2017

Accelerating Quantum Monte Carlo via Graphics Processing Units

Benjamin Evert Himberg
University of Vermont

Follow this and additional works at: http://scholarworks.uvm.edu/graddis

Part of the Condensed Matter Physics Commons

Recommended Citation
Himberg, Benjamin Evert, "Accelerating Quantum Monte Carlo via Graphics Processing Units" (2017). Graduate College Dissertations and Theses. 728.
http://scholarworks.uvm.edu/graddis/728

This Thesis is brought to you for free and open access by the Dissertations and Theses at ScholarWorks @ UVM. It has been accepted for inclusion in Graduate College Dissertations and Theses by an authorized administrator of ScholarWorks @ UVM. For more information, please contact donna.omalley@uvm.edu.
ACCELERATING QUANTUM MONTE CARLO VIA GRAPHICS PROCESSING UNITS

A Thesis Presented

by

Benjamin E. Himberg

to

The Faculty of the Graduate College

of

The University of Vermont

In Partial Fulfillment of the Requirements
for the Degree of Master of Science
Specializing in Physics

May, 2017

Defense Date: March 20th, 2017
Thesis Examination Committee:

Adrian Del Maestro, Ph.D., Advisor
Taras Lakoba, Ph.D., Chairperson
Dennis Clougherty, Ph.D.
Cynthia J. Forehand, Ph.D., Dean of the Graduate College
Abstract

An exact quantum Monte Carlo algorithm for interacting particles in the spatial continuum is extended to exploit the massive parallelism offered by graphics processing units. Its efficacy is tested on the Calogero-Sutherland model describing a system of bosons interacting in one spatial dimension via an inverse square law. Due to the long range nature of the interactions, this model has proved difficult to simulate via conventional path integral Monte Carlo methods running on conventional processors.

Using Graphics Processing Units, optimal speedup factors of up to 640 times are obtained for $N = 126$ particles. The known results for the ground state energy are confirmed and, for the first time, the effects of thermal fluctuations at finite temperature are explored.
To my parents, who taught me to ask uncomfortable questions and accept unexpected answers... you are the reason I’ve set myself down this path, and your encouragement has kept me on it even when I thought all was lost.
ACKNOWLEDGEMENTS

Foremost in my mind would be my advisor, Professor Del Maestro. I would like to thank you for the potential you saw in me and your encouragement to apply to the program. I would not have been accepted without your support, and your continued criticisms have made me a better physicist. Working with you continues to be a pleasure and most certainly has improved my skills as a programmer as well. Thank you for the opportunities you’ve provided, including a very interesting project.

Professor Clougherty, I’d like to thank you for keeping me honest, reminding me that I am a physicist first and that programming is simply a tool. While our discussion after my proposal was painful, I recognize it as necessary and accept pain as often the precursor to growth. I could do far worse than following your advice and I hope you recognize the changes you’ve encouraged in me.

I can think of no better person to act as chair of my committee than Professor Lakoba. You’ve shown yourself capable of remaining unbiased and speaking your mind. I sincerely thank you for your time and efforts with respect to my participation in the masters program.

Finally I’d like to thank two of my fellow graduate students who may never read this: Noah Wilson and Sanghita Sengupta. Noah shared an office with me for two years, putting up with my ramblings and collaborating on many a homework assignment. Sanghita was there for me when I began studying for my comprehensive, encouraging me to press on while also causing me to reevaluate my priorities.

This research was supported in part by the National Science Foundation under Award No. DMR-1553991.
### A Exactly Solvable Models

- A.1 Calogero’s Two-Body Model ........................................... 77
- A.2 Calogero’s Three-Body Model ........................................ 84
- A.3 Sutherland N-Body Model ............................................. 94
  - A.3.1 \( N = 2 \) ................................................................. 95
  - A.3.2 \( N = 3 \) ................................................................. 97
  - A.3.3 \( N = 4 \) ................................................................. 101
  - A.3.4 N Particles ........................................................... 102

### B Parallelization and GPUs

- B.1 Differences between a CPU and GPU ............................. 103
  - B.1.1 Background ......................................................... 104
  - B.1.2 Modern Hardware .................................................. 105
- B.2 OpenCL Core Concepts .................................................. 106
- B.3 Amdahl’s Law and GPU Performance ............................... 109
- B.4 Path Integral Monte Carlo on a GPU ............................... 111
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Serial evaluation of an N-particle pair interaction.</td>
<td>5</td>
</tr>
<tr>
<td>1.2</td>
<td>Parallel evaluation of an N-particle pair interaction.</td>
<td>6</td>
</tr>
<tr>
<td>2.1</td>
<td>Schematic of the N-body Calogero-Sutherland model.</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>Contour integral related to the periodic Sutherland potential.</td>
<td>11</td>
</tr>
<tr>
<td>3.1</td>
<td>Configuration of particles in the position basis.</td>
<td>21</td>
</tr>
<tr>
<td>3.2</td>
<td>Representation of the partition function in the position basis.</td>
<td>24</td>
</tr>
<tr>
<td>3.3</td>
<td>Particle in a box with periodic boundary conditions.</td>
<td>27</td>
</tr>
<tr>
<td>3.4</td>
<td>A one-dimensional model mapped to a torus.</td>
<td>30</td>
</tr>
<tr>
<td>3.5</td>
<td>Permutations of worldlines representing Bose statistics.</td>
<td>31</td>
</tr>
<tr>
<td>3.6</td>
<td>Illustration of a Markov chain.</td>
<td>35</td>
</tr>
<tr>
<td>4.1</td>
<td>A current state and center of mass generated state.</td>
<td>42</td>
</tr>
<tr>
<td>4.2</td>
<td>Slice of a current state and generated center of mass state.</td>
<td>43</td>
</tr>
<tr>
<td>4.3</td>
<td>Flow chart for the primitive action calculation on a CPU.</td>
<td>46</td>
</tr>
<tr>
<td>4.4</td>
<td>Flow chart for the primitive action calculation on a GPU.</td>
<td>47</td>
</tr>
<tr>
<td>4.5</td>
<td>Code for the fourth order potential action computation on a GPU.</td>
<td>49</td>
</tr>
<tr>
<td>4.6</td>
<td>Code for the fourth order potential action computation on a GPU, continued.</td>
<td>50</td>
</tr>
<tr>
<td>4.7</td>
<td>Code for the fourth order potential action computation on a GPU, final.</td>
<td>51</td>
</tr>
<tr>
<td>5.1</td>
<td>Imaginary time ($\tau$) scaling for the harmonic oscillator.</td>
<td>54</td>
</tr>
<tr>
<td>5.2</td>
<td>Temperature dependent results for the harmonic oscillator.</td>
<td>55</td>
</tr>
<tr>
<td>5.3</td>
<td>free bose gas temperature dependence.</td>
<td>57</td>
</tr>
<tr>
<td>5.4</td>
<td>Speedup factor demonstrated to more than 600x the speed of a single CPU core.</td>
<td>58</td>
</tr>
<tr>
<td>5.5</td>
<td>A comparison of runtime between the entire VACC and a single GPU.</td>
<td>59</td>
</tr>
<tr>
<td>5.6</td>
<td>Calogero-Sutherland model $\tau$ scaling for $\lambda_s = 4.0$, including running averages.</td>
<td>62</td>
</tr>
<tr>
<td>5.7</td>
<td>Calogero-Sutherland model $\tau$ scaling for $\lambda_s = 3.0$, including running averages.</td>
<td>63</td>
</tr>
<tr>
<td>5.8</td>
<td>Calogero-Sutherland model $\tau$ scaling for $\lambda_s = 2.0$, including running averages.</td>
<td>64</td>
</tr>
<tr>
<td>5.9</td>
<td>Calogero-Sutherland model $\tau$ scaling for $\lambda_s = 1.5$.</td>
<td>65</td>
</tr>
<tr>
<td>5.10</td>
<td>Asymptotic behavior of the Calogero-Sutherland model as the thermodynamic limit is approached.</td>
<td>67</td>
</tr>
<tr>
<td>5.11</td>
<td>Temperature dependent results for the Calogero-Sutherland model.</td>
<td>68</td>
</tr>
</tbody>
</table>
5.12 Energy dependence on interaction strength of the Calogero-Sutherland model at $T = 0.1\, \text{K}$. 

5.13 Energy dependence on interaction strength of the Calogero-Sutherland model at $T = 4.9\, \text{K}$. 

A.1 Schematic of the Calogero two-body model. 

A.2 Schematic of the Calogero three-body model. 

A.3 Schematic of the Sutherland three-body model. 

B.1 An intel 486 DX4. 

B.2 Two 3dfx Voodoo 2 graphics cards in SLI. 

B.3 Example of CPU code which is easily parallelized. 

B.4 CPU code which has been adapted to run on the GPU. 

B.5 OpenCL memory hierarchy. 

B.6 Representation of a workspace as a volume. 

B.7 Example of a PIMC state configuration.
LIST OF TABLES

5.1 Calogero-Sutherland model selected $\tau$ values per $\lambda_s$. . . . . . . . . . . . . . . . . 66
5.2 Calogero-Sutherland model comparison of energies. . . . . . . . . . . . . . . . . . . . . . . 70
B.1 Comparison of modern CPU and GPU hardware. . . . . . . . . . . . . . . . . . . . . . . . 105
Chapter 1

Introduction

While in principle a Hamiltonian may be written down for a many-body model, solving it for more than two particles is often a very difficult task, if even possible. This is what makes one-dimensional problems interesting: some are exactly solvable for an arbitrary number of particles N, such as the Ising Chain [1] or Lieb-Liniger [2] models. While often these solutions yield only the ground state energies, they are still celebrated due to their rarity. There are so many models which aren’t well understood that a many-body model with an exact solution, even if limited, may provide insight into such unsolved models. Bill Sutherland has an entire book dedicated to these systems titled *Beautiful Models* [3]. Moreover, with the advent of cold atom studies, these one-dimensional problems with exact solutions are now experimentally realizable [4].

Being exactly solvable and experimentally realizable, these models allow for studies combining experiment and theory, bridging the gap between experimentalists and theorists. A corollary is that these models serve as excellent testbeds for experimental
techniques. The Lieb-Liniger model

\[ H = -\frac{\hbar^2}{2m} \sum_{i=1}^{N} \frac{\partial^2}{\partial x_i^2} + \sum_{i<j} 2c\delta(x_i - x_j) \]

describes a system of particles interacting via the shortest range potential possible, a delta function \( \delta(x_i - x_j) \): interaction only occurs when the position of particle \( x_i \) is equal to the position of particle \( x_j \), with \( i \neq j \). This model has both ground state and finite temperature solutions [5] and is one of the first models to be experimentally realized utilizing cold atoms [6, 7]. At the other extreme of interaction range is the Sutherland model [8–10]

\[ H = -\frac{\hbar^2}{2m} \sum_{i=1}^{N} \frac{\partial^2}{\partial x_i^2} + \sum_{i<j} g(x_i - x_j)^2, \]  

(1.1)
of primary interest in this thesis. Interactions occur via an inverse-square potential, a power law. That a solution exists at all is surprising: intuition suggests a long range interaction shouldn’t easily yield an exact solution. Dating back some thirty years, the Sutherland model has been studied extensively since its conception, including recent numerical studies of correlation functions [11] and entanglement [12]. Explicit results at finite temperature have yet to be obtained.

The purpose of this thesis is to detail a computational scheme developed for studying long range interactions in one dimension at finite temperature. Algorithmic development in this area is highly desirable since studies of long range interactions are historically intractable due to unfavorable scaling as a function of the number of particles \( N \). Even models with solutions providing both the ground state and excitation spectrum benefit from numerical analysis since properties notoriously difficult to
evaluate exactly, such as particle entanglement or correlation functions, may be easily accessed numerically. While the Sutherland model is of primary interest here, the method developed is easily extended to other systems with long range interactions. Short range interactions also benefit in high particle density regimes such as those studied by Herdman et al. [13].

Consider the double sum in Eq. (1.1). This is a sum of pair interactions, the pair potential, and may be written generally as

$$V = \frac{1}{2} \sum_{i=1}^{N} \left[ \sum_{j=1}^{N} U(|x_i - x_j|) \right],$$  \hspace{1cm} (1.2)

where $i \neq j$ and $U(|x_i - x_j|)$ describes the interaction of a single pair of particles. For long range interactions, this summation is fully evaluated and scales as $O(N^2)$, leading to the poor scaling as a function of the number of particles $N$ mentioned earlier. An approximation may be applied for short range interactions if there exists some separation $R$ where $U(|x_i - x_j| > R)$ is small. The number of particles involved in the inner summation in Eq. (1.2) is then reduced to $N_{cut}$, where $N_{cut} < N$. The complexity of this approximation scales as $O(N \times N_{cut})$ and when employed allows for studies involving thousands of particles [14]. Such approximations fail for long range interactions since there exists no scale $R$ beyond which $U(|x_i - x_j| > R)$ is small. For comparison, the previously mentioned high density study [13] was limited to less than a hundred particles even with millions of CPU hours. For a general potential $(U(|x_i - x_j|))$, specifically one with long range interactions of interest here, Eq. (1.2) represents the limiting step in any computation and thus this thesis focuses on its acceleration.

Any numerical method involving this evaluation may thus be accelerated. There
are many such computational methods: Diffusion Monte Carlo [15], Variational Monte Carlo [16], Greens Function Monte Carlo [17] and Path Integral Monte Carlo [18]. Diffusion Monte Carlo and Variational Monte Carlo are ground state methods, while Greens Function Monte Carlo is difficult to apply at finite temperature. This leaves Path Integral Monte Carlo (PIMC), a numerically exact [19] method exploiting Feynman’s path integrals [20]. PIMC is well suited for studies of interaction many-body systems in the spatial continuum, requiring only a Hamiltonian in the position basis to simulate a model.

1.1 High Performance Computing on Graphics Processing Units

Acceleration of the pair potential evaluation in PIMC has been accomplished by more than just increasing the number of resources available: changes have been made to parallelize it such that a particular kind of resource, Graphics Processing Units (GPUs), may be utilized efficiently. This is achieved by focusing on the most computationally intensive portion of the algorithm involving the pair potential, seen in Eq. (1.2). The pair potential has been broken into discrete portions which are evaluated simultaneously on a GPU.

A conventional serial algorithm to evaluate the double sum in Eq. (1.2) would proceed as in Figure 1.1. Following the edges, there is a clear entry, exit, inner and outer loop. The inner most loop does the bulk of the work as it evaluates each term of the potential, adding the results to $V$. Indices $i$ and $j$ represent the particle labels. Beginning with start, the total potential $V$ and index $i$ are set to zero. The outer loop
Figure 1.1: Execution is sequential as the inner and outer loops are traversed. This is an example of a double sum calculation where $N$ represents the number of interacting particles with labels $i$ and $j$, $U(|x_i - x_j|)$ represents the pair potential between two particles and $V$ represents the summation of pair potentials for all particles.

is then entered, its first step setting index $j$ to zero before moving to the inner loop. If $j > N$, the total number of particles, the inner loop is immediately exited and an iteration of $i$ performed. A check if $i > N$ follows, in which case the summation has finished. Step by step, this is the same as writing down all the terms in the inner summation of Eq. (1.2), iterating $i$ and then writing down the next set of terms. This is a serial process where tasks are completed in order.

