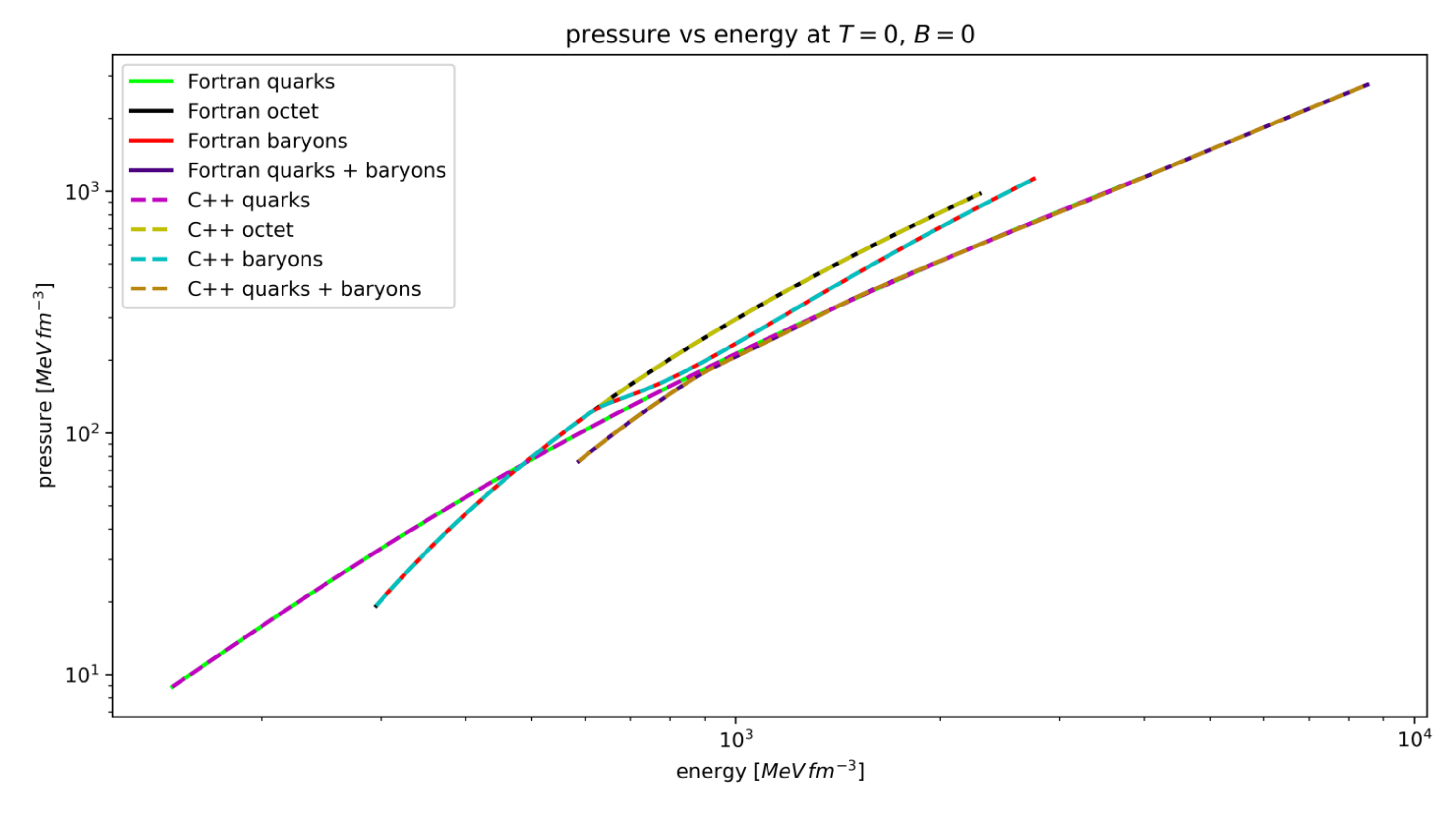


Zero temperature CMF C++ v0.4.0 is out!

