Parameterizing Molecular Models via Gaussian Approximation Methods

Vassia Kyrieri

Department of Mathematics and Applied Mathematics, University of Crete, Greece

Supervisors:

Prof. Vagelis Harmandaris

Dr. Evangelia Kalligiannaki

Members of committee:

Prof. Vagelis Harmandaris

Prof. George N. Makrakis

Dr. Yannis Pantazis

Outline

- Molecular Dynamics Simulations
- The Coarse-Graining technique
- Introduction to Machine Learning field and Gaussian Process Regression model
- From Gaussian Process Regression to Kernel Ridge Regression
- Toy problem of GPR
- Data pre-processing - Minimization problem
- Results of the methane system
- Conclusion and Future Work

Equations of Motion

Molecular Dynamics

Molecular Dynamics simulation is a technique for studying the movement of atoms in a classical many-body system. 'Classical' = Newton's law

N-body Problem at atomistic level

- Denote $\boldsymbol{q} = (q_1, ..q_N)$, $\quad \boldsymbol{q} \in \mathbb{R}^{3N}$ positions of N atoms and their momentum $\boldsymbol{p}$, $\boldsymbol{p}(t) \in \mathbb{R}^{3N}$
- Obey Newton's second law $\boldsymbol{F} = \boldsymbol{M}\frac{d^2\boldsymbol{q}}{d^2t} = -\boldsymbol{\nabla}_{\boldsymbol{q}} U(\boldsymbol{q})$ where
  - $\boldsymbol{M} = diag[m_1, \cdots, m_N]$: mass matrix
  - $\boldsymbol{F} \in \mathbb{R}^{3N}$: the total force applied to each atom's position
  - $U \in \mathbb{R}^N$: potential energy of the system

MD Algorithm

Verlet Algorithm: A method for solving the classical equations of motion

$$\boldsymbol{q}(t+\Delta t) - \boldsymbol{q}(t-\Delta t) = 2\boldsymbol{v}(t)\Delta t + O(\Delta t^3)$$

$$\boldsymbol{v}(t) = \frac{\boldsymbol{q}(t+\Delta t) - \boldsymbol{q}(t-\Delta t)}{2\Delta t} + O(\Delta t^2)$$

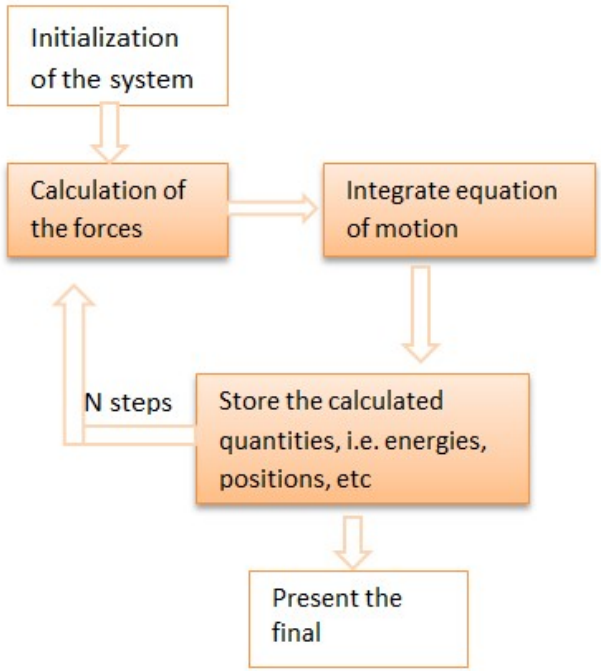


Figure: A Flow diagram of MD algorithm

The Challenge

- Complex molecular systems require a large-range of time and length scales to understand the properties.
- We reduce the number of degrees of freedom.
- One solution: coarse-graining.

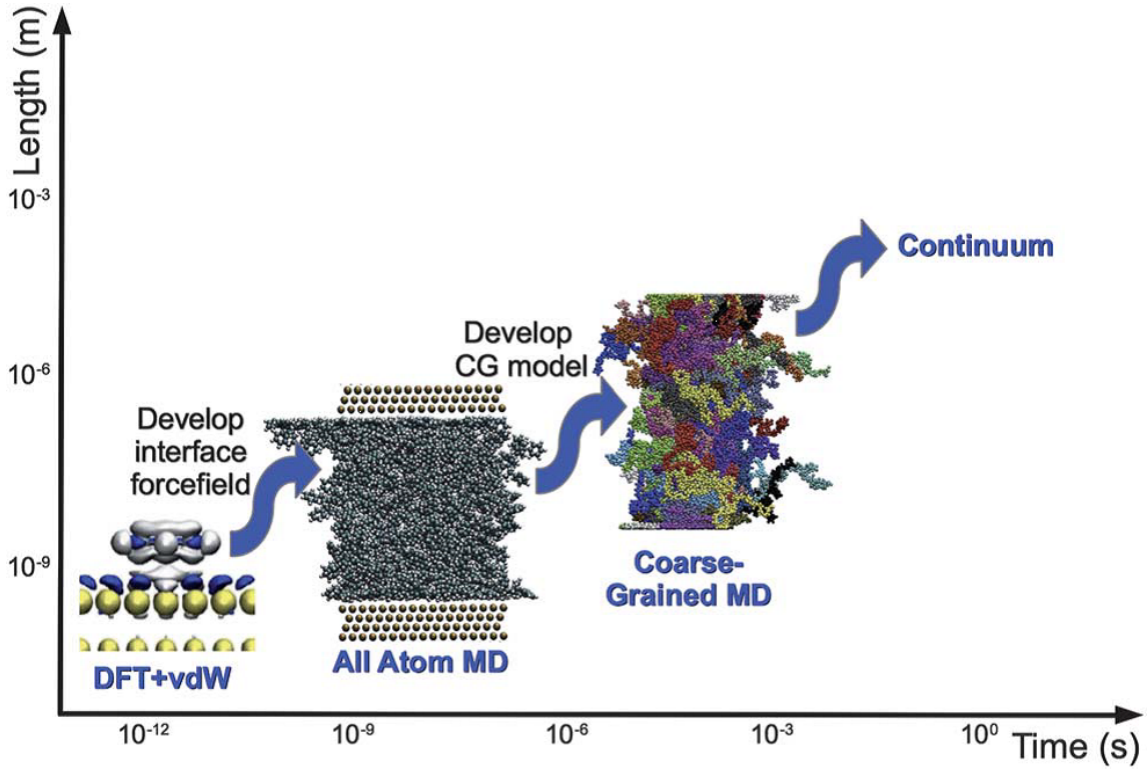


Figure: Levels of multiscale dynamics

Coarse-Grained (CG) Presentation

- M particles or "superatoms", $M < N$.
- Define a CG map $\Pi : \boldsymbol{q} \mapsto \mathcal{Q}$

  $\boldsymbol{q} = (\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N), \quad \mathcal{Q} = (\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_M)$
- Linear CG map,

  $\Pi : \mathbb{R}^{3N} \to \mathbb{R}^{3M}$

  $\boldsymbol{q} \mapsto \Pi(\boldsymbol{q}) = \mathcal{Q}, \; \boldsymbol{Q}_i = \Pi_i(\boldsymbol{q}_1, \cdots, \boldsymbol{q}_N)$

  $\boldsymbol{Q}_i = \sum_{j=1}^{N} w_{ij} \boldsymbol{q}_j, \quad i = 1, \cdots, M.$

  e.g. for the methane system

  $w_{ij} = \frac{m_j}{\sum_{k=1, k \in j}^{5} m_{ik}}$

- The CG potential $U(\mathcal{Q})$, describing the interactions between super-atoms, is yet unknown.
- $U(\mathcal{Q})$ is known as Potential of Mean Force.

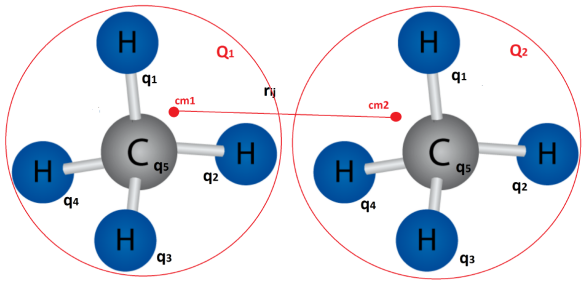


Figure: System of the two-body bulk methane

Potential of Mean Force

- Exact CG potential - Potential of Mean Force (PMF).

$$U(\mathcal{Q}) = -\frac{1}{\beta} \ln \int_{A(\mathcal{Q})} e^{-\beta U(\mathcal{Q})} d\mathcal{Q}.$$

  where $A(\mathcal{Q}) = \{\mathbf{q} \in \mathbb{R}^{3N} : \mathbf{\Pi}(\mathbf{q}) = \mathcal{Q}\}$
- We cannot compute the integral because of the high dimensionality.
- All CG methods attempt to estimate $U(\mathcal{Q})$.
- $U(\mathcal{Q})$ consists of M-body interactions. Most methods assume it is a function of few-body terms

$$U(\mathcal{Q}) = \sum_{ij} U_{ij}(\boldsymbol{Q}_i, \boldsymbol{Q}_j) + \sum_{ijk} U_{ijk}(\boldsymbol{Q}_i, \boldsymbol{Q}_j, \boldsymbol{Q}_k) + \sum_{ijkl} U_{ijkl}(\boldsymbol{Q}_i, \boldsymbol{Q}_j, \boldsymbol{Q}_k, \boldsymbol{Q}_l)$$

Machine Learning field

Machine learning algorithms are able to learn from previously gained data in order to predict or decide without being programmed to do so.