Now consider the same calculation on a GPU, as in Figure 1.2 where a serial $N$ calculation is reformulated as $N$ parallel calculations. Every iteration of the inner loop of Figure 1.1 is carried out simultaneously before iterating index $i$, since the $j$ index is scattered with each branch receiving a unique index. A key point, one that makes programming in such paradigms difficult, is that the order of execution is unknown. This is why there are additional nodes representing a scatter operation where the task is split and a gather operation where the results are reduced to a
Figure 1.2: The inner loop has been unrolled and a scatter operation assigns a unique j index to N workitems. These workitems evaluate their respective $U(|x_i - x_j|)$ interactions before a gather, also known as a reduction, is performed to sum the interactions. When multiple edges exit a node they do so simultaneously, while multiple edges entering a node may happen in any order: the gray edges. Emphasize that the order of execution is unknown.

single scalar. Traditional programming techniques allow inspection of single iterations of these loops, as in the inner loop of Figure 1.1, however with GPUs the entire computation has to be consider at once.

The question then becomes why bother with using GPU cores instead of CPU cores? Clusters, comprised of many CPUs each with multiple cores resulting in thousands of cores total, are at the heart of most HPC facilities and so widely available: their use would be most convenient. The answer is that GPUs also have thousands of cores and, while each of them is less capable than a single CPU core, they all reside on a single die: communication between the cores, and data shared by the cores, reside
on a single host machine. With a cluster, the particle positions \( x_i \) and \( x_j \) would have to be sent to every host for \( U(|x_i - x_j|) \) evaluation, while with a GPU all cores have access to the same memory which is located very close to the physical core.

In this thesis a speedup factor of over two orders of magnitude versus a single core is achieved by parallelizing the pair potential calculation. This reduces the runtime of a typical long range interaction simulation by months or even years as seen in chapter 5, and by extension can significantly increase the number of particles in a simulation given the same amount of time.

1.2 Breakdown of the Remaining Chapters

The thesis is organized as follows: in chapter 2 the Sutherland model is described and the energies of its ground state derived. In chapter 3, Path Integral Monte Carlo is introduced, first focusing on the formalism and nomenclature of path integrals before discussing Monte Carlo sampling. The works of Max Graves [21] and Llorenç Brualla Barberà [22] were particularly useful while preparing this chapter.

Chapter 4 will discuss in detail the computation of the action on Graphics Processing Units. In chapter 5 the quantitative power of this computational scheme is demonstrated, benchmarking it with well understood models and then applying it to the Calogero-Sutherland model. Conclusions are outlined in chapter 6, including a brief discussion of future work. Finally solutions to the models which inspired the Calogero-Sutherland model are included in appendix A.

All code developed as part of this thesis has been released as an open source project [23] and is available at http://www.github.com/bhimberg/gpupimc.
Figure 2.1: A schematic of the Periodic Sutherland Model. Many particles are confined to a ring, each interacting with all others via a long range inverse-square potential.

The Sutherland model as depicted in chapter 1 was simplified to show the power law scaling. The full Sutherland model (in units where $\hbar^2/2m = 1$) is

$$\mathcal{H} = -\sum_{i=1}^{N} \frac{\partial^2}{\partial x_i^2} + \sum_{i<j} \frac{g}{(x_i - x_j)^2} + \omega^2 \sum_{i=1}^{N} x_i^2,$$

which includes a harmonic well. The use of a harmonic well in the original Sutherland
model destroys Galilean invariance [10]. The purpose of the well is to confine the particles, a purpose which may instead be accomplished by replacing the potential of the Sutherland model with one which is periodic in nature. This can be visualized as constraining the particles to interact on a ring as in Figure 2.1, which will be referred to as the Calogero-Sutherland model.

The purpose of this chapter is to first derive an expression for the Calogero-Sutherland model in terms of its Hamiltonian by making modifications to the Sutherland model. This Hamiltonian will then be evaluated to arrive at its $T = 0K$ ground state energies.

2.1 Replacement of the Harmonic Well

Periodic boundary conditions are enforced by adding the interaction of pairs across multiple images, thus removing the necessity of a harmonic well to confine the particles. The full solution to the non-periodic model, known simply as the Sutherland model, can be found in appendix A. In order to incorporate periodic boundary conditions, begin with the potential from the non-periodic model

$$V(r) = g \frac{1}{r^2}, \quad r = x_i - x_j$$

and insert an integer multiplier $n$ and length $L$

$$V(r) = g \frac{1}{(r + nL)^2}$$
where the $n = 0$ case represents a single image, or put another way the minimum separation $r = x_i - x_j$ between two particles. Extending the potential to a summation over all $n \in \mathbb{Z}$

$$V(r) = g \sum_{n=-\infty}^{\infty} \frac{1}{(r + nL)^2}$$

includes all possible wrappings about L. Inspecting Figure 2.1, this can be visualized as taking a rod of length L with N-particles on it and bending the ends around to form a loop. Every particle on the rod interacts with every other particle once about the loop, twice about the loop, up to an infinite number of times about the loop in both positive and negative directions. Each time the interaction wraps about the loop, its contribution is significantly less and so this is a convergent series. Each wrapping about the loop represents an additional image, and these images serve to keep the particles from completely dispersing.

Simplifying this infinite series is accomplished via contour integration. First a factor of $L^2$ is pulled out before moving to the complex plane via a substitution of $z$ for $n$, followed by an integration over $\pi \cot(\pi z)dz$

$$V(r) = g \sum_{n=-\infty}^{\infty} \frac{1}{(r + nL)^2} = \frac{1}{L^2} \frac{1}{2\pi i} \oint_{C_1 + C_2} \frac{1}{(r/L + z)^2} \pi \cot(\pi z)dz \quad (2.2)$$

This places a singularity at every point in the series summation, the residue of which results in a respective term of the series. This is drawn in Figure 2.2, where the presence of a singularity outside contours $C_1$ and $C_2$ is noted. A third contour $C_3$ is drawn
Figure 2.2: Visualization of the contour integral in Eq. (2.2), note the presence of a singularity outside $C_1$ and $C_2$, at $-r/L$.

and its residue found to be

$$res(z = -r/L) = \lim_{z \to -r/L} \frac{d}{dz} \left[ (r/L + z)^2 \frac{\pi \cot(\pi z)}{(r/L + z)^2} \right]$$

$$= -\pi^2 \csc^2 \left( \frac{r\pi}{L} \right).$$

The integral in Eq. (2.2) is evaluated as

$$V(r) = \frac{1}{L^2} \frac{1}{2\pi} \left[ -2\pi i \sum z \, res(z) \right]$$

where the negative sign is a consequence of $C_3$’s direction. Now only a single residue remains, leading to

$$V(r) = \frac{\pi^2}{L^2} \left[ \sin \frac{\pi r}{L} \right]^{-2}$$
which is the potential portion of the Calogero-Sutherland model’s Hamiltonian

$$\mathcal{H} = -\sum_{i=1}^{N} \frac{\partial^2}{\partial x_i^2} + g \frac{\pi^2}{L^2} \sum_{i<j} \left[ \sin \left( \frac{\pi (x_i - x_j)}{L} \right) \right]^2. \quad (2.3)$$

Note that the resulting potential replaces separations along the line with a cord that connects particles, as in Figure 2.1. The effects of all the images slightly enhances the interaction between any two particles.

### 2.2 Ground State Energy

In Ref. [10] Sutherland presented an exact solution to this model for the general $N$ case via a Bethe-Ansatz approach starting with a wavefunction of the product form, a Bijl-Dingle-Jastrow wave function

$$\Psi = \prod_{i<j} [\psi(x_i - x_j)]^{\lambda_s}$$

and made use of the following notation

$$\phi_{ij} = \frac{\psi_{ij}'}{\psi_{ij}}, \quad (2.4)$$

$$\phi_{ij}' = \frac{\psi_{ij}''}{\psi_{ij}} - \left( \frac{\psi_{ij}'}{\psi_{ij}} \right)^2, \quad (2.5)$$

where in this notation derivatives are taken with respect to the first index, in this case $i$. The notation used here emphasizes the convenience of this type of wave function: in one-dimension the chain rule allows the results of the kinetic operator on such wave functions to be defined in a recursive manner. While this wavefunction
is normalizable, the numerical method later described has no dependence on the wavefunction, and so only ground state energies will be addressed here. A discussion on normalizability may be found in the literature \[10\]. Before attacking the case of general \( N \), \( N = 2, 3, 4 \) are addressed.

### 2.2.1 \( N = 2 \)

Utilizing the Hamiltonian derived in Eq. (2.3), the trial wave function is

\[ \Psi = \psi_{12}^\lambda \]

where \( \psi_{ij} = \sin \left[ \frac{\pi}{L} (x_i - x_j) \right] \) with first and second derivatives

\[
\begin{align*}
\frac{\partial \psi}{\partial x_1} &= \lambda_s \Psi \phi_{12} \\
\frac{\partial \psi}{\partial x_2} &= -\lambda_s \Psi \phi_{12} \\
\frac{\partial^2 \psi}{\partial x_1^2} &= \lambda_s (\lambda_s \phi_{12}^2 + \phi_{12}') \Psi \\
\frac{\partial^2 \psi}{\partial x_2^2} &= \lambda_s (\lambda_s \phi_{12}^2 + \phi_{12}') \Psi.
\end{align*}
\]

The following two conditions are relevant

\[
\begin{align*}
\frac{\partial \psi_{12}}{\partial x_1} &= - \frac{\partial \psi_{12}}{\partial x_2} \quad \text{(2.6)} \\
\frac{\partial \phi_{12}}{\partial x_1} &= - \frac{\partial \phi_{12}}{\partial x_2}. \quad \text{(2.7)}
\end{align*}
\]
The expected result of the kinetic operator working on this wave function is

$$-\sum_{i=1}^{2} \frac{\partial^2 \Psi}{\partial x_i^2} = -2\lambda_s(\phi'_{12} + \lambda_s\phi_{12}^2) \quad (2.8)$$

and immediately this is seen to be the case with

$$-\sum_{i=1}^{2} \frac{\partial^2 \Psi}{\partial x_i^2} = -\frac{\partial^2 \Psi}{\partial x_1^2} - \frac{\partial^2 \Psi}{\partial x_2^2}$$

$$= -2\lambda_s(\lambda_s\phi_{12}^2 + \phi'_{12})\Psi$$

A function is required which satisfies the definitions found in Eq. (2.4) to (2.7). One such function is $\sin(\theta_{ij})$, mentioned earlier, where $\theta_{ij} = \frac{\pi}{L}(x_i - x_j)$. A few identities

$$\psi_{ij} = \sin(\theta_{ij}) \quad (2.9)$$
$$\psi'_{ij} = \frac{\pi}{L} \cos(\theta_{ij}) \quad (2.10)$$
$$\phi_{ij} = \frac{\pi}{L} \cot(\theta_{ij}) \quad (2.11)$$
$$\phi'_{ij} = -\frac{\pi^2}{L^2} (1 + \cot^2(\theta_{ij})) \quad (2.12)$$

are necessary to work through this derivation.

Simplifying Eq. (2.8),

$$\lambda_s\phi_{ij}^2 + \phi'_{ij} = \lambda_s\frac{\pi^2}{L^2} \cot^2(\theta_{ij}) - \frac{\pi^2}{L^2} (1 + \cot^2 \theta_{ij})$$

$$= \frac{\pi^2}{L^2} ((\lambda_s - 1) \cot^2 \theta_{ij} - 1)$$

$$= \frac{\pi^2}{L^2} ((\lambda_s - 1) \csc^2 \theta_{ij} - \lambda_s)$$
leaves

\[- \sum \frac{\partial^2 \Psi}{\partial x_i^2} = -2\lambda_s((\lambda_s - 1)\frac{\pi^2}{L^2} \csc^2 \theta_{ij} + 2\lambda_s^2 \frac{\pi^2}{L^2})\]

an alternative form of the kinetic operator. Combining this with the potential operator, and simplifying

\[\left[ g - 2\lambda_s(\lambda_s - 1) \right] \sum_{i<j} \frac{1}{\sin(x_i - x_j)^2} \Psi + 2\lambda_s^2 \frac{\pi^2}{L^2} \Psi = E\Psi.\]

This describes the relation between the interaction strength \( g \) and the exponent \( \lambda_s \) as

\[ g = 2\lambda_s(\lambda_s - 1). \]

The expected energy under this condition is

\[ E = 2\lambda_s^2 \frac{\pi^2}{L^2}. \]

2.2.2 \( N = 3 \)

The expected three particle kinetic operator is

\[- \sum_i^3 \frac{\partial^2 \Psi}{\partial x_i^2} = -2\lambda_s[\lambda_s \sum_{i<j}^3 \phi_{ij}^2 + \sum_{i<j}^3 \phi_{ij}'^2] + 2\lambda_s^2(\phi_{31}\phi_{12} + \phi_{12}\phi_{23} + \phi_{23}\phi_{31}) \quad (2.13)\]

where the last set of terms is called a triple.

The trial wave function now has three components

\[ \Psi = [\psi_{12}\psi_{13}\psi_{23}]^{\lambda_s} \]
with first derivatives

\[
\frac{\partial \Psi}{\partial x_1} = \frac{\lambda_s \Psi}{\psi_{12}\psi_{13}\psi_{23}} (\psi'_{12}\psi_{13}\psi_{23} + \psi_{12}\psi'_{13}\psi_{23}) = \lambda_s \Psi (\phi_{12} + \phi_{13}) \\
\frac{\partial \Psi}{\partial x_2} = \frac{\lambda_s \Psi}{\psi_{12}\psi_{13}\psi_{23}} (-\psi'_{12}\psi_{13}\psi_{23} + \psi_{12}\psi'_{13}\psi_{23}) = \lambda_s \Psi (-\phi_{12} + \phi_{23}) \\
\frac{\partial \Psi}{\partial x_3} = \frac{\lambda_s \Psi}{\psi_{12}\psi_{13}\psi_{23}} (-\psi_{12}\psi'_{13}\psi_{23} - \psi_{12}\psi_{13}\psi'_{23}) = \lambda_s \Psi (-\phi_{13} - \phi_{23})
\]

and second derivatives

\[
\frac{\partial^2 \Psi}{\partial x_1^2} = \lambda_s \Psi (\phi'_{12} + \phi'_{13}) + \lambda_s^2 \Psi (\phi_{12} + \phi_{13})^2 \\
\frac{\partial^2 \Psi}{\partial x_2^2} = \lambda_s \Psi (\phi'_{12} + \phi'_{23}) + \lambda_s^2 \Psi (-\phi_{12} + \phi_{23})^2 \\
\frac{\partial^2 \Psi}{\partial x_3^2} = \lambda_s \Psi (\phi'_{13} + \phi'_{23}) + \lambda_s^2 \Psi (-\phi_{13} - \phi_{23})^2.
\]

Gathering the terms by power provides a glimpse of Eq. (2.13)

\[- \sum_{i=1}^{3} = -2\lambda_s [\lambda_s (\phi^2_{12} + \phi^2_{13} + \phi^2_{23}) + (\phi'_{12} + \phi'_{13} + \phi'_{23})] + 2\lambda_s^2 (-\phi_{13}\phi_{12} + \phi_{12}\phi_{23} - \phi_{23}\phi_{13}).\]

The cyclic permutations are of \(ijjk\) ordering (e.g. 1223 3112 and 2331) in Ref. [10]. The first term in the triple above is ordered \(ikij\). Indices can be swapped, producing
a sign change in the process, for the first and last term

\[-\sum_1^3 \frac{\partial^2 \Psi}{\partial x_i^2} = -2\lambda_s \sum_{i<j} [\lambda_s \phi_{ij}^2 + \phi_{ij}'] + 2\lambda_s^2 (\phi_{31} \phi_{12} + \phi_{12} \phi_{23} + \phi_{23} \phi_{31})\]

hence arrival at a form of Eq. (2.13), only differing by a count of pair interactions in the summation and thus having a single triples term.

