Approach to classical liquids based on functional renormalization group

Takeru Yokota RIKEN iTHEMS

Refs.

TY, Haruyama, Sugino, Functional-renormalization-group approach to classical liquids with short-range repulsion: A scheme without repulsive reference system, Phys. Rev. E 104, 014124 (2021)

TY, Physics-informed neural network for solving functional renormalization group on lattice, arXiv:2312.16038

Second Workshop on Fundamentals in Density Functional Theory (DFT2024), RIKEN Kobe campus, 22 Feb, 2024

About me

Main research topic: Functional renormalization group (FRG)

Graduate student at Kyoto U in 2014-2019

I started the study about
the combination of FRG and DFT.

w/ Prof. Kunihiro, Prof. Yoshida,
Dr. Naito, Mr. Kasuya

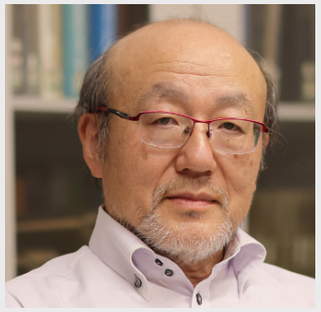
Prof. Yoshida

Dr. Naito

Researcher in the Sugino group at the Institute for Solid State Physics, U Tokyo in 2020-2021

I worked on FRG applications to
classical DFT. w/ Prof. Sugino, Dr. Haruyama

Prof. Sugino



Dr. Haruyama

Current position: Researcher in RIKEN iTHEMS

My recent interest is in **machine-learning approaches to FRG**.

About me

Main research topic: Functional renormalization group (FRG)

Graduate student at Kyoto U in 2014-2019

Prof. Yoshida

Dr. Naito

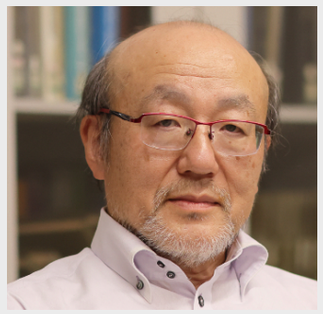
I started the study about
the combination of FRG and DFT.

w/ Prof. Kunihiro, Prof. Yoshida,
Dr. Naito, Mr. Kasuya

Today's main topic

Researcher in the Sugino group at the Institute for Solid State Physics, U Tokyo in 2020-2021

Prof. Sugino



Dr. Haruyama

I worked on FRG applications to
classical DFT. w/ Prof. Sugino, Dr. Haruyama

Current position: Researcher in RIKEN iTHEMS

My recent interest is in **machine-learning approaches to FRG**.

Related topic (if time allows...)

Contents

Functional renormalization group (FRG) for classical liquids

- FRG for classical scalar fields
- Path integral for classical DFT
- Modified FRG for classical liquids

Conventional approach: functional Taylor expansion

- Functional Taylor expansion & hierarchical eq.
- Approach for hard-core case
- Demonstration: 1D liquid

New approach: Machine learning

- Physics-informed neural network for solving FRG
- Demonstration: 0D O(N) model

Contents

Functional renormalization group (FRG) for classical liquids

- FRG for classical scalar fields
- Path integral for classical DFT
- Modified FRG for classical liquids

Conventional approach: functional Taylor expansion

- Functional Taylor expansion & hierarchical eq.
- Approach for hard-core case
- Demonstration: 1D liquid

TY, Haruyama, Sugino, PRE (2021)

New approach: Machine learning

- Physics-informed neural network for solving FRG
- Demonstration: 0D O(N) model

TY, arXiv:2312.16038

Contents

Functional renormalization group (FRG) for classical liquids

- FRG for classical scalar fields
- Path integral for classical DFT
- Modified FRG for classical liquids

Conventional approach: functional Taylor expansion

- Functional Taylor expansion & hierarchical eq.
- Approach for hard-core case
- Demonstration: 1D liquid

New approach: Machine learning

- Physics-informed neural network for solving FRG
- Demonstration: 0D O(N) model

Functional renormalization group (FRG) overview

FRG is…

Wegner, Houghton, PRA (1973), Wilson, Kogut,, PR (1974)
Polchinski, NPB (1984), Wetterich, PLB (1993)

Technique to analyze many-body systems

- Versatile tool: condensed matter, statistical phys., high-energy phys.…
- Non-perturbative approach
- Phase transition, phase diagram, thermodynamic quantities, correlations, …

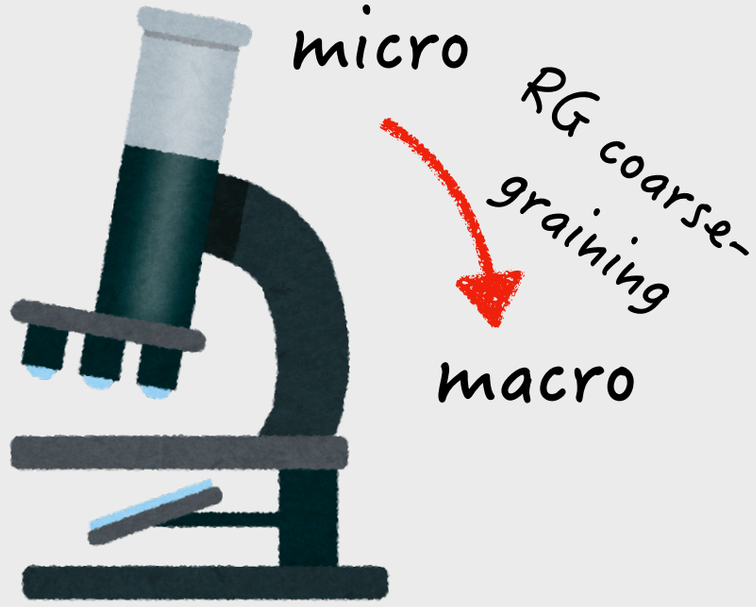
Rigorous formulation of RG flow

- Derivation: path-integral formulation
- Functional differential equation

Wetterich eq. *Wetterich, PLB (1993)*

$$\partial_k \Gamma_k[\phi] = \frac{1}{2}\mathrm{Tr}\left[\frac{\partial_k R_k}{\Gamma_k^{(2)}[\phi] + R_k}\right]$$

Governing eq. for "free energy" (I will explain it later)



Let's see the formulation in the classical case.

Classical scalar field theory & effective action

Let us consider scalar fields $\phi(x)$ statistically fluctuating following an action $S[\phi]$ (= Hamiltonian of the field $\beta H[\phi]$).

E.g.) 1. Classical ϕ^4 theory: $S[\phi] = \int_x \left(\frac{1}{2}\phi(\boldsymbol{x})(\nabla^2 + m^2)\phi(\boldsymbol{x}) + \frac{g}{4!}\phi(\boldsymbol{x})^4 \right)$

2. Action for classical DFT $\phi(\boldsymbol{x}) = \rho(\boldsymbol{x})$ (discussed later)

The partition function describes the statistical properties:

$$Z[J] = \int \mathscr{D}\phi e^{-S[\phi] + \int_x J(\boldsymbol{x})\phi(\boldsymbol{x})}$$

Path integral
= Summation over the configuration space

(recall $\sum_{s_1,\ldots,s_N} e^{-\beta H(\{s_i\})}$ in a spin system)

Classical scalar field theory & effective action

Let us consider scalar fields $\phi(x)$ statistically fluctuating following an action $S[\phi]$ (= Hamiltonian of the field $\beta H[\phi]$)**.**

E.g.) 1. Classical ϕ^4 theory: $S[\phi] = \int_x \left(\frac{1}{2}\phi(\boldsymbol{x})(\nabla^2 + m^2)\phi(\boldsymbol{x}) + \frac{g}{4!}\phi(\boldsymbol{x})^4 \right)$

2. Action for classical DFT $\phi(\boldsymbol{x}) = \rho(\boldsymbol{x})$ (discussed later)

The partition function describes the statistical properties:

$$Z[J] = \int \mathscr{D}\phi e^{-S[\phi] + \int_x J(\boldsymbol{x})\phi(\boldsymbol{x})}$$

Similarly, the effective action $\Gamma[\phi]$ describes them.

$$\Gamma[\phi] = \sup_J \left(\int_x J(\boldsymbol{x})\phi(\boldsymbol{x}) - \ln Z[J] \right)$$

- A kind of “free energy”
- It provides a variational principle for ϕ ($\langle\phi\rangle$ satisfies $\delta\Gamma[\phi]/\delta\phi(\boldsymbol{x}) = 0$)
- Its derivatives give the correlation functions.

Classical scalar field theory & effective action

Let us consider scalar fields $\phi(x)$ statistically fluctuating following an action $S[\phi]$ (= Hamiltonian of the field $\beta H[\phi]$)**.**

E.g.) 1. Classical ϕ^4 theory: $S[\phi] = \int_x \left(\frac{1}{2}\phi(\boldsymbol{x})(\nabla^2 + m^2)\phi(\boldsymbol{x}) + \frac{g}{4!}\phi(\boldsymbol{x})^4 \right)$

2. Action for classical DFT $\phi(\boldsymbol{x}) = \rho(\boldsymbol{x})$ (discussed later)

The partition funct
the statistical prop

The Wetterich's formalism of FRG is based on $\Gamma[\phi]$

Wetterich, PLB (1993)

Similarly, the effective action $\Gamma[\phi]$ describes them.

$$\Gamma[\phi] = \sup_J \left(\int_x J(\boldsymbol{x})\phi(\boldsymbol{x}) - \ln Z[J] \right)$$

- A kind of "free energy"
- It provides a variational principle for ϕ ($\langle\phi\rangle$ satisfies $\delta\Gamma[\phi]/\delta\phi(\boldsymbol{x}) = 0$)
- Its derivatives give the correlation functions.

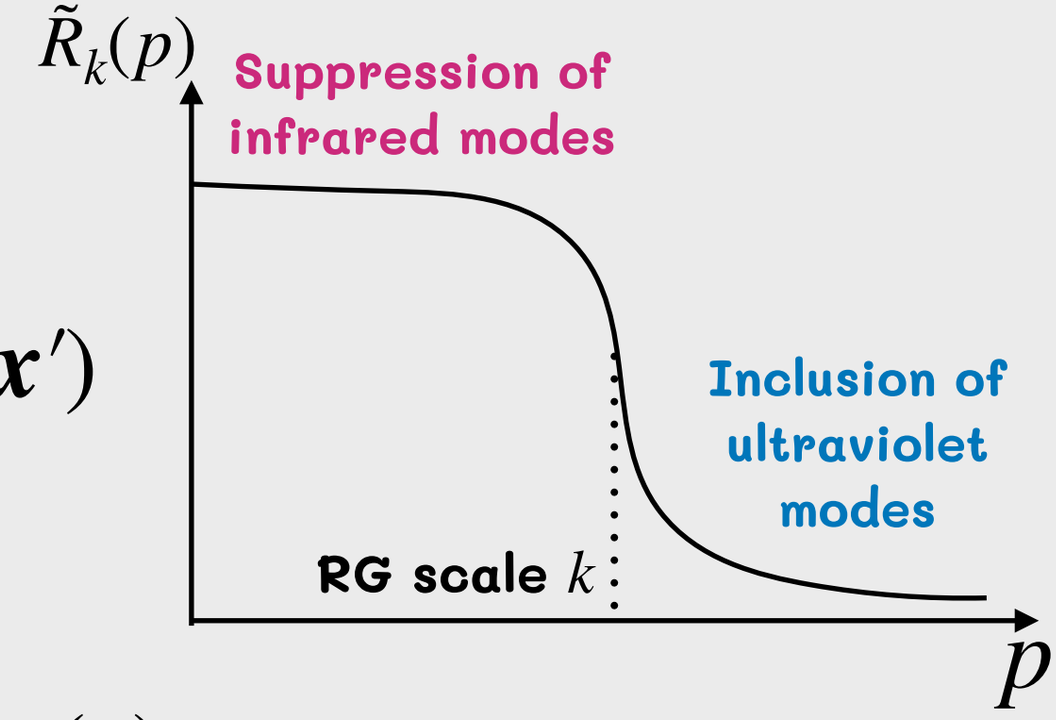
Wetterich's formalism of FRG

In RG, the fluctuations of $\phi(\boldsymbol{x})$ are gradually taken from high- to low-momentum components.

To implement this, we introduce **a regulator term**.

$$S_k[\phi] = S[\phi] + S_{\mathrm{reg},k}[\phi]$$

$$S_{\mathrm{reg},k}[\phi] = \frac{1}{2}\int_{\boldsymbol{x},\boldsymbol{x}'} \phi(\boldsymbol{x}) R_k(\boldsymbol{x}-\boldsymbol{x}')\phi(\boldsymbol{x}')$$



The regulator R_k is subject to

1. $\lim_{k_{\mathrm{UV}}\to\infty} \tilde{R}_{k_{\mathrm{UV}}}(p) = \infty$
2. $\lim_{k_{\mathrm{IR}}\to 0} \tilde{R}_{k_{\mathrm{IR}}}(p) = 0$

Flow equation

Scale-dependent effective action

$$\tilde{\Gamma}_k[\phi] = \sup_J \left(\int_x J(x)\phi(x) - \ln Z_k[J] \right) \qquad Z_k[J] = \int \mathscr{D}\phi e^{-S_k[\phi] + \int_x J(x)\phi(x)}$$

This interpolates $S[\phi]$ and $\Gamma[\phi]$:

For $k \to \infty$, $\tilde{\Gamma}_k[\phi] = S[\phi] + S_{\mathrm{reg},k}[\phi]$ $\because \lim_{k\to\infty} \tilde{R}_k(p) = \infty$

(saddle-point approx. in path int.)

For $k \to 0$, $\tilde{\Gamma}_k[\phi] = \Gamma[\phi]$ $\because \lim_{k\to 0} \tilde{R}_k(p) = 0$

The flow equation for $\tilde{\Gamma}_k[\phi]$ is derived as follows…

Derivation of flow eq.

$$\partial_k \tilde{\Gamma}_k[\phi] = -\frac{1}{Z_k[J_{\mathrm{sup},k}]}\partial_k Z_k[J_{\mathrm{sup},k}]$$

$$= \frac{1}{Z_k[J_{\mathrm{sup},k}]}\int \mathscr{D}\phi \partial_k S_{\mathrm{reg},k}[\phi] e^{-S_k[\phi]+\int_x J_{\mathrm{sup},k}(x)\phi(x)}$$

$$= \frac{1}{2}\int_{x,x'} \partial_k R_k(x-x') \frac{\int \mathscr{D}\phi' \phi'(x)\phi'(x') e^{-S_k[\phi']+\int_x J_{\mathrm{sup},k}(x)\phi'(x)}}{Z_k[J_{\mathrm{sup},k}]}$$

$$= \frac{1}{2}\int_{x,x'} \partial_k R_k(x-x') \frac{\delta^2 \ln Z_k[J_{\mathrm{sup},k}]}{\delta J(x)\delta J(x')} + \partial_k S_{\mathrm{reg},k}[\phi]$$

2pt propagator

Because $\dfrac{\delta\tilde{\Gamma}_k[\phi]}{\delta\phi(x)} = J_{\mathrm{sup},k}([\phi],x)$ **&** $\dfrac{\delta \ln Z_k[J_{\mathrm{sup},k}]}{\delta J(x)} = \phi(x)$

We have $\dfrac{\delta^2 \ln Z[J_{\mathrm{sup},k}]}{\delta J(x)\delta J(x')} = \left(\dfrac{\delta^2\tilde{\Gamma}_k[\phi]}{\delta\phi\delta\phi}\right)^{-1}(x,x')$

Functional inverse

$$\int_{x'} A(x,x')A^{-1}(x',x'') = \delta(x-x'')$$

Exact equation for RG flow

$$\partial_k \tilde{\Gamma}_k[\phi] = \frac{1}{2}\int_{\boldsymbol{x},\boldsymbol{x}'} \partial_k R_k(\boldsymbol{x}-\boldsymbol{x}')\left(\frac{\delta\tilde{\Gamma}_k[\phi]}{\delta\phi\delta\phi}\right)^{-1}(\boldsymbol{x}',\boldsymbol{x}) + \partial_k S_{\mathrm{reg},k}[\phi]$$

Initial condition: $\tilde{\Gamma}_{k_{\mathrm{UV}}}[\phi] = S[\phi] + S_{\mathrm{reg},k_{\mathrm{UV}}}[\phi]$

Solution: $\tilde{\Gamma}_0[\phi] = \Gamma[\phi]$

Notation We use a different notation from the Wetterich's one, where the flow equation is written as

$$\partial_k \Gamma_k[\phi] = \frac{1}{2}\mathrm{Tr}\left[\frac{\partial_k R_k}{\Gamma_k^{(2)}[\phi] + R_k}\right]$$

with the effective average action $\Gamma_k[\phi] = \tilde{\Gamma}_k[\phi] - S_{\mathrm{reg},k}[\phi]$.

Let us apply this formalism to classical liquids!