Categories of Machine Learning: Supervised learning, Unsupervised learning, Reinforcement learning

Gaussian Process Regression (GPR) Theory

GPR is a nonlinear, non-parametric regression, based on Bayesian probability theory. It is based on a large amount of data and a flexible function that fits the data and makes the predictions.

Two approaches will be presented: the weight-space and the function-space views of the GPR.

Introduction to GPR

Assume a training data set with elements x_n, where $n = 1, 2, \cdots, n_s$ with the corresponding target values $\{\tilde{t}_n\}$.
Goal: to predict the value of y for a new value of x, following the observation equation

$$\tilde{t} = y(x) + \epsilon, \quad \epsilon : \text{ observation error}$$

In the context of observations, we consider that $y = y(x)$ is a Gaussian process,

$$y(x) \sim \mathcal{GP}(m(x), k(x, x')),$$

$y(x)$ follows the Gaussian probability distribution function given by

$$(2\pi)^{-n/2}(\det \mathbf{K})^{-1/2} \exp\left[-\frac{1}{2} < \mathbf{K}^{-1}(\mathbf{y} - \mathbf{m}), \mathbf{y} - \mathbf{m} >\right], \mathbf{K}_{ij} = k(x_i, x_j), \mathbf{m}_i = m(x_i)$$

The mean value reflects the expected function value at input x,

$$m(x) = \mathbb{E}[y(x)] = 0.$$

The covariance function $k(x, x')$, the kernel of the Gaussian process, models the dependence between the function values at different input points x and x',

$$k(x, x') = \mathbb{E}[(y(x) - m(x))(y(x') - m(x'))].$$

Regression

Let $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_{n_s}), \mathbf{x}_i \in \mathbb{R}^d$ denote the known n_s independent input observations and output $y_i = y(\mathbf{x}_i), i = 1, ..n_s$ of value $\mathbf{x}_i$. $y : \mathbb{R}^d \to \mathbb{R}$.
The model for regression:

$$y(\boldsymbol{X}) = \phi(\boldsymbol{X})^{tr}\boldsymbol{a}, \quad \text{in matrix form } \mathbf{y} = \boldsymbol{\Phi}\mathbf{a}$$

The most common model is the linear regression:

$$y(\boldsymbol{X}) = \boldsymbol{X}^{tr}\boldsymbol{a}$$

- $\boldsymbol{a} = (a_1, \cdots, a_{n_s})$ the weight vector
- $\phi(\boldsymbol{X})$ A vector of fixed nonlinear basis function
- $\mathbb{E}[\mathbf{y}] = \boldsymbol{\Phi}\mathbb{E}[\mathbf{a}] = 0,$
- $\text{Cov}[\mathbf{y}] = \boldsymbol{\Phi}\mathbb{E}[\mathbf{a}\mathbf{a}^{tr}]\boldsymbol{\Phi}^{tr} = \frac{1}{\beta}\boldsymbol{\Phi}\boldsymbol{\Phi}^{tr} = \mathbf{K}$
- $p(\boldsymbol{a}) = N(\mathbf{0}, \beta^{-1}\boldsymbol{I})$
- $\mathbf{K}$ is the Gram matrix with elements $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\beta}\phi(\mathbf{x}_i)^{tr}\phi(\mathbf{x}_j)$
- The covariance function $k(x, x')$ The kernel of the Gaussian process

Modelling functions: the weight space view

According to a Bayesian viewpoint, we model functions with linear regression.

$$\tilde{t}_n = y(\mathbf{x}_n) + \epsilon_n = \mathbf{x}_n^{tr}\mathbf{a} + \epsilon_n \quad \text{where prior distribution } p(\boldsymbol{a}) = N(\mathbf{0}, \beta^{-1}\boldsymbol{I})$$

$$\epsilon_n \sim N(0, \lambda) \text{ and } \tilde{t}_n|y_n \sim N(y_n, \lambda), \quad \text{where } \lambda \text{ the precision of the noise.}$$

The joint distribution conditioned on $\mathbf{y}$

$$p(\tilde{\mathbf{t}}|\mathbf{y}) = p(\tilde{\mathbf{t}}|\mathbf{X}, \mathbf{a}) = N(\mathbf{y}, \lambda \mathbf{I}_{n_s}) = N(\mathbf{X}^{tr}\mathbf{a}, \lambda \mathbf{I}_{n_s}) \quad \mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_{n_s})$$

The posterior distribution

$$p(\mathbf{a}|\tilde{\mathbf{t}}, \mathbf{X}_n) \propto p(\tilde{\mathbf{t}}|\mathbf{X}, \mathbf{a})p(\mathbf{a}) = N(\lambda^{-1}\mathbf{A}^{-1}\mathbf{X}\tilde{\mathbf{t}}, \mathbf{A}_n^{-1})$$

where $\mathbf{A} = \beta\mathbf{I} + \lambda^{-1}\mathbf{X}\mathbf{X}^{tr}$

Weight space view of regression

As inference is performed over the weights, this is referred to as "the weight space view of regression". The new output $y_* = \tilde{t}_* - \epsilon_*$

$$p(y_*|\mathbf{x}_*, \mathbf{X}, \tilde{\mathbf{t}}) = N(\lambda^{-1}\mathbf{x}_*^{tr}\mathbf{A}^{-1}\mathbf{X}\tilde{\mathbf{t}}, \mathbf{x}_*^{tr}\mathbf{A}^{-1}\mathbf{x}_*)$$

Example of Modelling functions: the weight space view

Inputs and corresponding outputs

n	x_n	$\tilde{t}_n$
1	0.9	0.1
2	3.8	1.2
3	5.2	2.1
4	6.1	1.1
5	7.5	1.5
6	9.6	1.2

$$\tilde{t}_n = \mathbf{x}_n^{tr}\mathbf{a} + \epsilon_n$$

$$\mathbf{x}_n = \begin{bmatrix} 1 \\ x_n \end{bmatrix} \quad \mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}$$

According to

$$p(\mathbf{a}|\tilde{\mathbf{t}}_n, \mathbf{X}_n) = N(\lambda^{-1}\mathbf{A}_n^{-1}\mathbf{X}_n\tilde{\mathbf{t}}_n, \mathbf{A}_n^{-1})$$

where $\mathbf{A}_n = \Sigma + \lambda^{-1}\mathbf{X}_n\mathbf{X}_n^{tr}$, with $p(\boldsymbol{a}) = N(\mathbf{0}, \Sigma)$. We set

$$\mathbf{X}_n = \begin{bmatrix} 1 & 0.9 \\ 1 & 3.8 \\ \vdots & \\ 1 & 9.6 \end{bmatrix} \quad \tilde{\mathbf{t}}_n = \begin{bmatrix} 0.1 \\ 1.2 \\ \vdots \\ 1.2 \end{bmatrix}$$

and $\lambda = 0.1$, $\Sigma = [[0.1, 1.1], [1.1, 0.1]]$.

Thus $p(\mathbf{a}|\tilde{\mathbf{t}}_n, \mathbf{X}_n) = N([0.55, 0.11], \mathbf{A}_n^{-1})$ if $x_* = 3.0$ then $\mathbf{x}_* = [1.0, 3.0]$

$y(x^*) = \mathbf{x}_*^{tr}\mathbf{a}$, then $y(x^*) = 0.9037$.

Modelling functions: the function space view I

We have the function $y(\mathbf{x})$ as a Gaussian process ($\tilde{t}_n = y(\mathbf{x}_n) + \epsilon_n$):

$$\mathbf{y} \sim N(0, K(\mathbf{X}, \mathbf{X})) \sim N(\mathbf{0}, \mathbf{K})$$

The marginal distribution of $p(\tilde{\mathbf{t}})$

$$p(\tilde{\mathbf{t}}) = \int p(\tilde{\mathbf{t}}|\mathbf{y})p(\mathbf{y})d\mathbf{y} = \mathbb{E}_{\mathbf{y}}[p(\tilde{\mathbf{t}}|\mathbf{y})] = N(\mathbf{0}, \mathbf{C})$$

where the covariance matrix $\mathbf{C}$ has elements

$$C(\mathbf{x}_n, \mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m) + \lambda\delta_{nm}.$$

Goal: to predict the target variable $\tilde{t}_{n_s+1}$ for a new x_{n_s+1}, thus the predictive distribution

$$p(\tilde{t}_{n_s+1}|\tilde{\mathbf{t}}_{n_s}) = \frac{p(\tilde{\mathbf{t}}_{n_s+1})}{p(\tilde{\mathbf{t}}_{n_s})} = N(\mathbf{0}, \mathbf{C}_{n_s+1}) \quad \tilde{\mathbf{t}}_{n_s+1} = (\tilde{\mathbf{t}}_{n_s}, \tilde{t}_{n_s+1})$$

the covariance matrix $\mathbf{C}_{n_s+1} = \begin{pmatrix} \mathbf{C}_{n_s} & \mathbf{k} \\ \mathbf{k}^{tr} & c \end{pmatrix}$, $\quad c = k(\mathbf{x}_{n_s+1}, \mathbf{x}_{n_s+1}) + \lambda^{-1}$ (scalar) .