Any triples term can be expressed generically as

\[F(x, y, z) = \phi(x)\phi(y) - \phi(x + y)[\phi(x) + \phi(y)]\] (2.14)

where \(x = x_i - x_j, y = x_j - x_k, z = x_k - x_i\) and \(x + y + z = 0\). A function \(\phi(t)\) needs to be found which will satisfy Eq. (2.14), resulting in a constant evaluation of \(F(x, y, z)\). The cotangent function, when used with the cotangent addition identity

\[\cot(x + y)[\cot x + \cot y] = \cot x \cot y - 1\]

is adequate and confirms the form of \(\phi(t)\) in Eq. (2.11). Each triples term is replaced with

\[\phi_{ij}\phi_{jk} + \phi_{jk}\phi_{ki} + \phi_{ki}\phi_{ij} = 1\]

and, following from the two-body section,

\[[g - 2\lambda_s(\lambda_s - 1)] \sum_{i<j} \frac{1}{\sin(x_i - x_j)^2} \Psi + 2\lambda_s^2 \pi^2 [(3 + 1)/L^2] \Psi = E\Psi\] (2.15)

where three pair interactions and a single triples term have been counted. Zeroing
the summation, the energy for the three-body problem is

\[ E = 8\lambda_s^2\pi^2/L^2. \tag{2.16} \]

2.2.3 N Particles

The N-body case follows directly from the three-body case: consider Eq. (2.15)

\[ [g - 2\lambda_s(\lambda_s - 1)] \sum_{i<j}^3 \frac{1}{\sin(x_i - x_j)^2} \Psi + 2\lambda_s^2\pi^2[(3 + 1)/L^2] \Psi = E \Psi \]

and focus on the \(2\lambda_s^2\pi^2[(3 + 1)/L^2]\) term. Here the 3 represents the number of pair interactions for the three body model which can be taken as the upper triangular, subtracting out the trace, of a \(3 \times 3\) matrix. This is extended to an \(N \times N\) matrix as

\[ \text{# of pair interactions} = (N^2 - N)/2 = N(N - 1)/2. \]

While the 1 represents the number of triples. These can be counted by considering a rank 3 tensor with \(N \times N \times N\) where every element is an \(ijk\) permutation and each index is bound on \([1, N]\). Permutations without repeated indices are counted, of which there are less than \(N^3/2\). Subtracting the plane of the trace (that is the trace of \(N \times N \times N\) slices) as \(N^2\) before halving \(N^3\) and finally noting that there are only \(N(N - 1)\) non-repeating elements per slice in each upper-triangular results in

\[ \text{# of triples} = [(N^3 - N^2) - N(N - 1)]/6 = N(N - 1)(N - 2)/6 \]
where the division by three is necessary since it takes three unique, non-repeating, permutations to comprise a triple.

Utilizing these counts with $N = 3$ matches expectations and so

$$E_N = \frac{1}{3} \lambda_s^2 \pi^2 N(N^2 - 1)/L^2$$

(2.17)

is the energy of the ground state in terms of number of particles $N$, circumference $L$ and dimensionless variable $\lambda_s$. Recall that $\lambda_s$ began as an exponent and, having no units, marks $g$

$$g = 2\lambda_s(\lambda_s - 1)$$

(2.18)

$$\lambda_s = \frac{1}{2}[1 + (1 + 2g)^{1/2}]$$

(2.19)

as the bare interaction strength. Expressions for the potential and kinetic contributions to the total energy are derived in chapter 5.

Recall that these expressions are for the energies of the ground state, $T = 0.0 \text{ K}$. Beginning with the partition function, a numerical method will be described here in order to perform studies of the Calogero-Sutherland model at finite temperature.
Chapter 3

Path Integral Formalism

Having derived ground state energies for the Calogero-Sutherland model, development of the computational scheme necessary to study the model at finite temperature begins by formulating an integral representation of the partition function

\[ Z = \text{Tr} \hat{\rho}, \]

where \( \hat{\rho} = e^{-\beta \mathcal{H}} \) is the density matrix, represented as an exponentiated operator, \( \beta = 1/k_B T \), and \( \mathcal{H} = \mathcal{T} + \mathcal{V} \) is the Hamiltonian. The notation \( \text{Tr} \) signifies the sum of the diagonal entries of a matrix

\[ \text{Tr}(A) = \sum_i A_{ii}. \]

Exponentials of operators can be evaluated explicitly using a Taylor expansion
and working out the commutators. The following identities will need to be employed:

\[
\exp(i\mathcal{G}\epsilon)A\exp(-i\mathcal{G}\epsilon) = A + i\epsilon[\mathcal{G}, A] + \frac{i^2\epsilon^2}{2!}[\mathcal{G}, [\mathcal{G}, A]] + \ldots \quad (3.1)
\]

\[
\exp(\epsilon A + \epsilon B) = \exp(\epsilon A)\exp(\epsilon B)\exp\left(\frac{\epsilon}{2}[A, B]\right) \quad (3.2)
\]

where the second identity is true under the condition that \([A, [A, B]] = 0\).

![Figure 3.1: An example of one possible \(|R\rangle\) configuration for five particles in one dimension.](image)

While the density matrix may be used with any basis, the position basis is used here since the potential operator \(V\) is diagonal in such a basis. The density matrix is now written as

\[
\rho(R, R'; \beta) = \langle R | e^{-\beta H} | R' \rangle
\]

where \(R \equiv r_1, ..., r_N\) labels the positions of the \(N\) particles. The density matrix could be rewritten as

\[
\rho(R, R'; \beta) = \langle R | e^{-\beta T} e^{-\beta V} | R' \rangle
\]

if the kinetic and potential parts of the Hamiltonian were to commute. They do commute in the classical regime, and of course the Hamiltonian \(H\) commutes with itself. The density matrix can thus be written as

\[
\rho(R, R'; \beta) = \langle R | e^{-\frac{\beta}{2}(H+V)} | R' \rangle = \langle R | e^{-\frac{\beta}{2}H} e^{-\frac{\beta}{2}V} | R' \rangle. \quad (3.3)
\]
All particles must exist in the d-dimensional volume described by the position basis. This conservation of particles can be expressed as

\[ \int d^d r_1 \cdots d^d r_N |R⟩⟨R| = 1 = \int dR |R⟩⟨R|. \] (3.4)

The communicability of $\mathcal{H}$ with itself, the Hausdorff-Campbell-Baker relation and particle conservation expressed in Eq. (3.2) to (3.4) are used to write the density matrix as

\[ \rho(R, R'; \beta) = \langle R| e^{-\frac{\beta}{2} \mathcal{H}} e^{-\frac{\beta}{2} \mathcal{H}} |R'⟩ \]

\[ = \int dR'' \langle R| e^{-\frac{\beta}{2} \mathcal{H}} |R''⟩ \langle R''| e^{-\frac{\beta}{2} \mathcal{H}} |R'⟩ \]

\[ = \int dR'' \rho(R, R''; \beta/2) \rho(R'', R'; \beta/2). \]

This convolution relation represents a density matrix at some temperature $\beta$ as an integral of the product of two density matrices, each at a higher effective temperature with $\tau = \beta/2$. Combined with the definition of the partition function in the position basis,

\[ Z = \int dR \rho(R, R; \beta) \]

repeated application of the convolution relation expresses the partition function

\[ Z = \int dR_0 \cdots dR_{M-1} \cdots \rho(R_0, R_1; \beta/M) \rho(R_1, R_2; \beta/M) \cdots \rho(R_{M-1}, R_0; \beta/M) \] (3.5)

as an $N \times d \times M$-dimensional integral. This integral representation can be visualized along the lines of Figure 3.1 by drawing the $M$ unique $|R⟩$ configurations represented by Eq. (3.5) as $|R_\alpha⟩$, where $\alpha = 0, 1, \cdots, M - 1$. Note also that the first and last
terms of Eq. (3.5) share the \( |R_0 \rangle \) configuration as the first, and then second, argument of \( \rho(R, R'; \beta/M) \). This sharing of \( |R_0 \rangle \) represents the boundary condition in the path integral formalism which can be understood in the usual way as a quantum system mapping to a classical one in an additional dimension with a temperature dependent periodic boundary condition. Further, the form of \( \rho(R, R'; \beta/M) \) resembles that of the unitary time evolution operator

\[
\rho(R_\alpha, R_{\alpha+1}; \tau) = \langle R_\alpha | e^{-\tau H} | R_{\alpha+1} \rangle \\
= \langle R_\alpha | U(-i\hbar \tau) | R_{\alpha+1} \rangle
\]

where

\[
U(t) = e^{-itH/\hbar}
\]

having made the identification

\[
t = -i\hbar \tau
\]

which marks the extra dimension as imaginary time. Recognizing the periodic boundary conditions in imaginary time allows a schematic representation of the partition function to be drawn as in Figure 3.2. Consider \( M = 7 \) unique configurations, where the first and last row are the same configuration, and \( N = 5 \) particle labels. Following a single particle label as the imaginary time extent is traversed in the vertical direction forms what is called a *worldline*. Each worldline represents a single quantum particle.

This representation of the partition function is exact and stands as the basis of the path integral method. The individual high temperature density matrices may now be evaluated in some approximate manner.
3.1 Primitive Approximation

Writing the density matrix terms in a tractable form begins by making the primitive approximation, amounting to treating the operators $\mathcal{V}$ and $\mathcal{T}$ as if they commute. Eq. (3.2)

$$e^{-\tau \mathcal{H}} = e^{-\tau \mathcal{T}} e^{-\tau \mathcal{V}} e^{-\frac{\tau^2}{2} [\mathcal{T}, \mathcal{V}]}$$

(3.6)
demonstrates that as $\tau = \beta/M$ goes to zero the exponentiated commutator approaches unity. The error due to this approximation is

$$e^{-\tau H} = e^{-\tau T} e^{-\tau V} + \mathcal{O}(\tau^2)$$

which matches expectations: as $M \to \infty$, $\tau \to 0$, and the classical regime is approached for each density matrix term in Eq. (3.5) and so each thermal density matrix can be treated as a product of the kinetic density matrix and the potential density matrix, i.e. a high-temperature approximation. There exist higher-order approximations such as a fourth order method called the Generalized Suzuki Factorization [24] which is utilized in all numerical computations performed in this thesis with results reported in chapter 5. However, for the sake of clarity, subsequent derivations will be performed within the primitive approximation accurate to $\mathcal{O}(\tau^2)$.

### 3.1.1 Potential Density Matrix

Employing the primitive approximation and the particle normalization constraint, Eq. (3.4) and (3.7), a single transition amplitude from Eq. (3.5) can be represented as

$$\rho(R, R'; \tau) = \int dR'' \langle R | e^{-\tau T} | R'' \rangle \langle R'' | e^{-\tau V} | R' \rangle.$$  

Since $e^{-\tau V}$ is diagonal in the position basis $| R' \rangle$ it has eigenvalues

$$e^{-\tau V(R)} = \exp \left[ \sum_{i<j} U(|r_i - r_j|) \right]$$

(3.8)
where the potential portion of Eq. (2.3) is

\[ U(r_i - r_j) = g \frac{\pi^2}{L^2} \sum_{i<j} \left[ \sin \frac{\pi(x_i - x_j)}{L} \right]^{-2} \]

in the case of the Calogero-Sutherland model. Operating \( e^{-\tau V} \) to the right produces

\[ \rho(R, R'; \tau) = \int dR'' \langle R \mid e^{-\tau T} \mid R'' \rangle \langle R'' \mid R' \rangle \delta(R'' - R') e^{-\tau V(R')} = \langle R \mid e^{-\tau T} \mid R' \rangle e^{-\tau V(R')} \]

(3.9)

where

\[ \delta(R - R') = \prod_{i=1}^{N} \delta(r_i - r'_i). \]

This simplifies the expression, leaving the kinetic portion to be dealt with.

### 3.1.2 Free Particle Propagator

The kinetic portion of the density matrix is the free particle propagator, and can be evaluated for free particles in a d-dimensional box subject to periodic boundary conditions.

Consider the one dimensional, single particle, case

\[ \mathcal{H}\psi = -\lambda \frac{d^2\psi}{dx^2} = E\psi \]

where \( \lambda = \hbar^2/2m \), yielding

\[ \psi'' = -(E/\lambda)\psi. \]
There are no bound states, and so $E$ is positive. A solution to the above differential equation is

$$\psi(x) = Ae^{ikx} + Be^{-ikx}, \quad k^2 = E/\lambda$$

where the left term represents a wave travelling in the positive spatial direction and the right term a wave travelling in the negative spatial direction. A single particle travelling in either direction can be represented by setting $B = 0$ and allowing $k$ to take negative values. Enforcing the periodic boundary condition

$$\psi(x) = \psi(x + L)$$

results in

$$Ae^{ikx} = Ae^{ikx}e^{ikL}$$

$$1 = \cos(kL) + i\sin(kL)$$
where Euler’s formula is used to show that \( kL = 2\pi n \). The proposed solution is normalized

\[
\int_0^L \psi^* \psi \, dx = 1
\]

and yields

\[
A = L^{-1/2}
\]

leading to

\[
\psi(x) = L^{-1/2} \exp \left[ i \frac{2\pi}{L} n x \right], \quad n \in \mathbb{Z} \ldots \tag{3.10}
\]

with eigenvalues

\[
E_n = \lambda \left( \frac{2\pi}{L} \right)^2 n^2. \tag{3.11}
\]

Extension to the d-dimensional case is accomplished by recognizing the orthogonality of the spatial dimensions. The N-particle case is handled in a similar fashion since there are no interactions between particles. The N-body, d-dimensional wavefunction with eigenvalues is then

\[
\psi_n = \prod_{j=1}^{dN} \frac{1}{\sqrt{L}} \exp[i(2\pi/N)n_j x_j], \quad E_n = \lambda \left( \frac{2\pi}{L} \right)^2 \sum_{j=1}^{dN} n_j^2.
\]

Returning to Eq. (3.9) the eigenstates are written in terms of planewaves

\[
\langle R | e^{-T\mathcal{T}} | R' \rangle = \sum_n \psi_n^*(R) \psi_n(R') \exp \left[ -\tau \lambda \left( \frac{2\pi}{L} \right)^2 \sum_{j=1}^{dN} n_j^2 \right]
\]

and a second approximation made: \( \tau \lambda \ll L^2 \). The outer summation may now be
treated as an integration
\[ \sum_n \approx \prod_{j=1}^{dN} \int dn_j \]

therefore
\[ \langle R| e^{-\tau T} |R' \rangle = \prod_{j=1}^{dN} \frac{1}{L} \int_{-\infty}^{\infty} \exp \left[ -\tau \lambda \left( \frac{2\pi}{L} \right)^2 n_j^2 \right] \exp \left[ -2\pi i \frac{\left( r_j - r_{j}' \right) n_j}{L} \right] dn_j \]

which is a product of Gaussian integrals. Using the identity
\[ \int_{-\infty}^{\infty} e^{-ax^2} e^{-2bx} = \sqrt{\pi a} e^{b^2/a} \quad (a > 0) \]

with \( a = \tau \lambda (2\pi/L)^2 \) and \( b = i(\pi/L)(r_j - r_{j}') \) yields
\[ \langle R| e^{-\tau T} |R' \rangle = (4\tau \lambda \pi)^{-N_d/2} \exp \left[ -\frac{1}{4\lambda \tau} \sum_{i=1}^{N} |r_i - r_{i}'|^2 \right] = \rho_0(R, R'; \tau). \]

This is the free particle propagator and, when combined with Eq. (3.9)

\[ \rho(R, R'; \tau) = \rho_0(R, R'; \tau) e^{-\tau V(R)} \quad (3.12) \]

completes the expression for the imaginary time propagator in the primitive approximation. Since the configuration space is periodic in the spatial extent, it can be represented as a torus, where the worldlines run along the imaginary time extent, through the center of the torus, as in Figure 3.4.
3.2 Configurations and Permutations

The derivations to this point have been performed with distinguishable particles in mind, yet in the quantum regime like particles are indistinguishable. Henceforth the focus is identical particles with Bose statistics. This means taking all particle label permutations into account. The partition function then becomes

$$Z = (4\tau \lambda \pi)^{-MNd/2} \frac{1}{N!} \sum D R_\alpha e^{-|R_\alpha - R_{\alpha+1}|^2/4\tau \lambda} e^{-\tau V(R_\alpha)}$$  \hspace{1cm} (3.13)
where $\sum_P$ represents all such permutations, $\frac{1}{N!}$ is a normalization factor, and $R_\alpha$ are the positions of a single slice of imaginary time. Figure 3.5 shows one possible permutation.

