POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Reinforcement Learning Force Field

Laurea Magistrale in Computer Engineering - Ingegneria Informatica

Author: Alireza Ketabdari

Advisor: Prof. Marco Masseroli

Co-advisor: Prof. Modesto Orozco and Dr. Milosz Wieczor

Academic year: 2022-2023

1. Introduction

Molecular Dynamics (MD) simulations play a pivotal role in numerous scientific domains, with the accuracy of these simulations being largely dependent on the precision of Force Field (FF) parameters. Traditional optimization methods for these parameters, such as Gradient Descent (GD), often encounter challenges like getting stuck in local minima and difficulties in managing high-dimensional parameter spaces [1].
This thesis introduces a groundbreaking Reinforcement Learning (RL) technique using the Linear Q-function Approximation (LQFA) to optimize FF equations. Unlike conventional Q-learning that uses extensive Q-tables, the LQFA method employs a linear function to approximate Q-values, effectively addressing the computational challenges of vast state and action spaces. The approach hinges on the weight matrix initialization, ensuring the algorithm's efficiency and capturing state-action relationships effectively [2] [3].
To simplify the computational landscape, the methodology reduces the dimensionality of the parameter space, focusing primarily on sigma and epsilon parameters and limiting the number of atom types. The parameter space is visualized as an N-dimensional Cartesian coordinate system, structured in a grid-like manner. The results of this research highlighted a significant improvement in the helicity of the Alanine Oligopeptide (20-mer) (see the Figures 1 and 2), an indicator of its propensity to adopt a helical conformation. A peak helicity nearing 9 was observed, indicating the effectiveness of the LQFA approach. However, the maximal helicity value for this peptide is 16 , representing an ideal state of parameter optimization. This research, while noteworthy, was executed under specific conditions to ease computational demands. Future developments and refinements are discussed, aiming to achieve closer to the maximal helicity and fully realize the peptide's structural potential [4].
The algorithm's efficacy was further validated using real-world data, with the system's helicity being the primary evaluation metric. While the examples provided were limited to four dimensions, the algorithm inherently can handle even larger dimensional spaces, with computational processing time being the primary constraint. In conclusion, this thesis offers a novel approach to optimizing FF parameters in MD simulations using RL and LQFA, showcasing promising results and setting the stage for future advancements in the field.

2. Preliminary: Molecular Dynamics

Molecular dynamics (MD) simulations are computational tools used to simulate the motion and behavior of atoms and molecules by solving their equations of motion. These simulations are pivotal in fields like biochemistry and drug discovery, aiding in understanding the structural dynamics of biomolecules, their folding pathways, and interactions [5]. The primary objective of this project is to enhance MD simulations by combining top-down and bottom-up approaches, especially when experimental data doesn't align with atomistic trajectories. A typical MD simulation involves several steps [5]:

1. Initial Geometries: Using databases or molecular modeling software to obtain or create molecular structures.
2. Define Inter-Atomic Forces: Examining forces through Force Field (FF) equations.
3. Simulation Box Setup: Defining a volume to enclose the system, often using periodic boundary conditions.
4. Energy Minimization: Adjusting atomic positions to achieve a low-energy configuration.
5. Equilibration: Allowing the system to stabilize before data collection.
6. Production: Main simulation phase to gather data for analysis.
7. Analysis: Processing and interpreting simulation data to gain insights.

Among these, defining inter-atomic forces (Step 2) and the analysis phase (Step 7) present unique challenges. The former involves the intricate task of selecting appropriate FF parameters and equations, while the latter demands the processing and interpretation of vast amounts of raw data. Force Field (FF) in molecular modeling is a mathematical model that describes interatomic interactions during an MD simulation. It encompasses both functional forms and parameter sets used to calculate a system's potential energy. The accuracy and reliability of MD simulations heavily depend on the FF, making it a critical component. In MD simulations, interactions are categorized into bonded and non-bonded types. Bonded interactions include bond stretching, angle bending, and torsional rotation. Non-bonded interactions cover van der Waals forces and electrostatic interactions. The selection of FF parameters is complex due to the vast parameter space. To address this, the project focuses on a subset of the most significant parameters during simulations. This research specifically investigates the parameters sigma (σ) and epsilon (ϵ) [6]. To streamline parameter selection, sensitivity analysis is employed using the ThermoDiff library in Python. This technique calculates derivatives of free energy concerning specific FF parameters. The methodology produces a 'sensitivity matrix' that guides the modification of model attributes in line with existing data [7]. In conclusion, this thesis delves deep into the intricacies of MD simulations, emphasizing the importance of Force Field parameters and the challenges associated with their optimization. The research aims to enhance the accuracy and reliability of MD simulations, paving the way for more precise molecular studies in various scientific domains. Total Energy [8]:

$$E = E_{\text{bonded}} + E_{\text{non-bonded}} \tag{1}$$

Bonded Interactions:

$$\begin{aligned} E_{\text{bonded}} = &\sum_{\text{bonds}} K_b(r - r_0)^2 \\ &+ \sum_{\text{angles}} K_a(\theta - \theta_0)^2 \\ &+ \sum_{\text{torsions}} \frac{V_n}{2}[1 + \cos(n\phi - \gamma)] \end{aligned} \tag{2}$$

Non-bonded Interactions:

$$\begin{aligned} E_{\text{non-bonded}} = &\sum_{\text{pairs}} 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] \\ &+ \sum_{\text{pairs}} q_i q_j \frac{1}{4\pi\epsilon_0}\frac{1}{r_{ij}} \end{aligned} \tag{3}$$

Parameters (Constants):

- **Bonded Interactions**:
  - K_b: Bond force constant[9].
  - r_0: Equilibrium bond length[9].
  - K_a: Angle force constant[9].
  - θ_0: Equilibrium bond angle[9].
  - V_n: Barrier height for torsional rotation[9].

- n: Multiplicity of the torsional potential[9].
- γ: Phase angle for the torsional potential[9].

- **Non-bonded Interactions**:
  - ϵ_{ij}: Depth of the potential well for van der Waals interactions between atom pairs i and j[9].
  - σ_{ij}: Distance at which the potential energy between atom pairs i and j is zero, representing the position of the potential well[9].
  - q_i, q_j: Partial charges on atoms[9].
  - ϵ_0: Permittivity of free space (a universal constant)[9].

Variables:

- **Bonded Interactions**:
  - r: Current bond length[9].
  - θ: Current bond angle[9].
  - ϕ: Current torsional angle[9].
- **Non-bonded Interactions**:
  - r_{ij}: Current distance between atom pairs i and j[9].

3. Preliminary: Reinforcement Learning

Reinforcement Learning (RL) is a machine learning paradigm that focuses on an agent's interaction with its environment to achieve specific goals. The agent learns through this interaction, aiming to maximize cumulative rewards over time. The RL process involves components such as the agent, environment, state, action, reward, policy, value function, and optionally, a model of the environment. RL has found applications in diverse areas, from robotics to game playing and autonomous vehicles [2].

A significant focus of this thesis is on Q-learning, a type of RL algorithm. Q-learning is an off-policy learning method where the agent learns the optimal value function, known as the Q-function. This function represents the expected cumulative reward for taking a particular action in a given state and then following the optimal policy. The Q-learning process involves initializing a Q-table, balancing exploration and exploitation, selecting actions, and updating Q-values using the Bellman equation [2].

However, for large state and action spaces, maintaining a Q-table becomes computationally challenging. To address this, Linear Q-function Approximation (LQFA) is introduced. LQFA approximates Q-values using a linear function, representing them as a linear combination of feature values associated with state-action pairs. Instead of storing Q-values directly, LQFA uses feature and weight vectors. The Q-value of a state is then calculated as the scalar product of these vectors, as shown in the equation [3]:

$$Q(s,a) = \sum_{i=0}^{n} f_i(s,a) \cdot w_i^a$$

To implement LQFA in RL, two primary modifications are made to the standard algorithm: initialization and update. Initialization involves setting all weights to zero or assigning "good" weights. The update step focuses on adjusting the weights instead of Q-table values. The weight update formula is [3]:

$$w_i^a \leftarrow w_i^a + \alpha \cdot \Big[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \Big] \cdot f_i(s,a)$$

