TRIUMF

Extrapolating Nuclear Many-Body Calculations with Constrained Gaussian Processes

Peter Gysbers
M. Gennari, W. Fedorko, P. Navrátil

TRIUMF Ab Initio Workshop 2020

Discovery, accelerated

Context: *Ab Initio* Nuclear Theory

Goal: solve the nuclear eigenvalue problem

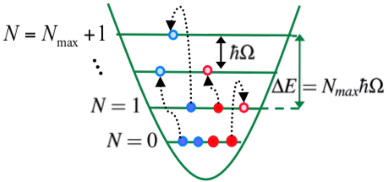
$$H\left|\Psi_k\right\rangle = E_k\left|\Psi_k\right\rangle \text{, where } H = \sum_i^A T_i + \sum_{i<j} V_{ij} + \sum_{i<j<f} V_{ijf} + \cdots$$

with nucleons as the degrees of freedom

The No-core Shell Model

Expand in anti-symmetrized products of harmonic oscillator single-particle states

$$\left|\Psi_k\right\rangle = \sum_{N=0}^{N_{max}} \sum_j c_{Nj}^k \left|\Phi_{Nj}\right\rangle$$

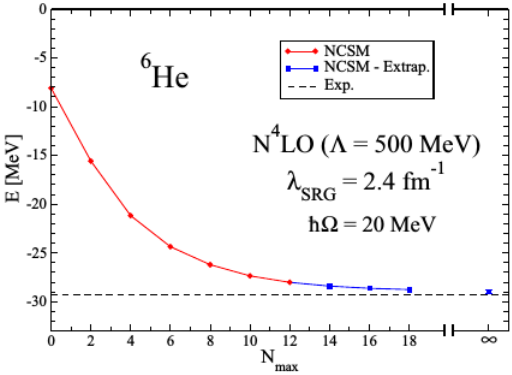


Calculations should converge to the exact value as $N_{max} \to \infty$

Motivation

- Computational complexity grows exponentially with basis size parameter N_{max}
- The functional form of convergence curve is not known
- Ad hoc extrapolation:

$$E = E_\infty + \alpha \exp(-bN_{max})$$



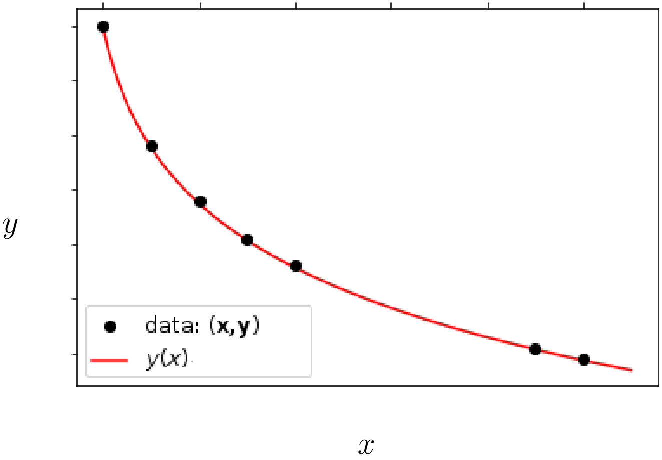
Goal: predict value at $N_{max} \to \infty$ with a meaningful error bar

Problem Statement

Given some data $\mathbf{y} = y(\mathbf{x})$, find the underlying function $y(x)$, i.e. predict $\mathbf{y}^* = y(\mathbf{x}^*)$

$y \sim E$

$x \sim N_{max}$



Gaussian Processes

Key Assumption (Prior):

$\mathbf{y}$ and $\mathbf{y}^*$ are drawn from a joint Gaussian distribution

$$p\left(\begin{bmatrix}\mathbf{y}\\ \mathbf{y}^*\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mu\\ \mu_*\end{bmatrix}, \begin{bmatrix}C & C_*\\ C_*^T & C_{**}\end{bmatrix}\right)$$

$$\begin{aligned} C &= C[\mathbf{y},\mathbf{y}] \\ &= r(\mathbf{x},\mathbf{x}) \\ C_* &= C[\mathbf{y},\mathbf{y}^*] \\ &= r(\mathbf{x},\mathbf{x}^*) \\ C_{**} &= r(\mathbf{x}^*,\mathbf{x}^*) \end{aligned}$$

Make predictions by conditioning on data:

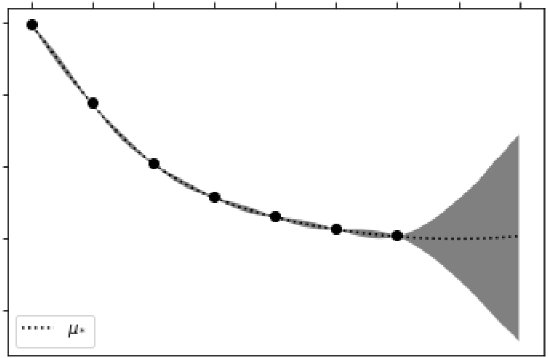
$$p(\mathbf{y}^*|\mathbf{y}) = \mathcal{N}\left(\mu_*, \Sigma_*\right)$$

where

$$\begin{aligned} \mu_* &= C_*^T C^{-1}\mathbf{y} \\ \Sigma_* &= C_{**} - C_*^T C^{-1} C_* \end{aligned}$$

Gaussian Processes give a distribution of predictions (within error band)

Problem: Error bars blow up outside of data!



Idea: Use information about derivatives

Using Derivatives

- The derivative of a Gaussian process is a Gaussian process
- i.e. ${\mathbf{y}'}_i \equiv \frac{\mathrm{d}y}{\mathrm{d}x}\big|_{x=x'_i}$ is also jointly Gaussian distributed (as is $\mathbf{y}''$)

$$p\left(\begin{bmatrix}\mathbf{y}^*\\ \mathbf{y}'\\ \mathbf{y}''\end{bmatrix}\Bigg|\mathbf{y}\right) = \mathcal{N}(\nu, \Sigma)$$

(ν and Σ are more complicated (see Extra Slides))

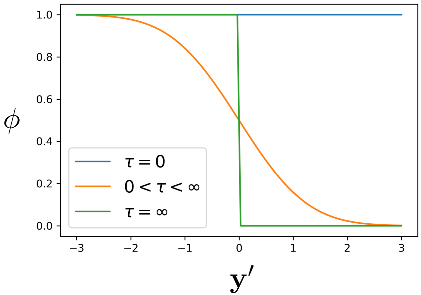
We want the posterior distribution:

$$p\left(\begin{bmatrix}\mathbf{y}^*\\ \mathbf{y}'\\ \mathbf{y}''\end{bmatrix}\Bigg|\mathbf{y}\right) \times p(\mathbf{y}', \mathbf{y}'') = \mathcal{N}(\nu, \Sigma) \times m(\mathbf{y}') \times n(\mathbf{y}'')$$

Use SMC!

Sequential Monte-Carlo / Particle Filter

Draw N samples ("particles": $\begin{bmatrix} \mathbf{y}^* \\ \mathbf{y}' \\ \mathbf{y}'' \end{bmatrix}$) from a GP



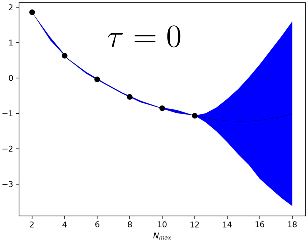
for τ_1, τ_2 from 0 to ∞:

- for each particle:
  - propose new $\begin{bmatrix} \mathbf{y}^* \\ \mathbf{y}' \\ \mathbf{y}'' \end{bmatrix}$ values "nearby" old values
  - accept or reject according to

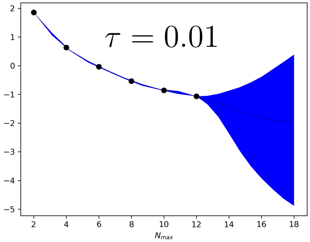
$$p\left(\begin{bmatrix} \mathbf{y}^* \\ \mathbf{y}' \\ \mathbf{y}'' \end{bmatrix} \middle| \mathbf{y}\right) \times p(\mathbf{y}', \mathbf{y}'' | \tau_1, \tau_2) = \mathcal{N}(\nu, \Sigma) \times \phi(-\tau_1 \mathbf{y}') \times \phi(\tau_2 \mathbf{y}'')$$

- resample: throw away "bad" particles and keep multiple copies of "good" particles (weighted by constraints)

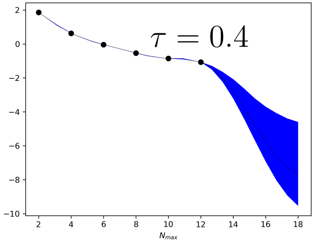
Results: He^4



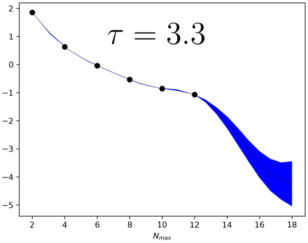
$\tau = 0$
2 1 0 −1 −2 −3
2 4 6 8 10 12 14 16 18
N_{max}