The partition function is represented as a path integral over all possible permutations and configurations of worldlines. Recall that a worldline is a set of configurations for a single particle label, each at a different point in imaginary time.

![Diagram showing permutation of worldlines](image)

*Figure 3.5: Each of the connected worldlines use many classical particles to represent a quantum particle. The particle positions have not changed, however the links have been swapped from the initial configuration (left) to generate a new configuration (right). This will change the kinetic energy, however it has no effect on the potential energy: for emphasis, note that the figure on the right two chains connected through periodic boundary conditions while the figure on the left has five chains.*

The components necessary to evaluate the integral form of an N-body, d-dimensional partition function now consist of an $N \times M \times d$ tensor containing the d-dimensional positions of N-particles at M imaginary-time slices and a $2 \times N \times M$ tensor allowing for the identification of imaginary time neighbors of $R_\alpha$ at $R_{\alpha-1}$ and $R_{\alpha+1}$. This second tensor consists of two $N \times M$ matrices, each of which maps the next (prior)
position of the particle in the positions matrix while traversing the imaginary time extent. Described in a simpler manner, the positions of every particle are stored in the first tensor while a description of the links connecting the particles is stored in the second tensor. These components are used to evaluate estimators, the most relevant for both benchmarking and studies of finite temperature being the energy of the N-particle system.

### 3.3 Energy Estimator

The thermodynamic energy estimator is given by

$$\langle E \rangle = -\frac{1}{Z} \frac{\partial Z}{\partial \beta}$$

where the partition function can be written as

$$Z = (4\tau\lambda\pi)^{-MN_d/2} \int \mathcal{D}R e^{-S_M}$$

with

$$S_M = \sum_{\alpha=0}^{M-1} S_\alpha, \quad S_\alpha = \frac{1}{4\tau\lambda} \left| R_\alpha - R_{\alpha+1} \right|^2 - \tau V(R_\alpha)$$

and the $N \times M \times d$ hyper-volume $\int \mathcal{D}R$ is defined to be

$$\int \mathcal{D}R = \prod_{\alpha=0}^{M-1} \prod_{i=1}^{N} \int d^d r_{i,\alpha}.$$  

$N$ denotes distinguishable particles: evaluated numerically, the sampling method will handle the indistinguishable case.
Noting that
\[ \beta = M \tau \]
and
\[ \frac{d}{d \beta} = \frac{1}{M} \frac{d}{d \tau} \]
leads to
\[
\langle E \rangle = -\frac{1}{Z} \frac{\partial Z}{\partial \beta} = \frac{N d}{2 \tau} + \frac{1}{M} \int DR \left( -\frac{\partial S_M}{\partial \tau} \right) e^{-S_M}.
\]
By definition:
\[
\left\langle -\frac{\partial S_M}{\partial \tau} \right\rangle = \frac{\int DR \left( -\frac{\partial S_M}{\partial \tau} \right) e^{-S_M}}{\int DR e^{-S_M}}
\]
and when substituted into Eq. (3.17) gives
\[
\langle E \rangle = \frac{N d}{2 \tau} + \frac{1}{M} \left\langle -\frac{\partial S_M}{\partial \tau} \right\rangle.
\]
Taking the derivative of \( S_M \) with respect to \( \tau \) produces
\[
\langle E \rangle = \left\langle \frac{N d}{2 \tau} + \frac{1}{M} \sum_{\alpha=0}^{M-1} \frac{|R_\alpha - R_{\alpha+1}|^2}{4 \lambda \tau^2} + \frac{1}{M} \sum_{\alpha=0}^{M-1} V(R_\alpha) \right\rangle
\]
which is a complete expression for the thermodynamic energy, evaluated by integrating over all configurations and permutations of the worldlines.

There are an infinite number of configurations in a path integral written in the position basis. Not all configurations contribute to the energy equally, with their respective weights being controlled by the factor \( e^{-S_M} \) in Eq. (3.15). Were this a classical many-body system, the configurations would be weighted according to the
Boltzmann factor

\[ p(X) \propto \exp[-\beta E(X)], \quad \beta = 1/(k_B T) \]

where \( X \) represents a single configuration, and \( E(X) \) would be the energy of that configuration. While this may not be a classical many-body simulation, the exponential in Eq. (3.15) serves the same purpose, weighting configurations.

Recognition of the partition function’s integrand in equation Eq. (3.15) as a probability distribution is the first step towards developing a method of sampling these configurations, a method necessary to evaluate estimators such as in Eq. (3.18).

3.4 Sampling with Markov Chain Monte Carlo

Recall that a configuration, henceforth \( X \), is a set of \( M \)-slices, each slice describing the spatial position of \( N \)-particles. It has already been noted that the partition function in Eq. (3.15)

\[ Z = (4\pi \tau \lambda \tau_i \pi)^{-MN^d/2} \int D \mathbf{R} e^{-S_M} \]

contains a probability distribution function

\[ p_0(S_M) \propto e^{-S_M}, \quad S_M = \sum_{\alpha=0}^{M-1} S_\alpha \]

where \( S_\alpha \) is the action of a single slice \( R_\alpha \)

\[ S_\alpha = \frac{1}{4\tau \lambda} |R_\alpha - R_{\alpha+1}|^2 + \tau V(R_\alpha), \quad (3.19) \]

34
\( \tau = \beta/M \) and \( V(R_\alpha) \) is the potential. The goal is to devise a method to sample configurations \( X \) governed by this probability distribution. Expectation values (such as the one derived for the energy in section 3.3) can then be replaced with a simple sum over configurations:

\[
\langle E \rangle = \frac{1}{N_X} \sum_{\{X\}} E(X)
\]

(3.20)

where \( N_X \) is the number of configurations.

Measuring observables using estimators such as the thermodynamic energy estimator in section 3.3 require many configurations to be sampled from this probability distribution. Since it is high-dimensional, sampling is best performed using a Monte Carlo technique [25]. The method described here utilizes the Metropolis-Hastings algorithm [26], generating new configurations with a probability distribution based on the previous configuration. This is a random walk following a Markov chain as in Figure 3.6 where edges possess a transition probability \( T(X \rightarrow X') \) describing

![Figure 3.6: Some generic Markov chain. Note how the arrows on the edges are unidirectional in this case.](image)

the probability of generating configuration \( X' \) from \( X \) and each node represents a single configuration. The probability of a specific random walk then consists of the
probabilities of traversing a sequence of $n$ nodes, or configurations

\[ P_n(X_1, X_2, \cdots, X_n) = P_1(X_1)T(X_1 \rightarrow X_2)T(X_2 \rightarrow X_3)\cdots T(X_{M-1} \rightarrow X_M) \]  

(3.21)

where $P_1(X_1)$ is the independent probability of configuration $X_1$ occurring, and the transition probabilities are normalized

\[ \sum_{X'} T(X \rightarrow X') = 1 \]

such that the sum of all edges exiting a node is unity. Note that each $X_i$ configuration contains all the worldline degrees of freedom described above, i.e. the positions of $N$ particles at $M$ time slices and their connectivity.

As an example, consider the probability of transitioning to node $X_n$ from node $X_1$. A transition matrix can be built representing all such transitions. This is accomplished by using the row and column indices $i$ and $j$ to represent node labels, and with $T_{ij} = T(X_i \rightarrow X_j)$ yields

\[
\begin{pmatrix}
T_{11} & T_{12} & T_{13} & \cdots & T_{1N} \\
T_{21} & T_{22} & T_{23} & \cdots & T_{2N} \\
T_{31} & T_{32} & T_{33} & \cdots & T_{3N} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
T_{N1} & T_{N2} & T_{N3} & \cdots & T_{NN}
\end{pmatrix}
\]

where normalization is again enforced by each column and each row summing to unity. For example, the transition matrix for a four-state uniform distribution would
then be

\[
\begin{pmatrix}
1/4 & 1/4 & 1/4 & 1/4 \\
1/4 & 1/4 & 1/4 & 1/4 \\
1/4 & 1/4 & 1/4 & 1/4 \\
1/4 & 1/4 & 1/4 & 1/4
\end{pmatrix}
\]

allowing any state to lead to itself or any other state with equal probability.

The goal is to generate a Markov chain of configurations with distribution \( p(X) = \exp\{-S_M\}/\mathcal{Z} \). The distribution simulated by this Markov chain should be independent of the current position in the chain, and independent of the initial position in the chain. Put another way, the transition matrix of the Markov chain should be constant. This can be satisfied by linking all configurations such that any configuration \( X' \) is accessible beginning with any other configuration \( X \) after some finite number of \( t \) steps, and ensuring there are no traps such that a set of nodes \( \{X_1, X_2, \cdots, X_n\} \) are visited in a periodic matter. Essentially all nodes must have at least one edge entering and one edge exiting. Such a Markov chain is called \textit{ergodic}.

One method of obtaining an ergodic Markov chain is to enforce \textit{detailed balance}. Consider the \textit{master equation}

\[
p(X, t + 1) - p(X, t) = -\sum_{X'} T(X \rightarrow X')p(X, t) + \sum_{X'} T(X' \rightarrow X)p(X', t)
\]

where \( p(X, t) \) is the probability of configuration \( X \) occurring at time step \( t \). Since the distribution needs to be stationary, possessing a constant transition matrix, it is required that \( p(X, t + 1) = p(X, t) \). This means that the probability of some configuration \( X \) occurring is independent of the time step. The master equation is
reduced to

\[ \sum_{X'} T(X \rightarrow X')p(X, t) = \sum_{X'} T(X' \rightarrow X)p(X', t) \]

where one possible solution is to match the terms in each sum, leading to an expression for detailed balance

\[ T(X \rightarrow X')p(X) = T(X' \rightarrow X)p(X') \] (3.22)

where the time step dependence has been removed.

In order to proceed, the transition probability may be written as

\[ T(X \rightarrow X') = \omega_{XX'}A_{XX'} \] (3.23)

where \( \omega_{XX'} \) represents an element of a trial step probability matrix and \( A_{XX'} \) represents an element of an acceptance probability matrix. Matrix \( \omega \) has elements \( 0 \leq \omega_{XX'} \leq 1 \) with every column summing to unity, as well as every row. The matrix \( A \) also has elements \( 0 \leq A_{XX'} \leq 1 \). Returning to Eq. (3.22), Eq. (3.23) is substituted in to find

\[ A_{XX'} = \frac{\omega_{X'X}p(X')}{\omega_{XX'}p(X)} \] (3.24)

where \( A_{X'X} = 1 \). The generalized Metropolis procedure calculates the acceptance probability \( A_{XX'} \), and generates a random number \( r \) uniformly between 0 and 1. This number is then compared to \( A_{XX'} \). The new configuration is accepted if \( r < A_{XX'} \), otherwise it is rejected. Note that \( \omega_{X'X} \) and \( \omega_{XX'} \) are determined by the type of update, or move, used. Before discussing moves however, an occurrence probability or "weight" \( p(X) \) must be determined.
Recall this section began with a discussion of the action in Eq. (3.19)

\[ S_\alpha = \frac{1}{4\tau \lambda} |R_\alpha - R_{\alpha+1}|^2 + \tau V(R_\alpha) \]

which is actually a sum of the kinetic action

\[ S^k_\alpha = \frac{1}{4\tau \lambda} |R_\alpha - R_{\alpha+1}|^2 \]

and the potential action

\[ S^v_\alpha = \tau V(R_\alpha). \]

These are the actions of two separate probability distributions. The kinetic portion is not involved in the metropolis sampling stage since the free particle propagator, as derived in section 3.1.2, can be exactly sampled as a product of Gaussian distributions [27] to generate new configurations \( X' \) from current configuration \( X \). That leaves the potential portion, which takes the roll of \( p(X) \) in Eq. (3.24) leaving

\[ A_{XX'} = \frac{\omega_{X'X}}{\omega_{XX'}} \exp\{-\tau [V(R'_\alpha) - V(R_\alpha)]\} \tag{3.25} \]

which involves the evaluation of a single slice: most updates involve multiple slices.

The application of the Metropolis-Hasting algorithm, as described here, to the path integral form of the partition function is well established in the literature. Comprehensive details are included in Ref. [28] with the more modern worm algorithm described in Refs. [19, 29]. These algorithms introduce a set of Monte Carlo updates (open, close, insert, remove, staging, center of mass, advance, recede and swap) which, when combined, allow for the efficient simulation of thousand of particles (with short
range interactions).

The majority of computational time is spent evaluating the acceptance probability of these new configurations (Eq. (3.25)) due to the pair potential calculation and in the next chapter we describe the main advance of this work: accelerating PIMC on Graphics Processing Units.
Chapter 4

Implementation Details

The time scaling of Path Integral Monte Carlo is of order $O(N^2 M)$. While the scaling cannot be changed, the time in which an $N \times M$ configuration, a set $N$ particles described in the position basis across an imaginary time extent $M$ (see chapter 3), has been reduced. This has been accomplished by parallelizing the computation of the effective action that appears in the configuration weights.