Modelling functions: the function space view II

The second order statistics, namely the mean and covariance, is given by $\tilde{t}_{n_s+1} \sim N(m(\mathbf{x}_{n_s+1}), \sigma^2(\mathbf{x}_{n_s+1}))$

$$m(\mathbf{x}_{n_s+1}) = \mathbf{k}^{tr}\mathbf{C}_{n_s}^{-1}\tilde{\mathbf{t}}$$

where $\mathbf{k}$ is the vector with elements $k(x_i, x_{n_s+1}), i = 1, \cdots, n_s$.

$$\sigma^2(\mathbf{x}_{n_s+1}) = c - \mathbf{k}^{tr}\mathbf{C}_{n_s}^{-1}\mathbf{k}.$$

An expansion in radial basis function gives the following form:

$$m(\mathbf{x}_{n_s+1}) = \sum_{n=1}^{n_s} a_n k(\mathbf{x}_n, \mathbf{x}_{n_s+1})$$

where a_n is the n^{th} of $\mathbf{C}_{n_s}^{-1}\tilde{\mathbf{t}}$.

In the weight space view of the previous section, we concentrate our interest on distributions over weights. In Gaussian process regression, we focus directly on such distributions over functions.

Example of Modelling functions: the function space view

In this method, we have the same data from the previous example and we can calculate the weights $\mathbf{a} = [\mathbf{K} + \lambda \mathbf{I}]^{-1} \tilde{\mathbf{t}}_n$.

Parameters: $\lambda = 0.1, \delta = 1,\ \theta = 1.1$ of the kernel function $k(\mathbf{x}, \mathbf{x}') = \delta^2 \exp\left(-\frac{|\mathbf{x}-\mathbf{x}'|^2}{2\theta^2}\right)$

New input point $x_* = 3$.

n	x_n	$\tilde{t}_n$	a_n	$k(x_n, x_*)$	$a_n k(x_n, x_*)$
1	0.9	0.1	0.08	1.6e-01	1.3e-02
2	3.8	1.2	0.20	7.6e-01	1.5e-02
3	5.2	2.1	2.47	1.3e-01	3.3e-01
4	6.1	1.1	-1.23	1.8e-03	-2.3e-02
5	7.5	1.5	1.48	2.3e-04	3.4e-04
6	9.6	1.2	0.87	1.5e-08	1.3e-08
				$y(x^*) = \sum_{n=1}^{6} a_n k(x_n, x_*)$:	0.48

If $\theta = 2.2 \rightarrow y(x^*) = 0.92$.

By changing the set of hyperparameters, the $y(x^*)$ value changes, and by applying the optimal set we can better approximate the value of the least square problem (0.9039).

Switching back to the weight view

We want to approximate $y(\mathbf{x})$ by $f(\mathbf{x})$ expressed as a linear combination of n_s basis functions.

$$f(\mathbf{x}) = m(\mathbf{x}_n) = \sum_{i=1}^{n_s} a_i k(\mathbf{x}, \mathbf{x}_i)$$

One very popular choice of a kernel

$$k(\mathbf{x}, \mathbf{x}_i) = \delta^2 \exp\left(-\frac{|\mathbf{x} - \mathbf{x}_i|^2}{2\theta^2}\right)$$

where

- $\mathbf{x}_i$ a previously observed input value in $\mathbf{X}_n$
- $\mathbf{a} = [\mathbf{K} + \lambda \mathbf{I}]^{-1} \tilde{\mathbf{t}}_n$
- $\mathbf{K} + \lambda \mathbf{I}$ corresponds to the matrix $\mathbf{C}_{n_s}$

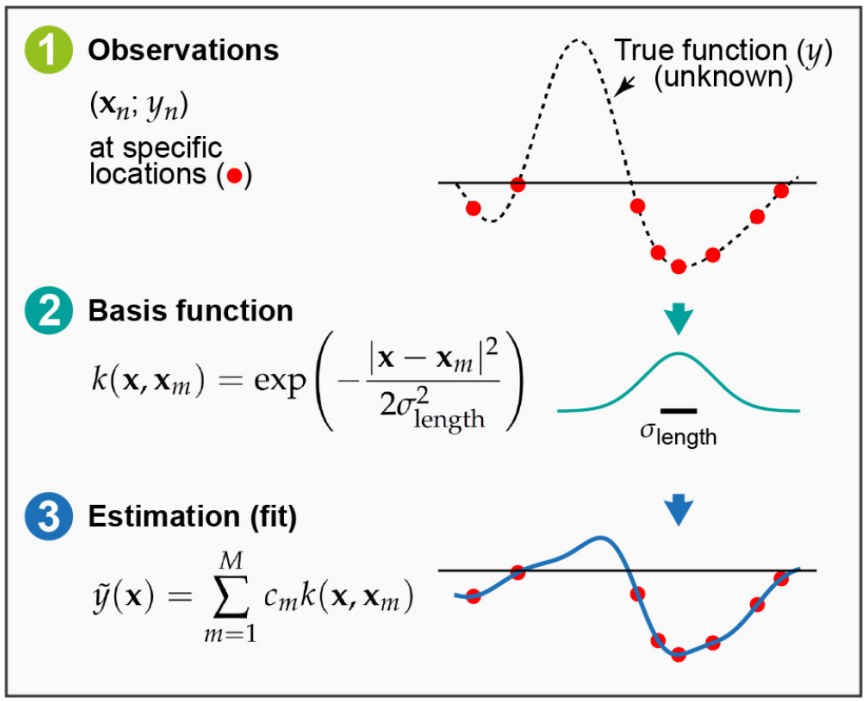


Figure: The prediction of a GPR model

Relation to Kernel Ridge Regression (KRR)

The fitting of the Gaussian process regression model can also be written as a (kernel) ridge regression problem

Goal: to find a by minimizing the loss (or cost) function

$$L = \sum_{n=1}^{n_s} \frac{[y_n - f(\mathbf{x}_n)]^2}{\sigma_i^2} + \lambda \mathbf{R}, \quad \mathbf{R} \text{ regularization parameter}, \sigma \text{ the relative importance}$$

We worked with 2 regularization terms:

$$\mathbf{R} = \sum_{n,n'}^{n_s} a_n k(x_n, x_{n'}) a_{n'} = \mathbf{a}^{tr} \mathbf{K}_{n_s n_s} \mathbf{a} \quad \mathbf{R} = \sum_{n,n'}^{n_s} a_n a_{n'}.$$

Key difference between kernel ridge regression and Gaussian process regression

The interpretation of regularization. In KRR: An empirical regularization parameter λ that needs to be optimized. In GPR: the regularization parameter can be identified with the variance of noisy observations, λ.

The loss function:

$$L = (\mathbf{y} - \mathbf{K}_{n_s n_s} \mathbf{a})^{tr} \mathbf{\Sigma}^{-1} (\mathbf{y} - \mathbf{K}_{n_s n_s} \mathbf{a}) + \lambda \mathbf{a}^{tr} \mathbf{K}_{n_s n_s} \mathbf{a}, \quad \Sigma_{ii} = \sigma_i^2$$

$$\nabla_{\mathbf{a}^{tr}} L = 0 \rightarrow -\mathbf{K}_{n_s n_s} \mathbf{\Sigma}^{-1} \mathbf{y} + \mathbf{K}_{n_s n_s} \mathbf{\Sigma}^{-1} \mathbf{K}_{n_s n_s}^{tr} \mathbf{a} + \lambda \mathbf{K}_{n_s n_s} \mathbf{a} = 0.$$

Force Matching problem - Minimization problem

Force Matching

$$\min_{\mathbf{a}} L(\Phi;\mathbf{a}) = \min_{\mathbf{a}} \mathbb{E}_{\mu}\left[||(\mathbf{F}o\Pi)(\cdot) - \mathbf{F}_{CG}(\cdot;\mathbf{a})||^2_{3M}\right]$$

Here, we have observations/samples $\{\mathbf{q}_i, \mathbf{F}_i\}_{i=1}^{n_s}$ and according to the CG map the $\mathcal{Q} = \Pi\mathbf{q}$,

$$\min_{\mathbf{a}} \frac{1}{n_s}\sum_{l=1}^{n_s}\sum_{l=1}^{M} ||(\mathbf{F}o\Pi_l)(\mathbf{q}^{(l)}) - \mathbf{F}_{CG,l}(\Pi\mathbf{q}^{(l)};\mathbf{a})||_3^2$$

The force is calculated by

$$\mathbf{F}_{CG,I}(\mathcal{Q};\mathbf{a}) = -\nabla_{\mathbf{Q}_I} U_{CG}(\mathcal{Q};\mathbf{a}), \quad \mathbf{F}_{CG,I}(\mathcal{Q};\mathbf{a}) \in \mathbb{R}^3$$

where

$$U_{CG}(\mathcal{Q};\mathbf{a}) = \sum_{j,k} W_{dimer}(D_{dimer}(\mathbf{Q}_j,\mathbf{Q}_k);\hat{\mathbf{a}}) + \sum_{j,k,l} W_{trimer}(D_{trimer}(\mathbf{Q}_j,\mathbf{Q}_k,\mathbf{Q}_l);\mathbf{a}^*)$$

where $D_{dimer}(\mathbf{Q}_j,\mathbf{Q}_k)$, $D_{trimer}(\mathbf{Q}_j,\mathbf{Q}_k,\mathbf{Q}_l)$: the descriptors

Application of Gaussian Process Regression - Only Dimer Descriptor I

In the first technique, we describe the total force including only pair contributions.