Classical liquids

Hamiltonian

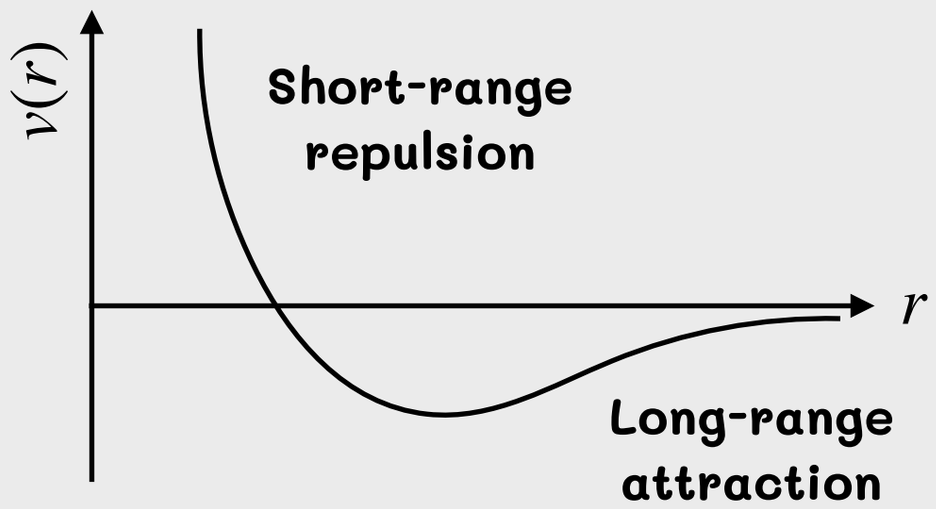
$$H_N = \sum_{i=1}^{N} \frac{\boldsymbol{p}_i^2}{2m} + V_N(\{\boldsymbol{x}_i\}_i)$$

N-body potential $V_N(\{\boldsymbol{x}_i\}_i) = \sum_{i<j} v(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{i<j<k} v_3(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k) + \cdots$

Everywhere around us!

Simple liquids

We only consider two-body interactions: $V_N(\{\boldsymbol{x}_i\}_i) = \sum_{i<j} v(|\boldsymbol{x}_i - \boldsymbol{x}_j|)$



Density field

We consider grand canonical ensemble (GCE).

Density (=positions of particles) is the fundamental degree of freedom.

$$\phi(\boldsymbol{x}) = \rho(\boldsymbol{x}) = \left\langle \sum_{i=1}^{\hat{N}} \delta(\boldsymbol{x} - \hat{\boldsymbol{x}}_i) \right\rangle_{\mathrm{GCE}}$$

$\hat{N}, \hat{\boldsymbol{x}}_i$ are averaged following GCE.

**In quantum cases, Hohenberg-Kohn thm. is required for this statement.*

“Density functional theory” naturally appears.

Let us find the action of density.

Partition function

Lue & Prausnitz, JCP (1998), Caillol, Mol. Phys. (2011)

To find the action, we consider the path integral of the grand partition function

$$\Xi = \sum_{N=0}^{\infty} \frac{1}{N!} \int_{\boldsymbol{p}_1,\ldots,\boldsymbol{p}_N} \int_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N} e^{-\beta H_N(\{\boldsymbol{x}_i\},\{\boldsymbol{p}_i\})+\beta\mu N}$$

$$= \sum_{N=0}^{\infty} \frac{n_0^N}{N!} \int_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N} e^{-\beta\left(\frac{1}{2}\int_{\boldsymbol{x},\boldsymbol{x}'} \rho_N(\boldsymbol{x}) v(\boldsymbol{x}-\boldsymbol{x}')\left(\rho_N(\boldsymbol{x}')-\delta(\boldsymbol{x}-\boldsymbol{x}')\right)\right)}$$

Density to be averaged $\rho_N(\boldsymbol{x}) = \sum_{i=1}^{N} \delta(\boldsymbol{x}-\boldsymbol{x}_i)$

$n_0 = \Lambda^{-3} e^{\beta\mu}$ with de Broglie thermal wavelength $\Lambda = \sqrt{2\pi\beta\hbar^2/m}$

[Ξ]

Lue & Prausnitz, JCP (1998)

$$= \sum_{N=0}^{\infty} \frac{n_0^N}{N!} \int_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N} e^{-\beta\left(\frac{1}{2}\int_{\boldsymbol{x},\boldsymbol{x}'} \rho_N(\boldsymbol{x}) v(\boldsymbol{x}-\boldsymbol{x}')\left(\rho_N(\boldsymbol{x}')-\delta(\boldsymbol{x}-\boldsymbol{x}')\right)\right)}$$

Path integral

$$\int \mathscr{D}\rho \prod_{\boldsymbol{y}} \delta(\rho(\boldsymbol{y}) - \rho_N(\boldsymbol{y})) = 1$$

$$= \int \mathscr{D}\rho e^{-\beta\left(\frac{1}{2}\int_{\boldsymbol{x},\boldsymbol{x}'} \rho(\boldsymbol{x}) v(\boldsymbol{x}-\boldsymbol{x}')\left(\rho(\boldsymbol{x}')-\delta(\boldsymbol{x}-\boldsymbol{x}')\right)\right)}$$

$$\times \sum_{N=0}^{\infty} \frac{n_0^N}{N!} \int_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N} \prod_{\boldsymbol{y}} \delta(\rho(\boldsymbol{y}) - \rho_N(\boldsymbol{y}))$$

Integral exp. of δ

$$\prod_{\boldsymbol{y}} \delta(\rho(\boldsymbol{y}) - \rho_N(\boldsymbol{y})) = \int \mathscr{D}\eta e^{i\beta \int d\boldsymbol{x} \eta(\boldsymbol{x})(\rho(\boldsymbol{x}) - \rho_N(\boldsymbol{x}))}$$

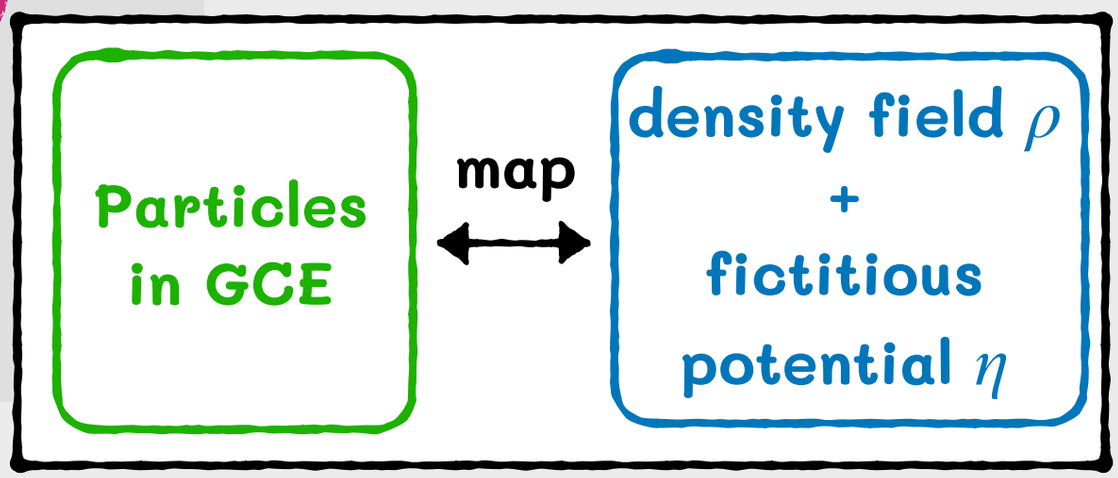
$$= \int \mathscr{D}\rho e^{-\beta\left(\frac{1}{2}\int_{\boldsymbol{x},\boldsymbol{x}'} \rho(\boldsymbol{x}) v(\boldsymbol{x}-\boldsymbol{x}')\left(\rho(\boldsymbol{x}')-\delta(\boldsymbol{x}-\boldsymbol{x}')\right)\right)}$$

$$\times \sum_{N=0}^{\infty} \frac{n_0^N}{N!} \int_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N} \int \mathscr{D}\eta e^{i\beta \int d\boldsymbol{x} \eta(\boldsymbol{x})(\rho(\boldsymbol{x}) - \rho_N(\boldsymbol{x}))}$$

Summation w.r.t. N

$$= \int \mathscr{D}\rho e^{-\beta\left(\frac{1}{2}\int_{\boldsymbol{x},\boldsymbol{x}'} \rho(\boldsymbol{x}) v(\boldsymbol{x}-\boldsymbol{x}')\left(\rho(\boldsymbol{x}')-\delta(\boldsymbol{x}-\boldsymbol{x}')\right)\right)}$$

$$\times \int \mathscr{D}\eta e^{\int_{\boldsymbol{x}} \left(i\beta\eta(\boldsymbol{x})\rho(\boldsymbol{x}) + n_0 e^{-i\beta\eta(\boldsymbol{x})}\right)}$$



Action for classical liquids

$$\Xi = \int \mathscr{D}\rho \, e^{-S[\rho]}$$

The action

$$S[\rho] = S_{\text{free}}[\rho] + \frac{\beta}{2} \int_{\boldsymbol{x},\boldsymbol{x}'} \rho(\boldsymbol{x}) v(\boldsymbol{x}-\boldsymbol{x}') \Big(\rho(\boldsymbol{x}') - \delta(\boldsymbol{x}-\boldsymbol{x}') \Big)$$

$$S_{\text{free}}[\rho] = \ln \left(\int \mathscr{D}\eta e^{\int_{\boldsymbol{x}} \left(i\beta\eta(\boldsymbol{x})\rho(\boldsymbol{x}) + n_0 e^{-i\beta\eta(\boldsymbol{x})} \right)} \right)$$

Action for classical liquids

$$\Xi = \int \mathscr{D}\rho \, e^{-S[\rho]}$$

The action

$$S[\rho] = S_{\text{free}}[\rho] + \frac{\beta}{2} \int_{\boldsymbol{x},\boldsymbol{x}'} \rho(\boldsymbol{x}) v(\boldsymbol{x}-\boldsymbol{x}') \Big(\rho(\boldsymbol{x}') - \delta(\boldsymbol{x}-\boldsymbol{x}') \Big)$$

$$S_{\text{free}}[\rho] = \ln \left(\int \mathscr{D}\eta e^{\int_x \left(i\beta\eta(\boldsymbol{x})\rho(\boldsymbol{x}) + n_0 e^{-i\beta\eta(\boldsymbol{x})} \right)} \right)$$

Complicated!! 😵

Practically usable? I will come back to this point later.

Effective action

Grand partition function
w/ one-body ext. field U

$$\Xi[U] = \int \mathscr{D}\rho e^{-S[\rho]+\int_{\boldsymbol{x}} \beta U(\boldsymbol{x})\rho(\boldsymbol{x})}$$

Grand potential

$$\Omega[U] = -\beta^{-1} \ln \Xi[U]$$

Effective action (=Free-energy density functional $\times \beta$)

$$\Gamma[\rho]\Big(= \beta F[\rho]\Big) = -\sup_{U} \left[\int_{\boldsymbol{x}} \rho(\boldsymbol{x})\beta(U(\boldsymbol{x}) - \mu) - \beta\Omega[U] \right]$$

This comprehensively describes the system.

Variational principle: $\delta F[\rho]/\delta\rho(\boldsymbol{x}) = \mu$
Thermodynamic quantities, correlations (derivatives)

Problem! Regulator & initial condition

$$S_{\mathrm{reg},k}[\rho] = \frac{1}{2}\int_{\boldsymbol{x},\boldsymbol{x}'} \rho(\boldsymbol{x}) R_k(\boldsymbol{x}-\boldsymbol{x}')\rho(\boldsymbol{x}')$$

By choosing a regulator satisfying $R_{k_{\mathrm{UV}}} \to \infty$, we have the initial condition $\Gamma_{k_{\mathrm{UV}}}[\rho] = S[\rho] + S_{\mathrm{reg},k_{\mathrm{UV}}}[\rho]$.

But the form of $S[\rho]$ is complicated and does not look useful.

$$S[\rho] = \ln\left(\int \mathscr{D}\eta e^{\int_x \left(i\beta\eta(\boldsymbol{x})\rho(\boldsymbol{x}) + n_0 e^{-i\beta\eta(\boldsymbol{x})}\right)}\right) + \frac{\beta}{2}\int_{\boldsymbol{x},\boldsymbol{x}'} \rho(\boldsymbol{x}) v(\boldsymbol{x}-\boldsymbol{x}')\Big(\rho(\boldsymbol{x}') - \delta(\boldsymbol{x}-\boldsymbol{x}')\Big)$$

How about discarding the condition $R_{k_{\mathrm{UV}}} \to \infty$ and introducing another type of flow?

Deformation of the two-body potential

$$S[\rho] + S_{\mathrm{reg},k}[\rho] = S_{\mathrm{free}}[\rho] + \frac{1}{2}\int_{\boldsymbol{x},\boldsymbol{x}'} \rho(\boldsymbol{x})\beta v(\boldsymbol{x}-\boldsymbol{x}')\big(\rho(\boldsymbol{x}')-\delta(\boldsymbol{x}-\boldsymbol{x}')\big) + \frac{1}{2}\int_{\boldsymbol{x},\boldsymbol{x}'} \rho(\boldsymbol{x})R_k(\boldsymbol{x}-\boldsymbol{x}')\rho(\boldsymbol{x}')$$

quadratic

Both the interaction and regulator terms are quadratic. Therefore, the regulator term can be regarded as **the deformation of** $\beta v(\boldsymbol{x})$.

For more simplification, we modify $S_{\mathrm{reg},k}[\rho]$ **as**

$$S_{\mathrm{reg},k}[\rho] = \frac{1}{2}\int_{\boldsymbol{x},\boldsymbol{x}'} \rho(\boldsymbol{x})R_k(\boldsymbol{x}-\boldsymbol{x}')\big(\rho(\boldsymbol{x}')-\delta(\boldsymbol{x}-\boldsymbol{x}')\big)$$

Then

$$S[\rho] + S_{\mathrm{reg},k}[\rho] = S_{\mathrm{free}}[\rho] + \frac{1}{2}\int_{\boldsymbol{x},\boldsymbol{x}'} \rho(\boldsymbol{x})\beta v_k(\boldsymbol{x}-\boldsymbol{x}')\big(\rho(\boldsymbol{x}')-\delta(\boldsymbol{x}-\boldsymbol{x}')\big)$$

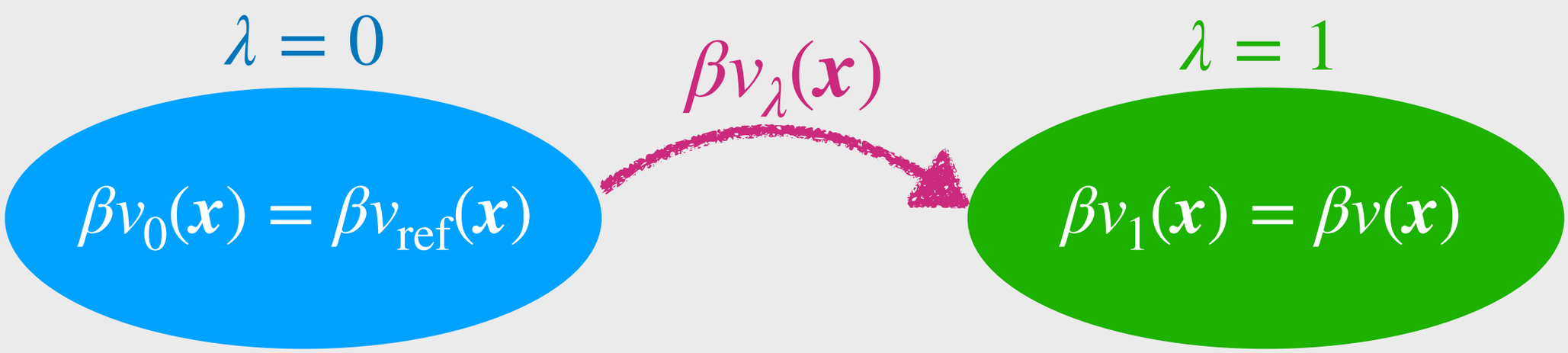
with $\beta v_k(\boldsymbol{x}) = R_k(\boldsymbol{x}) + \beta v(\boldsymbol{x})$

Deformation of the two-body potential

We no longer consider the original RG flow. Thus we replace k with a deformation parameter $\lambda \in [0,1]$.

$$S[\rho] + S_{\mathrm{reg},\lambda}[\rho] = S_{\mathrm{free}}[\rho] + \frac{1}{2}\int_{\boldsymbol{x},\boldsymbol{x}'} \rho(\boldsymbol{x})\beta v_\lambda(\boldsymbol{x}-\boldsymbol{x}')\Big(\rho(\boldsymbol{x}') - \delta(\boldsymbol{x}-\boldsymbol{x}')\Big)$$

With the condition $\beta v_{\lambda=1}(\boldsymbol{x}) = \beta v(\boldsymbol{x})$, the flow corresponds to the deformation from a reference system with βv_0 to the system of interest.



Flow eq.

The flow eq. is obtained in the same manner as the Wetterich eq.

$$\partial_\lambda \tilde{\Gamma}_\lambda[\rho] = \frac{1}{2}\int_{x,x'} \partial_\lambda R_\lambda(x-x')\left(\frac{\delta^2\tilde{\Gamma}_\lambda[\rho]}{\delta\rho\delta\rho}\right)^{-1}(x,x') + \partial_\lambda S_{\mathrm{reg},\lambda}[\rho]$$

$\tilde{\Gamma}_\lambda[\rho] = \beta F_\lambda[\rho]$ (the free-energy density functional with v_λ). Thus,

$$\partial_\lambda \beta F_\lambda[\rho] = \frac{1}{2}\int_{x,x'} \partial_\lambda \beta v_\lambda(x-x')\left(\rho(x)\rho(x') + \left(\frac{\delta^2\beta F_\lambda[\rho]}{\delta\rho\delta\rho}\right)^{-1}(x,x') - \rho(x)\delta(x-x')\right)$$

Similar methods based on flow eqs.