Section 4.1 focuses on the primitive action, derived from making the primitive approximation in section 3.1. The fourth order action, derived from the Generalized Suzuki Factorization mentioned in that same section, is utilized in production code the details of which can be found in section 4.2. Additionally it should be noted that the code written in section 4.2 performs work on only a single configuration at a time, while the primitive action example code here evaluates the weights of the current configuration and the proposed configuration simultaneously.
4.1 **Primitive Action**

Consider a center of mass move as an example, which proposes a new system configuration via a uniform translation of an entire worldline. This is depicted schematically in Figure 4.1 where there is a proposed state $X'$ and a current state $X$. The probability of transitioning from $X$ to $X'$ is the acceptance probability

$$A_{XX'} = \frac{\omega_{X'X} p(X')}{\omega_{XX'} p(X)}$$  \hspace{1cm} (4.1)

where the center of mass move provides its own detailed balance and so $\omega_{X'X} = \omega_{XX'}$. Consulting Eq. (4.1), this leaves the acceptance probability dependent on the ratio of the probabilities of being in states $X$ or $X'$ respectively

$$A_{XX'} = \frac{p(X')}{p(X)}$$  \hspace{1cm} (4.2)

*Figure 4.1: Two states: $X$ is the current state and $X'$ is the proposed state generated from $X$ using a center of mass (com) move. Every bead on $r_3$ has been displaced by a constant.*
This ratio $p(X')/p(X)$ depends only on the potential

$$\frac{p(X')}{p(X)} = \sum_{\alpha=1}^{M} e^{-\tau[V(R'_\alpha) - V(R_\alpha)]}$$

(4.3)

where

$$V(R) = \sum_{i < j}^{N} g \frac{\pi^2}{L^2} \left[ \sin \frac{\pi}{L} (r_i - r_j) \right]^{-2}$$

(4.4)

is the Calogero-Sutherland model potential. Evaluating Eq. (4.3) yields the relative weights of state $X$ and state $X'$. Since the primary interest here is the evaluation of Eq. (4.4), focus instead on the contribution from a single slice $\alpha$ in Figure 4.2 and note the interaction potential

![Figure 4.2: A single slice of the current state $X$ has $N = 4$ particles at positions $r_1, r_2, r_3, r_4$ and is used by a center of mass update to generate a new slice in proposed state $X'$. This new slice has the same 4 particles in positions $r_1, r_2, r'_3, r_4$: a single particle has moved since $r_3 \neq r'_3$.](image)

may be represented as a sum of all elements of an upper triangular $4 \times 4$ matrix. The current configuration $X \equiv |R\rangle$ is represented using positions $r_1, r_2, r_3$ and $r_4$

$$
\begin{pmatrix}
0 & V(r_1, r_2) & V(r_1, r_3) & V(r_1, r_4) \\
V(r_2, r_1) & 0 & V(r_2, r_3) & V(r_2, r_4) \\
V(r_3, r_1) & V(r_3, r_2) & 0 & V(r_3, r_4) \\
V(r_4, r_1) & V(r_4, r_2) & V(r_4, r_3) & 0
\end{pmatrix}
$$
and the new configuration is represented using positions $r_1, r_2, r_3', r_4$

$$
\begin{pmatrix}
0 & V(r_1, r_2) & V(r_1, r_3') & V(r_1, r_4) \\
V(r_2, r_1) & 0 & V(r_2, r_3') & V(r_2, r_4) \\
V(r_3', r_1) & V(r_3', r_2) & 0 & V(r_3', r_4) \\
V(r_4, r_1) & V(r_4, r_2) & V(r_4, r_3') & 0 \\
\end{pmatrix}
$$

where the primes denote elements which have changed in the transition. Taking the difference of these two matrices leaves

$$
\begin{pmatrix}
0 & 0 & V(r_1, r_3') - V(r_1, r_3) & 0 \\
0 & 0 & V(r_2, r_3') - V(r_2, r_3) & 0 \\
V(r_3, r_1') - V(r_3, r_1) & V(r_3, r_2') - V(r_3, r_2) & 0 & V(r_3', r_4) - V(r_3, r_4) \\
0 & 0 & V(r_4, r_3) - V(r_4, r_3) & 0 \\
\end{pmatrix}
$$

which can always be rearranged such that the N-1 terms are along the upper edge,

$$
\begin{pmatrix}
\Delta V^0_{31} & \Delta V^0_{32} & \Delta V^0_{34} \\
\end{pmatrix}
$$

and a shorthand notation has been adopted

$$
\Delta V^\alpha_{ij} = V(r_{i,\alpha'}, r_{j,\alpha}) - V(r_{i,\alpha}, r_{j,\alpha})
$$

where the prime denotes a modified position.

Recall that this is only for a single slice. Incorporating all $M$ slices and $N$ particles
results in $M \times (N - 1)$ terms. Considering the $N = 4$ and $M = 7$ slice case produces

$$
\begin{pmatrix}
\Delta V_{31}^6 & \Delta V_{32}^6 & \Delta V_{34}^6 \\
\Delta V_{31}^5 & \Delta V_{32}^5 & \Delta V_{34}^5 \\
\Delta V_{31}^4 & \Delta V_{32}^4 & \Delta V_{34}^4 \\
\Delta V_{31}^3 & \Delta V_{32}^3 & \Delta V_{34}^3 \\
\Delta V_{31}^2 & \Delta V_{32}^2 & \Delta V_{34}^2 \\
\Delta V_{31}^1 & \Delta V_{32}^1 & \Delta V_{34}^1 \\
\Delta V_{31}^0 & \Delta V_{32}^0 & \Delta V_{34}^0 \\
\end{pmatrix}
$$

where all elements are then summed on each slice

$$
\begin{pmatrix}
\Delta V_{31}^6 + \Delta V_{32}^6 + \Delta V_{34}^6 \\
\Delta V_{31}^5 + \Delta V_{32}^5 + \Delta V_{34}^5 \\
\Delta V_{31}^4 + \Delta V_{32}^4 + \Delta V_{34}^4 \\
\Delta V_{31}^3 + \Delta V_{32}^3 + \Delta V_{34}^3 \\
\Delta V_{31}^2 + \Delta V_{32}^2 + \Delta V_{34}^2 \\
\Delta V_{31}^1 + \Delta V_{32}^1 + \Delta V_{34}^1 \\
\Delta V_{31}^0 + \Delta V_{32}^0 + \Delta V_{34}^0 \\
\end{pmatrix}
$$

to represent the change in potential of each slice. This column may then be summed, multiplied by $\tau$ and exponentiated to produce a single weight. This weight is the acceptance probability $A_{XX'}$.

These steps are represented in Figure 4.3 as a flow chart.
Figure 4.3: Flow chart description of CPU code used to calculate the primitive difference in potential action. Note that the index $i$ marks the particle that has been moved by an update. The left block represents each $esXi$ element in the chain on the right. Reading this flow chart begins at start, where the edge enters the first $esXi$ block $esli$, then starting at enter and traversing to exit before moving to the second $esXi$ block, $es2i$.

where the index $i$ labels the particle whose position has been modified between states $X$ and $X'$. This operation can be trivially parallelized by considering the flowchart in Figure 4.4 where
Figure 4.4: Flow chart description of GPU code used to calculate the primitive difference in potential action. Note that the index $i$ marks the particle that has been moved by an update. All arrows leaving a node, unmarked by 'yes' or 'no', happen in parallel, except gray arrows which serve two purposes: they indicate that synchronization is possible between each branch, and that the order of execution is unknown between each branch. The block on the left represents each $esXi$ element on the right, all of which are executed in parallel.

it becomes clear where the advantage lies with GPUs: massively parallel execution.

All elements of our $M \times (N - 1)$ reduced difference in the potential action are calculated simultaneously, their results stored in an array which can be summed and
exponentiated for the total action. Please refer to appendix B for some details on modern GPU performance metrics.

4.2 Production Code

The schematic flowcharts in section 4.1 describe the general algorithm employed to evaluate the potential action used in Monte Carlo simulations. In this section, simplified code is presented to provide further detail on the parallel implementation. The code included in Figure 4.5 and 4.6 has been simplified by removing the external potential calculation, assuming distinguishable particles and not enforcing periodic boundary conditions.
int m = get_local_id(0);  // each workgroup gets a m and contains
int j = get_global_id(1); // N workitems, each with a unique j<N

float bareU, Fint1, totF2 = 0.0f;

local float scratch[3*N];
element(scratch, 1, j) = element(R, m, j); // the local array is filled
barrier(CLK_LOCAL_MEM_FENCE); // workitems stall until they all encounter this

const float ri = element(scratch, 1, i); // recall that i marks the
const float rj = element(scratch, 1, j); // modified particle

if (j != i){
bareU += Vint(ri - rj);
    if (m % 2 == 1){ // only evaluate corrections on odd m
        float Fint2 = gradVint(ri - rj);
        Fint1 -= Fint2;
        float Fint3 = 0.0f;
        for (int k = 0; k < N; ++k){
            float rk = element(scratch, 1, k);
            if (k != i && k != j)
                Fint3 += gradVint(rj - rk);
        }
totF2 += Fint2*Fint2 + 2.0f*Fint2*Fint3;
    }
} barrier(CLK_LOCAL_MEM_FENCE);

Figure 4.5: GPU code used to generate elements in an arrays S and C containing the
bare difference in potential (S), which is equivalent S in the primitive approximation, and
the fourth order correction from the Generalized Suzuki Factorization. Note that only odd
slices are used in the correction, and that the entire slice is read into local memory since all
workitems involve a loop which reads the positions of all particles on the slice. This lowers
the utilization of global memory, which is costly. Workitems will wait at the barrier until
all are present: this is called synchronization.

It serves as an example of the complexity of the production code, even simplified,
and a general framework upon which an alternative implementation may be based.
Note that the bare potential action is stored in array S and the correction is stored in
array C. Additionally this code is designed to be called twice per update: once with
the current configuration, and then a second time with the proposed configuration.
Figure 4.6: The remainder of the GPU code from Figure 4.5. Here a gather operation is performed, utilizing all workitems to sum the bareU, totF2 and Fint1 arrays into their respective first element. The final portion of the computation is then completed, the host combining the terms as necessary and then exponentiating them to provide a per slice difference in potential action.

The resulting $S_c, S_p, C_c, C_p$ arrays are then used to calculate the total difference in potential action on the host. Figure 4.5 details the larger portion of the calculation, while Figure 4.6 performs a gather operation before completing the computation and storing the results.
const double norm = tau*tau*lambda;
const double Vfactor[2] = {2.0/3.0,4.0/3.0};
const double gradVfactor[2] = {0.0,2.0/9.0};

for (int m = 0; m < M; ++m){
    S[m] = 0.0; // start with 0.0
    int eo = m % 2; // tracking even and odd slices
    if (norm*gradVfactor[eo]*fabs(Cp[m])
       < Vfactor[eo]*fabs(Sp[m])){
        S[m] += Vfactor[eo]*Sp[m];
        S[m] += norm*gradVfactor[eo]*Cp[m];
    } else
        S[m] += Sp[m];

    if (norm*gradVfactor[eo]*fabs(Cc[m])
       < Vfactor[eo]*fabs(Sc[m])) {
        S[m] -= Vfactor[eo]*Sc[m];
        S[m] -= norm*gradVfactor[eo]*Cc[m];
    } else
        S[m] -= Sc[m];
}

Figure 4.7: The remainder of the fourth order computation from Figure 4.5 and 4.6 is completed on the CPU. Here corrections \( Cc \) and \( Cp \) which are greater than their respective \( Sc \) and \( Sp \) are ignored, details on \( gradVfactor \) and \( Vfactor \) can be found in the literature [24] and lambda refers to \( \hbar/2m \).

While in the case of the primitive approximation examples above, the \( S \) array contains the difference in potential action, the algorithm here leaves the task of combining, summing and exponentiating the arrays to the CPU, detailed in Figure 4.7.

This is an open source project, and so complete details can be found in the source code available at http://www.github.com/bhimberg/gpupimc.
Chapter 5

Results

Two models, the harmonic oscillator and the free bose gas, are used to benchmark the PIMC implementation. The harmonic oscillator will confirm the implementation is functional for $N = 1$ while the free bose gas tests the N-body case and an imaging method to deal with the

$$\tau \lambda \ll L^2$$

approximation made in section 3.1.2.

An analysis of how much faster the GPU implementation is when compared to a published CPU implementation is then performed before presenting the Calogero-Sutherland model results.

5.1 Benchmarking

The models of interest are comprised of external potentials, such as a harmonic well, and interaction potentials such as the Calogero-Sutherland model. When no potential
is present only the kinetic operator,

\[ \mathcal{T} = -\lambda \frac{\partial^2}{\partial x^2} \]

remains, where \( \lambda = \hbar^2 / 2m \). The two models used as benchmarks are the harmonic oscillator \( \mathcal{V} = \frac{1}{2} m \omega^2 x^2 \) and the free bose gas \( \mathcal{V} = 0 \).

### 5.1.1 Harmonic Oscillator

The harmonic oscillator’s Hamiltonian

\[ \mathcal{H} = -\lambda \frac{d^2}{dx^2} + \frac{1}{2} m \omega^2 x^2 \]

is rearranged

\[ \mathcal{H} \frac{k_B}{k_B} = -\frac{\lambda}{k_B} \frac{d^2}{dx^2} + \frac{k_B}{4\lambda} \left( \frac{\hbar \omega}{k_B} \right)^2 x^2 \]

in order to yield energies in units kelvin. For the results to follow, \( \lambda/k_B = 1 \). The exact solution at finite temperature gives the energy

\[ \frac{E}{k_B} = \frac{\hbar \omega}{2k_B} \coth \left( \frac{\hbar \omega}{2k_B T} \right) \]

where

\[ \frac{\hbar \omega}{k_B} = 1. \]

As the path integral method implemented contains systematic time step (Trotter) error, a suitably small value of \( \tau \) must be determined such that estimators will reproduce known properties of the model being studied within stochastic error. This
value is found through a process called $\tau$ scaling. The error can be represented, in the primitive approximation, by Eq. (3.6)

$$e^{\tau H} = e^{-\tau T} e^{-\tau V} + O(\tau^2)$$

The larger the value of $[T, V]$, the smaller the necessary value of $\tau$ and the greater the computational intensity. Once a value of $\tau$ is found at the lowest temperature of interest, that same value of $\tau$ is applicable to higher temperatures.

Figure 5.1: $\tau$ scaling for the harmonic oscillator in both kinetic and potential energy. The grey bars represent the analytic solution with $\hbar \omega/k_B = 1$ and energy in units kelvin at $T = 0.1$ K. Note that as $\tau$ decreases the PIMC results approach the analytic results.

Figure 5.1 shows that $\tau = 0.08 \text{K}^{-1}$ is acceptable since both the kinetic and potential energy of the analytic results, indicated by the horizontal lines, are within error bars of the PIMC results.
Figure 5.2: These harmonic oscillator results were produced using $\hbar \omega / 2k_B = 1$, $\lambda / k_B = 1$ and $L = 20\,\text{Å}$ with a single particle at $\tau = 0.08\,\text{K}^{-1}$. The potential energy per particle is shown on the left while the kinetic energy per particle is shown on the right.

Having selected a $\tau$, many runs are performed at different temperatures to produce the temperature dependent results in Figure 5.2 which confirm that, for a single particle subject to an external potential, the PIMC implementation is convergent on the expected results.

5.1.2 FREE BOSE GAS

Consider the free bose gas with Hamiltonian

$$\frac{\mathcal{H}}{k_B} = -\frac{\lambda}{k_B} \frac{d}{dx^2}$$

where setting $k_B = 1$ gives energy in units K. The exact solution at finite temperature for a single particle is

$$\frac{E_1}{k_B} = -\frac{1}{k_B} \frac{Z_1(\beta)}{Z_0(\beta)}$$
where
\[ Z_e(\beta) = -\sum_{n=0}^{\infty} \lambda \left( \frac{2\pi}{L} n \right)^2 e^{-\beta \lambda \left( \frac{2\pi}{L} n \right)^2} \]

and
\[ Z_1(\beta) = -\sum_{n=0}^{\infty} e^{-\beta \lambda \left( \frac{2\pi}{L} n \right)^2} \]

Solutions to the two and three particle cases can be found using the recursion relation
\[ Z_N(\beta) = \frac{1}{N} \sum_{k=1}^{N} Z_1(k\beta) Z_{N-k}(\beta) \]
as derived by Borrman and Franke [30, 31]. Note that these can also be expressed as Jacobi Theta Functions of the 3rd kind.