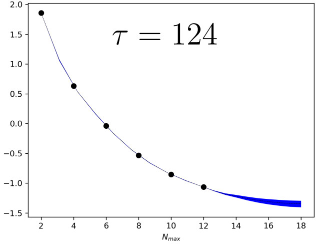
$\tau = 0.01$
2 1 0 −1 −2 −3 −4 −5
2 4 6 8 10 12 14 16 18
N_{max}



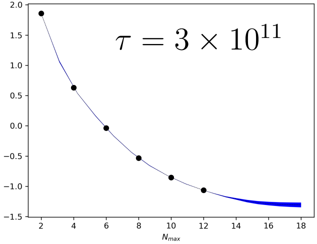
$\tau = 0.4$
2 0 −2 −4 −6 −8 −10
2 4 6 8 10 12 14 16 18
N_{max}



$\tau = 3.3$
2 1 0 −1 −2 −3 −4 −5
2 4 6 8 10 12 14 16 18
N_{max}

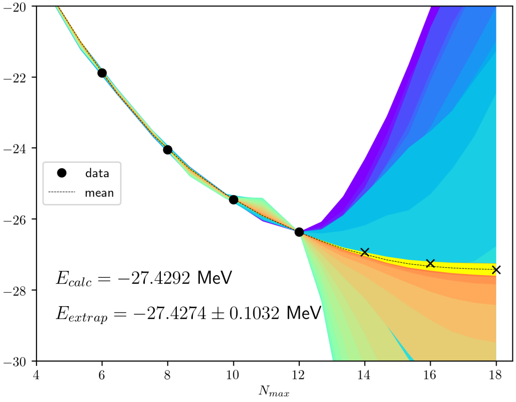


$\tau = 124$
2.0 1.5 1.0 0.5 0.0 −0.5 −1.0 −1.5
2 4 6 8 10 12 14 16 18
N_{max}

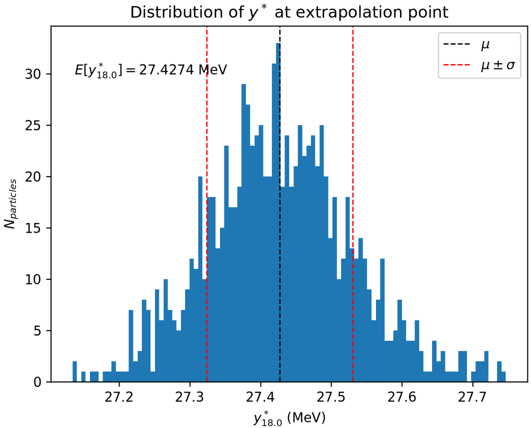


$\tau = 3 \times 10^{11}$
2.0 1.5 1.0 0.5 0.0 −0.5 −1.0 −1.5
2 4 6 8 10 12 14 16 18
N_{max}

Results: He^4

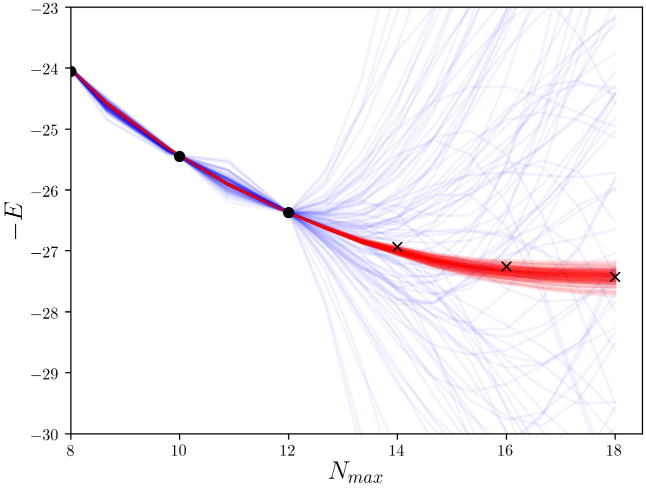


−20
−22
−24
−26
−28
−30
4 6 8 10 12 14 16 18
N_{max}
data
mean
$E_{calc} = -27.4292$ MeV
$E_{extrap} = -27.4274 \pm 0.1032$ MeV