git clone –b 0.4.0 https://gitlab.com/nsf-muses/module-cmf/cmf.git

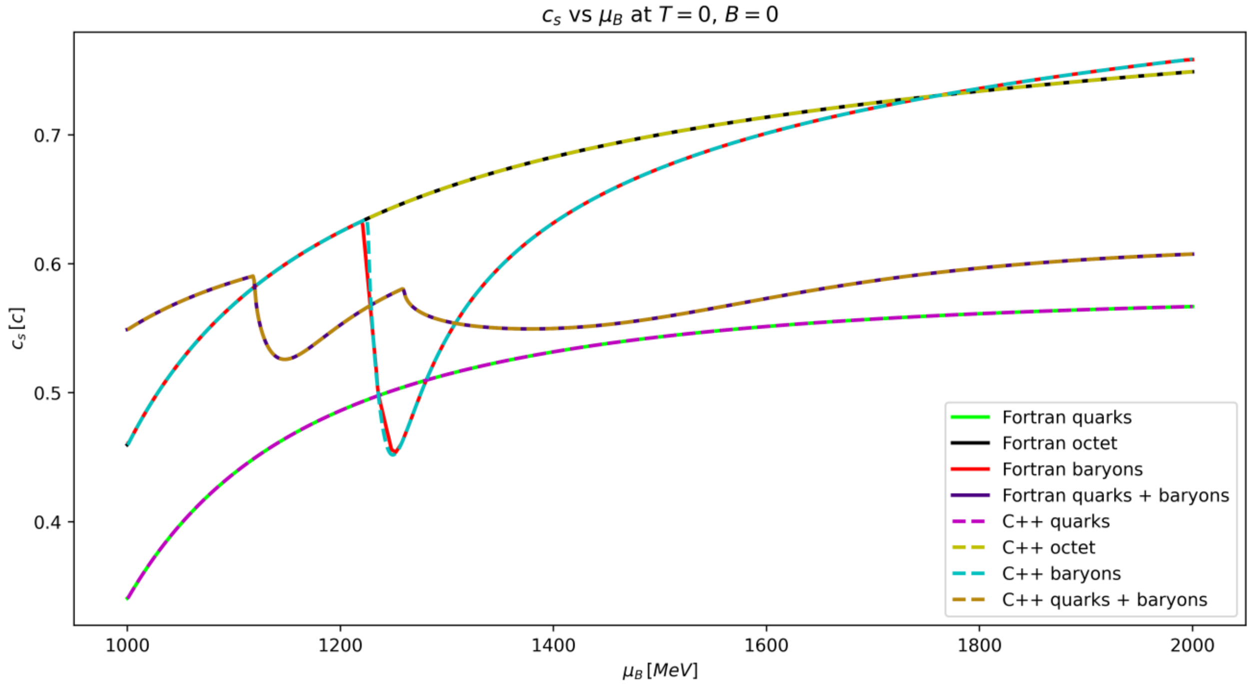
CMF 0.40. – simplified flowchart

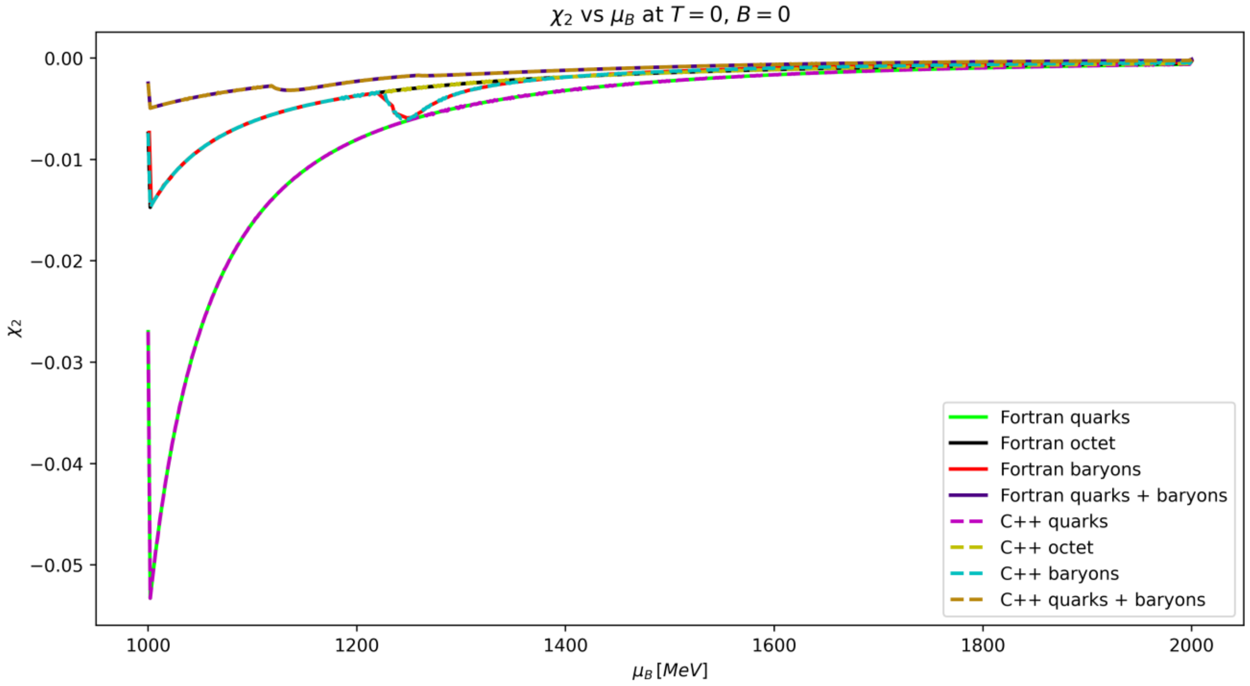