Recall from the derivation for the imaginary time propagator in section 3.1.2 that the kinetic portion of the Hamiltonian is sampled directly. Without a potential, as in the free bose gas case, the Hamiltonian is the kinetic operator and so all moves are accepted. This means the selection of \( \tau \) only affects error bars, yet multiple slices are needed to sample all permutations of identical particles and so \( \tau = 0.05 \) K\(^{-1} \) is selected. The mass is arbitrary: an \( m \) is set such that \( \hbar^2/2mk_B = \lambda = 0.5 \).

While free particles are sampled directly, an approximation was made involving periodic boundary conditions when deriving the free particle propagator. A method of images is used, called winding sectors [18], to sample an increasing number of neighboring boxes as \( \tau \lambda \to L^2 \). This method is benchmarked as well by confining the free bose gas to a small box.
Figure 5.3: free bose gas temperature dependent results for $N = 1, 2, 3$, $\lambda/k_B = 0.5$, $\tau = 0.05\,\text{K}^{-1}$ and $L = 5\,\text{Å}$.

The temperature dependent results in Figure 5.3 were produced using $L = 5.0\,\text{Å}$, resulting in an increasing density as the number of particles is increased from from $N = 1 - 3$.

Results from well understood models have been reproduced and so the remaining sections are dedicated to the Calogero-Sutherland model.
5.2 Speedup Factor

Long range interactions such as in the Calogero-Sutherland model are difficult to simulate due to poor scaling with particle number $N$. A primary motivation for this project was significantly reducing the computational time for these long range interactions, allowing for simulations with a greater number of particles to be run.

![Graph showing speedup factor for different $\tau$ and $N$ values.](image)

Figure 5.4: Speedup factor demonstrated for an entire range of $\tau$ and $N$ values. Points on the grid indicate measured times and every run was done using the Calogero-Sutherland model potential with $\lambda_s = 2$ and $\lambda = 1$. Two GPUs are compared: the fastest (slowest) has a speedup factor of 640 (350). This data was also used to estimate runtime in Figure 5.5.

Often the refinement of a tool is as important as its application. Figure 5.4 shows a speedup factor of over 600 for 128 particles at $\tau = 0.002 \text{K}^{-1}$. Speedup factor is defined in terms of wallclock time: for a given set of parameters, the wallclock time of a target device like the GPUs above is divided by the wallclock time of a single CPU core. This is neglecting equilibration time: the GPU code only needs to equilibrate a single state, which it does at over 600 times the speed of a single CPU core, before then taking measurements of the state again at over 600 times the speed of a single CPU core.
Figure 5.5: Equilibration is costly: the cluster runtime (left) of a single point, evaluated using 2728 CPU cores, is estimated to take 10 times longer than the single GPU runtime (right) of that same point.

The implication is that 600+ CPU cores are required to keep up with the fastest GPU tested, yet only a single core would be capable of equilibrating at 1/600th the speed of that GPU. This is best shown as a comparison of two runs using reasonable parameters in Figure 5.5 where the entire Vermont Advanced Computing Core (VACC) is compared to a single GPU. Equilibration is treated as running on a single device in both cases, while sampling utilizes all devices which happens to be the best case scenario for the CPU. A few of the data points below were significantly more difficult than than the one used in this comparison.

5.3 Calogero-Sutherland Model

Data has been gathered for interaction strengths of $g = 2 \lambda_s (\lambda_s - 1)$ where $\lambda_s = 1.5, 2.0, 3.0, 4.0$, $N = 2, 6, 14, 30, 62$ and $T = 0.1 \text{K} \rightarrow 4.9\text{K}$ in increments of $0.2\text{K}$. All runs were done with $\hbar^2/2mk_B = 1.0$ and the density $n = 0.4 \text{Å}^{-1}$ held constant.

The attentive reader may wonder why the values of $N$ were chosen. The PIMC imple-
mentation developed here can run any value of $N$, however for performance reasons binary values are preferred ($2, 4, 8, 16, 32, 64$ etc). Some of the moves implemented require an empty worldline to grow into utilizing the memory space of two ficticious particles \cite{19, 29}.

Recall the Calogero-Sutherland model, Eq. (2.3),

$$\mathcal{H} = -\sum_{i=1}^{N} \frac{\partial^2}{\partial x_i^2} + g \frac{\pi^2}{L^2} \sum_{i<j}^{N} \left[ \sin \frac{\pi(x_i - x_j)}{L} \right]^{-2},$$

and note that the potential portion is linear in $g$. This allows the extraction of $\langle \mathcal{V} \rangle$

$$\langle \mathcal{V} \rangle = g \left\langle \frac{\partial}{\partial g} \mathcal{H} \right\rangle = g \frac{\partial}{\partial g} \langle \mathcal{H} \rangle = g \frac{\partial}{\partial g} E_N,$$

where $E_N$, Eq. (2.17), is

$$E_N = \frac{1}{3} \frac{\pi^2}{L^2} \lambda_s^2 N (N^2 - 1).$$

Recognition that $\langle \mathcal{T} \rangle = \langle \mathcal{H} \rangle - \langle \mathcal{V} \rangle$, gives

$$\langle \mathcal{V} \rangle = \frac{2}{3} \left( \frac{\lambda_s^2 \pi^2}{2\lambda_s - 1} \right) \frac{N(N^2 - 1)}{L^2} (\lambda_s - 1)$$

$$\langle \mathcal{T} \rangle = \frac{1}{3} \left( \frac{\lambda_s^2 \pi^2}{2\lambda_s - 1} \right) \frac{N(N^2 - 1)}{L^2}.$$

Additional steps provide the energies per particle in the thermodynamic limit. First a factor of $N$ is divided through, followed by the substitution of $N^2/L^2 = n^2$ where $n$ is the particle density. Finally the energies per particle in the thermodynamic limit
are:

\[
\begin{align*}
\langle V \rangle &= \frac{2}{3} \left( \frac{\lambda_s^2 \pi^2 n^2}{2\lambda_s - 1} \right) (\lambda_s - 1) \\
\langle T \rangle &= \frac{1}{3} \left( \frac{\lambda_s^2 \pi^2 n^2}{2\lambda_s - 1} \right).
\end{align*}
\]

These energies provide the necessary bar against which \( \tau \) scaling will be performed: a \( \tau \) at \( T = 0.1 \) K will be sought which reproduces the ground state energies derived above.

### 5.3.1 \( \tau \) Scaling

Possessing an analytic solution to the ground state, \( \tau \) scaling is performed by comparing the energies of the ground state to the Monte Carlo results at a temperature of 0.1 K. The number of particles was set to \( N = 6 \) for all \( \tau \) scaling results: \( \tau \) scaling is considered independent of the number of particles, as long enough are present to fully exercise the potential.

The strongest interaction strength used, \( \lambda_s = 4.0 \), is presented in Figure 5.6
Figure 5.6: Calogero-Sutherland model $\tau$ scaling for $\lambda_s = 4.0$, performed at $N = 6$ with a constant density $n = 0.4 \, \text{Å}^{-1}$. Each point represents the average and standard error of 200k measurements, each corresponding to a "MC bin" of the kinetic (left) or potential (right) energy. The running averages for selected $\tau$ (bottom) show convergence to the expected values (grey lines), with the shaded areas representing the standard error of the final point.

where $\tau = 0.01 \, \text{K}^{-1}$ was observed to reproduce ground state energies within stochastic errorbars. The next strongest interaction strength of $\lambda_s = 3.0$ is shown in in Figure 5.7
Figure 5.7: Calogero-Sutherland model τ scaling for $\lambda_s = 3.0$, performed at $N = 6$ with a constant density $n = 0.4 \text{Å}^{-1}$. Each point represents the average and standard error of 400k measurements, each corresponding to a "MC bin" of the kinetic (left) or potential (right) energy. The running averages for selected $\tau$ (bottom) show convergence to the expected values (grey lines), with the shaded areas representing the standard error of the final point.

and took significantly longer to converge confirming $\tau = 0.004 \text{K}^{-1}$ could be selected.

The number of samples was extended to show that, given a proper value of $\tau$, PIMC will eventually converge. An interaction strength of $\lambda_s = 2.0$ was the smallest interaction strength chosen for a complete analysis. A value of $\tau = 0.0005 \text{K}^{-1}$ was deemed necessary by considering Figure 5.8.
Due to this low value of $\tau$, some of the data points used in the temperature dependent results took weeks to produce.

A pattern emerges: increasingly lower values of $\tau$ are selected as the interaction strength decreases. Note also that as $\tau$ decreases the standard error in kinetic energy increases. Revisiting the energy estimator derivation in section 3.3, this matches our expectations since the kinetic portion of the estimator possesses a factor of $1/\tau^2$.

Finally consider Figure 5.9
Figure 5.9: Calogero-Sutherland model $\tau$ scaling for $\lambda_s = 1.5$, each point run at $T = 0.1 \text{K}$, $N = 6$ and $n = 0.4 \text{Å}^{-1}$. Note that the Monte Carlo results are not convergent to the ground state potential energy even at $\tau = 0.00008 \text{K}^{-1}$, represented by the gray bar: the shaded area represents the standard error of the final point and does not overlap with the gray bar.

where an interaction strength of $\lambda_s = 1.5$ has been chosen. Convergence was not attained at this interaction strength even for the lowest value of $\tau = 0.00008 \text{K}^{-1}$.

Table 5.1 matches each interaction strength to the selected value of $\tau$, including the non-convergent $\lambda_s = 1.5$ interaction strength. The last column indicates how many slices, or high temperature density matrices, are used in the simulation as related to the selected value of $\tau$. The $\lambda_s = 1.5$ results act as motivation for an improved action calculation, discussed in chapter 6. The action used here, which determines whether
<table>
<thead>
<tr>
<th>$\lambda_s$</th>
<th>$\tau$ [K$^{-1}$]</th>
<th>slices</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.0</td>
<td>0.01</td>
<td>1,000</td>
</tr>
<tr>
<td>3.0</td>
<td>0.004</td>
<td>2,500</td>
</tr>
<tr>
<td>2.0</td>
<td>0.0005</td>
<td>20,000</td>
</tr>
<tr>
<td>1.5</td>
<td>0.00008</td>
<td>125,000</td>
</tr>
</tbody>
</table>

Table 5.1: Calogero-Sutherland model interaction strengths $\lambda_s$ and the selected $\tau$ values. The slices column describes the number of high-temperature density matrices, or slices, used per configuration.

a proposed configuration is accepted, is inadequate for $\lambda_s < 2.0$ due to performance considerations: it took over a month to complete a $\tau = 0.0005$ run so a $\tau = 0.00008$ run would take near half a year, and that is just for $\lambda_s = 1.5$.

5.3.2 Temperature Dependence of the Energy

The temperature dependence in the thermodynamic limit is considered. This means looking at the asymptotic behavior shown in Figure 5.10
Figure 5.10: Asymptotic behavior of the Calogero-Sutherland model the thermodynamic limit (the grey bar) is approached: represented points are the reciprocals of $N = 62, 30, 14, 6, 2$ at $T = 0.1\, \text{K}$ and $n = 0.4\, \text{Å}^{-1}$.

as $N$ increases and the density $n$ is held constant.
Next the temperature dependent results are presented in Figure 5.10

\begin{center}
\begin{tabular}{|c|c|}
\hline
$\lambda_s = 4.0$ & $\lambda_s = 3.0$ \\
$\tau = 0.01 K^{-1}$ & $\tau = 0.004 K^{-1}$ \\
\hline
\end{tabular}
\end{center}

**Figure 5.11:** Energy dependence on temperature for the Calogero-Sutherland model. These plots were produced with $N = 62$ and $n = 0.4 \text{Å}^{-1}$. Note the roughness of the lower left plot: this is due to the use of $\tau = 0.0005 K^{-1}$ for these points. The grey bar represents the ground state energy in the thermodynamic limit.
and find that these limits are approached as $N \to 62$ particles for all interaction strengths.

### 5.3.3 Interaction Strength Dependence of Energy

Finally consider the interaction strength: how do the kinetic, potential and total energy change as a function of $\lambda_s$? The ground state in the thermodynamic limit is compared to the $T = 0.1 \text{ K}$ runs with $N = 62$ particles, seen in Figure 5.12.

*Figure 5.12: Interaction Strength dependence of the Calogero-Sutherland model at $T = 0.1 \text{ K}$ and $n = 0.4 \AA^{-1}$, represented by points, compared to the ground state energies in the thermodynamic limit.*
Table 5.2 shows that the lowest interaction strength is within two standard deviations deviations of the thermodynamic limit results at \( T = 0.1 \) K for both kinetic and potential energy and within five standard deviations at all interaction strengths when compared to the \( N = 62 \) analytic results. All points also agree to within several decimal places, and again this is a comparison of the ground state at \( T = 0.0 \) K to the \( T = 0.1 \) K state.

Producing the same ground state plot as Figure 5.12, but using the \( T = 4.9K \) data at the other extreme, Figure 5.13

<table>
<thead>
<tr>
<th>( \lambda )</th>
<th>Source</th>
<th>Kinetic Energy [K]</th>
<th>Potential Energy [K]</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.0</td>
<td>Analytic ( N = 62 )</td>
<td>1.203</td>
<td>7.2170</td>
</tr>
<tr>
<td></td>
<td>Thermodynamic Limit</td>
<td>1.203</td>
<td>7.2189</td>
</tr>
<tr>
<td></td>
<td>Monte Carlo Result</td>
<td>1.202(1)</td>
<td>7.2171(2)</td>
</tr>
<tr>
<td>3.0</td>
<td>Analytic ( N = 62 )</td>
<td>0.947</td>
<td>3.7889</td>
</tr>
<tr>
<td></td>
<td>Thermodynamic Limit</td>
<td>0.947</td>
<td>3.7899</td>
</tr>
<tr>
<td></td>
<td>Monte Carlo Result</td>
<td>0.947(2)</td>
<td>3.7892(1)</td>
</tr>
<tr>
<td>2.0</td>
<td>Analytic ( N = 62 )</td>
<td>0.702</td>
<td>1.4033</td>
</tr>
<tr>
<td></td>
<td>Thermodynamic Limit</td>
<td>0.702</td>
<td>1.4037</td>
</tr>
<tr>
<td></td>
<td>Monte Carlo Result</td>
<td>0.704(5)</td>
<td>1.4038(1)</td>
</tr>
</tbody>
</table>

Table 5.2: A table comparing the Monte Carlo results for the Calogero-Sutherland model at \( N = 62, n = 0.4 \text{ Å}^{-1} \) with \( T = 0.1 \) K to the analytic results in the thermodynamic limit and at \( N = 62, L = 155 \text{ Å}, \) with \( T = 0.0 \) K.
Figure 5.13: Interaction Strength dependence of the Calogero-Sutherland model at $T = 4.9\,\text{K}$ and $n = 0.4\,\text{Å}^{-1}$, represented by points, compared to the ground state energies in the thermodynamic limit.

suggests that kinetic and potential energy have exchanged places in relative magnitude at $\lambda_s = 2.0$. 
Chapter 6

Conclusion and Future Work

The ability to run path integral Monte Carlo on a long range interaction model has been demonstrated, providing an insight into the finite temperature properties of the Calogero-Sutherland model. This was achieved through a speedup factor of over 600x, a factor applied to both the equilibration and sampling stages of the path integral Monte Carlo algorithm.