The pair force is equal to the derivative of the W function.

$$\hat{f}(x) = \frac{dW_{dimer}(x)}{dx} \text{ where } x = D(\mathbf{Q}_j, \mathbf{Q}_k) = |\mathbf{Q}_j - \mathbf{Q}_k| \in \mathbb{R}, \quad j, k = 1, 2, \ldots, M$$

We choose a sparser model with far fewer kernel basis functions than the input data points where the locations of these basis functions (which we call the representative set) need not coincide with the input data locations. Thus,

$$\hat{f}(|\mathbf{Q}_j - \mathbf{Q}_k|) = \sum_{i}^{n_p} \hat{\alpha}_i \hat{k}(x_i, |\mathbf{Q}_j - \mathbf{Q}_k|)$$

where

- n_p is the number of basis points
- x_i takes values of a grid, from $r_{min} - \epsilon$ to r_{cut}, where $\epsilon > 0$
- $\hat{k}(x_i, x) = \hat{\delta}^2 \exp\left(-\frac{|x - x_i|^2}{2\hat{\theta}^2}\right)$.

Application of Gaussian Process Regression - Only Dimer Descriptor II

Thus,

$$
\begin{aligned}
\mathbf{F}_{CG,i}(\mathcal{Q};\mathbf{a}) &\approx \hat{\mathbf{F}}_i(\mathcal{Q};\hat{\mathbf{a}}) = -\nabla_{\mathbf{Q}_i} U_{CG}(\mathcal{Q};\hat{\mathbf{a}}) \\
&= -\nabla_{\mathbf{Q}_i}\Big(\sum_{j=1}^{M}\sum_{k=j+1}^{M} W_{dimer}(D(\mathbf{Q}_j,\mathbf{Q}_k))\Big) \\
&= -\sum_{j>k}^{M} \frac{dW_{dimer}(D(\mathbf{Q}_j,\mathbf{Q}_k))}{dD(\mathbf{Q}_j,\mathbf{Q}_k)} \nabla_{\mathbf{Q}_i} D(\mathbf{Q}_j,\mathbf{Q}_k) \\
&= -\sum_{j>k}^{M} \hat{f}(|\mathbf{Q}_j-\mathbf{Q}_k|)\nabla_{\mathbf{Q}_i} D(\mathbf{Q}_j,\mathbf{Q}_k) \\
&= -\sum_{j>k}^{M}\sum_{l=1}^{n_p} \hat{a}_l \hat{k}(x_l,|\mathbf{Q}_j-\mathbf{Q}_k|)\nabla_{\mathbf{Q}_i}(|\mathbf{Q}_j-\mathbf{Q}_k|)
\end{aligned}
$$

- Kernel function $\hat{k}(x_i,|\mathbf{Q}_j-\mathbf{Q}_k|)$
- Dimer Descriptor $D(\mathbf{Q}_j,\mathbf{Q}_k) = \sqrt{|Q_{j1}-Q_{k1}|^2+|Q_{j2}-Q_{k2}|^2+|Q_{j3}-Q_{k3}|^2}$

- $i=k \rightarrow \nabla_{\mathbf{Q}_i} D(\mathbf{Q}_j,\mathbf{Q}_k) = -\frac{\mathbf{Q}_j-\mathbf{Q}_i}{|\mathbf{Q}_j-\mathbf{Q}_i|}$
- $i=j \rightarrow \nabla_{\mathbf{Q}_i} D(\mathbf{Q}_j,\mathbf{Q}_k) = \frac{\mathbf{Q}_j-\mathbf{Q}_i}{|\mathbf{Q}_j-\mathbf{Q}_i|}$
- $i\neq k,j \rightarrow \nabla_{\mathbf{Q}_i} D(\mathbf{Q}_j,\mathbf{Q}_k) = (0,0,0)$

Application of Gaussian Process Regression - Only Dimer Descriptor III

Minimization problem

$$\min_{\hat{\mathbf{a}}} \frac{1}{n_s} \sum_{l=1}^{n_s} \sum_{I=1}^{M} ||\mathbf{F}(\mathbf{Q}_I^l) - \sum_{j<k}^{M} \sum_{i=1}^{n_p} \hat{a}_i k(x_i, |\mathbf{Q}_j^l - \mathbf{Q}_k^l|)(-\nabla_{\mathbf{Q}_I} D(\mathbf{Q}_j^l, \mathbf{Q}_k^l))||_3^2$$

The term $\sum_{j<k}^{M} \sum_{i=1}^{n_p} a_i \hat{k}(x_i, r_{jk})(-\nabla_{\mathbf{Q}_I} r_{jk})$ corresponds to the linear term $\hat{\mathbf{K}}\hat{\mathbf{a}}$, Thus, to solve the minimization problem, we solve the linear problem

$$\mathbf{y} = \hat{\mathbf{K}}\hat{\mathbf{a}}, \quad \mathbf{y} \in \mathbb{R}^{3M \cdot n_s}, \hat{\mathbf{a}} \in \mathbb{R}^{n_p}, \hat{\mathbf{K}} \in \mathbb{R}^{(3M \cdot n_s) \times n_p}$$

where

- $\mathbf{y} = [F_{1,x}(\mathbf{Q}^1), F_{1,y}(\mathbf{Q}^1), \cdots, F_{M,z}(\mathbf{Q}^1), F_{1,x}(\mathbf{Q}^2), \cdots, F_{M,z}(\mathbf{Q}^{n_s}]^{tr}$
- $\hat{\mathbf{a}} = [\hat{a}_1, \hat{a}_2, \cdots, \hat{a}_{n_p}]^{tr}$
- $\hat{\mathbf{K}} = \begin{bmatrix} \sum_{j<k}^{M} \hat{k}(x_1, r_{jk}^{(1)})(-\frac{dr_{jk}^{(1)}}{d\mathbf{Q}_{1,x}^{(1)}}) & \cdots & \sum_{j<k}^{M} \hat{k}(x_{n_p}, r_{jk}^{(1)})(-\frac{dr_{jk}^{(1)}}{d\mathbf{Q}_{1,x}^{(l)}}) \\ \vdots & \ddots & \vdots \\ \sum_{j<k}^{M} \hat{k}(x_1, r_{jk}^{(n_s)})(-\frac{dr_{jk}^{(n_s)}}{d\mathbf{Q}_{M,z}^{(n_s)}}) & \cdots & \sum_{j<k}^{M} \hat{k}(x_{n_p}, r_{jk}^{(n_s)})(-\frac{dr_{jk}^{(n_s)}}{d\mathbf{Q}_{M,z}^{(n_s)}}) \end{bmatrix}$

Application of Gaussian Process Regression - Both Descriptors I

In the second technique, dimer and trimer descriptors are included. From

$$\mathbf{F}_i(\mathcal{Q};\hat{\mathbf{a}},\mathbf{a}^*) = \hat{\mathbf{F}}_i(\mathcal{Q};\hat{\mathbf{a}}) + \mathbf{F}_i^*(\mathcal{Q};\mathbf{a}^*)$$

We have already calculate the term $\hat{\mathbf{F}}_i(\mathcal{Q};\hat{\mathbf{a}})$.

For three-body contributions

We choose a sparser model with far fewer kernel basis functions.

$$f^*(D(\mathbf{Q}_j,\mathbf{Q}_k,\mathbf{Q}_l)) = \sum_{I}^{\mathbf{n_p}^*} \alpha_I^* k^*(\mathbf{x}_I, D(\mathbf{Q}_j,\mathbf{Q}_k,\mathbf{Q}_l))$$

- $D(\mathbf{Q}_j,\mathbf{Q}_k,\mathbf{Q}_l) = [d_1,d_2,d_3] = [r_{ij}+r_{ik},(r_{ij}-r_{ik})^2,r_{jk}]$
- $\mathbf{n_p}^* = (\cdot,\cdot,\cdot)$: a vector with the number of basis points in each dimension
- $\mathbf{x}_I$: a 3-d vector and each element takes values of a different grid
- I is a multi-index $I=(i,j,k)$, $i,j,k = 1,\ldots,n_p^*$ and takes $n_p^* = \#\mathbf{n_p}^*$
  e.x. if we choose 3 basis points for each element, then the multi-index I takes $n_p^* = 27$ values: $\mathbf{n_p}^* = (1,1,1),(1,1,2),\cdots,(1,3,3),(2,1,1),\cdots,(3,3,3)$
- Kernel function for trimer $k^*(\mathbf{x},\mathbf{x}') = \delta_*^2 \exp\left[\sum_{i=1}^3(-\frac{|x_i-x_i'|^2}{2(\theta_*^i)^2})\right]$, $\theta_* = [\theta_*^1,\theta_*^2,\theta_*^3]$