EoS agreement between Fortran and C++ (Polyakov off)

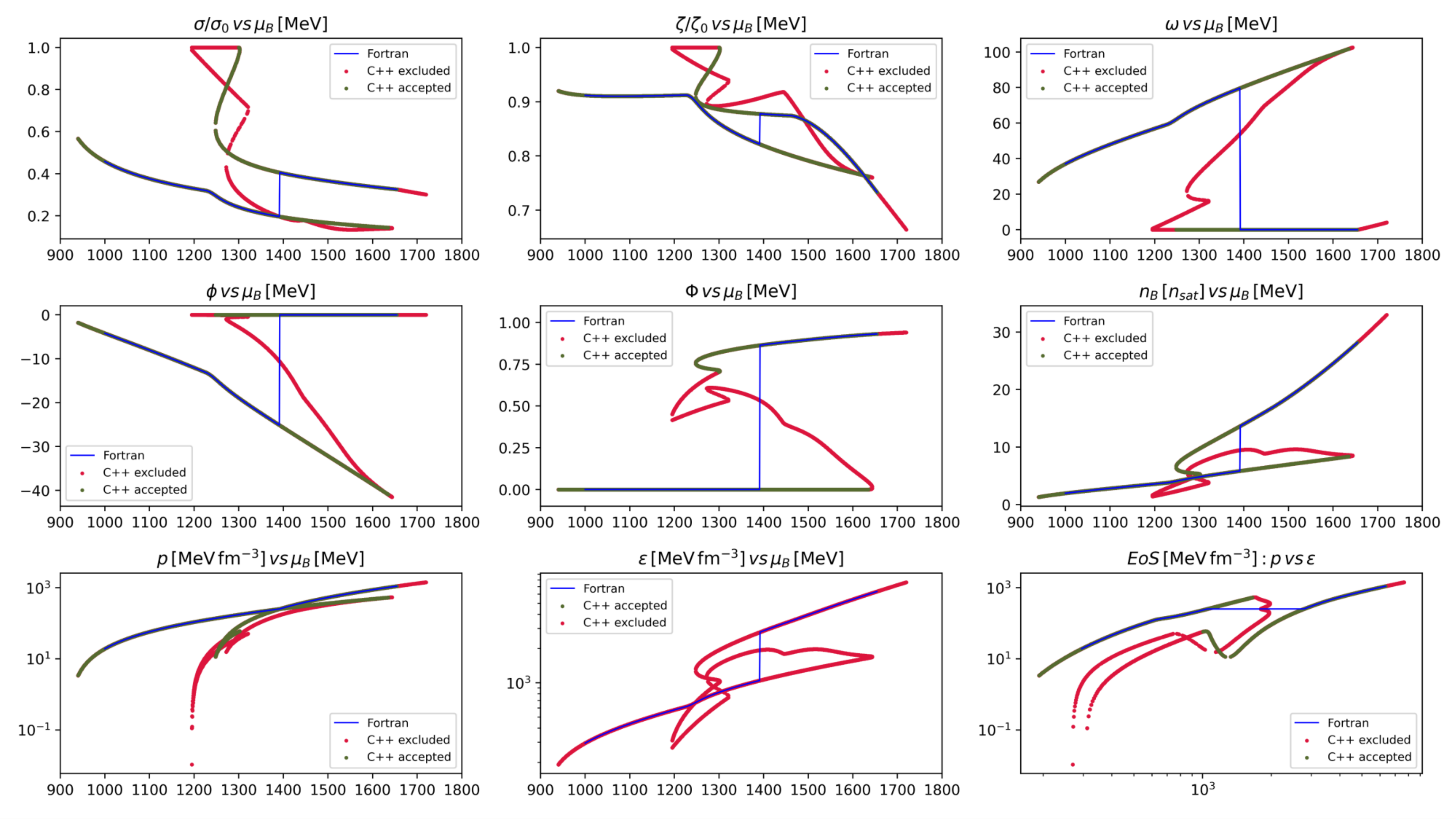
Results: Fortran vs C++ 0.4.0 – Polyakov off