The implementation itself was developed to be robust and can be generalized: it currently can be used to study any one dimensional model and will be made available to the community at large[23]. With this implementation two issues become apparent: one mentioned earlier being the need for a better action calculation and the other issue being an open question on how to deal with attractive potentials.

Attractive potentials are pathological in path integral Monte Carlo algorithms, however there may be methods to deal with this [32]. The action used, a fourth order approximation called the Generalized Suzuki Factorization, is ill suited for the Sutherland model. It relies on an expansion of the $[T, V]$ commutator as a sign alternating infinite sum. Since only the first term in the expansion is used and the
potential is singular as the separation between particles goes to zero the correction is often larger, and thereby certainly incorrect, than the primitive action. An alternative action calculation is possible using a pair product method which essentially samples the two-body density matrix exactly, approximating all interactions as two-body interactions [28]. While this is a third order method, it doesn’t rely on gradients of the potential as any commutation based correction would.

Once pair product is implemented exploration of the lower interaction strength regime will be possible. Additionally path integral Monte Carlo lends itself well to the estimation of correlation functions, spatial entanglement and the structure factor. Some work has been done regarding entanglement of the Sutherland model [12] in the ground state.

Finally, implementation of a GPU path integral Monte Carlo algorithm with an interaction cutoff length could improve equilibration time significantly, allowing equilibration on a GPU prior to sampling with a significantly more powerful cluster.

Overall there are many avenues to explore. This first step merely confirmed these possibilities: the future is bright.
BIBLIOGRAPHY


Appendix A

Exactly Solvable Models

The Sutherland model has its roots in a one dimensional three-body problem studied by F. Calogero [33]. The author mentions it had been known for some time that problems with harmonic and inverse-square pair interactions are separable and therefore have closed-form solutions [34, 35].

The two-body problem serves as an introduction to exactly solvable models, many details of which apply to the three-body problem. The full details of both are reproduced below before considering an extension of the Calogero model to the N-body problem, called the Sutherland model.

A.1 Calogero’s Two-Body Model

Consider the two-body Hamiltonian

\[ \mathcal{H} = -\frac{\partial^2}{\partial x_1^2} - \frac{\partial^2}{\partial x_2^2} + \frac{1}{8} \omega^2 (x_1 - x_2)^2 + \frac{g}{(x_1 - x_2)^2} \]  

(A.1)
where there is a harmonic interaction between the two particles, serving to confine them, and a $1/r^2$ potential with interaction strength $g$ as in Figure A.1. Note that

![Figure A.1: A schematic of the Calogero two-body model where the harmonic interaction is represented as a spring with strength $k$ which pulls the particles together and the inverse-square interaction as a rod with interaction strength $g$ which pushes the particles apart. Note that the particles are confined to move on the straight rod and there are no periodic boundary conditions.](image)

$h^2/2m = 1$ for this and all following derivations.

The goal being to obtain the eigenvalues and eigenfunctions of this Hamiltonian, the first step is moving to center of mass coordinates

$$R = \frac{1}{2}(x_1 + x_2) \quad x = \frac{1}{\sqrt{2}}(x_1 - x_2)$$

and applying the kinetic energy operator to a dummy function $f(x_i)$. The chain rule

$$\frac{\partial f}{\partial x_i} = \frac{\partial R}{\partial x_i} \frac{\partial f}{\partial R} + \frac{\partial x}{\partial x_i} \frac{\partial f}{\partial x}$$

leads to

$$\frac{\partial^2 f}{\partial x_i^2} = \frac{\partial^2 R}{\partial x_i^2} \frac{\partial f}{\partial R} + \left( \frac{\partial R}{\partial x_i} \right)^2 \frac{\partial^2 f}{\partial R^2} + \frac{\partial^2 x}{\partial x_i^2} \frac{\partial f}{\partial x} + \left( \frac{\partial x}{\partial x_i} \right)^2 \frac{\partial^2 f}{\partial x^2} + 2 \frac{\partial x}{\partial x_i} \frac{\partial R}{\partial x_i} \frac{\partial^2 f}{\partial x \partial R}$$
which motivates a refactoring of the Hamiltonian in Eq. (A.1) as

$$\mathcal{H} = -\frac{\partial^2}{\partial x^2} + \frac{1}{4}\omega^2 x^2 + \frac{1}{2}gx^{-2}$$  \hspace{1cm} (A.2)

where the center of mass term is discarded. While this is unnecessary in the two body problem, which is completely separable into a center of mass coordinate and a relative coordinate, the two body problem serves as an introduction to the more complex problems: there is no interest in the center of mass term.

A two body problem has now become a one body problem in the center of mass frame. This is a second order differential equation and so

$$\Psi'' = \left(\frac{1}{4}\omega^2 x^2 + \frac{g}{2x^2} - E\right)\Psi.$$  \hspace{1cm} (A.3)

Examining the asymptotic behavior, there are two regimes of interest: $0 \leq |x| \ll 1$ and $|x| \gg 1$. Considering $|x| \gg 1$ first,

$$\Psi'' \approx \frac{1}{4}\omega^2 x^2 \Psi$$

which lends itself to solutions of the (approximate) form

$$\Psi \approx Ae^{\omega x^2/4} + Be^{-\omega x^2/4}.$$  

The $A$ term is not normalizable. Using the $B$ term a wavefunction of the form

$$\Psi(x) = h(x) \exp\left(-\omega x^2/4\right)$$  \hspace{1cm} (A.4)
is proposed which, taking the first and second derivatives

\[
\Psi' = \left[ \frac{h'}{h} - \frac{1}{2} \omega x \right] \Psi
\]

\[
\Psi'' = \left[ \frac{h''}{h} - \frac{1}{2} \omega + \frac{1}{4} \omega^2 x^2 - \frac{h'}{h} \omega x \right] \Psi
\]

and substituting these back into Eq. (A.3) leaves

\[
\frac{h''}{h} - \frac{1}{2} \omega - \frac{h'}{h} \omega x = \frac{g}{2x^2} - E
\]

leading to a new differential equation

\[
h'' = h' \omega x + \left( \frac{1}{2} \omega + \frac{g}{2x^2} - E \right) h.
\] (A.5)

Some asymptotic behavior of our model has been recognized and incorporated directly into the wavefunction: while this was motivated by an approximation, there exists an \( h(x) \) in Eq. (A.4) for which it is exact. The process is repeated, beginning with Eq. (A.5), by now considering \( 0 \leq |x| \ll 1 \)

\[
h'' \approx \frac{g}{2x^2} h
\]

with solutions

\[
h(x) = Ax^{a+1/2}
\]

where the proposed form of \( h(x) \) need not be normalizable: Eq. (A.4) must meet that condition. Note that \( a \) is arbitrary here: it generalizes the proposed \( h(x) \).

Recall also that \( x \) is the separation between two particles: generally speaking
$|\psi(x)|^2$ and $\psi(x)\psi'(x)$ must be continuous for a wave function to be considered physically acceptable [33], and since $x$ is a separation it can be zero. The parameter $a$ is thus restricted as $a \geq 0$, except for the special case of $a = \pm 1/2$ resulting in an interaction strength $g = 0$, and reproducing the harmonic oscillator as shown later.

With this in mind, an $h(x)$ and the corresponding wavefunction is proposed

$$h(x) = q(x)x^{a+1/2}$$

$$\Psi(x) = q(x)x^{a+1/2}e^{-\omega x^2/4}$$

(A.6)

with first and second derivates (in $h(x)$)

$$h' = \left[ \frac{q'}{q} + \frac{a + 1/2}{x} \right] h$$

$$h'' = \left[ \frac{q''}{q} + \frac{a^2 - 1/4}{x^2} + 2\frac{q' a + 1/2}{q} \right] h$$

which when substituted back into Eq. (A.5) to yield

$$\frac{q''}{q} + \frac{a^2 - 1/4}{x^2} + 2\frac{q' a + 1/2}{q} = wx \left[ \frac{q'}{q} + \frac{a + 1/2}{x} \right] + \frac{1}{2} \omega + \frac{g}{2x^2} - E.$$ 

Some simplification leads to

$$xq'' + \left[ 2(a + 1/2) - wx^2 \right] q' + \left[ E - wa - \omega \right] xq + \left[ (a^2 - 1/4) - \frac{1}{2} g \right] x^{-1}q = 0$$

(A.7)

from which a constraint, relating interaction strength $g$ with a dimensionless constant $a$

$$a = \pm \frac{1}{2} (2g + 1)^{1/2} \quad g = 2a^2 - \frac{1}{2}$$

(A.8)
is found.

Even having zeroed the third term in Eq. (A.7) using this constraint, after stripping off the asymptotic behavior in both the small $x$ and large $x$ regimes, there remains the quadratic term $\omega x^2$ to deal with. This suggests a substitution

$$u = \frac{1}{2} \omega x^2 \quad \frac{du}{dx} = \omega x \quad \frac{d^2 u}{dx^2} = \omega$$

with the chain rule providing

$$\frac{dq}{dx} = \omega x \frac{dq}{du}$$
$$\frac{d^2 q}{dx^2} = \omega \frac{dq}{du} + 2u\omega \frac{d^2 q}{du^2}.$$  

Substituting these back into Eq. (A.7) and simplifying yields

$$uq'' + (a + 1 - u)q' + \frac{1}{2}(E/\omega - a - 1)q = 0$$  

(A.9)

which is in the form of the generalized Laguerre polynomials

$$uq'' + (a + 1 - u)q' + nq = 0, \quad q(u) = L^a_n.$$  

(A.10)

The eigenstates and energies of this two-body problem are now described as

$$\Psi_n(x) = L^a_n \left( \frac{1}{2} \omega x^2 \right) x^{a+1/2} e^{-\omega x^2/4}$$

$$E_n = \omega(2n + a + 1), \quad n = 0, 1, 2, 3 \cdots.$$  

(A.11)

(A.12)

Note what happens when $a = \pm 1/2$: $g$, the strength of the inverse-square inter-
action, is zero and leaves a Hamiltonian

\[ \mathcal{H} = -\frac{\partial^2}{\partial x^2} + \frac{1}{4} \omega^2 x^2 \]

with eigen states

\[ \Psi_{2n}(x) = L_{n}^{-1/2} \left( \frac{1}{2} \omega x^2 \right) e^{-\omega x^2 / 4} \]
\[ \Psi_{2n+1}(x) = x L_{n}^{1/2} \left( \frac{1}{2} \omega x^2 \right) e^{-\omega x^2 / 4} \]

and energies

\[ E_{2n} = \omega(2n + 1/2) \]
\[ E_{2n+1} = \omega(2n + 1 + 1/2). \]

These even and odd eigenstates can be thought of as the even and odd rungs of a ladder which, when related to the Hermite polynomials through

\[ H_{2n}(x) \propto L_{n}^{-1/2}(x^2) \]
\[ H_{2n+1}(x) \propto x L_{n}^{-1/2}(x^2), \]

are the full set of states belonging to the Harmonic oscillator with energies

\[ E_m = \omega(m + 1/2) \]

where the even rungs of the ladder are \( 2n = m \) and the odd rungs are \( 2n + 1 = m \) from Eq. (A.13) and (A.14).
A.2 Calogero’s Three-Body Model

![Figure A.2: A schematic of the Calogero three-body model where the springs with strength $k$ represent harmonic interactions and the inverse-square interactions of strength $g$ are represented by rods connecting the particles, where the particles are confined to move on the straight rod.](image)

The Hamiltonian of the schematic in Figure A.2 is an extension of the two body model

$$
\mathcal{H} = -\frac{\partial^2}{\partial x_1^2} - \frac{\partial^2}{\partial x_2^2} - \frac{\partial^2}{\partial x_3^2} + \frac{1}{8} \omega^2 ((x_1 - x_2)^2 + (x_2 - x_3)^2 + (x_3 - x_1)^2) \\
+ \frac{g_3}{(x_1 - x_2)^2} + \frac{g_1}{(x_2 - x_3)^2} + \frac{g_2}{(x_3 - x_1)^2} \tag{A.15}
$$

where $\hbar^2/2m = 1$.

Start with the methods used in the two body solution by moving to the center of mass frame and treating these as like-mass particles. This is done using Jacobi coordinates

$$
R = \frac{1}{N} \sum_{k=1}^{N} x_k \\
r_i = \frac{1}{i} \sum_{k=1}^{i} x_k - x_{i+1}, \quad i = 1, 2, ..., N - 1
$$
which would yield

\[
R = \frac{1}{3}(x_1 + x_2 + x_3)
\]

\[
r_1 = x_1 - x_2
\]

\[
r_2 = \frac{1}{2}(x_1 + x_2) - x_3
\]

however substitutions of this nature can carry any coefficients found convenient, and so Calogero’s coordinates \cite{33}

\[
R = \frac{1}{3}(x_1 + x_2 + x_3)
\]

\[
x = \frac{1}{\sqrt{2}}(x_1 - x_2)
\]

\[
y = \frac{1}{\sqrt{6}}(x_1 + x_2 - 2x_3)
\]

are used.

These can be rearranged to give the relative coordinates in terms of \(x\) and \(y\)

\[
x_1 - x_2 = \sqrt{2}x
\]

\[
x_2 - x_3 = -\frac{\sqrt{2}}{2}(x - \sqrt{3}y)
\]

\[
x_3 - x_1 = -\frac{\sqrt{2}}{2}(x + \sqrt{3}y)
\]
and the partials work out to

\[
\begin{align*}
\frac{\partial^2}{\partial x_1^2} &= \frac{1}{9} \frac{\partial^2}{\partial R^2} + \frac{1}{2} \frac{\partial^2}{\partial x^2} + \frac{1}{6} \frac{\partial^2}{\partial y^2} \\
\frac{\partial^2}{\partial x_2^2} &= \frac{1}{9} \frac{\partial^2}{\partial R^2} + \frac{1}{2} \frac{\partial^2}{\partial x^2} + \frac{1}{6} \frac{\partial^2}{\partial y^2} \\
\frac{\partial^2}{\partial x_3^2} &= \frac{1}{9} \frac{\partial^2}{\partial R^2} + \frac{4}{3} \frac{\partial^2}{\partial y^2}.
\end{align*}
\]

The resulting Hamiltonian

\[
\mathcal{H} = -\frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2} + \frac{1}{8} \omega^2 (x^2 + y^2) + \frac{g_3}{2x^2} + \frac{2g_1}{(\sqrt{3}y - x)^2} + \frac{2g_2}{(\sqrt{3}y + x)^2}
\]

(A.16)

is relieved of a single degree of freedom since it has been moved to the center of mass frame and, assuming no acceleration, this is now a two-body problem.