Application of Gaussian Process Regression - Both Descriptors II

Thus,

$$
\begin{aligned}
\mathbf{F}_i^*(\mathcal{Q};\mathbf{a}^*) &= -\nabla_{\mathbf{Q}_i} U_{CG}(\mathcal{Q};\mathbf{a}^*) \\
&= -\nabla_{\mathbf{Q}_i}\Big(\sum_{j=1}^{M}\sum_{k=j+1}^{M}\sum_{l=k+1}^{M} W_{trimer}(D(\mathbf{Q}_j,\mathbf{Q}_k,\mathbf{Q}_l))\Big) \quad (\text{here } f^*(\mathbf{x}) = W_{trimer}(\mathbf{x})) \\
&= -\nabla_{\mathbf{Q}_i}\Big[\sum_{j<k<l}^{M} W_{trimer}(d_1(\mathbf{Q}_j,\mathbf{Q}_k,\mathbf{Q}_l), d_2(\mathbf{Q}_j,\mathbf{Q}_k,\mathbf{Q}_l), d_3(\mathbf{Q}_j,\mathbf{Q}_k,\mathbf{Q}_l))\Big] \\
&= -\sum_{j<k<l}^{M}\Big[\frac{\partial W_{tr}(d_1,d_2,d_3)}{\partial d_1}\nabla_{\mathbf{Q}_i} d_1(\cdot) + \frac{\partial W_{tr}(d_1,d_2,d_3)}{\partial d_2}\nabla_{\mathbf{Q}_i} d_2(\cdot) + \frac{\partial W_{tr}(d_1,d_2,d_3)}{\partial d_3}\nabla_{\mathbf{Q}_i} d_3(\cdot)\Big] \\
&= -\sum_{j<k<l}^{M}\sum_{l}^{\mathbf{n_p}^*}\alpha_l^*\Big[\frac{\partial k^*(\mathbf{x}_l,(d_1,d_2,d_3))}{\partial d_1}\nabla_{\mathbf{Q}_i} d_1(\cdot) + \frac{\partial k^*(\mathbf{x}_l,(d_1,d_2,d_3))}{\partial d_2}\nabla_{\mathbf{Q}_i} d_2(\cdot) \\
&\quad + \frac{\partial k^*(\mathbf{x}_l,(d_1,d_2,d_3))}{\partial d_3}\nabla_{\mathbf{Q}_i} d_3(\cdot)\Big] \\
&= -\sum_{j<k<l}^{M}\sum_{l}^{\mathbf{n_p}^*}\alpha_l^*\sum_{z}^{3}\frac{\partial k^*(\mathbf{x}_l,(d_1,d_2,d_3))}{\partial d_z}\nabla_{\mathbf{Q}_i} d_z(\cdot)
\end{aligned}
$$

Application of Gaussian Process Regression - Both Descriptors III

Minimization problem

$$\min_{\hat{\mathbf{a}},\mathbf{a}^*} \frac{1}{n_s}\sum_{s=1}^{n_s}\sum_{l=1}^{M}||(\mathbf{F}(\mathbf{Q}_l^s) - \sum_{j<k}^{M}\sum_{i=1}^{n_p}\hat{a}_i k(x_i, r_{jk}^s)(-\nabla_{\mathbf{Q}_l} r_{jk}^s) - \sum_{j<k<l}^{M}\sum_{\mathbf{i}}^{\mathbf{n_p}^*}\alpha_{\mathbf{i}}^* \sum_{z}^{3}\frac{\partial k^*(\mathbf{x_i}, (d_1(\cdot), d_2(\cdot), d_3(\cdot)))}{\partial d_z(\cdot)}(-\nabla_{\mathbf{Q}_l} d_z(\mathbf{Q}_j^s, \mathbf{Q}_k^s, \mathbf{Q}_l^s))||_3^2$$

Thus, we solve the corresponding equation

$$\mathbf{y} = \mathbf{K}\mathbf{a}, \quad \mathbf{K} = [\hat{\mathbf{K}} \quad \mathbf{K}^*] \text{ and } \mathbf{a} = \begin{bmatrix}\hat{\mathbf{a}} \\ \mathbf{a}^*\end{bmatrix}$$

where

- $\mathbf{y} \in \mathbb{R}^{3M\cdot n_s}$
- $\mathbf{a} \in \mathbb{R}^{n_p + {n_p}^*}$
- $\mathbf{K} \in \mathbb{R}^{(3M\cdot n_s)\times(n_p+{n_p}^*)}$

The normal equations

A linear problem arises in both cases.

$$\mathbf{F} = \mathbf{K} \cdot \mathbf{a}$$

Using the normal equations

$$\mathbf{F}^{tr}\mathbf{F} = \mathbf{F}^{tr}\mathbf{K} \cdot \mathbf{a}$$

Goal: to minimize the $g(\mathbf{a}) = ||\mathbf{F} - \mathbf{K}\mathbf{a}||_2^2 = (\mathbf{F} - \mathbf{K}\mathbf{a})^{tr}(\mathbf{F} - \mathbf{K}\mathbf{a})$ Thus, Thus,

$$\begin{aligned} \nabla g(\mathbf{a}) &= 2\mathbf{K}^{tr}\mathbf{K}\mathbf{a} - 2\mathbf{K}^{tr}\mathbf{F} = 0 \\ \mathbf{K}^{tr}\mathbf{K}\mathbf{a} &= \mathbf{K}^{tr}\mathbf{F} \end{aligned}$$

The regularization parameter

$$(\mathbf{K}^{tr} \cdot \mathbf{K} + \lambda \mathbf{I}) \cdot \mathbf{a} = \mathbf{K}^{tr} \cdot \mathbf{F}$$

The regularization matrix

$$(\mathbf{K}^{tr} \cdot \mathbf{K} + \lambda \mathbf{K}_{n_p n_p}) \cdot \mathbf{a} = \mathbf{K}^{tr} \cdot \mathbf{F}$$

Hyperparameters tuning

There are three approaches to learning the hyperparameters:

- a grid search,
- maximizing the likelihood and
- with a Bayesian approach.

For our project, we applied the grid search.

For the model with only the dimer descriptor:

- Number of basis points: 15, 20, 48.
- Parameter δ : 1.0, 2.0, 6.0.
- Parameter θ : 2.5, 3.5, 5.0.
- Parameter λ : 0.0, 0.1, 0.01, 0.001.

Other representations of the pair potential

Lennard-Jones Model

The Lennard-Jones potential models soft repulsive and attractive interactions. This is a straightforward, parametric way to approximate the total force of the system.

$$u_{LJ}(x) = 4\epsilon\Big[\Big(\frac{a_0}{x}\Big)^{12} - \Big(\frac{a_1}{x}\Big)^{6}\Big]$$

$$\mathbf{F}_i = -\nabla_{\mathbf{Q}_i} U_{CG}(\mathcal{Q};\mathbf{a}) = -\sum_{j=1}^{M}\sum_{k=j+1}^{M} \frac{du_{LJ}(D(\mathbf{Q}_j,\mathbf{Q}_k);\mathbf{a})}{dD}\nabla_{\mathbf{Q}_i} D(\mathbf{Q}_j,\mathbf{Q}_k)$$

Linear splines Basis Model

A spline model is referred to as a representation of the pair potential $u(x)$ with piecewise linear functions,

$$g_i(x) = \begin{cases} \frac{x-x_{i-1}}{\Delta x} & x_{i-1} < x \leq x_i \\ \frac{x_{i+1}-x}{\Delta x} & x_i < x \leq x_{i+1} \end{cases}$$

where x_i is a basis point, $\Delta x = x_{i+1} - x_i$ is the distance between two consecutive basis points and x is the distance of two CG particles.

$$u(x) = \sum_{l}^{n_p} \alpha_l g_l(x)$$

$$\mathbf{F}_i = -\nabla_{\mathbf{Q}_i} V(\mathcal{Q};\mathbf{a}) = -\sum_{j=1}^{M}\sum_{k=j+1}^{M}\sum_{l}^{n_p} \alpha_l g_l(|\mathbf{Q}_j - \mathbf{Q}_k|)\nabla_{\mathbf{Q}_i} D(\mathbf{Q}_j,\mathbf{Q}_k)$$

Quantities to check the quality of the model

Wasserstein metric

Let μ, ν denote two probability measures on Ω, a measurable space with σ-algebra. For $\Omega = \mathbb{R}$, if $\mathcal{H}, \mathcal{G}$ are the distribution functions of μ, ν, respectively, the Kantorovich metric is defined by

$$d_W(\mu, \nu) := \int_{-\infty}^{\infty} |\mathcal{H}(x) - \mathcal{G}(x)| dx = \int_0^1 |\mathcal{H}^{-1}(t) - \mathcal{G}^{-1}(t)| dt$$

If the metric space is separable, then the previous equation is equivalent to

$$d_W(\mu, \nu) := \sup\{| \int z d\mu - \int z d\nu| : ||z||_L \leq 1\},$$

the supremum being taken over all z satisfying the Lipschitz condition $|z(x) - z(y)| \leq d(x, y)$ where d is the metric on $\mathbb{R}$.