High-resolution agreement on derived observables like the speed of sound and susceptibilities for diverse particle sets

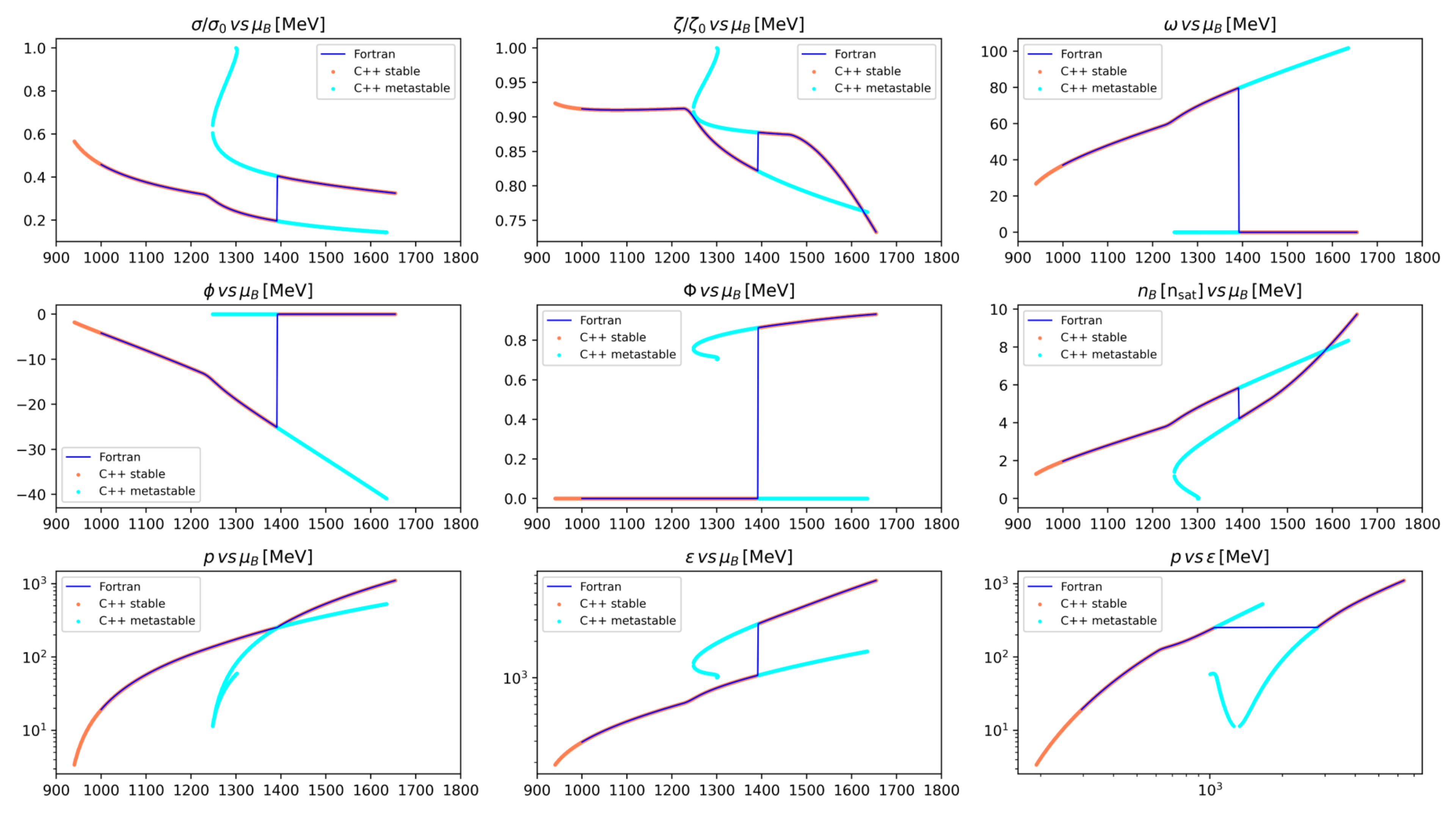
Results: Fortran vs C++ 0.4.0 – Polyakov on