The description of the variables and parameters of the above equations are listed blow:

- w_i^a: Represents the weight associated with feature i for action a. The weight is being updated in both equations.
- α: The learning rate. It determines the step size in the weight update. A value between 0 and 1, it dictates how much of the new weight estimate we adopt.
- r: The immediate reward received after taking action a in state s.
- γ: The discount factor. It's a value between 0 and 1 that determines the present value of future rewards. A value closer to 0 makes the agent "short-sighted" (cares more about immediate rewards), while a value closer to 1 makes the agent "far-sighted" (cares more about future rewards).
- $Q(s,a)$: The Q-value of state s and action a. It represents the expected cumulative reward of taking action a in state s and following the optimal policy thereafter.
- $\max_{a'} Q(s',a')$: Represents the maximum Q-value for the next state s' over all possible actions a'. It's used to estimate the future reward.
- $f_i(s,a)$: A feature function for state s and action a corresponding to feature i. It maps the state-action pair to a real value, which is used in the linear approximation of the Q-value.

- δ: The temporal difference (TD) error. In the context of the given equations, δ is defined as the difference between the estimated Q-value and the updated Q-value based on the received reward and the estimated future reward.

In conclusion, this thesis delves into the intricacies of RL, emphasizing Q-learning and the advantages of LQFA in handling large state and action spaces. The research aims to provide a comprehensive understanding of these techniques and their applicability in real-world scenarios.

4. The Model

The primary objective of this research is to optimize the Force Field (FF) parameters using a Reinforcement Learning (RL) approach. The focus is on the Q-function (QF) method, which is adept at handling multiple parameters and discovering the optimal solution in the solution space. To manage the high dimensionality of the problem, two parameter reduction steps are employed. Initially, the emphasis is on the sigma and epsilon parameters due to their adjustable nature. Subsequently, the number of atom types in the system is reduced using sensitivity analysis with the ThermoDiff library in Python.

To visualize the parameter optimization problem, an analogy to a video game is drawn. The environment, analogous to the game setting, is where the algorithm explores different parameter values. This environment is represented as an N-dimensional Cartesian coordinate system, with each dimension symbolizing a specific parameter. To navigate this space, it's transformed into a grid-like structure, referred to as an ND grid [10]. This discretization ensures a more controlled exploration of parameter values. Each configuration of parameters is termed a 'state'. For every state visited, a simulation runs based on its parameter values. The outcome of this simulation, varying based on the observables considered, serves as the reward. In this research, the helicity calculation of an Alanine Oligopeptide (20-mer) is utilized as a trial reward function. In the broader application, this will be replaced with a measure of agreement between simulation estimates and experimental reference values [11].

The agent in this RL setup is an abstraction of the FF, interacting with the environment by receiving rewards, influencing the Q-function parameters, and deciding subsequent actions based on a defined policy. The goal is to discover optimal parameter values for the FF equation. Actions, in this context, are changes in parameter values. The agent's possible actions are influenced by specific strategies, and these actions can involve moving to a neighboring state's location.

The policy adopted combines a greedy-based action with a 70% probability, Gradient Descent (GD) with a 20% probability, and a random walk with a 10% probability. This policy introduces diverse movements to the agent's decision-making, aiming to avoid over-fitting and enabling exploration of efficient paths in the simulation. By integrating GD, the agent can find local minima in the state space. However, reliance on GD alone is not advisable due to potential challenges like noisy gradients and a highly non-linear reward function with multiple minima [12] [13] [1].

The reward function quantifies the agreement between simulation outcomes and experimental data. It emphasizes larger discrepancies between simulated targets and experimental data, ensuring sensitivity to significant deviations. This function offers a quantitative measure of how closely the simulation aligns with real-world experimental data, making it a valuable tool for refining the simulation's accuracy.

The trajectory duration is a pivotal constant in this research. It balances the need for observing system changes over time while maintaining reasonable simulation lengths. The time constant, indicative of a system's rate of change, is derived from an exponential decay function. This constant signifies the time required for the system to decay to approximately 63.2% of its initial state [14].