Chi-squared error

The form of the χ^2-error is adjusted for the number of coordinates, samples, and particles. It is defined as:

$$\chi^2 = \frac{1}{3 * M * n_s} ||(\mathbf{F} o \Pi)(\cdot) - \mathbf{F}_{CG}(\cdot; \mathbf{a})||^2 = \frac{1}{3 * M * n_s} ((\mathbf{F} o \Pi)(\cdot) - \mathbf{F}_{CG}(\cdot; \mathbf{a}))^{tr} ((\mathbf{F} o \Pi)(\cdot) - \mathbf{F}_{CG}(\cdot; \mathbf{a}))$$

Bulk methane system

- A system of 512 CH_4
- At the NVT ensemble at 100K, density $\rho = 0.38 g/cm^3$,
- Cut-off distance of 10 Angstroms
- 7400 frames (observations) were collected

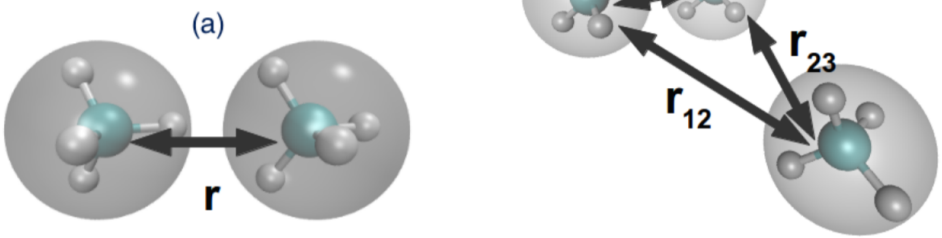


Figure: Example of methane molecules in atomistic and CG description

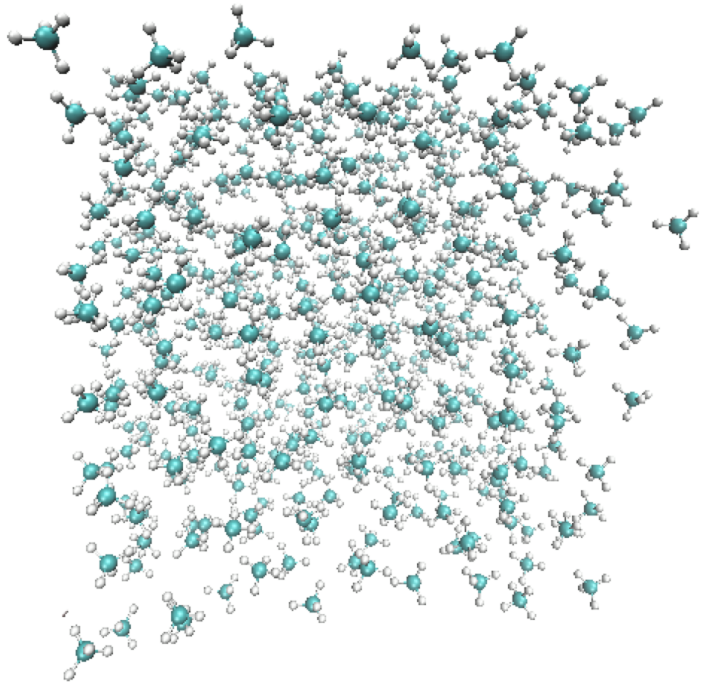
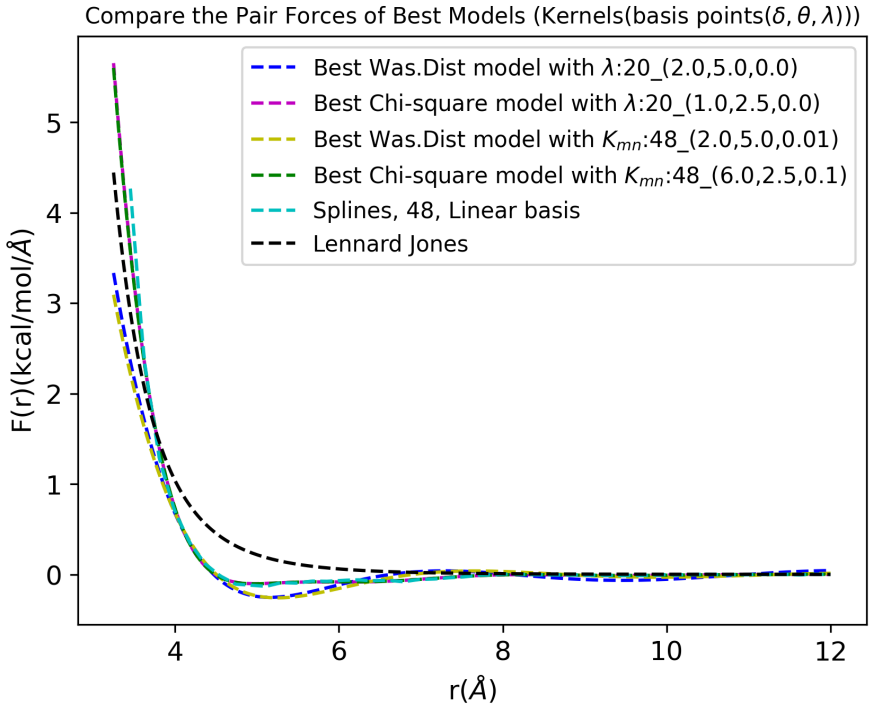


Figure: Methane system

The simulations were done by Antonis Chazirakis.

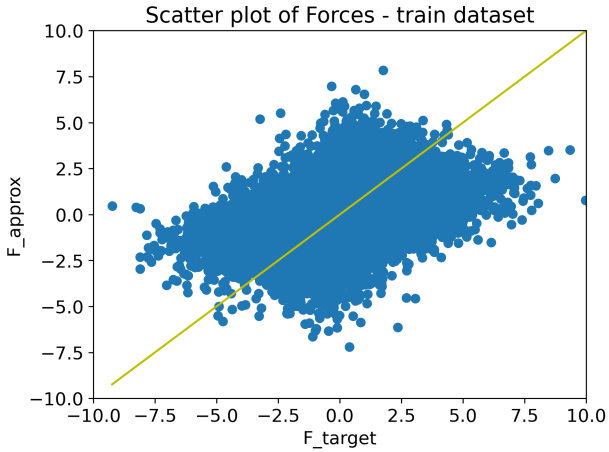
Results with Dimer Descriptor (only) I

- 100 samples (90% training set - 10% testing set)
- Models with minimum Wasserstein value, χ^2 seem to increase quite a bit
- According to the figure, the initial values differ because in this region we have little data

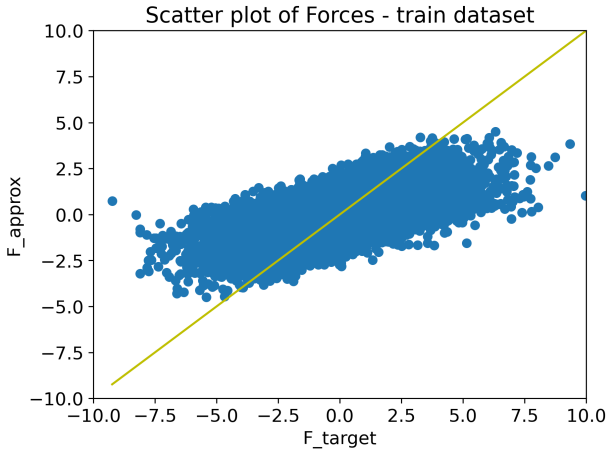


(basis points, δ, θ, λ)	Wasserstein train	Wasserstein test	Chi-Square train	Chi-Square test
λI (20 2.0 5.0 0.0)	0.09	0.10	1.33	1.36
λI (20 1.0 2.5 0.0)	0.23	0.24	0.77	0.80
K_{mn} (48 2.0 5.0 0.01)	0.07	0.07	1.73	1.73
K_{mn} (48 6.0 2.5 0.1)	0.23	0.24	0.77	0.80
Splines 48	0.23	0.25	0.76	0.80
LJ	0.25	0.27	0.78	0.81

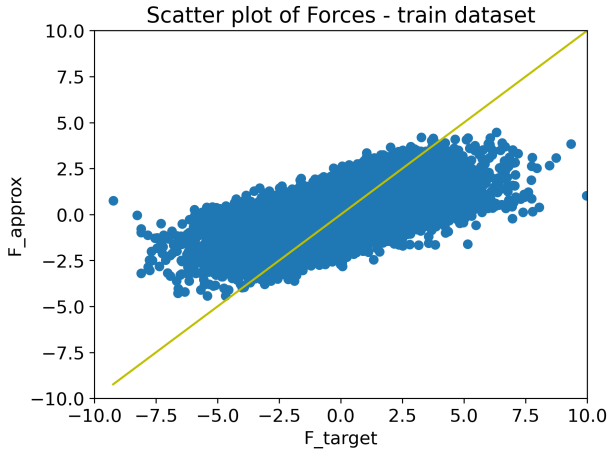
Results with Dimer Descriptor (only) - Scatter plots



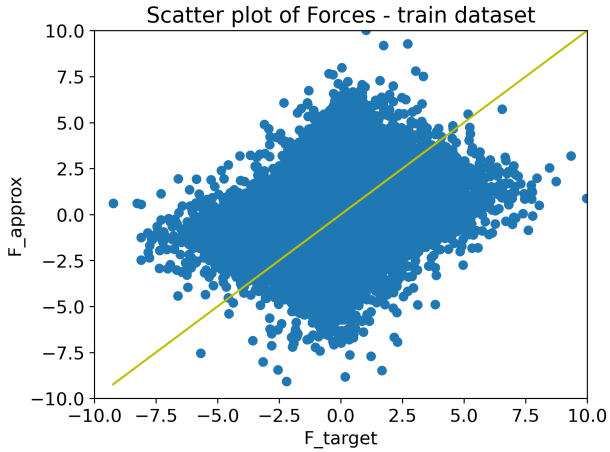
(a) λI (20 2.0 5.0 0.0)