Hierarchical reference theory (HRT)

Parola, Reatto, PRA (1985)

- Almost the same equation is derived.
- v_λ is decomposed as $v_\lambda = v_{\mathrm{repulsive}} + v_{\mathrm{att},\lambda}$
- RG-like treatment of $v_{\mathrm{att},\lambda}$ (inclusion from high- to low-momentum components $\tilde{v}_{\mathrm{att}}(p)$)
- Phase transition (phase coexistence curve, critical exponent…)

Density RG

Iso, Kawana, PTEP (2019)

- Flow eq associated with the spacial scale transformation

Quantum version: FRG-DFT

Polonyi, Sailer (2002), Schwenk, Polonyi (2004)

- 1D nucleons [*TY, Yoshida, Kunihiro PRC (2019)*], 3D electrons [*TY, Naito PRR (2021)*]

How to solve functional differential eq.?

$$\partial_\lambda \beta F_\lambda[\rho] = \frac{1}{2}\int_{\boldsymbol{x},\boldsymbol{x}'} \partial_\lambda \beta v_\lambda(\boldsymbol{x}-\boldsymbol{x}')\left(\rho(\boldsymbol{x})\rho(\boldsymbol{x}') + \left(\frac{\delta^2 \beta F_\lambda[\rho]}{\delta\rho\delta\rho}\right)^{-1}(\boldsymbol{x},\boldsymbol{x}') - \rho(\boldsymbol{x})\delta(\boldsymbol{x}-\boldsymbol{x}')\right)$$

Initial condition: $F_{\lambda=0}[\rho] = F_{\mathrm{ref}}[\rho]$

Solving functional differential eqs. is formidable!

Solution method?

- **Conventional approximation: Taylor expansion**
  (Next section) *TY, Haruyama, Sugino, PRE (2021)*
- **New approach: Machine learning**
  (Next to next section) *TY, arXiv:2312.16038*

Contents

Functional renormalization group (FRG) for classical liquids

- FRG for classical scalar fields
- Path integral for classical DFT
- Modified FRG for classical liquids

Conventional approach: functional Taylor expansion

- Functional Taylor expansion & hierarchical eq.
- Approach for hard-core case
- Demonstration: 1D liquid

TY, Haruyama, Sugino, PRE (2021)

New approach: Machine learning

- Physics-informed neural network for solving FRG
- Demonstration: 0D O(N) model

Functional Taylor expansion

Liquids are almost homogeneous. We expand $F_\lambda[\rho]$ around ρ_{hom}.

$$\partial_\lambda \beta F_\lambda[\rho] = \frac{1}{2}\int_{\boldsymbol{x},\boldsymbol{x}'} \partial_\lambda \beta v_\lambda(\boldsymbol{x}-\boldsymbol{x}')\left[\rho(\boldsymbol{x})\rho(\boldsymbol{x}') + \left(\frac{\delta^2 \beta F_\lambda[\rho]}{\delta\rho\delta\rho}\right)^{-1}(\boldsymbol{x},\boldsymbol{x}') - \rho(\boldsymbol{x})\delta(\boldsymbol{x}-\boldsymbol{x}')\right]$$

$$\beta F_\lambda[\rho] = \beta F_\lambda[\rho_{\mathrm{hom}}] + \int d\boldsymbol{x}\beta F_\lambda^{(1)}([\rho_{\mathrm{hom}}],\boldsymbol{x})(\rho(\boldsymbol{x})-\rho_{\mathrm{hom}}) + \cdots$$

Hierarchical eq. for $F_\lambda[\rho_{\mathrm{hom}}]$, $F_\lambda^{(1)}[\rho_{\mathrm{hom}}]$, $F_\lambda^{(2)}[\rho_{\mathrm{hom}}]$, …

This should be truncated at some order to facilitate numerical computations.

Hierarchical eq. for correlation functions

$F_\lambda[\rho_{\rm hom}]$, $F_\lambda^{(1)}[\rho_{\rm hom}]$, $F_\lambda^{(2)}[\rho_{\rm hom}]$, … are related to the free energy, chemical potential, and distribution functions through variational eq. & Ornstein-Zernike eq.

Hierarchical eq. for distribution functions

0th order

Free energy

$$\partial_\lambda \frac{\beta F_\lambda}{N} = -\frac{\rho_{\rm hom}}{2}\int_{\mathbf{r}} \partial_\lambda \beta v_\lambda(\mathbf{r}) g_\lambda^{(2)}(\mathbf{r})$$

1st order

Chemical potential

$$\partial_\lambda \beta\mu_\lambda = \frac{\rho_{\rm hom}\int_{\mathbf{r}} \partial_\lambda \beta v_\lambda(\mathbf{r}) g_\lambda^{(2)}(\mathbf{r}) + \frac{\rho_{\rm hom}^2}{2}\int_{\mathbf{r},\mathbf{r}'} \partial_\lambda \beta v_\lambda(\mathbf{r}-\mathbf{r}')\left[g_\lambda^{(3)}(\mathbf{r},\mathbf{r}') - g_\lambda^{(2)}(\mathbf{r}-\mathbf{r}')\right]}{1+\rho_{\rm hom}\int_{\mathbf{r}}\left[g_\lambda^{(2)}(\mathbf{r})-1\right]}$$

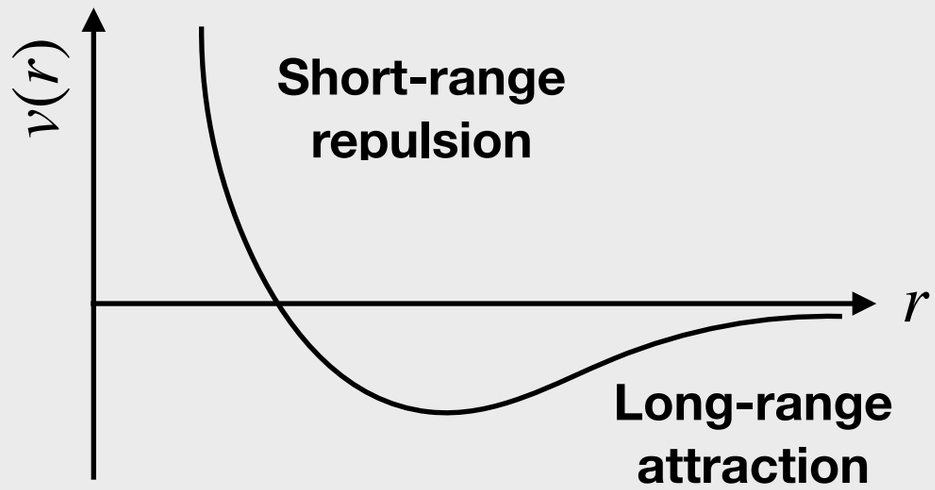
$n(\geq 2)$th order

n-particle distribution function

$$\begin{aligned}\partial_\lambda g_\lambda^{(n)}(\mathbf{r}_1,\cdots,\mathbf{r}_{n-1}) = &-\sum_{i<j}^{n}\partial_\lambda\left[\beta v_\lambda(\mathbf{r}_i-\mathbf{r}_j)\right] g_\lambda^{(n)}(\mathbf{r}_1,\cdots,\mathbf{r}_{n-1}) \\ &+ n g_\lambda^{(n)}(\mathbf{r}_1,\cdots,\mathbf{r}_{n-1})\partial_\lambda\mu_\lambda + \rho_{\rm hom}\partial_\lambda\mu_\lambda\int_{\mathbf{r}}\left[g_\lambda^{(n+1)}(\mathbf{r}_1,\cdots,\mathbf{r}_n,\mathbf{r}) - g_\lambda^{(n)}(\mathbf{r}_1,\cdots,\mathbf{r}_{n-1})\right] \\ &-\rho_{\rm hom}\int_{\mathbf{r}}\sum_{i=1}^{n}\partial_\lambda\left[\beta v_\lambda(\mathbf{r}-\mathbf{r}_i)\right] g_\lambda^{(n+1)}(\mathbf{r},\mathbf{r}_1,\cdots,\mathbf{r}_{n-1}) \\ &-\frac{\rho_{\rm hom}^2}{2}\int_{\mathbf{r},\mathbf{r}'}\partial_\lambda\left[\beta v_\lambda(\mathbf{r}-\mathbf{r}')\right]\left[g_\lambda^{(n+2)}(\mathbf{r},\mathbf{r}',\mathbf{r}_1,\cdots,\mathbf{r}_{n-1}) - g_\lambda^{(2)}(\mathbf{r}-\mathbf{r}')g_\lambda^{(n)}(\mathbf{r}_1,\cdots,\mathbf{r}_{n-1})\right]\end{aligned}$$

Hard-core divergence?

$$\partial_\lambda \frac{\beta F_\lambda}{N} = -\frac{\rho_{\text{hom}}}{2} \int_{\mathbf{r}} \partial_\lambda \beta v_\lambda(\mathbf{r}) g_\lambda^{(2)}(\mathbf{r})$$



At first glance, the repulsive core leads to a divergence.

In HRT, the repulsive core is treated as the reference in order to avoid to treat it in the flow eq.

$$v_\lambda = v_{\text{ref}} + v_{\text{att},\lambda} \quad \Rightarrow \quad \partial_\lambda v_\lambda = \partial_\lambda v_{\text{att},\lambda}$$

However, this divergence does not occur since $g_\lambda^{(2)}(\mathbf{r})$ is suppressed.

Cavity distribution function

In fact, the divergence is removed when we introduce

the cavity distribution function $y_\lambda^{(n)}(\mathbf{r},\cdots,\mathbf{r}_{n-1}) = e^{\beta\sum_{i<j}^{n} v_\lambda(\mathbf{r}_i-\mathbf{r}_j)} g_\lambda^{(n)}(\mathbf{r},\cdots,\mathbf{r}_{n-1})$

Exact eqs. for cavity distribution functions

No divergence!
(Only Mayer function $f_\lambda(\mathbf{r},\mathbf{r}') = e^{-\beta v_\lambda(\mathbf{r},\mathbf{r}')} - 1$ appears)

Method w/o hard-core reference!

0th

$$\partial_\lambda \frac{\beta F_\lambda}{N} = \frac{\rho_{\mathrm{hom}}}{2}\int_{\mathbf{r}} \partial_\lambda f_\lambda(\mathbf{r}) y_\lambda^{(2)}(\mathbf{r})$$

1st

$$\partial_\lambda \beta\mu_\lambda = \frac{-\rho_{\mathrm{hom}}\int_{\mathbf{r}} \partial_\lambda f_\lambda(\mathbf{r}) y_\lambda^{(2)}(\mathbf{r}) - \frac{\rho_{\mathrm{hom}}^2}{2}\int_{\mathbf{r},\mathbf{r}'} \partial_\lambda f_\lambda(\mathbf{r}-\mathbf{r}')\left[e^{-\beta v_\lambda(\mathbf{r})-\beta v_\lambda(\mathbf{r}')} y_\lambda^{(3)}(\mathbf{r},\mathbf{r}') - y_\lambda^{(2)}(\mathbf{r}-\mathbf{r}')\right]}{1+\rho_{\mathrm{hom}}\int_{\mathbf{r}}\left[e^{-\beta v_\lambda(\mathbf{r})} y_\lambda^{(2)}(\mathbf{r}) - 1\right]}$$

nth

$$\begin{aligned}\partial_\lambda y_\lambda^{(n)}(\mathbf{r}_1,\cdots,\mathbf{r}_{n-1}) = \partial_\lambda\beta\mu_\lambda \Bigg(& n y_\lambda^{(n)}(\mathbf{r}_1,\cdots,\mathbf{r}_{n-1}) + \rho_{\mathrm{hom}}\int_{\mathbf{r}}\left[e^{-\beta\sum_i^{n-1} v_\lambda(\mathbf{r}_i-\mathbf{r})-\beta v_\lambda(\mathbf{r})} y_\lambda^{(n+1)}(\mathbf{r}_1,\cdots,\mathbf{r}_{n-1},\mathbf{r}) - y_\lambda^{(n)}(\mathbf{r}_1,\cdots,\mathbf{r}_{n-1})\right]\Bigg) \\ & + \rho_{\mathrm{hom}}\int_{\mathbf{r}} \partial_\lambda\left[e^{-\beta\sum_i^{n-1} v_\lambda(\mathbf{r}_i-\mathbf{r})-\beta v_\lambda(\mathbf{r})}\right] y_\lambda^{(n+1)}(\mathbf{r},\mathbf{r}_1,\cdots,\mathbf{r}_{n-1}) \\ & + \frac{\rho_{\mathrm{hom}}^2}{2}\int_{\mathbf{r},\mathbf{r}'} \partial_\lambda f_\lambda(\mathbf{r}-\mathbf{r}')\left[e^{-\beta\sum_i^{n-1} v_\lambda(\mathbf{r}_i-\mathbf{r})-\beta\sum_i^{n-1} v_\lambda(\mathbf{r}_i-\mathbf{r}')-\beta v_\lambda(\mathbf{r})-\beta v_\lambda(\mathbf{r}')} y_\lambda^{(n+2)}(\mathbf{r},\mathbf{r}',\mathbf{r}_1,\cdots,\mathbf{r}_{n-1}) - y_\lambda^{(2)}(\mathbf{r}-\mathbf{r}') y_\lambda^{(n)}(\mathbf{r}_1,\cdots,\mathbf{r}_{n-1})\right]\end{aligned}$$

The repulsive reference is unnecessary.

We use the free-gas reference $v_0 = 0$. $\beta F_0[\rho] = \int_{\boldsymbol{x}} \rho(\boldsymbol{x})\Big(\ln\left(\Lambda^3\rho(\boldsymbol{x})\right) - 1\Big)$

Truncation up to 2nd order

0th order

$$\partial_\lambda \frac{\beta F_\lambda}{N} = \frac{\rho_{\mathrm{hom}}}{2} \int_{\mathbf{r}} \partial_\lambda f_\lambda(\mathbf{r}) y_\lambda^{(2)}(\mathbf{r})$$

1st order

$$\partial_\lambda \beta\mu_\lambda = \frac{-\rho_{\mathrm{hom}} \int_{\mathbf{r}} \partial_\lambda f_\lambda(\mathbf{r}) y_\lambda^{(2)}(\mathbf{r}) - \frac{\rho_{\mathrm{hom}}^2}{2} \int_{\mathbf{r},\mathbf{r}'} \partial_\lambda f_\lambda(\mathbf{r}-\mathbf{r}') \left[e^{-\beta v_\lambda(\mathbf{r}) - \beta v_\lambda(\mathbf{r}')} y_\lambda^{(3)}(\mathbf{r},\mathbf{r}') - y_\lambda^{(2)}(\mathbf{r}-\mathbf{r}') \right]}{1 + \rho_{\mathrm{hom}} \int_{\mathbf{r}} \left[e^{-\beta v_\lambda(\mathbf{r})} y_\lambda^{(2)}(\mathbf{r}) - 1 \right]}$$

2nd order

$$\partial_\lambda y_\lambda^{(2)}(\mathbf{r}_1) = \partial_\lambda \beta\mu_\lambda \left(2 y_\lambda^{(2)}(\mathbf{r}_1) + \rho_{\mathrm{hom}} \int_{\mathbf{r}} \left[e^{-\beta v_\lambda(\mathbf{r}_1-\mathbf{r}) - \beta v_\lambda(\mathbf{r})} y_\lambda^{(3)}(\mathbf{r}_1,\mathbf{r}) - y_\lambda^{(2)}(\mathbf{r}_1) \right] \right)$$

$$+ \rho_{\mathrm{hom}} \int_{\mathbf{r}} \partial_\lambda \left[e^{-\beta v_\lambda(\mathbf{r}_1-\mathbf{r}) - \beta v_\lambda(\mathbf{r})} \right] y_\lambda^{(3)}(\mathbf{r},\mathbf{r}_1)$$

$$+ \frac{\rho_{\mathrm{hom}}^2}{2} \int_{\mathbf{r},\mathbf{r}'} \partial_\lambda f_\lambda(\mathbf{r}-\mathbf{r}') \left[e^{-\beta v_\lambda(\mathbf{r}_1-\mathbf{r}) - \beta v_\lambda(\mathbf{r}_1-\mathbf{r}') - \beta v_\lambda(\mathbf{r}) - \beta v_\lambda(\mathbf{r}')} y_\lambda^{(4)}(\mathbf{r},\mathbf{r}',\mathbf{r}_1) - y_\lambda^{(2)}(\mathbf{r}-\mathbf{r}') y_\lambda^{(2)}(\mathbf{r}_1) \right]$$

Approximation is needed

Truncation up to 2nd order

0th order

$$\partial_\lambda \frac{\beta F_\lambda}{N} = \frac{\rho_{\mathrm{ho}}}{2}$$

We close the hierarchy using

Kirkwood superposition approximation (KSA)