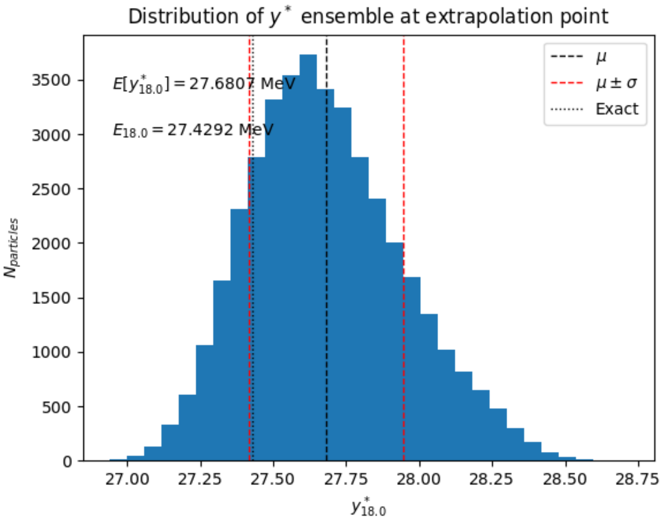


Distribution of y^* at extrapolation point
$E[y^*_{18.0}] = 27.4274$ MeV
μ
$\mu \pm \sigma$
$N_{particles}$
0 5 10 15 20 25 30
27.2 27.3 27.4 27.5 27.6 27.7
$y^*_{18.0}$ (MeV)

Results: He^4



Results: He^4, Ensemble of SMC Runs



Summary

- Demonstrated viability of extrapolation by constrained Gaussian processes

Recent Progress

- Re-factored code
- Tuned SMC schedules give better results

Next Steps

- Test reproducibility
- Add ”far-point” constraint: $\mathbf{y}' \to 0$ at high N_{max}
- Try ”log-kernels”

TRIUMF

Thank you
Merci

Discovery,
accelerated

Extra Slide: Including Derivatives

$$p\left(\begin{bmatrix}\mathbf{y}\\ \mathbf{y}^*\\ \mathbf{y}'\\ \mathbf{y}''\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mu\\ \mu_*\\ \mu_1\\ \mu_2\end{bmatrix}, \begin{bmatrix}C & C_* & C_1 & C_2\\ C_*^T & C_{**} & C_{1*} & C_{2*}\\ C_1^T & C_{*1} & C_{11} & C_{12}\\ C_2^T & C_{*2} & C_{21} & C_{22}\end{bmatrix}\right)$$

then

$$p\left(\begin{bmatrix}\mathbf{y}^*\\ \mathbf{y}'\\ \mathbf{y}''\end{bmatrix} |\mathbf{y}\right) = \mathcal{N}(\nu, \Sigma)$$

where

$$\nu = [C_*, C_1, C_2]C^{-1}\mathbf{y}$$

$$\Sigma = \begin{bmatrix}C_{**} & C_{1*} & C_{2*}\\ C_{*1} & C_{11} & C_{12}\\ C_{*2} & C_{21} & C_{22}\end{bmatrix} - [C_*, C_1, C_2]C^{-1}\begin{bmatrix}C_*\\ C_1\\ C_2\end{bmatrix}$$

$$\begin{aligned} C_{*1} &= C[\mathbf{y}^*, \mathbf{y}'] \\ &= \frac{\partial}{\partial x_j} r(\mathbf{x}^*{}_i, \mathbf{x}'{}_j) \\ &\vdots \\ C_{22} &= C[\mathbf{y}'', \mathbf{y}''] \\ &= \frac{\partial^2}{\partial x_i^2}\frac{\partial^2}{\partial x_j^2} r(\mathbf{x}''{}_i, \mathbf{x}''{}_j) \end{aligned}$$

Extra Slide: Constraints on Derivatives

Weight probability of samples:

$$p(\mathbf{y}^*|\mathbf{y}) \sim \mathcal{N}(\mu_*, \Sigma_*) \times m(\mathbf{y}') \times n(\mathbf{y}'')$$

based on criteria:

$$m(\mathbf{y}') = \sum_i \left(m(y_i') = \begin{cases} 1 & \text{if } y_i' < 0 \\ 0 & \text{otherwise} \end{cases} \right)$$

$$n(\mathbf{y}'') = \sum_i \left(n(y_i'') = \begin{cases} 1 & \text{if } y_i'' > 0 \\ 0 & \text{otherwise} \end{cases} \right)$$

How to compute derivatives?