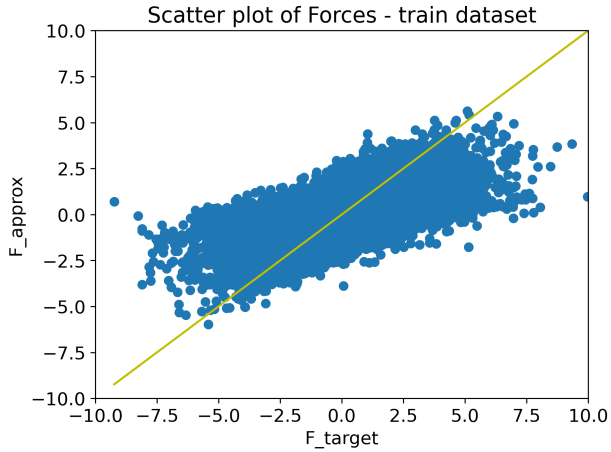
(b) λI (20 1.0 2.5 0.0)



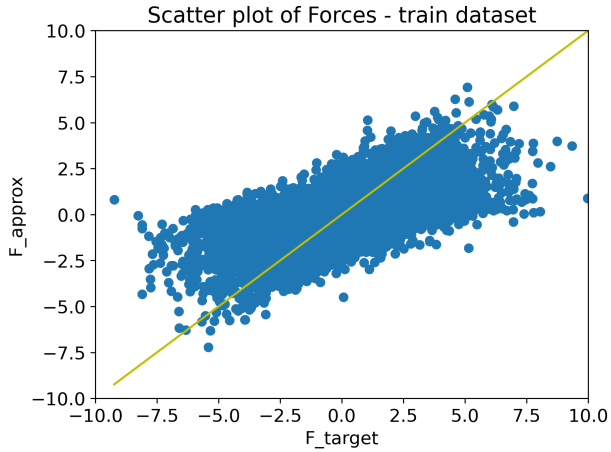
(c) K_{mn} (48 6.0 2.5 0.1)



(d) K_{mn} (48 2.0 5.0 0.01)

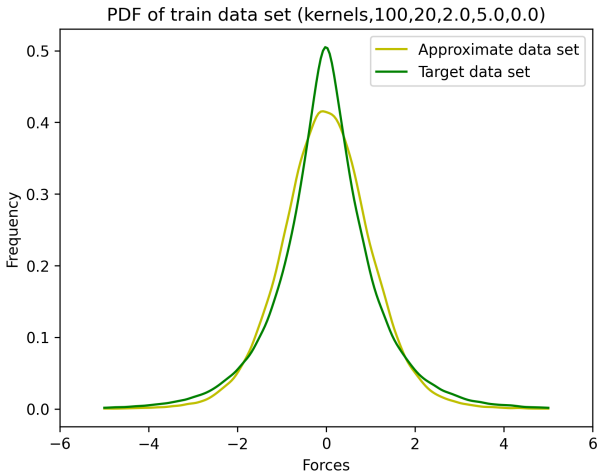


(e) Splines 48

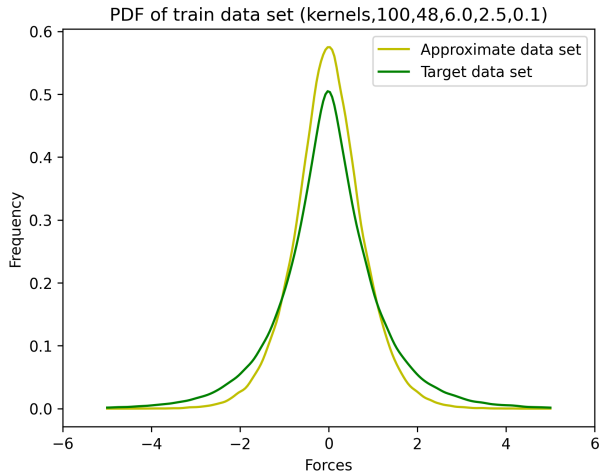


(f) LJ

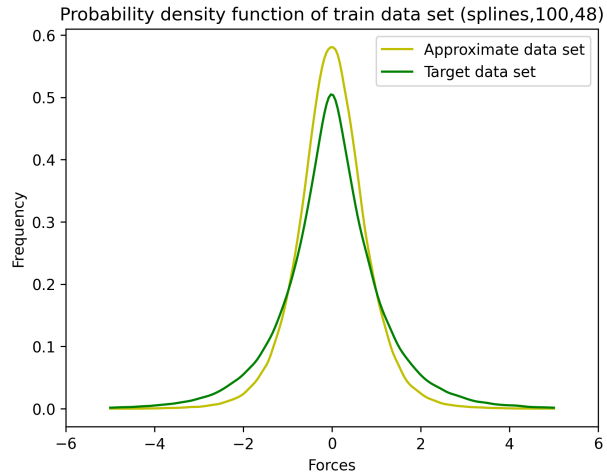
Results with Dimer Descriptor (only) - PDF plots



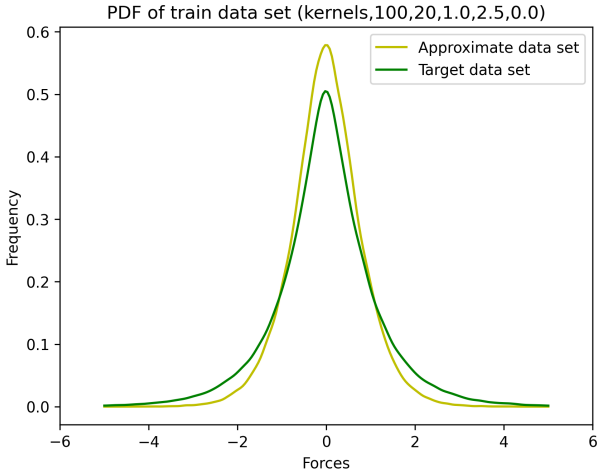
(a) λI (20 2.0 5.0 0.0)



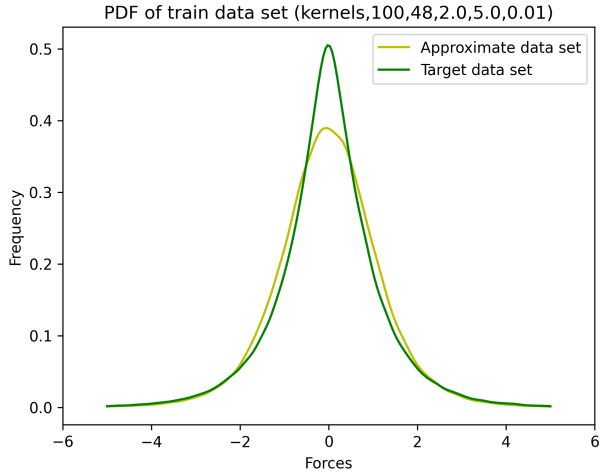
(c) K_{mn} (48 6.0 2.5 0.1)



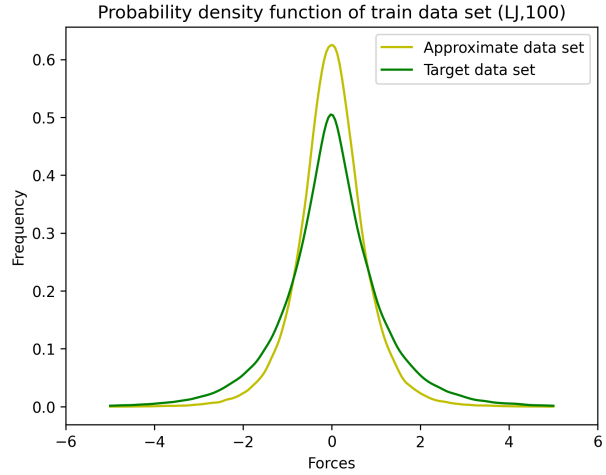
(e) Splines 48



(b) λI (20 1.0 2.5 0.0)



(d) K_{mn} (48 2.0 5.0 0.01)



(f) LJ

Results with Dimer Descriptor (only) IV

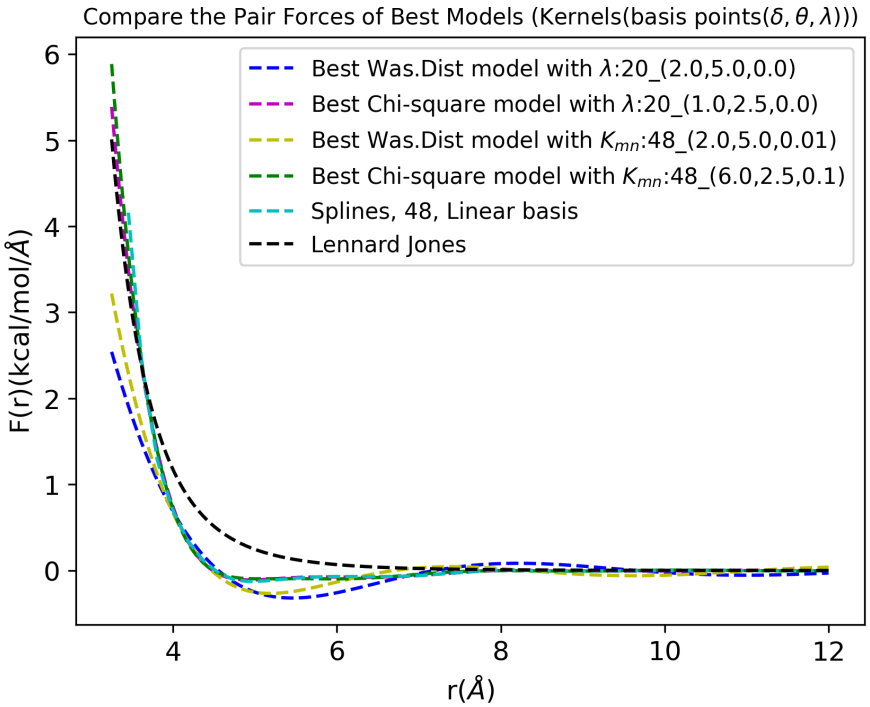
Models with a Regularization parameter for 100 samples