Kirkwood, JCP (1935)

- Ignorance of mean many-body potential
- Exact for low densities

$$y_\lambda^{(3)}(\mathbf{r},\mathbf{r}_1) \approx y_\lambda^{(2)}(\mathbf{r}_1) y_\lambda^{(2)}(\mathbf{r}) y_\lambda^{(2)}(\mathbf{r}-\mathbf{r}_1)$$

$$y_\lambda^{(4)}(\mathbf{r},\mathbf{r}',\mathbf{r}_1) \approx y_\lambda^{(2)}(\mathbf{r}) y_\lambda^{(2)}(\mathbf{r}') y_\lambda^{(2)}(\mathbf{r}_1) y_\lambda^{(2)}(\mathbf{r}-\mathbf{r}') y_\lambda^{(2)}(\mathbf{r}'-\mathbf{r}_1) y_\lambda^{(2)}(\mathbf{r}_1-\mathbf{r})$$

1st order

$$\partial_\lambda \beta\mu_\lambda = \frac{-\rho_{\mathrm{hom}} \int_{\mathbf{r}} \partial_\lambda f_\lambda(\mathbf{r}) y_\lambda^{(2)}(\mathbf{r}) - \frac{\rho_{\mathrm{hom}}^2}{2} \int_{\mathbf{r},\mathbf{r}'} \partial_\lambda f_\lambda(\mathbf{r}-\mathbf{r}') \left[e^{-\beta v_\lambda(\mathbf{r}) - \beta v_\lambda(\mathbf{r}')} y_\lambda^{(3)}(\mathbf{r},\mathbf{r}') - y_\lambda^{(2)}(\mathbf{r}-\mathbf{r}') \right]}{1 + \rho_{\mathrm{hom}} \int_{\mathbf{r}} \left[e^{-\beta v_\lambda(\mathbf{r})} y_\lambda^{(2)}(\mathbf{r}) - 1 \right]}$$

2nd order

$$\partial_\lambda y_\lambda^{(2)}(\mathbf{r}_1) = \partial_\lambda \beta\mu_\lambda \left(2 y_\lambda^{(2)}(\mathbf{r}_1) + \rho_{\mathrm{hom}} \int_{\mathbf{r}} \left[e^{-\beta v_\lambda(\mathbf{r}_1-\mathbf{r}) - \beta v_\lambda(\mathbf{r})} y_\lambda^{(3)}(\mathbf{r}_1,\mathbf{r}) - y_\lambda^{(2)}(\mathbf{r}_1) \right] \right)$$

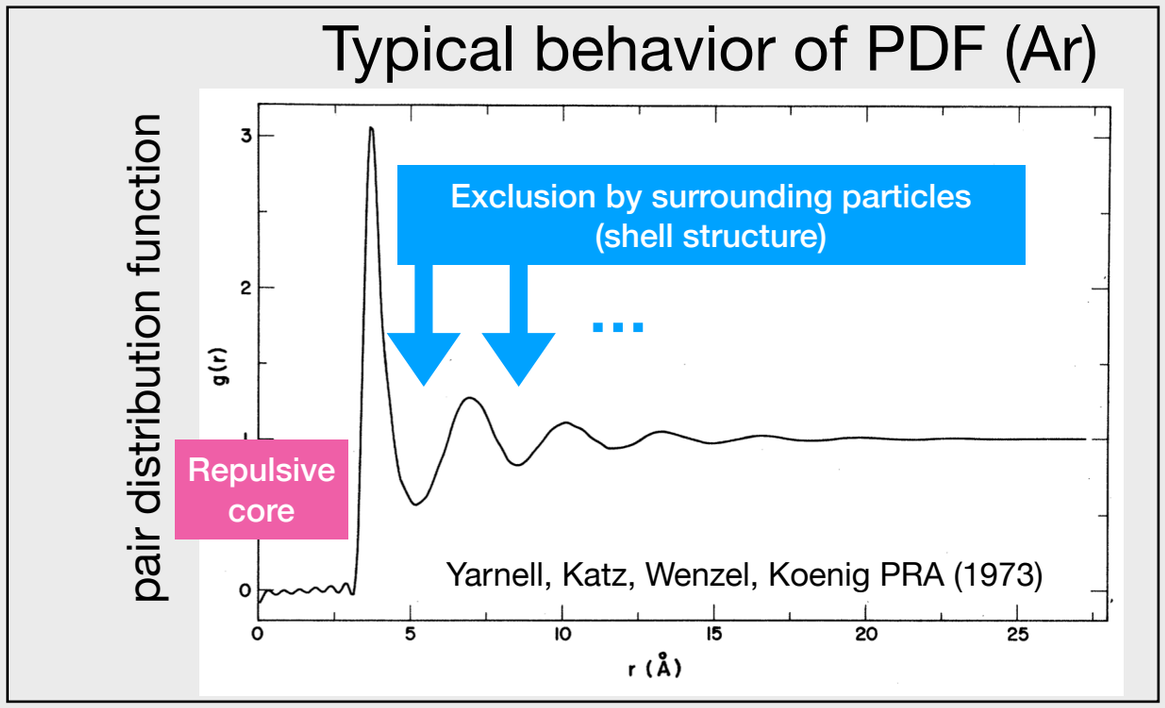
$$+ \rho_{\mathrm{hom}} \int_{\mathbf{r}} \partial_\lambda \left[e^{-\beta v_\lambda(\mathbf{r}_1-\mathbf{r}) - \beta v_\lambda(\mathbf{r})} \right] y_\lambda^{(3)}(\mathbf{r},\mathbf{r}_1)$$

Approximation is needed

$$+ \frac{\rho_{\mathrm{hom}}^2}{2} \int_{\mathbf{r},\mathbf{r}'} \partial_\lambda f_\lambda(\mathbf{r}-\mathbf{r}') \left[e^{-\beta v_\lambda(\mathbf{r}_1-\mathbf{r}) - \beta v_\lambda(\mathbf{r}_1-\mathbf{r}') - \beta v_\lambda(\mathbf{r}) - \beta v_\lambda(\mathbf{r}')} y_\lambda^{(4)}(\mathbf{r},\mathbf{r}',\mathbf{r}_1) - y_\lambda^{(2)}(\mathbf{r}-\mathbf{r}') y_\lambda^{(2)}(\mathbf{r}_1) \right]$$

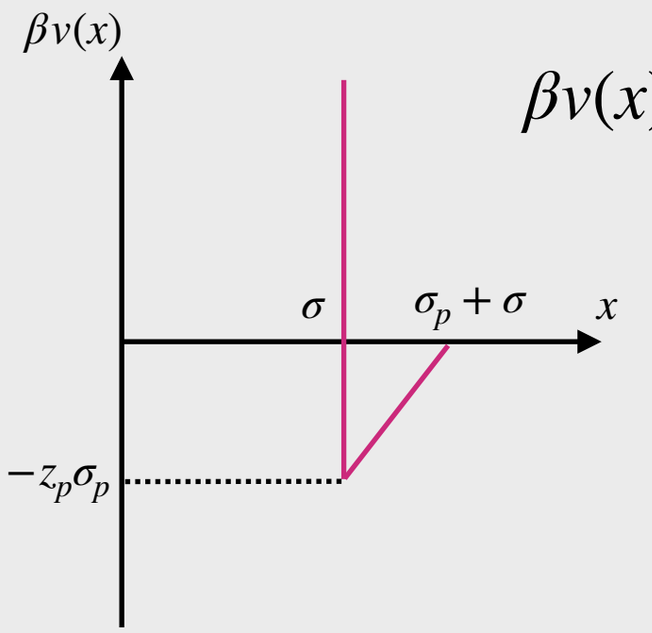
1D solvable hard rod

Let us apply our method to the derivation of the pair distribution function (PDF) & free energy in a 1D solvable model.



Model

Archer, Chacko, Evans, J. Chem. Phys. (2017)



$$\beta v(x) = \begin{cases} \infty & (|x| \le \sigma) \\ -z_p(\sigma + \sigma_p - |x|) & (\sigma < |x| \le \sigma + \sigma_p) \\ 0 & (\sigma + \sigma_p \le |x|) \end{cases}$$

In the case of $\sigma_p < \sigma$, thermodynamic quantities & pair correlation function are obtained exactly.

Takahashi, Proc. Phys.-Math. Soc. Jpn. (1942)
Percus, J. Stat. Phys. (1982)
Brader, Evans, Physica A (2002)

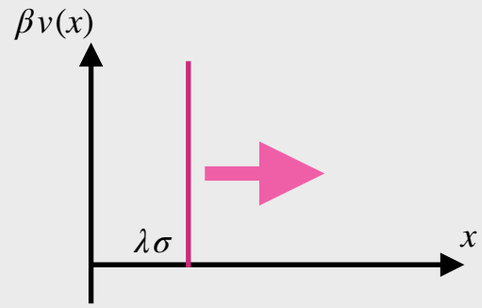
Choice of evolution

Naive adiabatic connection cannot be used for hard core

$f_\lambda(\boldsymbol{x}) = e^{-\lambda\beta v(\boldsymbol{r})} - 1$ suddenly changes at $\lambda = 0$

Instead, we introduce the evolution of **hard-core diameter**

$$\beta v_\lambda^{\mathrm{hard}}(x) = \begin{cases} \infty & (|x| \le \lambda\sigma) \\ 0 & (\lambda\sigma < |x|) \end{cases}$$



This is compatible with KSA.

$\rho_{\mathrm{hom}}\sigma$ is the only dimensionless parameter

→ Small σ corresponds to small ρ_{hom}

At the beginning of the flow (small σ), KSA is very accurate

The evolution of the attractive part is the adiabatic connection.

Benchmarks

We calculate the free energy & pair distribution function.

- Runge-Kutta & numerical spatial integral

Benchmarks

- **Exact result**
- Integral-equation method A widely used method for classical liquids.

Ornstein-Zernike eq. (Schwinger-Dyson eq.)

$$g(\mathbf{x}) = 1 + c(\mathbf{x}) + \rho_0 \int d\mathbf{x}' c(|\mathbf{x}-\mathbf{x}'|)\left[g(\mathbf{x}')-1\right]$$

+

Closure (approximation)

$$g(\boldsymbol{x}) \approx g(\boldsymbol{x}, c(\boldsymbol{x}))$$

$$c(\boldsymbol{x}-\boldsymbol{x}') = \delta(\boldsymbol{x}-\boldsymbol{x}') - \frac{\delta\beta F[\rho]}{\delta\rho(\boldsymbol{x})\delta\rho(\boldsymbol{x}')}$$

Closure

Hypernetted chain (HNC)

$$g(\mathbf{x}) = \exp\left[-\beta v(\mathbf{x}) + g(\mathbf{x}) - c(\mathbf{x}) - 1\right]$$

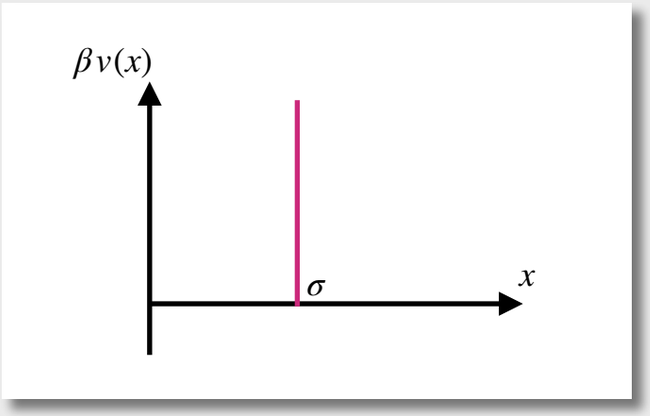
- Based on 2nd order functional Taylor expansion for $\beta F[\rho]$
  & $g^{(2)}(\boldsymbol{x}, \boldsymbol{x}') = \rho_0^{-1}\rho(\boldsymbol{x}\,|\,\boldsymbol{x}')$ (Percus' test particle method)

Percus-Yevick equation (PY)

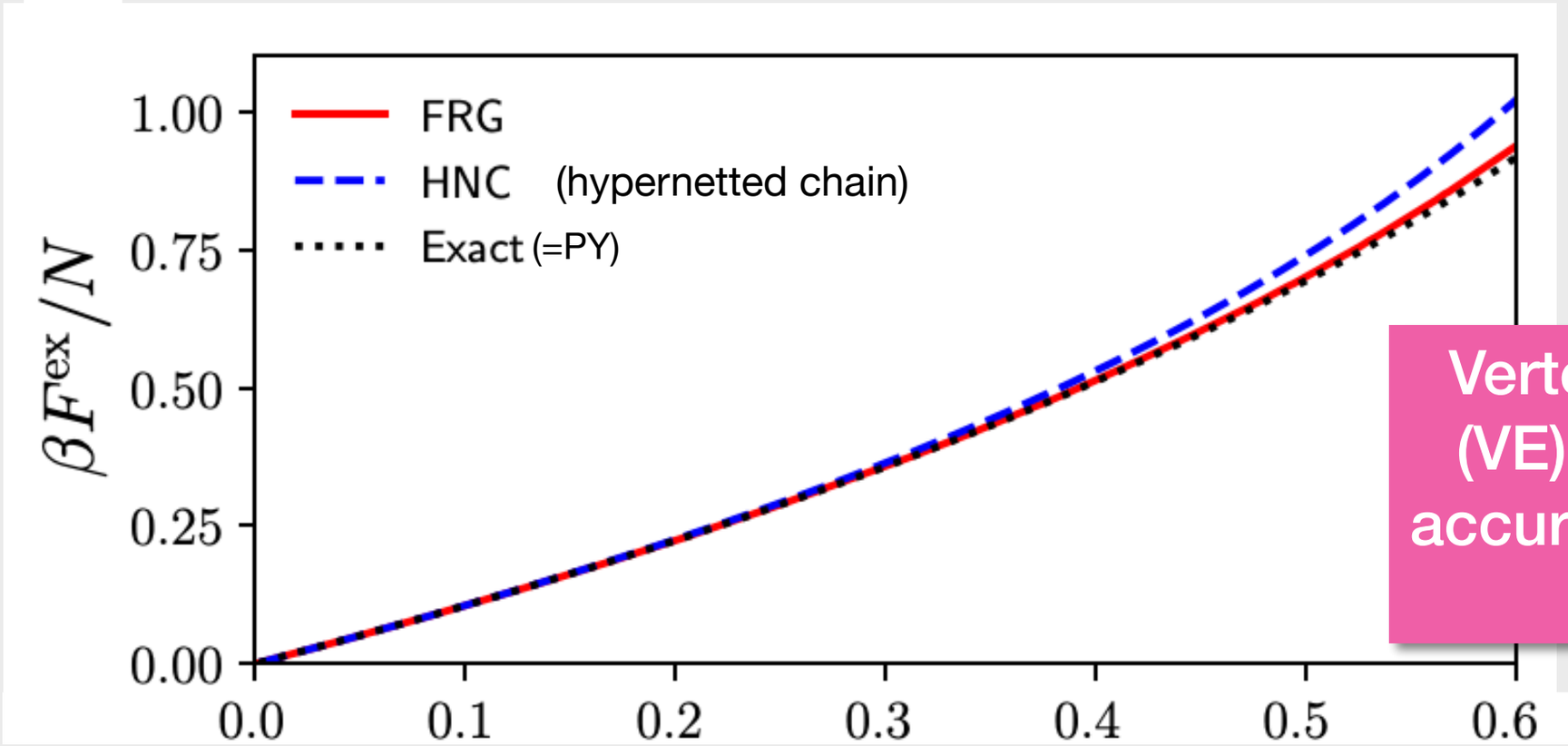
$$g(\mathbf{x}) = \exp\left[-\beta v(\mathbf{x})\right]\left[g(\mathbf{x}) - c(\mathbf{x})\right]$$

- Linearization of HNC
- **Exact for one-dimensional hard rod w/o attractive forces**

Free energy for hard rod ($z_p\sigma = 0$)