The Linear Q-function Approximation (LQFA) is indispensable for managing extensive environments with numerous states. In our execution, we deal with two distinct radii: a global-radius and a local-radius. The global-radius defines the ranges of the steps in each dimension, while the local-radius determines the neighbors impacted by any alterations in the current state [3].

Gradient Descent (GD) is an optimization method employed to find the closest local ex-

trema of multidimensional functions. In the context of molecular simulations, it's used as an iterative optimization algorithm to minimize discrepancies between simulated results and experimental data. To apply GD for parameter optimization, sensitivity analysis is initially conducted. This analysis calculates the local estimate of the gradient, quantifying the impact of individual parameters on the FF equation [13].
Lastly, essential initializations for our simulation process include determining the number of FF parameters, grid size, step size derived from the PDB file analysis, weights matrix, local-radius, and parameters of the Q-function and GD. These initializations are crucial for the effective execution and optimization of the simulation process [15] [16].

5. Results

The primary objective of this thesis was to optimize Force Field (FF) parameters using a Reinforcement Learning (RL) approach. The research journey began with the foundational task of setting up the necessary code for running simulations. This involved creating a trajectory, applying forces to the system, and observing atom responses. Notably, certain atom types, like solvents and ions, were excluded from sensitivity analysis to maintain focus on the molecule's structure and interactions.
The optimization algorithm scaled each gradient component by a learning rate, $-\alpha_{\text{gr}}$, which determined the step size for optimization. This learning rate was pivotal for the speed and accuracy of the algorithm's convergence. The continuous parameter space was discretized into a grid to establish an environment for the RL model. Each grid point represented a state, and the magnitude of actions within the RL model was influenced by the derivative values. Actions were derived from gradients, and a loop was employed to iteratively adjust the FF parameters based on these actions.
A simplified example relying solely on Gradient Descent (GD) was executed to assess its effectiveness (see the Figure 3). The results showed that while GD could guide the agent to a local maximum, the maximum attainable helicity was not reached. The research then transitioned to using artificial rewards in a two-dimensional parameter space, which was later expanded to an N-dimensional space. The use of artificial rewards was to evaluate the performance of the Q-learning algorithm.
The Linear Q-function Approximation (LQFA) played a crucial role in the optimization of FF parameters. The initialization of the weights matrix in LQFA was of paramount importance for the efficacy of the algorithm. The research demonstrated that initializing the weights matrix with high values promoted exploration of various regions of the parameter space.
In the N-dimensional space tests, the agent successfully localized desired areas within approximately 20 iterations. The algorithm effectively distinguished high-potential areas from those with lower potential. The results prepared the groundwork for extending the implementation to accommodate an N-dimensional parameter space.
The final segment of the research evaluated the comprehensive algorithm using real-world data, focusing on the system's helicity as the primary metric. The complete LQFA was employed to identify optimal parameters within an N-dimensional parameter space. The results showed that the agent was capable of identifying areas with high reward potential and distinguishing them from lower potential areas (see the Figure 4).
In conclusion, the research successfully demonstrated the potential of combining Q-learning, GD, and Random moves for optimizing FF parameters. The agent was able to pinpoint and consistently track regions with high reward potentials, making this approach promising for future applications and studies.

6. Figures

7. Conclusions

This research has pioneered the integration of Reinforcement Learning (RL) with the Linear Q-function Approximation (LQFA) to optimize Force Field (FF) parameters in Molecular Dynamics (MD) simulations. Traditional methods, such as Gradient Descent (GD), often faced challenges in high-dimensional parameter spaces. The novel approach introduced in this thesis effectively addresses these challenges by employing LQFA, which uses a linear function to approximate Q-values, thereby managing vast

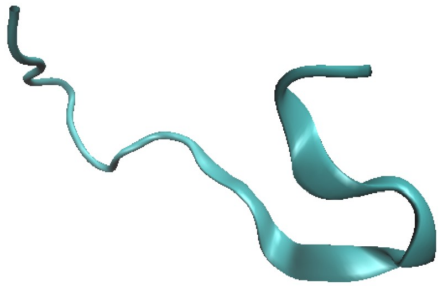

Figure 1: Unhelix structure of Alanine Oligopeptide (20-mer). Helicity is a crucial metric for evaluating a system, as it directly relates to the stability of the Alanine structure. A higher degree of helicity often indicates a more stable conformation of the Alanine Oligopeptide.