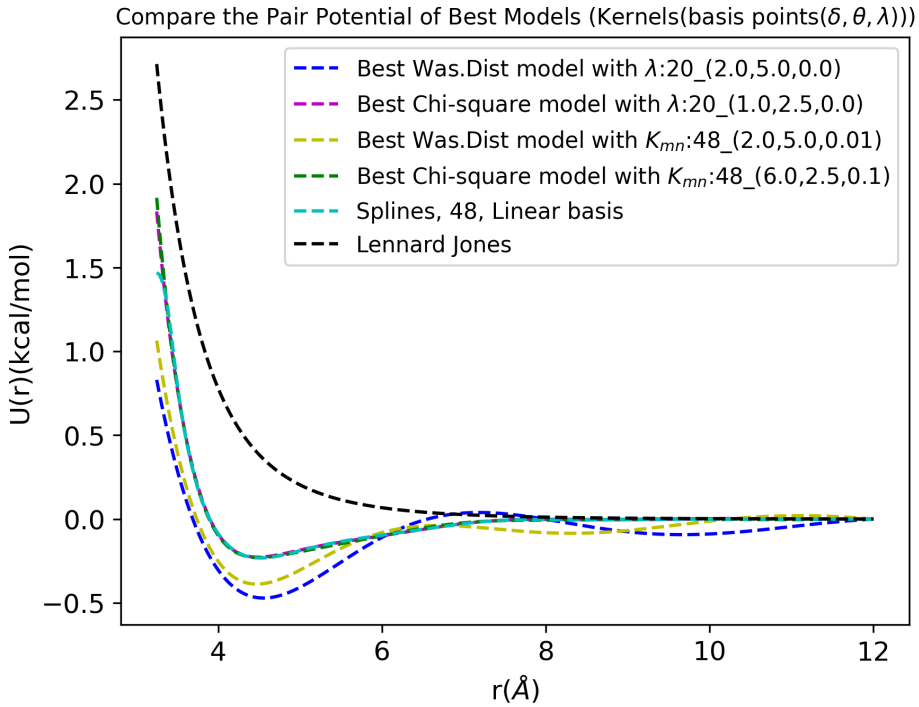
	#basis point	δ	θ	λ	W_{tr}	W_{te}	χ^2_{tr}	χ^2_{te}
W_{tr}	20	2.0	5.0	0.0	0.09	0.10	1.33	1.36
	48	2.0	5.0	0.0	0.14	0.13	2.02	2.01
	20	2.0	3.5	0.0	0.15	0.16	0.92	0.95
	20	1.0	5.0	0.0	0.15	0.16	1.18	1.20
	48	1.0	2.5	0.0	0.16	0.17	0.89	0.92
W_{te}	20	2.0	5.0	0.0	0.09	0.10	1.33	1.36
	48	2.0	5.0	0.0	0.14	0.13	2.02	2.01
	20	2.0	3.5	0.0	0.15	0.16	0.92	0.95
	20	1.0	5.0	0.0	0.15	0.16	1.18	1.20
	48	1.0	2.5	0.0	0.16	0.17	0.89	0.92
χ^2_{tr}	20	1.0	2.5	0.0	0.2318	0.2439	0.7756	0.8078
	48	6.0	2.5	0.0	0.2319	0.2440	0.7758	0.8083
	20	2.0	2.5	0.0	0.2318	0.2437	0.7759	0.8079
	15	6.0	2.5	0.0	0.2311	0.2432	0.7759	0.8081
	15	1.0	2.5	0.0	0.2305	0.2425	0.7762	0.8089
χ^2_{te}	20	1.0	2.5	0.0	0.2318	0.2439	0.7756	0.8078
	20	2.0	2.5	0.0	0.2318	0.2437	0.7759	0.8079
	15	6.0	2.5	0.0	0.2311	0.2432	0.7759	0.8081
	48	6.0	2.5	0.0	0.2319	0.2440	0.7758	0.8083
	15	1.0	2.5	0.0	0.2305	0.2425	0.7762	0.8089

Models with a Regularization matrix for 100 samples

	# basis point	δ	θ	λ	W_{tr}	W_{te}	χ^2_{tr}	χ^2_{te}
W_{tr}	48	2.0	5.0	0.01	0.07	0.07	1.73	1.73
	48	1.0	3.5	0.001	0.09	0.09	1.20	1.24
	20	2.0	5.0	0.0	0.09	0.10	1.33	1.36
	48	6.0	3.5	0.1	0.11	0.12	1.08	1.09
	48	2.0	2.5	0.001	0.12	0.12	1.06	1.10
W_{te}	48	2.0	5.0	0.01	0.07	0.07	1.73	1.73
	48	1.0	3.5	0.001	0.09	0.09	1.20	1.24
	20	2.0	5.0	0.0	0.09	0.10	1.33	1.36
	48	2.0	2.5	0.001	0.12	0.12	1.06	1.10
	48	6.0	3.5	0.1	0.11	0.12	1.08	1.09
χ^2_{tr}	48	6.0	2.5	0.1	0.2318	0.2440	0.7752	0.8077
	48	6.0	2.5	0.001	0.2319	0.2444	0.7755	0.8073
	20	1.0	2.5	0.0	0.2318	0.2439	0.7756	0.8078
	48	6.0	2.5	0.0	0.2319	0.2440	0.7758	0.8083
	20	1.0	2.5	0.001	0.2317	0.2437	0.7758	0.8078
χ^2_{te}	48	6.0	2.5	0.001	0.2319	0.2444	0.7755	0.8073
	48	6.0	2.5	0.1	0.2378	0.2440	0.7752	0.8077
	20	1.0	2.5	0.0	0.2318	0.2439	0.7756	0.80783
	20	1.0	2.5	0.001	0.2317	0.2437	0.7758	0.80784
	20	1.0	2.5	0.01	0.2318	0.2436	0.7759	0.80786

Results with Dimer Descriptor - 4000 samples





Method & Samples	Wasserstein train	Wasserstein test	Chi-Square train	Chi-Square test
λ**I** (20 2.0 5.0 0.0)	0.27	0.27	0.97	0.97
λ**I** (20 1.0 2.5 0.0)	0.23	0.23	0.78	0.78
K_{mn} (48 2.0 5.0 0.01)	0.23	0.22	0.89	0.89
K_{mn} (48 6.0 2.5 0.1)	0.23	0.23	0.78	0.78
Splines 48	0.24	0.23	0.77	0.77
LJ	0.26	0.26	0.79	0.78

The implementation of the two models was accomplished in Python. The codes are linked in https://github.com/VassiaKyr/Master_Thesis.git.

Conclusion and Future Work

Conclusion

- Gaussian kernel models produce results of similar quality to the splines method
- In a simple molecular system like the methane one, Gaussian and Splines models require an almost equal computational cost
- In the Gaussian method, the set of hyperparameters is a determining factor
- The models with a minimum value of Chi-square error seem to have better results.
- The preparation of the trimer Kernel matrix costs computationally.

Future Work

- Learning the hyperparameters by maximizing the likelihood function
- Except for the Wasserstein metric we can test the results for another probability density distance
- Test our models for more samples and also for more complex systems
- Include the three-body interactions and apply a cut-off function to reduce the complexity of the algorithm

Bibliography

- "The geometry of generalized force matching and related information metrics in coarse-graining of molecular systems", "E. Kalligiannaki and V. Harmandaris and M.A. Katsoulakis and P. Plech", 2015
- "Gaussian process regression for materials and molecules", Deringer, Volker L and Bartók, Albert P and Bernstein, Noam and Wilkins, David M and Ceriotti, Michele and Csányi, Gábor, 2021
- "Many-body coarse-grained interactions using Gaussian approximation potentials", John, ST and Csányi, Gábor., 2017
- "A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions", Schulz, Eric and Speekenbrink, Maarten and Krause, Andreas, 2018
- "Pattern recognition and machine learning", Bishop, Christopher M and Nasrabadi, Nasser M, 2006
- "Neural Network Potential Surfaces: A Comparison of two Approaches", Chazirakis, Anthony and Kirieri, Vassia and Sarris, Ilias-Marios and Kalligiannaki, Evangelia and Harmandaris, Vagelis, 2020

Acknowledgements

Thank you all!

This project has received funding from the Hellenic Foundation for Research and Innovation (HFRI) and the General Secretariat for Research and Technology (GSRT), under grant agreement No [52], SISDECS, and from the project "Computational Modeling of Complex Molecular Systems" funded by Goodyear.