Agreement is observed with an extra feature: C++ has more solutions around the metastable phase

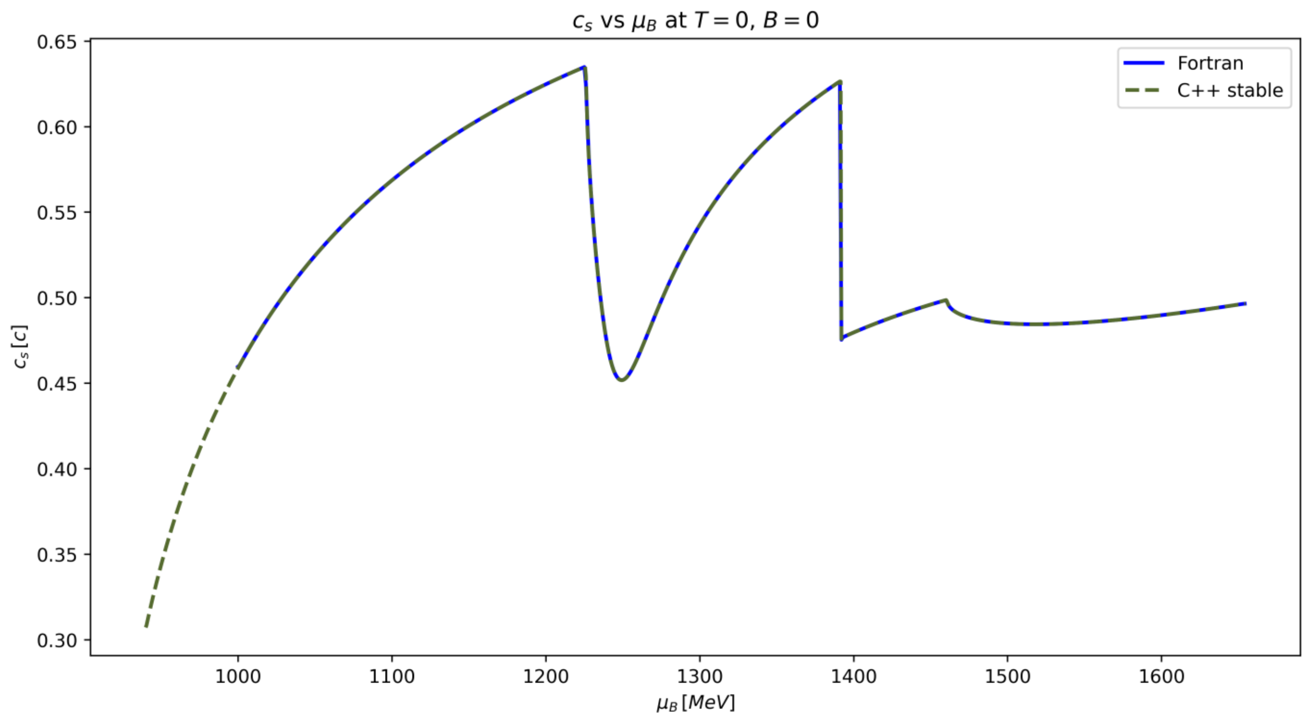
In red: unphysical data removed

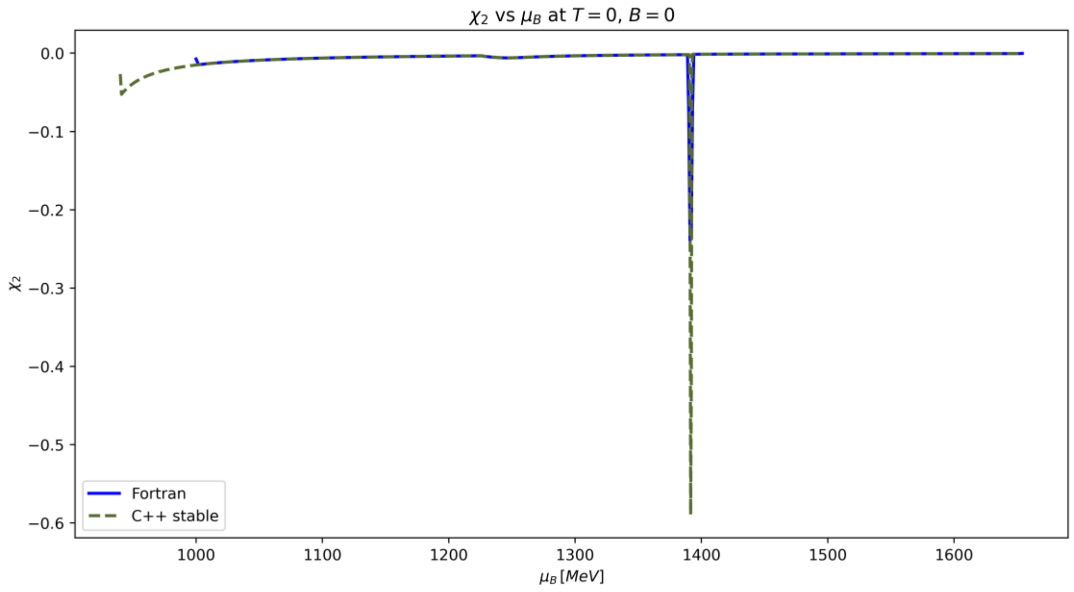
Results: Fortran vs C++ 0.4.0 – Metastable solutions