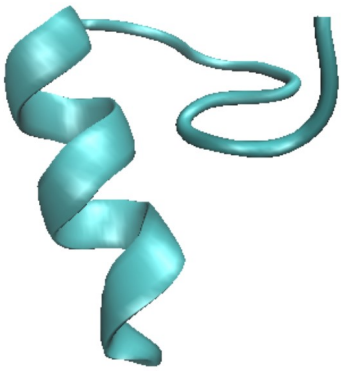

Figure 2: helix structure of Alanine Oligopeptide (20-mer).

state and action spaces efficiently.
The methodology adopted streamlined the computational landscape by reducing the dimensionality of the parameter space, focusing on key parameters like sigma and epsilon. The results showcased a marked improvement in the helicity of the Alanine Oligopeptide (20-mer), a measure of its structural conformation. While the research achieved significant milestones, it also highlighted areas for future exploration and refinement, aiming to achieve optimal helicity values.

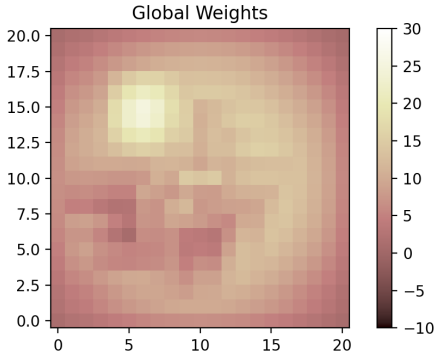

Figure 3: In the aforementioned figure, the outcomes of the agent's exploration-exploitation activities on the Weights matrix associated with each state, is illustrated. Areas depicted in darker shades signify regions with a low probability of yielding high rewards. Conversely, lighter shades indicate regions where there is a higher likelihood of encountering substantial rewards.

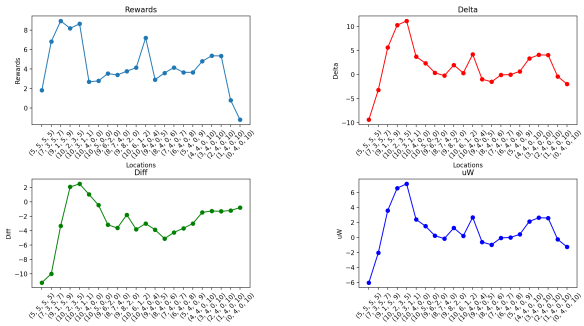

Figure 4: The agent's growth trajectory over four states, achieving values of 6.82, 8.93, 8.17, and 8.64, peaking near 9. This contrasts with GD results from Section 3.1, where non-averaged helicity peaked around 5. While GD alone can lead to local maxima, integrating Q-learning, GD, and Random moves enhances exploration. This is evident as the agent diverges from areas like (7,3,5,7) to explore distant states such as (10,2,3,5) due to random influences. After identifying a peak, the algorithm's knowledge aids in pinpointing other promising regions, as seen with a subsequent peak at (10,6,1,2) with 7.2. The exploration continues, revealing values around 5.2 in states like (5,4,0,9), but a notable drop is observed at (2,4,0,10).

Furthermore, the research emphasized the importance of the initialization of the weights matrix in LQFA and demonstrated its impact on

the algorithm's efficiency. The algorithm's applicability was further validated using real-world data, emphasizing the system's helicity as the primary evaluation metric.
In essence, this thesis has laid a robust foundation for the optimization of FF parameters in MD simulations using a combination of RL, LQFA, and GD. The results obtained are promising, indicating the potential of this approach for future research and applications in the realm of molecular simulations.

References

[1] Pierre Baldi. Gradient descent learning algorithm overview: a general dynamical systems perspective. *IEEE Transactions on Neural Networks, IEEE*, 6(1):182–195, January 1995. doi: 10.1109/72.363438.

[2] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Adaptive computation and machine learning series. The MIT Press, Cambridge, Massachusetts and London, England, 2 edition, 2018. ISBN 9780262039246.