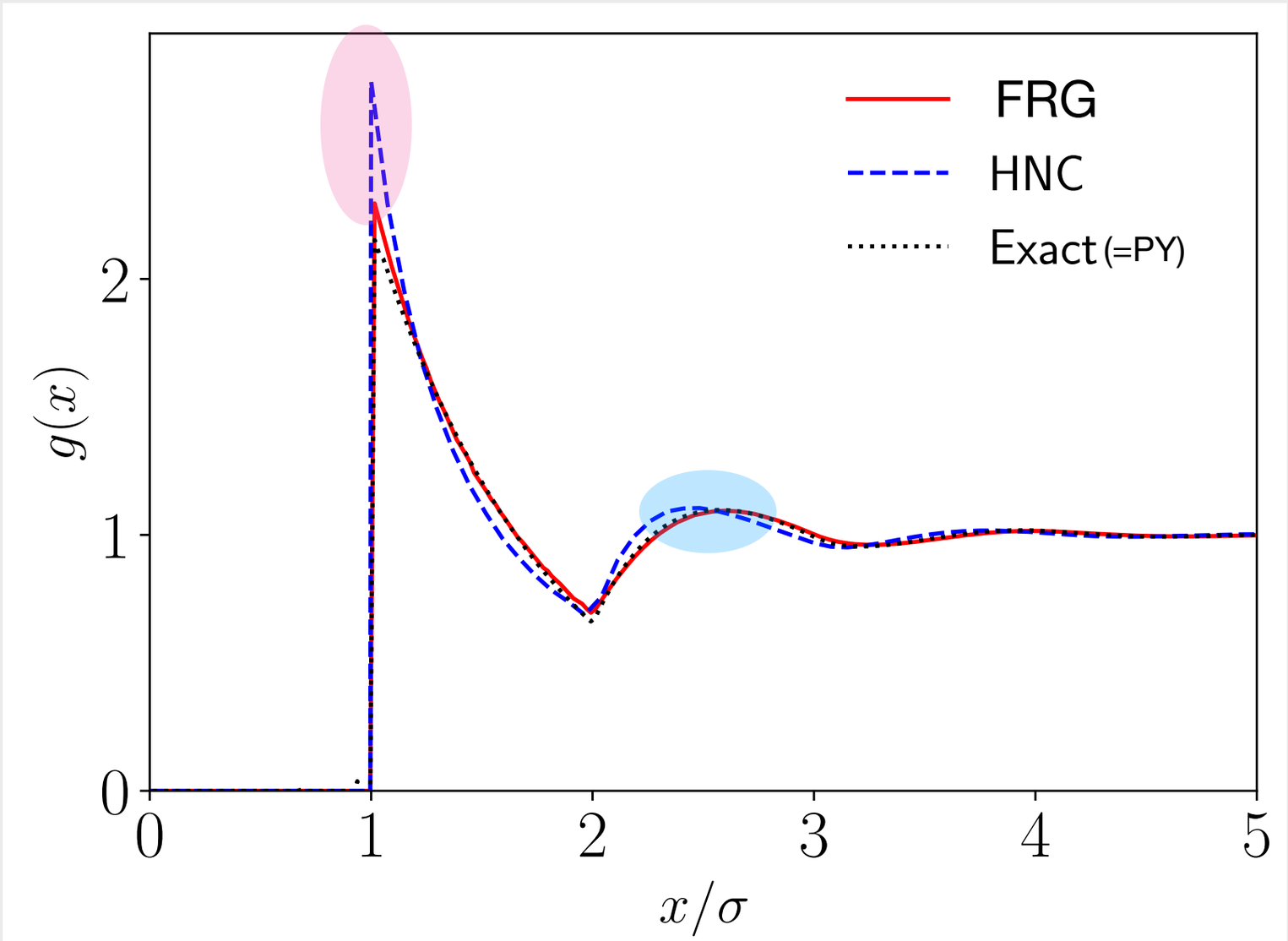
Excess free energy



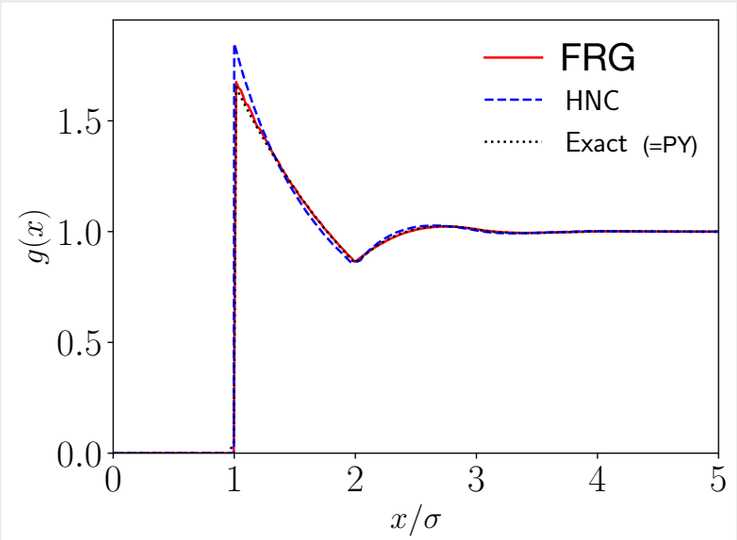
Vertex expansion (VE) shows more accurate result than HNC

Distribution functions for Hard rod ($z_p\sigma = 0$)

Two-particle distribution function $\rho_{\mathrm{hom}}\sigma = 0.55$



c.f.) lower density $\rho_{\mathrm{hom}}\sigma = 0.4$



Height of 1st peak

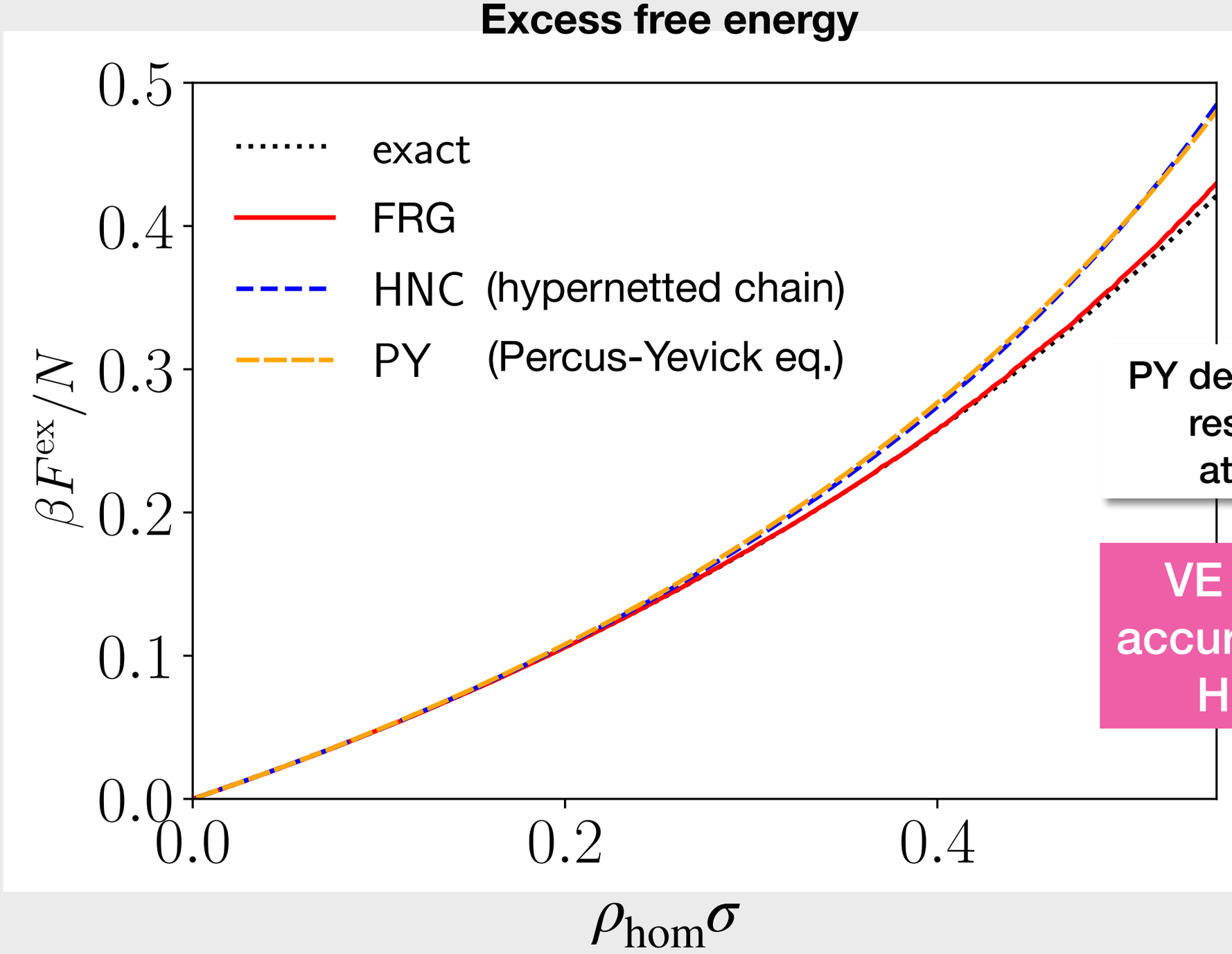
&

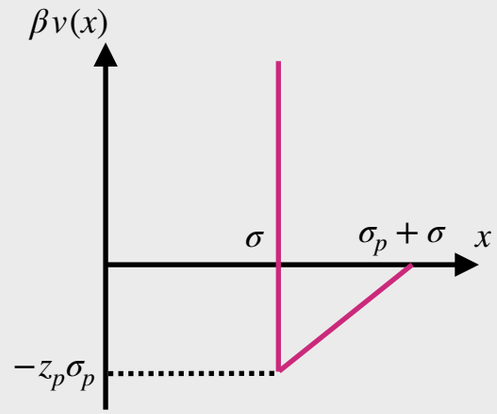
position of 2nd peak

are described by VE more accurately than HNC

KSA + diameter evolution successfully describe hard part!

Free energy with attractive force ($z_p\sigma = 1$, $\sigma_p/\sigma = 0.9$)



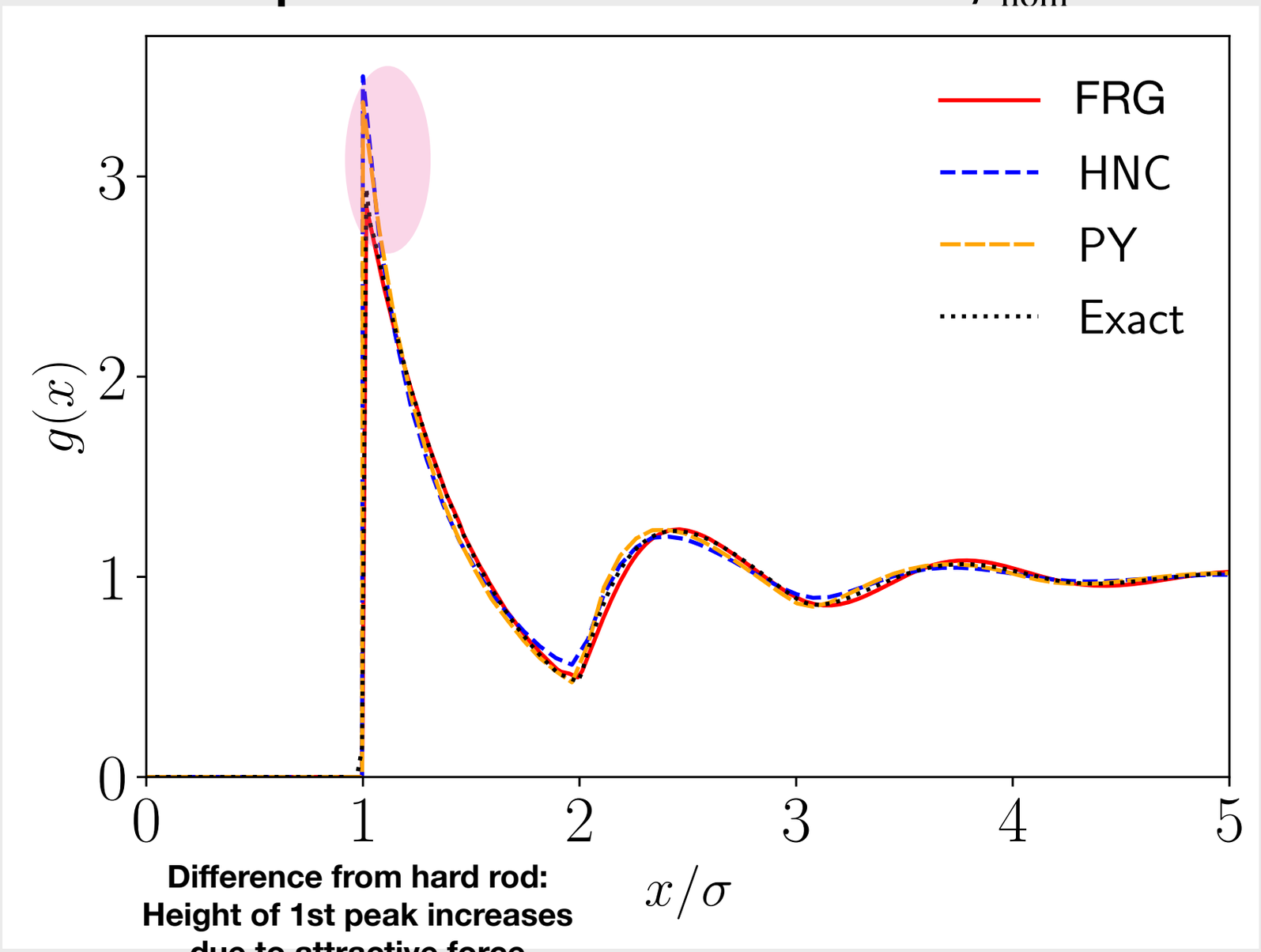


PY deviates from exact result due to the attractive force

VE shows more accurate result than HNC and PY

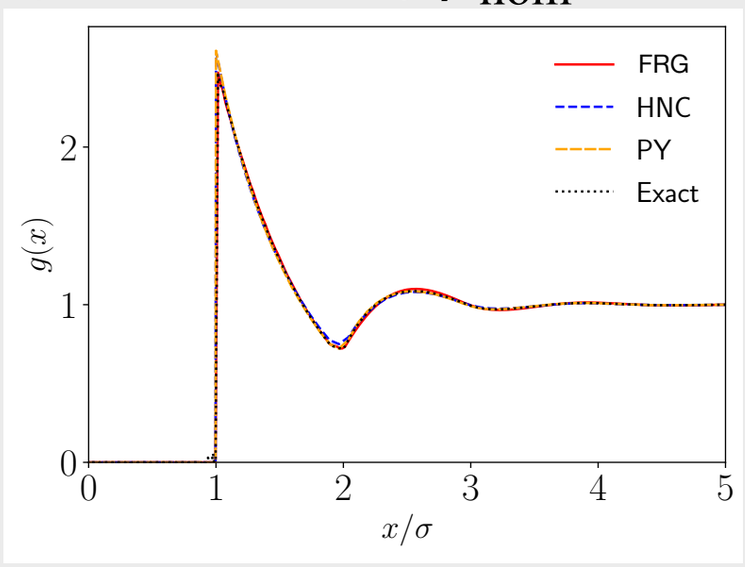
Distribution functions ($z_p\sigma = 1$, $\sigma_p/\sigma = 0.9$)

Two-particle distribution function $\rho_{\mathrm{hom}}\sigma = 0.55$



Difference from hard rod: Height of 1st peak increases due to attractive force

c.f.) lower density $\rho_{\mathrm{hom}}\sigma = 0.4$



Height of 1st peak

Is described by VE more accurately than HNC and PY

VE accurately incorporates not only the hard part but also the attractive part!

Contents

Functional renormalization group (FRG) for classical liquids

- FRG for classical scalar fields
- Path integral for classical DFT
- Modified FRG for classical liquids

Conventional approach: functional Taylor expansion

- Functional Taylor expansion & hierarchical eq.
- Approach for hard-core case
- Demonstration: 1D liquid

New approach: Machine learning

- Physics-informed neural network for solving FRG
- Demonstration: 0D O(N) model

TY, arXiv:2312.16038

Taylor-expansion-based method and its limitations

Let us switch the notation to Wetterich's original one.

$$\partial_k \Gamma_k[\varphi] = \frac{1}{2}\mathrm{Tr}\left[\partial_k R_k \left(\frac{\delta^2 \Gamma_k}{\delta\varphi\delta\varphi}[\varphi] + R_k\right)^{-1}\right]$$

$$\Gamma_k[\varphi] = \tilde{\Gamma}_k[\varphi] - S_{\mathrm{reg},k}[\varphi]$$

Tr: trace w.r.t. coordinates & internal DOF

We have applied the functional Taylor expansion to the flow equation.

But, there are limitations…

❌ **The solution is applicable to limited configurations of $\varphi(x)$**

- $\varphi(x) \approx \varphi_{\mathrm{target}}(x)$
- High cost for the analysis of complicated field configurations, such as crystal.

❌ **Usually, improving the order of the truncation is not easy.**

- Many coordinate (momentum) integrals

Other solution methods?

Another attempt: FDE as high-dim. PDE

FDE: Infinite-dimensional partial differential equation (PDE)

To realize numerical analysis, the input dimensions should be truncated.

Method 1) Introduction of finite spatial lattice

$$\boldsymbol{\varphi} = \{\varphi_{n,\alpha}\}_{n,\alpha},\ n = (n_1, \ldots, n_d),\ 0 \le n_i < L,\ \alpha = 1, \ldots, N_{\mathrm{IDOF}} \text{ (internal DOF)}$$

Total DOF of φ: $N_{\mathrm{DOF}} = L^d N_{\mathrm{IDOF}}$

Method 2) Basis function expansion

See, e.g., *Venturi, PR (2018), Venturi, Dektor, Res. Math. Sci (2021)*

$$\varphi(x,\alpha) \approx \sum_{i=1}^{N_{\mathrm{DOF}}} \varphi_i b_i(x,\alpha) = \boldsymbol{\varphi} \cdot \boldsymbol{b}(x,\alpha)\ (b_i(x,\alpha)\text{: orthonormal basis function})$$

$$\partial_k \Gamma_k[\varphi] = \frac{1}{2}\mathrm{Tr}\left[\partial_k R_k \left(\frac{\delta^2 \Gamma_k}{\delta\varphi\delta\varphi}[\varphi] + R_k\right)^{-1}\right] \quad \Rightarrow \quad \text{($N_{\mathrm{DOF}}+1$)-dim. PDE}$$

$$\partial_k \Gamma_k(\boldsymbol{\varphi}) = \frac{1}{2}\mathrm{tr}\left[\partial_k R_k \left(\frac{\partial^2 \Gamma_k}{\partial\varphi\partial\varphi}(\boldsymbol{\varphi}) + R_k\right)^{-1}\right]$$

The original continuum theory is obtained for increasing N_{DOF}.