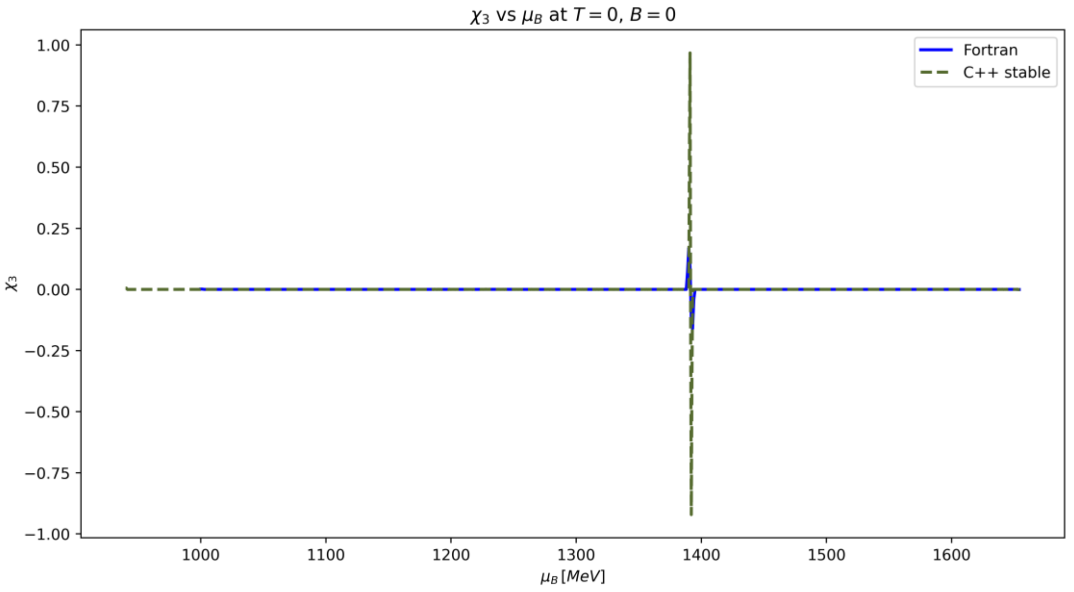


After removing unphysical data, and applying a maximum pressure filter, the stable and metastable regions are clearly seen

Results: Fortran vs C++ 0.4.0 – Polyakov on, observables







The stable region fully overlaps
for the speed of sound

The transition chemical potential
agrees with χ_2 and χ_3

Next steps

- Finish low-level source code documentation.
- Finish script to compute higher-order derivatives using central differences.
- Couple to Flavor equilibration module. (Z. Zhang)
- Test CMF 0.4.0+QLIMR workflow inside CE 0.10.0 (C. Conde).
- Extend for finite temperature.
- Include thermal mesons interactions (R. Kumar).
- Extend for magnetic field effects (J. Peterson).
- Couple to Lepton module.
- Parallelize via OMP+MPI. (R. Haas)

CMF
CHIRAL MEAN FIELD
+
QLIMR

Summary

Zero temperature implementation of the Chiral Mean Field model done in modern C++20.

High precision agreement with legacy Fortran77 version

Metastable region now accessible with the new code

Agreement for observables of the stable branch

Offline coupled to QLIMR

Source code containerized using Docker and automatization bash script for default Docker case created.

Quick note: if you need help creating a container of your current source code, please contact me in a coffee break or after lunch ☺

Questions?

Backup slides

CMF Lagrangian in detail

$$\mathcal{L} = \mathcal{L}_{kin} + \mathcal{L}_{int} + \mathcal{L}_{Self} + \mathcal{L}_{SB} - U$$

$$\mathcal{L}_{kin} = -i\overline{N}\gamma_i \nabla^i N - \frac{1}{2} \sum_{\varphi=\sigma,\zeta,\chi,\omega,\rho,A} \nabla_i \varphi \nabla^i \varphi$$

$$L_{Int} = -\sum_i \bar{\psi}_i [\gamma_0 (g_{i\omega}\omega + g_{i\phi}\phi + g_{i\rho}\tau_3\rho) + M_i^*]\psi_i,$$