[3] Francisco S. Melo and M. Isabel Ribeiro. *Learning Theory*, chapter Q-Learning with Linear Function Approximation, pages 319–333. Lecture Notes in Artificial Intelligence. Springer, Berlin Heidelberg New York, 2007. ISBN 978-3-540-72925-9.

[4] Terence T. L. Tang and Lori A. Passmore. Recognition of poly(a) rna through its intrinsic helical structure. *Cold Spring Harb Symp Quant Biol*, 84:21–30, 2019. doi: 10.1101/sqb.2019.84.039818. Epub 2020 Apr 15.

[5] Anita Rácz, Levente M. Mihalovits, Dávid Bajusz, Károly Héberger, and Ramón Alain Miranda-Quintana. Molecular dynamics simulations and diversity selection by extended continuous similarity indices. *J. Chem. Inf. Model.*, 62(14):3415–3425, 7 2022.

[6] Sung Bo Hwang, Chang Joon Lee, Sehan Lee, Songling Ma, Young-Mook Kang, Kwang Hwi Cho, Su-Yeon Kim, Oh Young Kwon, Chang No Yoon, Young Kee Kang, Jeong Hyeok Yoon, Ky-Youb Nam, Seong-Gon Kim, Youngyong In, Han Ha Chai, William E. Acree Jr., J. Andrew Grant, Ken D. Gibson, Mu Shik Jhon, Harold A. Scheraga, and Kyoung Tai No. Pmff: Development of a physics-based molecular force field for protein simulation and ligand docking. *J. Phys. Chem. B, American Chemical Society*, 124(6):974–989, 2020. doi: 10.1021/acs.jpcb.9b10339.

[7] Shefov Konstantin Sergeevich and Stepanova Maria M. Sensitivity analysis in a problem of reaxff molecular-dynamic force field optimization. In *Proceedings of the VIII International Conference "Distributed Computing and Grid-technologies in Science and Education" (GRID 2018)*, page 595. St. Petersburg State University, September 2018.

[8] M.A. González. Force fields and molecular dynamics simulations. *Collection SFN, EDP Sciences*, 12:169–200, 2011.

[9] Thijs van Westen, Thijs J. H. Vlugt, and Joachim Gross. Determining force field parameters using a physically based equation of state. *J. Phys. Chem. B, American Chemical Society*, 115(24):7872–7880, 2011. doi: 10.1021/jp2026219.

[10] Martin Roesch, Christian Linder, Roland Zimmermann, Andreas Rudolf, Andrea Hohmann, and Gunther Reinhart. Smart grid for industry using multi-agent reinforcement learning. *Appl. Sci., MDPI*, 10(19):6900, Oct 2020. doi: 10.3390/app10196900.

[11] Chikako T Nakazawa, Atsushi Asano, and Takuzo Kurotsu. Structural studies of amphiphilic oligopeptides composed of alternating alanine and ionizable amino-acid residues using cd and 13c cp/mas nmr spectroscopy. *Polymer Journal, Nature Publishing Group*, 44:882–887, June 2012.

[12] Hariharan N and Paavai Anand G. A brief study of deep reinforcement learning with epsilon-greedy exploration. *International Journal of Computing and Digital Systems*, 11(1), 1 2022. ISSN 2210-142X. doi: 10.12785/ijcds/110144.

[13] Chuanlei Zhang, Minda Yao, Wei Chen, Shanwen Zhang, Dufeng Chen, and Yuliang Wu. Gradient descent optimization in deep learning model training based on multistage and method combination strategy. *Security and Communication Networks, Hindawi*, 2021:1–15, 7 2021. doi: 10.1155/2021/9956773. Academic Editor: Chi-Hua Chen.

[14] Science News Research Reviews Encyclopedia. Time constant. URL `https://academic-accelerator.com/encyclopedia/time-constant`.

[15] Yong Song, Yi bin Li, Cai hong Li, and Gui fang Zhang. An efficient initialization approach of q-learning for mobile robots. *International Journal of Control, Automation and Systems, Springer*, 10:166–172, February 2012.

[16] Xi Min Zhang, Yan Qiu Chen, Nirwan Ansari, and Yun Q. Shi. Mini-max initialization for function approximation. *Neurocomputing, Elsevier*, 57:389–409, Mar 2004.