Failure of computational grids

How can we solve $(N_{\mathrm{DOF}}+1)$-dim. PDE with large N_{DOF}

Taking computational grids for φ is infeasible.

# of grid points $\sim$ exp(N_{DOF}) if grids are assigned for each $\varphi_{n,\alpha}$

Grid-based methods (finite-element method, Runge-Kutta, …) can not be used.

c.f.) Some people attempt to use tensor decomposition to mitigate computational complexity of representing the solution for linear FDE, such as Hopf eq.

But only the results with $N_{\mathrm{DOF}} \lesssim 6$ has been reported…

Venturi, PR (2018), Venturi, Dektor, Res. Math. Sci (2021)

Grid-free method for high-dim. PDE is required

Physics-informed neural network (PINN)

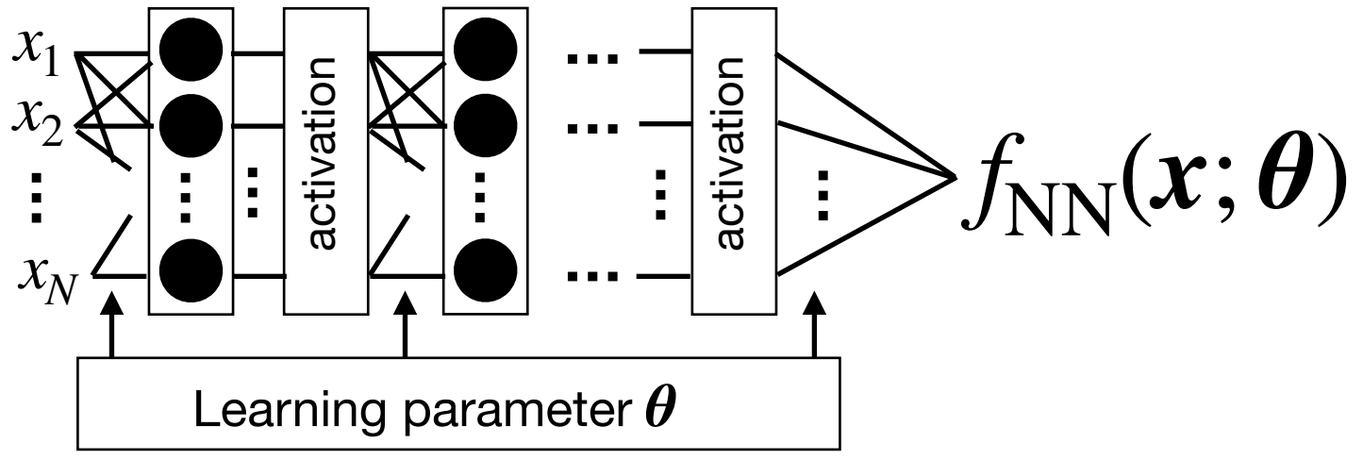
ML-based grid-free approach: PINN

I.E. Lagaris, A. Likas, D. I. Fotiadis, IEEE Transactions on Neural Networks (1998)
M. Raissi, P. Perdikaris, G. E. Karniadakis (2017)
M. Raissi, P. Perdikaris, G.E. Karniadakis, Journal of Computational Physics (2019)

PDE: $I_{\mathrm{DE}}(f(\boldsymbol{x}), \partial_\mu f(\boldsymbol{x}), \cdots) = 0 \qquad \boldsymbol{x} \in \Omega$ (Domain)

Boundary condition: $I_{\mathrm{BC}}(f(\boldsymbol{x}), \partial_\mu f(\boldsymbol{x}), \cdots) = 0 \qquad \boldsymbol{x} \in \partial\Omega$

NN assumption for solution $f(\boldsymbol{x}) \approx f_{\mathrm{NN}}(\boldsymbol{x}; \boldsymbol{\theta})$



Loss function (taking the minimum when PDE & BC are satisfied)

例: $L(\boldsymbol{\theta}) = \underset{\boldsymbol{x}\in\Omega}{\mathrm{average}} \| I_{\mathrm{DE}}(f_{\mathrm{NN}}(\boldsymbol{x};\boldsymbol{\theta}), \partial_\mu f_{\mathrm{NN}}(\boldsymbol{x};\boldsymbol{\theta}), \cdots) \|^2 + \underset{\boldsymbol{x}\in\partial\Omega}{\mathrm{average}} \| I_{\mathrm{BC}}(f_{\mathrm{NN}}(\boldsymbol{x};\boldsymbol{\theta}), \partial_\mu f_{\mathrm{NN}}(\boldsymbol{x};\boldsymbol{\theta}), \cdots) \|^2$

PINN's capability of handling high-dim. PDEs

Advantage of PINN: Grid free → Applicable to high-dim. PDEs

Applications to high-dim. PDEs

- *L. Guo, H. Wu, X. Yu, T. Zhou, Computer Methods in Applied Mechanics and Engineering (2022)*
- *D. He, S. Li, W. Shi, X. Gao, J. Zhang, J. Bian, L. Wang, T.-Y. Liu, In International Conference on Artificial Intelligence and Statistics (2023)*
- *J. Cen, X. Chen, M. Xu, Q. Zou, arXiv:2305.06863.*
- *K. Tang, X. Wan, C. Yang, Journal of Computational Physics, 476 (2023)*
- *Z. Hu, K.Shukla, G. E. Karniadakis, K. Kawaguchi, arXiv:2307.12306*
- *Z. Hu, Z. Yang, Y. Wang, G. E. Karniadakis, K. Kawaguchi, arXiv:2311.15283*
- *Z. Hu, Z. Shi, G. E. Karniadakis, K. Kawaguchi, arXiv:2312.14499*

10^5-dim. PDE

Hamilton-Jacobi-Bellman eq.

$$\partial_t u(\boldsymbol{x},t) + \Delta u(\boldsymbol{x},t) - \frac{1}{d}\sum_{i=1}^{d} |\partial_{\boldsymbol{x}_i} u|^c = -2, \quad \boldsymbol{x} \in \mathbb{R}^d, t \in [0,T]$$

Black-Scholes-Barenblatt eq.

$$u_t = -\frac{1}{2}\sigma^2 \sum_{i=1}^{d} \boldsymbol{x}_i^2 u_{\boldsymbol{x}_i \boldsymbol{x}_i} + r(u - \sum_{i=1}^{d} \boldsymbol{x}_i u_{\boldsymbol{x}_i}),$$

PDE	10^2	10^3	10^4	10^5
HJB-Lin	34min	68min	81min	310min
BSB	31min	57min	118min	41min

Table 1: This table presents the convergence time required by our SDGD for different PDEs. In the HJB-Lin equation, as the dimensionality increases from 100 to 100,000, the dimensionality grows by a factor of 1000, while the time only increases by a factor of ten. This indicates that our method can withstand the curse of dimensionality. In the second BSB equation, surprisingly, the high-dimensional case converges faster than the low-dimensional case, demonstrating the so-called blessing of dimensionality and that our method can harness the blessing of dimensionality.

Moreover, the solution is obtained simultaneously for a **domain** of the inputs rather than one input.

PINN for Wetterich equation

PINN allows us to solve the Wetterich equation with large N_{DOF}!

$$(k, \boldsymbol{\varphi}) \longrightarrow \boxed{\mathrm{NN}(\theta)} \longrightarrow \Gamma_k^{\boldsymbol{\theta}}(\boldsymbol{\varphi})$$

$$L_{\boldsymbol{\theta}} = L_{\boldsymbol{\theta}}^{\mathrm{PDE}} + L_{\boldsymbol{\theta}}^{\mathrm{BC}}$$

$$L_{\boldsymbol{\theta}}^{\mathrm{PDE}} = \mathop{\mathbb{E}}_{\substack{\boldsymbol{\varphi} \sim \mathscr{P}_{\boldsymbol{\varphi}} \\ k \sim \mathscr{P}_k}} \left[\left(\partial_k \Gamma_k^{\boldsymbol{\theta}}(\boldsymbol{\varphi}) - \frac{1}{2} \mathrm{tr} \left[\partial_k R_k \left(\frac{\partial^2 \Gamma_k^{\boldsymbol{\theta}}(\boldsymbol{\varphi})}{\partial \boldsymbol{\varphi} \partial \boldsymbol{\varphi}} + R_k \right)^{-1} \right] \right)^2 \right]$$

$$L_{\boldsymbol{\theta}}^{\mathrm{BC}} = \mathop{\mathbb{E}}_{\boldsymbol{\varphi} \sim \mathscr{P}_{\boldsymbol{\varphi}}} \left[\left(\Gamma_{k_{\mathrm{UV}}}^{\boldsymbol{\theta}}(\boldsymbol{\varphi}) - S(\boldsymbol{\varphi}) \right)^2 \right]$$

- The expectation is evaluated on a finite number of collocation points $(k, \boldsymbol{\varphi})$.
- $\mathscr{P}_{\boldsymbol{\varphi},k}$ is some probability distribution.

PINN for Wetterich equation

PINN allows us to solve the Wetterich equation with large N_{DOF}!

$$(k, \boldsymbol{\varphi}) \xrightarrow{\mathrm{NN}(\theta)} \Gamma_k^{\boldsymbol{\theta}}(\boldsymbol{\varphi})$$

$$L_{\boldsymbol{\theta}} = L_{\boldsymbol{\theta}}^{\mathrm{PDE}} + \cancel{L_{\boldsymbol{\theta}}^{\mathrm{BC}}}$$

$$L_{\boldsymbol{\theta}}^{\mathrm{PDE}} = \mathop{\mathbb{E}}_{\substack{\boldsymbol{\varphi} \sim \mathscr{P}_{\boldsymbol{\varphi}} \\ k \sim \mathscr{P}_k}} \left[\left(\partial_k \Gamma_k^{\boldsymbol{\theta}}(\boldsymbol{\varphi}) - \frac{1}{2} \mathrm{tr} \left[\partial_k R_k \left(\frac{\partial^2 \Gamma_k^{\boldsymbol{\theta}}(\boldsymbol{\varphi})}{\partial \boldsymbol{\varphi} \partial \boldsymbol{\varphi}} + R_k \right)^{-1} \right] \right)^2 \right]$$

$$\cancel{L_{\boldsymbol{\theta}}^{\mathrm{BC}} = \mathop{\mathbb{E}}_{\boldsymbol{\varphi} \sim \mathscr{P}_{\boldsymbol{\varphi}}} \left[\left(\Gamma_{k_{\mathrm{UV}}}^{\boldsymbol{\theta}}(\boldsymbol{\varphi}) - S(\boldsymbol{\varphi}) \right)^2 \right]}$$

The initial condition may be implemented directly in NN.

Hereafter, we omit $L_{\boldsymbol{\theta}}^{\mathrm{BC}}$.

Lagaris, Likas, Fotiadis, IEEE Transactions on Neural Networks (1998)

- The expectation is evaluated on a finite number of collocation points $(k, \boldsymbol{\varphi})$.
- $\mathscr{P}_{\boldsymbol{\varphi},k}$ is some probability distribution.

0-dim. $\mathrm{O}(N)$ model

$$S(\boldsymbol{\varphi}) = \frac{1}{2}m^2\boldsymbol{\varphi}^2 + \frac{g}{4!}(\boldsymbol{\varphi}^2)^2$$

- $N_{\mathrm{DOF}} = N$

The Wetterich equation is an $(N+1)$-dim. PDE.

We can investigate the scalability with N_{DOF} by increasing N.

* We do not reduce Wetterich eq. to 2-dim. PDE with k and $\rho = \varphi^2/2$

- Exact results and results by perturbative, large-N expansions are available.

  E.g., *Keitel, Bartosch, JPA (2012)*

  - The perturbative region is given by $\tilde{g} = Ng/m^4 \ll 1$
  - We calculate $\gamma(k, \boldsymbol{\varphi})$ and self-energy $\sigma = \partial^2_{\boldsymbol{\varphi}}\gamma(k, \boldsymbol{\varphi})$.

- Regulator: $R_k^{\alpha\alpha'} = k_{\mathrm{UV}}^2 e^{-2l},\ l = \ln(k_{\mathrm{UV}}/k)$
- Mass squared: $m^2/k_{\mathrm{UV}}^2 = 0.01$ ($\ll 1$ to validate the UV saddle-point cond.)

Neural network for effective action

We replace the **"nontrivial part"** of the effective action with an NN.

$$\Gamma_k^{\theta}(\varphi) = S(\varphi) + \Delta S_{\text{free}}(k) + \gamma_{\theta}(k, \varphi)$$

Constant shift associated with free part

$$\partial_k \Delta S_{\text{free}}(k) = \frac{1}{2}\text{tr}\left[\partial_k R_k \left(\frac{\partial^2 S_{\text{free}}(\varphi)}{\partial\varphi\partial\varphi} + R_k\right)^{-1}\right]$$

Interaction-induced RG part ("nontrivial part")

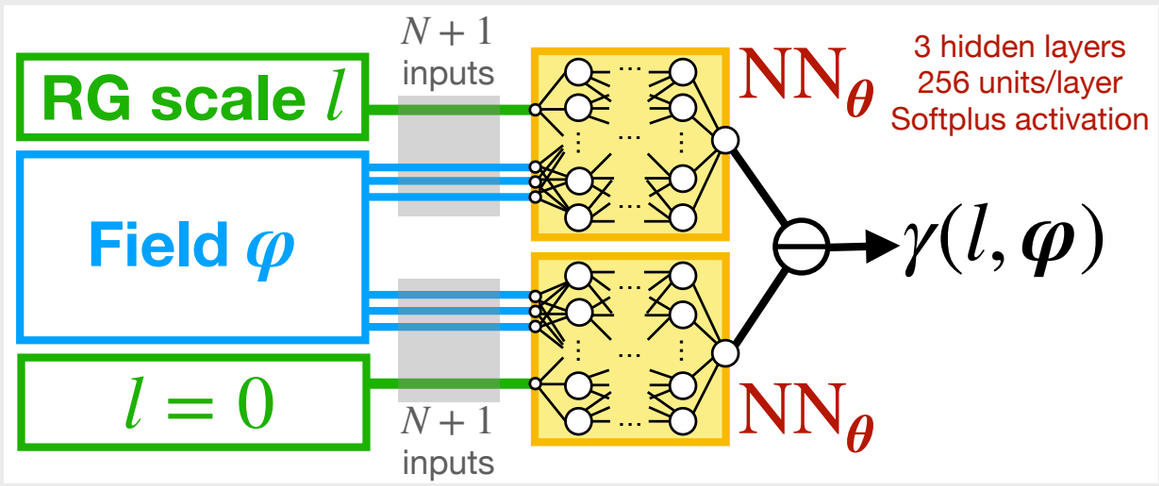
This satisfies $\gamma_{\theta}(k_{\text{UV}}, \varphi) = 0$

NN ansatz subject to $\gamma_{\theta}(k_{\text{UV}}, \varphi) = 0$ (hard implementation of the init. cond.)

$$\gamma_{\theta}(k, \varphi) \approx \text{NN}_{\theta}(l, \varphi) - \text{NN}_{\theta}(0, \varphi) \qquad l = \ln(k_{\text{UV}}/k)$$

$\text{NN}_{\theta}(l, \varphi)$

- 3 hidden layers
- 256 units/layer
- Differentiable softplus activation



Pretraining

$$L_{\boldsymbol{\theta}} = \mathop{\mathbb{E}}_{\substack{\boldsymbol{\varphi} \sim \mathscr{P}_{\boldsymbol{\varphi}} \\ l \sim \mathscr{P}_l}} \left[\left(\partial_l \Gamma^{\boldsymbol{\theta}}_l(\boldsymbol{\varphi}) - \frac{1}{2} \mathrm{tr} \left[\partial_l R_l \left(\frac{\partial^2 \Gamma^{\boldsymbol{\theta}}_l(\boldsymbol{\varphi})}{\partial \boldsymbol{\varphi} \partial \boldsymbol{\varphi}} + R_l \right)^{-1} \right] \right)^2 \right]$$

The matrix must be regular during the training.

In our experience, this is frequently broken with randomly chosen $\boldsymbol{\theta}$.