CMF Lagrangian in detail

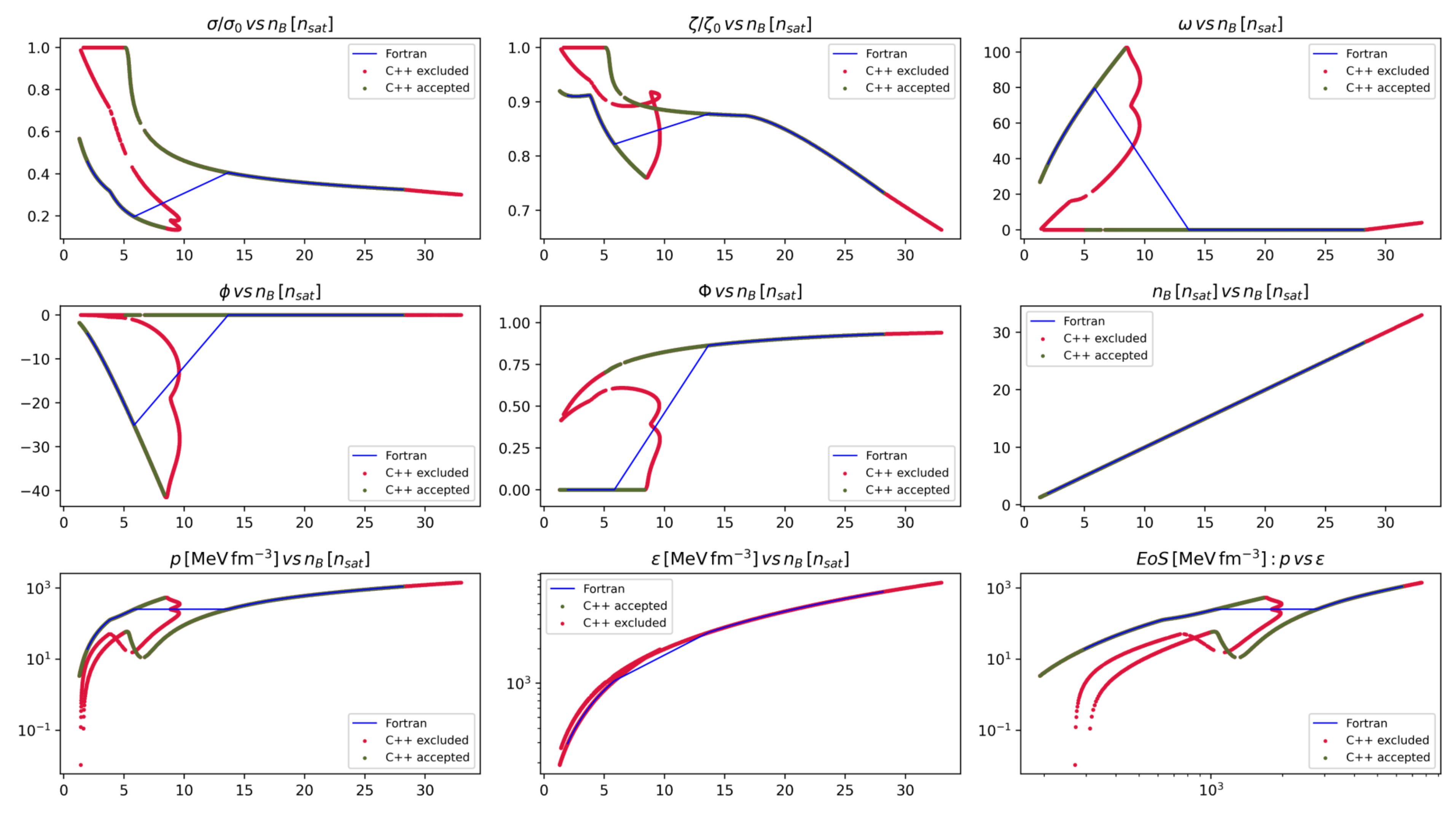
$$\mathcal{L} = \mathcal{L}_{kin} + \mathcal{L}_{int} + \mathcal{L}_{Self} + \mathcal{L}_{SB} - U$$

$$\begin{aligned}
L_{Self} &= -\tfrac{1}{2}(m_\omega^2\omega^2 + m_\rho^2\rho^2 + m_\phi^2\phi^2) \\
&+g_4\left(\omega^4 + \frac{\phi^4}{4} + 3\omega^2\phi^2 + \frac{4\omega^3\phi}{\sqrt{2}} + \frac{2\omega\phi^3}{\sqrt{2}}\right) \\
&+k_0(\sigma^2+\zeta^2+\delta^2) + k_1(\sigma^2+\zeta^2+\delta^2)^2 \\
&+k_2\left(\frac{\sigma^4}{2} + \frac{\delta^4}{2} + 3\sigma^2\delta^2 + \zeta^4\right) + k_3(\sigma^2-\delta^2)\zeta \\
&+k_4 \ \ln\frac{(\sigma^2-\delta^2)\zeta}{\sigma_0^2\zeta_0},
\end{aligned}$$

$$L_{SB} = m_\pi^2 f_\pi \sigma + \left(\sqrt{2}m_k^2 f_k - \frac{1}{\sqrt{2}}m_\pi^2 f_\pi\right)\zeta,$$

$$\begin{aligned}
U &= (a_0T^4 + a_1\mu^4 + a_2T^2\mu^2)\Phi^2 \\
&+a_3T_0^4\log(1-6\Phi^2+8\Phi^3-3\Phi^4).
\end{aligned}$$

Results: Fortran vs C++ 0.4.0 – Polyakov on



Agreement is observed with an extra feature: C++ has more solutions around the metastable phase

In red: unphysical data removed

Results: Fortran vs C++ 0.4.0 – Metastable