Pretraining with some approximate analytic results remedies this problem

We use 1st-order perturbative result:

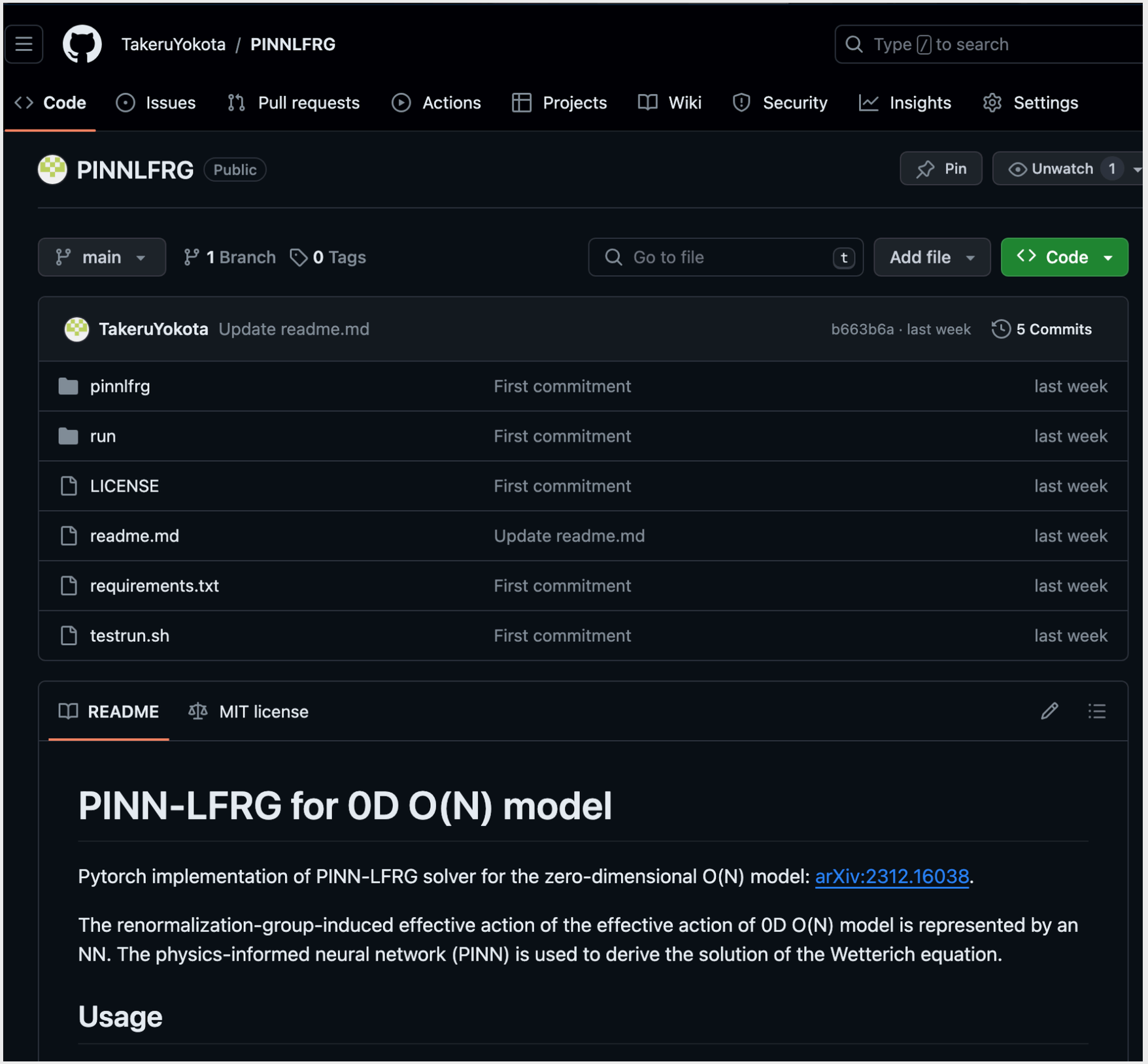
$$L^{\mathrm{pre}}_{\boldsymbol{\theta}} = \mathop{\mathbb{E}}_{\substack{\boldsymbol{\varphi} \sim \mathscr{P}_{\boldsymbol{\varphi}} \\ l \sim \mathscr{P}_l}} \left[\left(\gamma(l, \boldsymbol{\varphi}; \boldsymbol{\theta}) - \gamma^{\mathrm{1pt}}(l, \boldsymbol{\varphi}) \right)^2 \right]$$

Other details of numerical procedure

$$L_{\boldsymbol{\theta}} = \mathop{\mathbb{E}}_{\substack{\boldsymbol{\varphi} \sim \mathscr{P}_{\boldsymbol{\varphi}} \\ l \sim \mathscr{P}_l}} \left[\left(\partial_l \Gamma_l^{\boldsymbol{\theta}}(\boldsymbol{\varphi}) - \frac{1}{2} \mathrm{tr} \left[\partial_l R_l \left(\frac{\partial^2 \Gamma_l^{\boldsymbol{\theta}}(\boldsymbol{\varphi})}{\partial \boldsymbol{\varphi} \partial \boldsymbol{\varphi}} + R_l \right)^{-1} \right] \right)^2 \right]$$

- $\mathscr{P}_l$: uniform distribution in $[0, l_{\mathrm{end}}]$ with $l_{\mathrm{end}} = 5$
- $\mathscr{P}_{\boldsymbol{\varphi}}$: $\|\boldsymbol{\varphi}\|$ is sampled following $N(0, N/m^2)$ (w/o sign)
  $\hat{\boldsymbol{n}} = \boldsymbol{\varphi}/\|\boldsymbol{\varphi}\|$ is uniformly sampled
  * Other choices such as $N(\mathbf{0}, m^{-2}\mathbf{1})$ fail to sample the neighborhoods of $\boldsymbol{\varphi} = \mathbf{0}$ due to curse of dimensionality
- 500 collocation points are used to evaluate the expectation.
- Adam optimizer
- Pytorch
- The matrix inverse is evaluated by direct method
  - This may not be efficient but is easy to implement (torch.linalg.inv)
  - More efficient way: Hutchinson trace estimator (future work) *Hutchinson, Simul. Comp. (1990)*

Code: https://github.com/TakeruYokota/PINNLFRG



Computational time & convergence

We conduct computations for all the combinations of

$$N = 1,10,100 \text{ and } \tilde{g} = 0.1,1,10 \qquad \tilde{g} = Ng/m^4$$

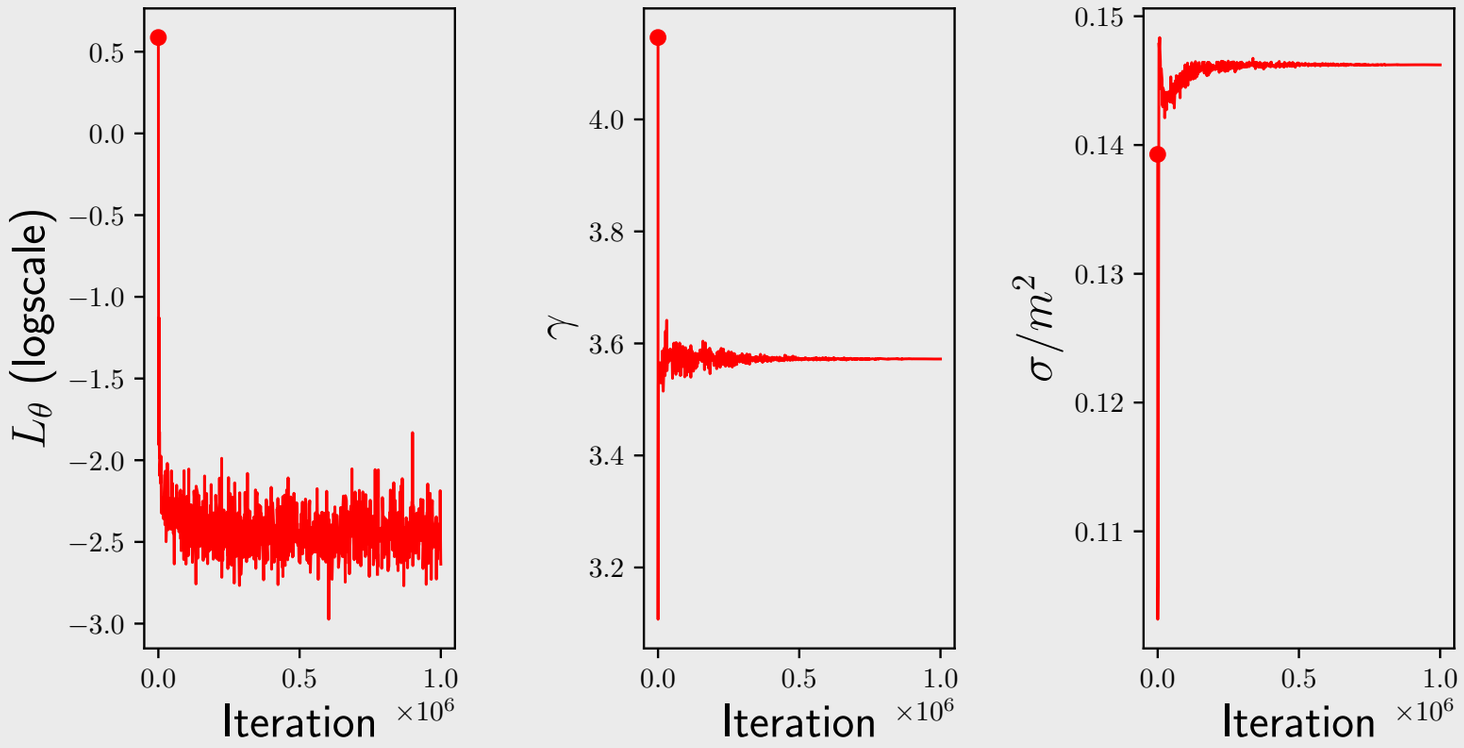
- Learning rate (Wetterich): 10^{-4} with exponential decay factor 0.99999
- Learning rate (pretraining): 10^{-3}

Comp. time on NVIDIA A100 GPU

N	1	10	100
Pretraining	4m	4m	6m
Wetterich	6h	7h	11h

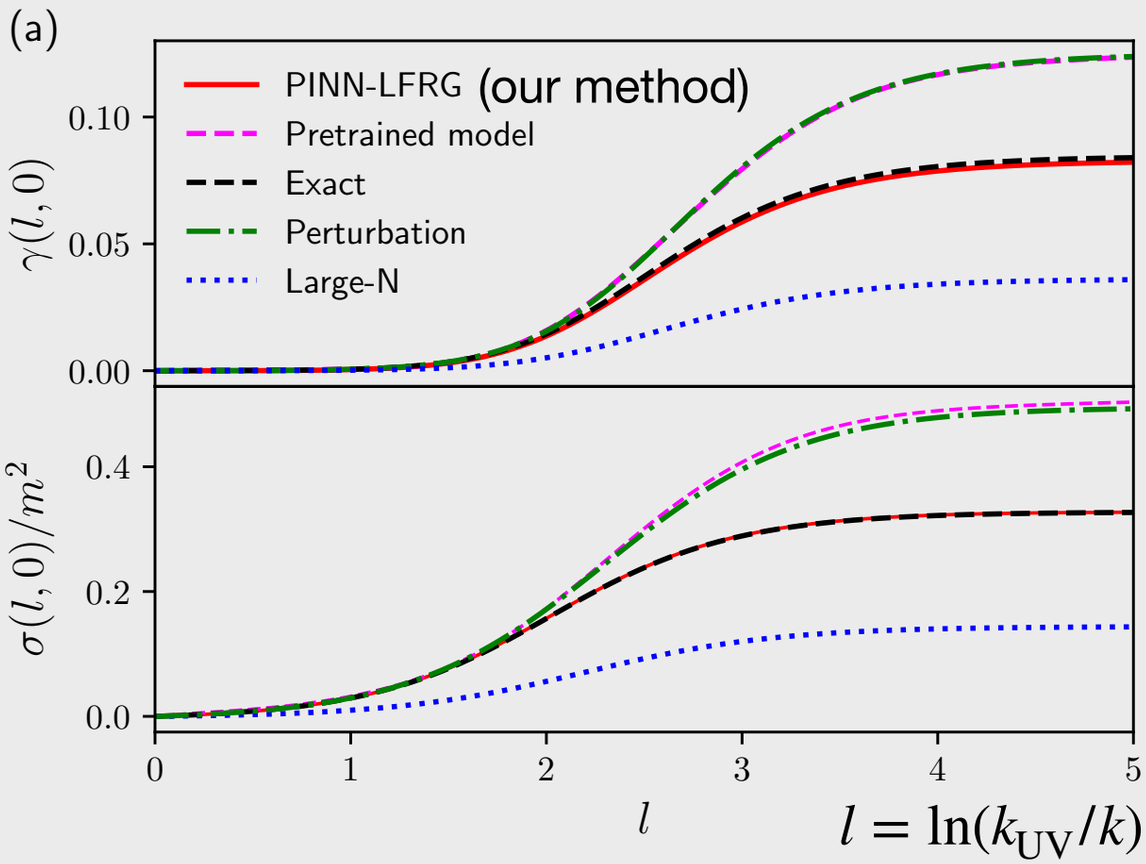
Learning curve & histories of physical quantities

($N = 100$ & $\tilde{g} = 1$ case)

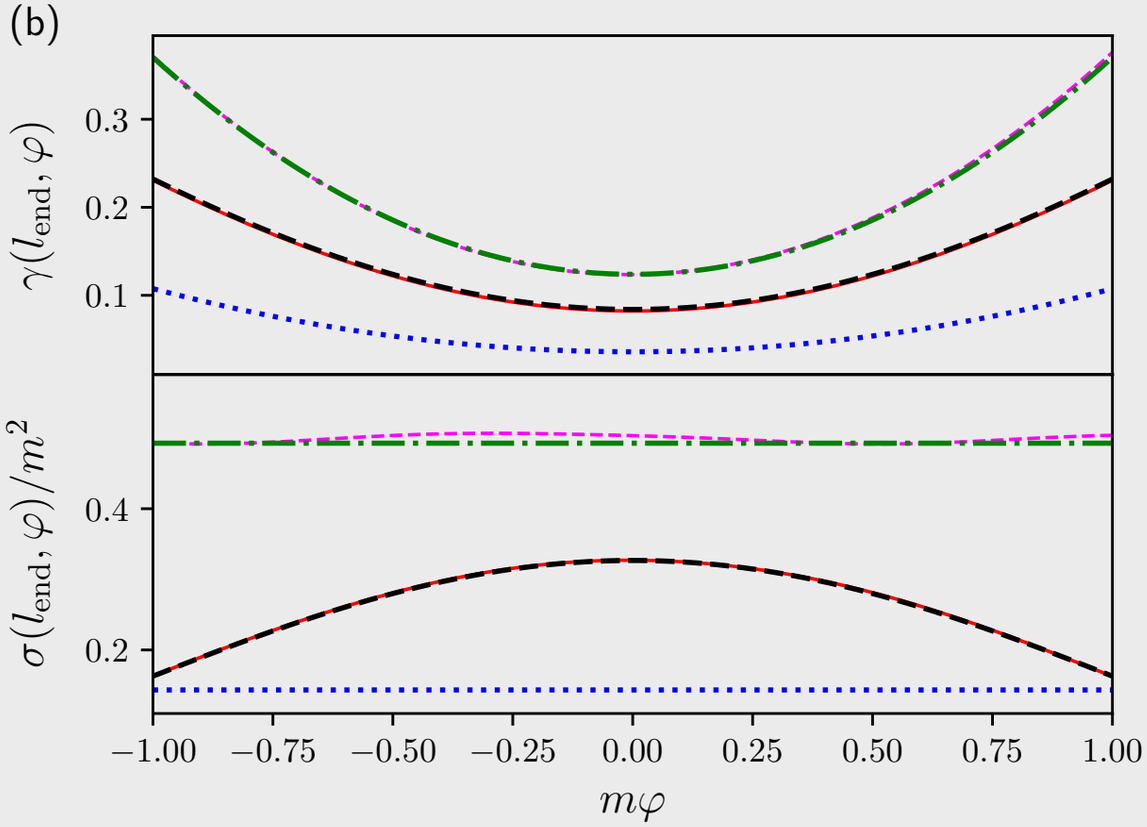


$N = 1$ and $\tilde{g} = 1$ case

RG scale l dependence at $\varphi = 0$



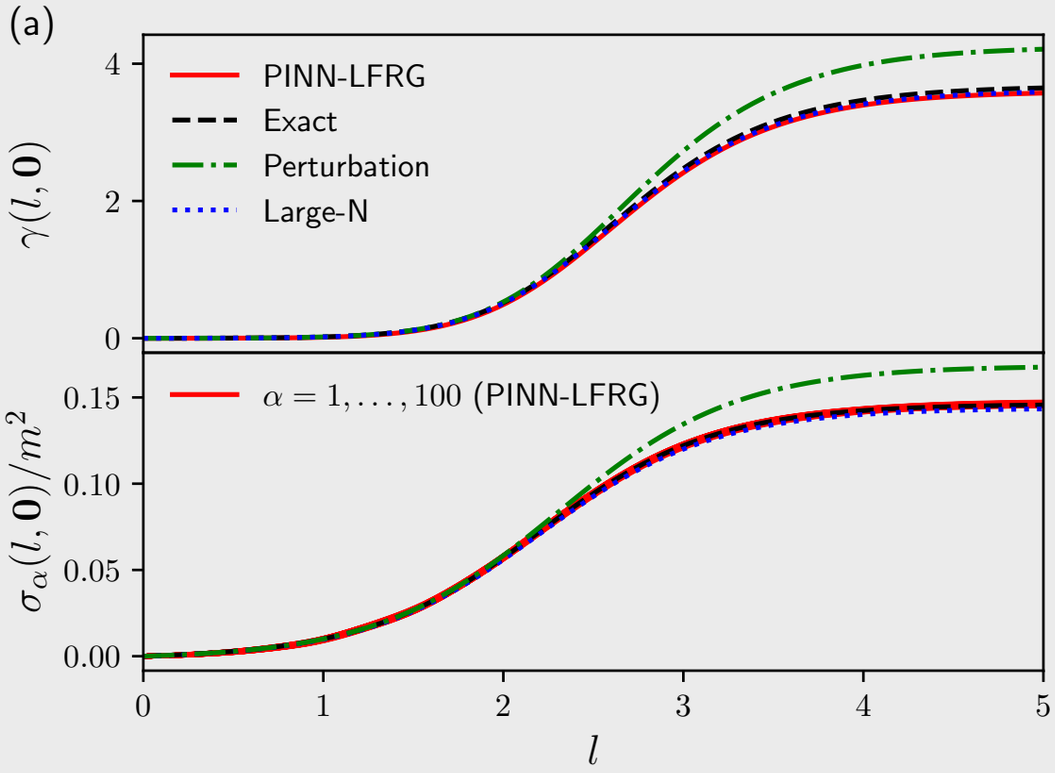
φ dependence at $l = l_{\mathrm{end}}$



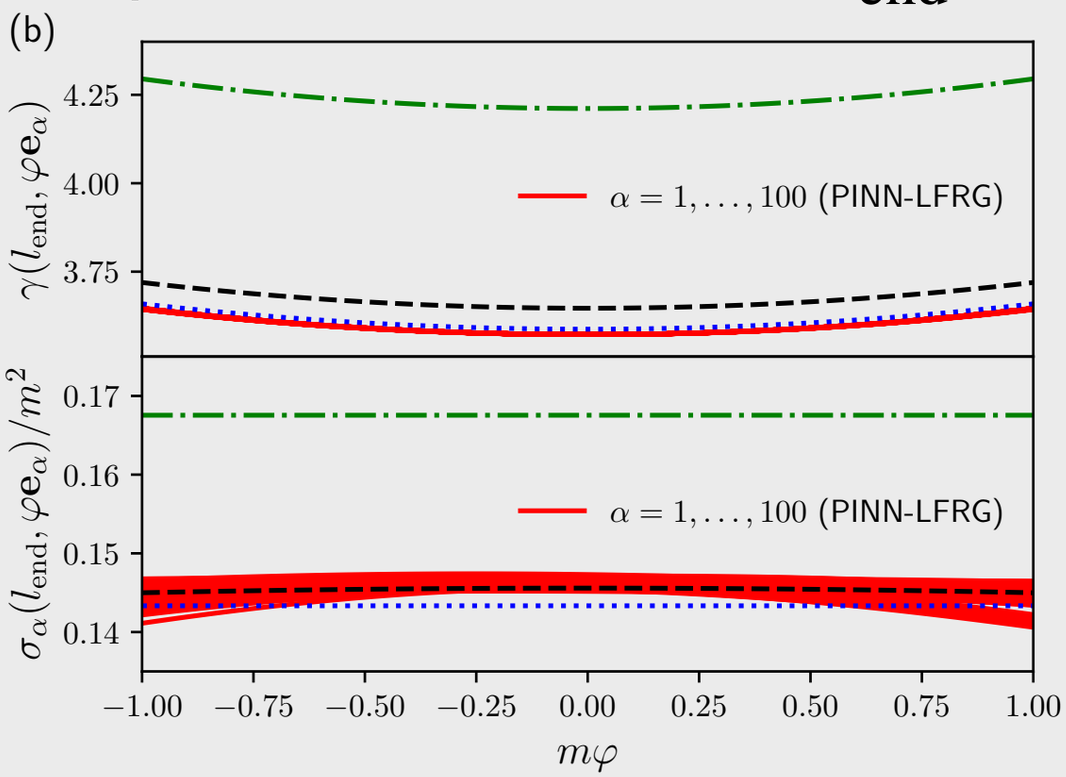
- $\gamma(l, \varphi)$ is simultaneously obtained for a domain of φ in our method (PINN-LFRG).
- PINN-LFRG shows accurate results compared to 1st-order perturbation & leading-order large-N expansion

$N = 100$ and $\tilde{g} = 1$ case

RG scale l dependence at $\varphi = 0$



φ dependence at $l = l_{\text{end}}$



Reduced self-energy $\sigma_\alpha(l, \boldsymbol{\varphi}) = \dfrac{\partial^2}{\partial \varphi_\alpha^2} \gamma(l, \boldsymbol{\varphi})$

- Except for $\gamma(l, \mathbf{0})$, PINN-LFRG results are given by $N = 100$ lines corresponding to the N-direction in $\boldsymbol{\varphi}$ space.
- PINN-LFRG shows comparable results with large-N expansion, which should be accurate for $N = 100$.
- $\mathrm{O}(N)$ symmetry is reproduced in PINN-LFRG.

Results at $\varphi = 0$ for different N and $\tilde{g}$

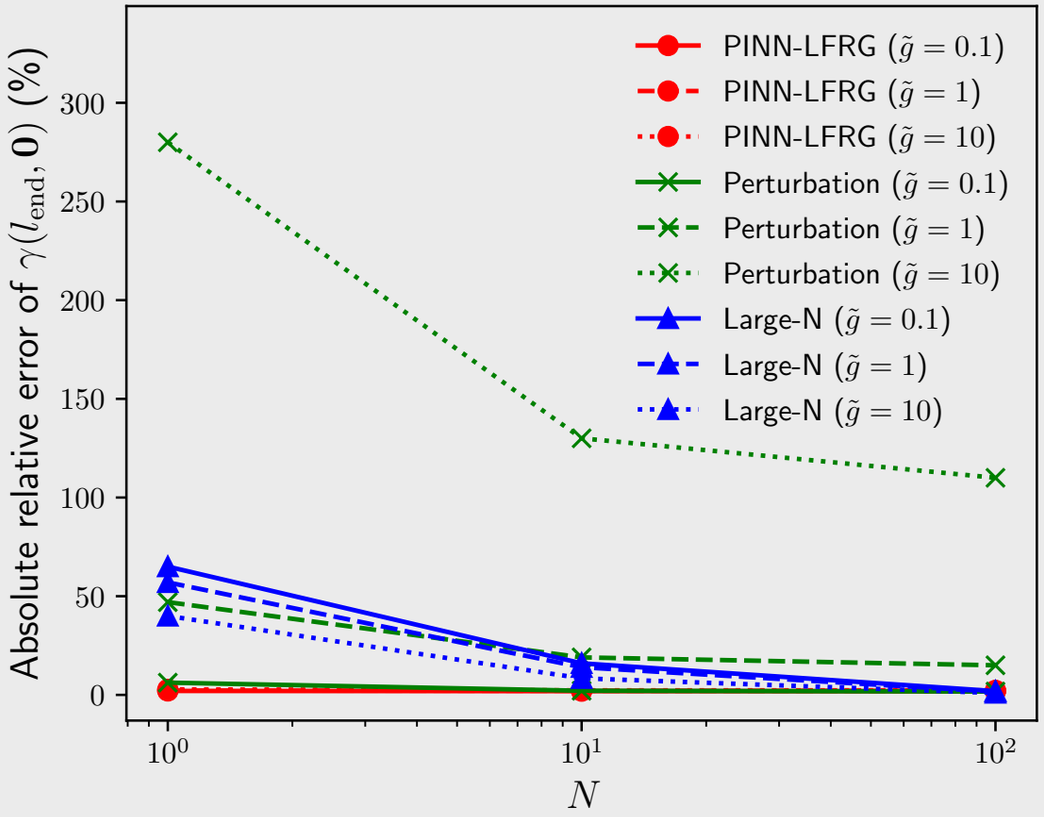
Relative error compared to exact results (minus indicates underestimation)

N		1			10			100		
$\tilde{g}$		0.1	1	10	0.1	1	10	0.1	1	10
Perturb. (%)	γ	6.2	47	275	2.1	19	129	1.7	15	110
	σ	7.6	51	228	2.3	19	109	1.7	15	92
Large-N (%)	γ	-65	-57	-40	-16	-14	-8.4	-1.9	-1.6	-0.95
	σ	-65	-56	-42	-16	-13	-8.2	-1.9	-1.5	-0.89
PINN-LFRG (%)	γ	-2.0	-2.2	-2.8	-1.9	-2.1	-2.3	-1.9	-2.0	-2.3
	σ	-0.17	0.12	0.76	0.16	0.46	0.42	-0.011	0.44	0.50
	$\Delta\sigma$	0	0	0	0.27	0.18	0.24	0.38	0.29	0.26

$$\gamma = \gamma(l_{\text{end}}, \mathbf{0})$$

$$\sigma = \frac{1}{N}\sum_{\alpha=1}^{N} \sigma_\alpha(l_{\text{end}}, \mathbf{0})$$

$$\Delta\sigma = \sqrt{\frac{1}{N}\sum_{\alpha=1}^{N} \left(\sigma_\alpha(l_{\text{end}}, \mathbf{0}) - \sigma\right)^2}$$



- For all $\tilde{g}$ and N, the errors of PINN-LFRG are within 3% for γ and 1% for σ.

- Even when $1/N$ and $\tilde{g}$ are not small, PINN-LFRG shows accurate results.

NNs are promising approximations independent of the existence of a small parameter.

Summary

I showed
a **functional renormalization group (FRG) formalism for classical liquids.**

||

FRG for classical DFT

- FRG for classical scalar fields
- Action for classical DFT
- Modified FRG for classical liquids

<u>Two approaches for solving FRG</u>

Functional Taylor expansion

TY, Haruyama, Sugino, PRE (2021)

- Solution method with no need for a hard-core reference
- Demonstration in 1D liquids.

Physics-informed neural network solver

TY, arXiv:2312.16038

- FRG eq: a kind of high-dimensional PDE
- PINN: a cutting-edge solver for high-D PDE
- Demonstration in the 0D scalar model

<u>Outlook</u>

- 3D liquids
- PINN for classical DFT

Backup

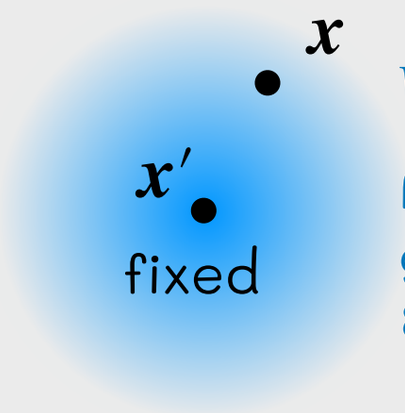
Kirkwood superposition approximation (KSA)

Free particle's distribution in a potential $U(\boldsymbol{x})$

$$\rho(\boldsymbol{x}) = \rho_0 e^{-\beta U(\boldsymbol{x})}$$

Pair distribution
= particle distribution around a fixed particle

$$g^{(2)}(\boldsymbol{x}, \boldsymbol{x}') = \rho_0^{-1}\rho(\boldsymbol{x}\,|\,\boldsymbol{x}') = e^{-\beta W(\boldsymbol{x}-\boldsymbol{x}')}$$



$W(\boldsymbol{x}-\boldsymbol{x}')$

Mean force potential (MFP) generated by the fixed & surrounding particles

N-particle distribution
= (particle distribution around $N-1$ fixed particle) x ($N-1$)-particle distribution

$$g^{(3)}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) = \rho_0^{-1}\rho(\boldsymbol{x}_1\,|\,\boldsymbol{x}_2, \boldsymbol{x}_3)g^{(2)}(\boldsymbol{x}_2, \boldsymbol{x}_3) = e^{-\beta W(\boldsymbol{x}_1|\boldsymbol{x}_2,\boldsymbol{x}_3)}e^{-\beta W(\boldsymbol{x}_2-\boldsymbol{x}_3)}$$

KSA = Superposition of MFPs

$$W(\boldsymbol{x}_1\,|\,\boldsymbol{x}_2, \boldsymbol{x}_3) \approx W(\boldsymbol{x}_1\,|\,\boldsymbol{x}_2) + W(\boldsymbol{x}_1\,|\,\boldsymbol{x}_3)$$

Removal of mean many-body forces

$$\Rightarrow\ g^{(3)}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) \approx g^{(2)}(\boldsymbol{x}_1, \boldsymbol{x}_2)g^{(2)}(\boldsymbol{x}_2, \boldsymbol{x}_3)g^{(2)}(\boldsymbol{x}_3, \boldsymbol{x}_1)$$

4-body $\quad W(\boldsymbol{x}_1\,|\,\boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) \approx W(\boldsymbol{x}_1\,|\,\boldsymbol{x}_2) + W(\boldsymbol{x}_1\,|\,\boldsymbol{x}_3) + W(\boldsymbol{x}_1\,|\,\boldsymbol{x}_4)$

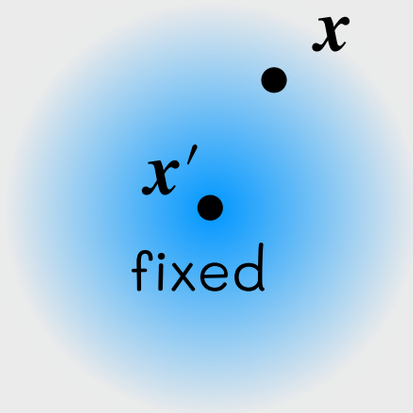
$$g^{(4)}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) = \rho_0^{-1}\rho(\boldsymbol{x}_1\,|\,\boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4)g^{(3)}(\boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) = e^{-\beta W(\boldsymbol{x}_1|\boldsymbol{x}_2,\boldsymbol{x}_3,\boldsymbol{x}_4)}g^{(3)}(\boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4)$$
$$= g^{(2)}(\boldsymbol{x}_1, \boldsymbol{x}_2)g^{(2)}(\boldsymbol{x}_1, \boldsymbol{x}_3)g^{(2)}(\boldsymbol{x}_1, \boldsymbol{x}_4)g^{(2)}(\boldsymbol{x}_2, \boldsymbol{x}_3)g^{(2)}(\boldsymbol{x}_1, \boldsymbol{x}_4)g^{(2)}(\boldsymbol{x}_3, \boldsymbol{x}_4)$$

Hypernetted chain

Percus's test particle method

$$n(\boldsymbol{x}\,|\,\boldsymbol{x}') = \rho_0 g^{(2)}(\boldsymbol{x}, \boldsymbol{x}')$$

i.e., $g^{(2)}(\,\cdot\,, \boldsymbol{x}')$ is obtained as the density in the presence of the external field $U_{\boldsymbol{x}'}(\boldsymbol{x}) = v(\boldsymbol{x} - \boldsymbol{x}')$



$$U_{\boldsymbol{x}'}(\boldsymbol{x}) = v(\boldsymbol{x} - \boldsymbol{x}')$$

Variational eq.

$$\frac{\delta \beta F}{\delta \rho(\boldsymbol{x})}[\rho_0 g^{(2)}(\,\cdot\,, \boldsymbol{x}')] = \beta(\mu - v(\boldsymbol{x} - \boldsymbol{x}'))$$

Functional Taylor expansion of the excess part around $\rho(\boldsymbol{x}) = \rho_0$

$$\beta F[\rho] \approx \int_{\boldsymbol{x}} \rho(\boldsymbol{x})\left(\ln \rho(\boldsymbol{x})/\rho_0 - 1\right) + \beta F_{\mathrm{ex}}[\rho_0] + \int_{\boldsymbol{x}} \beta\mu(\rho(\boldsymbol{x}) - \rho_0) - \frac{1}{2}\int_{\boldsymbol{x},\boldsymbol{x}'} c(\boldsymbol{x} - \boldsymbol{x}')(\rho(\boldsymbol{x}) - \rho_0)(\rho(\boldsymbol{x}') - \rho_0)$$

Closure

$$\beta F[\rho] = \beta F_{\mathrm{free}}[\rho] + \beta F_{\mathrm{ex}}[\rho]$$

From variational eq. & Taylor exp.,

$$g^{(2)}(\boldsymbol{x}) = \exp\left[-\beta v(\boldsymbol{x}) + \rho_0 \int_{\boldsymbol{x}''} c(\boldsymbol{x} - \boldsymbol{x}'')(g^{(2)}(\boldsymbol{x}, \boldsymbol{x}'') - 1)\right] \xrightarrow{\text{OZ eq.}} g^{(2)}(\boldsymbol{x}) = \exp\left[-\beta v(\boldsymbol{x}) + g^{(2)}(\boldsymbol{x}) - c(\boldsymbol{x}) - 1\right]$$

Machine learning for partial differential equations (PDE)

Recently, there have been many applications of machine learning to PDEs

- **Physics-informed neural network (PINN)**

  Psichogios, Ungar, AIChE (1992)
  Lagaris, Likas, Fotiadis, IEEE Transactions on Neural Networks (1998)
  Raissi, Perdikaris, Karniadakis (2017)
  Raissi, Perdikaris, Karniadakis, Journal of Computational Physics (2019),…

Applicable to various types of PDEs

- **Rayleigh-Ritz variational method**

  E, Yu, Comm. Math. Stat. (2018)
  Khoo, Lu, Ying, Res. Math. Sci. (2018),…

- **Backward stochastic differential equation**

  E, Han, Jentzen, Comm. Math. Stat. (2017)
  Han, Jetzen, E (2018)
  Rassi (2018)
  Beck, E, Jentzen, J. Nonlinear Science (2019),…

⋮