Next perform a 2d-‘polar’ coordinate transform with

\[
\begin{align*}
r^2 &= x^2 + y^2 \\
x &= r \sin \phi \\
y &= r \cos \phi \\
\phi &= \arctan(x/y).
\end{align*}
\]

The \(\sqrt{3}y \pm x\) terms in Eq. (A.16) can be formulated like so

\[
\begin{align*}
\sqrt{3}y - x &= 2r \left[ -\cos\left(\frac{2\pi}{3}\right) \cos(\phi) - \sin\left(\frac{2\pi}{3}\right) \sin(\phi) \right] = -2r \sin\left(\phi + \frac{2\pi}{3}\right) \\
\sqrt{3}y + x &= 2r \left[ \cos\left(\frac{\pi}{3}\right) \cos(\phi) + \sin\left(\frac{\pi}{3}\right) \sin(\phi) \right] = -2r \sin\left(\phi + \frac{4\pi}{3}\right)
\end{align*}
\]
by using the sine and cosine addition identities. Finally including the partials

\[
\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} = \frac{\partial^2 r}{\partial x^2 \partial r} + \frac{\partial^2 r}{\partial y^2 \partial r} + \frac{\partial^2 \phi}{\partial x^2 \partial \phi} + \frac{\partial^2 \phi}{\partial y^2 \partial \phi} + \\
\left( \frac{\partial r}{\partial x} \right)^2 \frac{\partial^2}{\partial r^2} + \left( \frac{\partial r}{\partial y} \right)^2 \frac{\partial^2}{\partial r^2} + \left( \frac{\partial \phi}{\partial x} \right)^2 \frac{\partial^2}{\partial \phi^2} + \left( \frac{\partial \phi}{\partial y} \right)^2 \frac{\partial^2}{\partial \phi^2}
\]

\[
= x^2 + y^2 \frac{\partial^2}{\partial \phi^2} + \frac{x^2 + y^2 \frac{\partial^2}{\partial r^2} + \frac{x^2 + y^2 \partial}{(x^2 + y^2)^{3/2} \partial r}}{r^2 \partial^2} + \frac{1}{r^2} \frac{\partial^2}{\partial r^2}
\]

leaves

\[
\mathcal{H} = -\frac{\partial^2}{\partial r^2} + \frac{3}{r^2} \frac{\omega^2 r^2}{8} + M
\]

\[
\mathcal{M} = -\frac{\partial^2}{\partial \phi^2} + \frac{1}{2} \left[ \frac{g_3}{\sin^2 \phi} + \frac{g_1}{\sin^2(\phi + \frac{2\pi}{3})} + \frac{g_2}{\sin^2(\phi + \frac{4\pi}{3})} \right]
\]

where \(M\) is a separable portion with its own eigenvalues.

Tackling the radial portion first, consider the asymptotic behavior at \(r \gg 1\) by searching for solutions to

\[
\left( -\frac{\partial^2}{\partial r^2} + \frac{3}{8} \omega^2 r^2 \right) R(r) = 0
\]

which take the (approximate) form

\[
R(r) \approx Ae^{-\frac{1}{2}kr^2}, \quad k = \sqrt{\frac{3}{8} \omega}.
\]
Now incorporate this asymptotic behavior into the wave equation

\[
R(r) = h(r)e^{-\frac{1}{2}kr^2}
\]

\[
\frac{\partial R}{\partial r} = \left(\frac{h'}{h} - kr\right)R
\]

\[
\frac{\partial^2 R}{\partial r^2} = \left(\frac{h''}{h} - 2kr\frac{h'}{h} + k^2r - k\right)R
\]

to arrive at a simplified differential equation

\[
h'' = h' \left(2kr - \frac{1}{r}\right) + h \left(2k - E + \frac{M}{r^2}\right).
\]

Next consider the \( r \ll 1 \) regime with differential equation

\[
h'' \approx \frac{M}{r^2}h = \frac{b_i^2}{r^2}h
\]

where \( b_i^2 \) are eigenvalues of \( M \). This has an approximate solution \( h \approx Ar^{b_i} \) which is incorporated into \( h(r) \)

\[
h(r) = q(r)r^{b_i}
\]

\[
\frac{\partial h}{\partial r} = \left(\frac{q'}{q} + \frac{b_i}{r}\right)h
\]

\[
\frac{\partial^2 h}{\partial r^2} = \left(\frac{q''}{q} + \frac{b_i(b_i - 1)}{r^2} + 2b_i\frac{1}{rq'}\right)h
\]

and, after some simplification, produces the differential equation

\[
q'' + \left(2\frac{b_i}{r} - 2kr + \frac{1}{r}\right)q' - (2k(b_i + 1) - E) = 0.
\]
After a coordinate transform with \( u = kr^2 \), giving us the final form of the radial differential equation

\[
uf'' + (b_l + 1 - u)f' - \left[ (b_l + 2)/2 - \frac{E}{4k} \right] f = 0
\]

which has a solution in terms of laguerre polynomials with \( n = -(b_l + 1)/2 + E/4k \).

The eigen-energies of \( \mathcal{H} \) are therefore

\[
E = \left[ \frac{1}{2}(b_l + 1) + n \right] 4k = \sqrt{\frac{3}{2}} \omega (b_l + 1 + 2n).
\] (A.19)

The angular portion is not so straight forward. Begin by looking at the (presumably) simpler problem with \( g_1 = g_2 = 0 \) and \( g_3 = g \). The operator

\[
\mathcal{M} = -\frac{\partial^2}{\partial \phi^2} + \frac{g}{2 \sin^2 \phi}
\]

is simplified yet again by considering the asymptotic behavior using the small angle approximation \( \sin(\phi) \rightarrow \phi \), and yields

\[
\mathcal{M} f(\phi) \approx \left[ -\frac{\partial^2}{\partial \phi^2} + \frac{g}{2 \phi^2} \right] f(\phi) = b_l^2 f(\phi).
\]

A solution to this differential equation is

\[
f(\phi) = A\phi^{a+1/2}
\]
\[ \frac{\partial f}{\partial \phi} = \frac{a + 1/2}{\phi} f \]
\[ \frac{\partial^2 f}{\partial \phi^2} = \frac{(a + 1/2)(a - 1/2)}{\phi^2} f \]

which, for small enough \( \phi \), is exact and details the relation between \( a \) and \( g \) as

\[ \frac{g}{2} = (a + \frac{1}{2})(a - \frac{1}{2}). \quad (A.20) \]

A trial eigenfunction of

\[ f(\phi) = h(\phi)(\sin \phi)^{a+1/2} \]
\[ \frac{\partial f}{\partial \phi} = \left[ \frac{h'}{h} + (a + 1/2) \cot \phi \right] f \]
\[ \frac{\partial^2 f}{\partial \phi^2} = \left[ \frac{h''}{h} + 2 \frac{h'}{h} (a + 1/2) \cot \phi + \frac{(a + 1/2)(a - 1/2)}{\sin^2 \phi} - (a + 1/2)^2 \right] f \]

when used with Eq. (A.20), produces

\[ \frac{\partial^2}{\partial \phi^2} + (2a + 1) \cot \phi \frac{\partial}{\partial \phi} - (a + 1/2)^2 = b_i^2. \]

While this doesn’t appear particularly simplified, a final coordinate transform \( u = \sin^2 \phi \) gives

\[ u(1 - u) \frac{\partial^2 f}{\partial u^2} + \left[ \frac{1}{2} - u + \left( \frac{1}{2} + a \right)(1 - u) \right] \frac{\partial f}{\partial u} - \frac{1}{4} [(a + 1/2)^2 - b_i^2] = 0, \]
solutions to which are the hypergeometric function $F(A,B;C;u)$ with parameters

\begin{align*}
A &= \frac{1}{2}(a + \frac{1}{2} - b_l) \\
B &= \frac{1}{2}(a + \frac{1}{2} + b_l) \\
C &= 1 + a \\
u &= \sin^2 \phi.
\end{align*}

The eigenfunctions of $\mathcal{M}$ are evidently

\[ f(\phi) = (\sin \phi)^{a+1/2} F\left(\frac{1}{2}(a + \frac{1}{2} - b_l), \frac{1}{2}(a + \frac{1}{2} - b_l); 1 + a; \sin^2 \phi \right). \]

Note that they are singular for non-negative, non-integer, values of $A$ as described in Eq. (A.21). This provides the final constraint necessary to find the eigenvalues of Eq. (A.18). Using $m$ to denote the second quantum number, the following identity \cite{36}

\[ C_{2m}^\lambda(x) = AF(-m, m + \lambda; \lambda + 1/2; 1 - x^2) \]

gives $f(\phi)$ in terms of Gegenbauer polynomials for the even states with

\[ 2m = b_l - a - 1/2 = l \]

eigenvalues.
Getting at the odd states requires two additional identities \[36\]

\[ F(A, B; C; t) = (1 - t)^{C-A-B} F(C - B, C - A; C; t) \]

\[ C_{2m+1}^\lambda(x) = Ax F(-m, m + \lambda + 1; \lambda + 1/2; 1 - x^2) \]

resulting in a second expression for the eigenfunctions

\[ f(\phi) = \cos \phi (\sin \phi)^{a+1/2} F\left(\frac{1}{2}(a + \frac{3}{2} - b_l), \frac{1}{2}(a + \frac{2}{2} - b_l); 1 + a; \sin^2 \phi \right) \]

and the odd eigenvalues

\[ 2m + 1 = b_l - a - 1/2 = l. \quad (A.26) \]

Using Eq. (A.25) and (A.26) in terms of \( l \) instead of \( m \), the eigenvalues and eigenfunctions of \( \mathcal{M} \) are

\[ f(\phi) = (\sin \phi)^{a+1/2} C_{l+1/2}^a(\cos \phi) \]

\[ b_l = a + l + 1/2. \]

Note that this solution is only on the interval \( 0 \leq \phi \leq \pi \). Displacing \( \phi \) by \( \pi \) and considering exchange symmetry to extends it to the full range, however the eigenvalues are unaffected.

Substituting the above eigenvalues into Eq. (A.19) in place of \( b_l \), the eigenenergies
for the three-body model are

$$E_{l+2n} = \sqrt{\frac{3}{2}} \omega (a + l + 2n + 3/2).$$

Recall however that this was the for the special case of $g_1 = g_2 = 0, g_3 = 1$, effectively turning off the interactions between all but a pair of particles. This is extended to general $g_1 = g_2 = g_3 = g$ case by an obscure identity [33]

$$\frac{1}{\sin^2 \phi} + \frac{1}{\sin^2 (\phi + 2\pi/3)} + \frac{1}{\sin^2 (\phi + 4\pi/3)} = \frac{9}{\sin^2 (3\phi)}$$

which when substituted into Eq. (A.18) produces

$$\mathcal{M} = -\frac{\partial^2}{\partial \phi^2} + \frac{1}{2} \left( \frac{9g}{\sin^2 (3\phi)} \right).$$

Evidently $B_l = 3b_l$, where $B_l$ are eigenvalues for the angular portion of the full three-body problem. Hence one final substitution and the energies of the equal interaction-strength three-body problem

$$E_{3l+2n} = \sqrt{\frac{3}{2}} \omega (3l + 2n + 3a + 5/2).$$

Calogero suggested that the number of particles in these models might be further extended, which is precisely what Bill Sutherland did [8] by switching to a harmonic well, rather than continuing with harmonic interactions, as a method of confining the particles.
A.3 Sutherland N-Body Model

The Calogero model was extended by Bill Sutherland through substitution of a harmonic well for the harmonic interaction, as shown in Figure A.3. This N-body model is described by the Hamiltonian

$$\mathcal{H} = -\sum_{i=1}^{N} \frac{\partial^2}{\partial x_i^2} + \sum_{i<j} \frac{g}{(x_i - x_j)^2} + \omega^2 \sum_{i=1}^{N} x_i^2, \quad (A.27)$$

is long range, many body and has closed form solutions to the ground state energy for any number of particles.

The wave function is of the Bijl-Dingle-Jastrow [37] type

$$\Psi = \exp \left( -\frac{1}{2} \omega \sum_{i=1}^{N} x_i^2 \right) \prod_{i<j}^{N} |x_i - x_j|^\lambda, \quad (A.28)$$
or product in form. Splitting it into two functions

\[
\phi = \prod_{i<j}^N |x_i - x_j|^{\lambda_s}
\]

\[
\varphi = \exp \left( -\frac{1}{2} \omega \sum_{i=1}^{N} x_i^2 \right)
\]

allows each to be treated separately. This notation will be used to arrive at the energies for the \( N = 2, 3 \) and 4 cases, motivating the general \( N \) particle solution.

### A.3.1 \( N = 2 \)

The nature of the Sutherland model solution is that the kinetic portion of the Hamiltonian partially cancels the potential portion. The bulk of the work is present in the derivation of the kinetic term from the Bethe Ansatz, or guess of the wave-function’s form.

Expand the kinetic portion of Eq. (A.27) in terms of \( \phi \) and \( \varphi \)

\[
-\sum_i^2 \frac{\partial^2}{\partial x_i^2} \Psi = -\left( \frac{\partial^2 \phi}{\partial x_1^2} + \frac{\partial^2 \phi}{\partial x_2^2} \right) \varphi - 2 \left( \frac{\partial \phi}{\partial x_1} \frac{\partial \varphi}{\partial x_1} + \frac{\partial \phi}{\partial x_2} \frac{\partial \varphi}{\partial x_2} \right) - \left( \frac{\partial^2 \varphi}{\partial x_1^2} + \frac{\partial^2 \varphi}{\partial x_2^2} \right) \phi.
\]

(A.28)

Next take the first derivates of \( \phi \) and \( \varphi \) with respect to \( x_1 \) and \( x_2 \)

\[
\frac{\partial \phi}{\partial x_1} = \lambda_s \phi \frac{(x_1 - x_2)}{|x_1 - x_2|^2}
\]

\[
\frac{\partial \varphi}{\partial x_1} = -\omega x_1 \varphi
\]

\[
\frac{\partial \phi}{\partial x_2} = -\lambda_s \phi \frac{(x_1 - x_2)}{|x_1 - x_2|^2}
\]

\[
\frac{\partial \varphi}{\partial x_2} = -\omega x_2 \varphi,
\]
and the second derivatives,

\[
\frac{\partial^2 \phi}{\partial x_1^2} = \lambda_s(\lambda_s - 1) \frac{1}{(x_1 - x_2)^2} \phi
\]

\[
\frac{\partial^2 \varphi}{\partial x_1^2} = \omega (\omega x_1^2 - 1) \varphi
\]

and

\[
\frac{\partial^2 \phi}{\partial x_2^2} = \lambda_s(\lambda_s - 1) \frac{1}{(x_1 - x_2)^2} \phi
\]

\[
\frac{\partial^2 \varphi}{\partial x_2^2} = \omega (\omega x_2^2 - 1) \varphi.
\]

The first term in Eq. (A.28) then evaluates to

\[
\left( \frac{\partial^2 \phi}{\partial x_1^2} + \frac{\partial^2 \phi}{\partial x_2^2} \right) \varphi = 2\lambda_s(\lambda_s - 1) \frac{1}{(x_1 - x_2)^2} \Psi,
\]

while the third term yields

\[
\left( \frac{\partial^2 \varphi}{\partial x_1^2} + \frac{\partial^2 \varphi}{\partial x_2^2} \right) \phi = \left( \omega^2 (x_1^2 + x_2^2) - 2\omega \right) \Psi.
\]

The second term is the most interesting. As the number of particles increases, the number of cross terms, is \(N(N-1)\). Their evaluation

\[
2 \left( \frac{\partial \phi}{\partial x_1} \frac{\partial \varphi}{\partial x_1} + \frac{\partial \phi}{\partial x_2} \frac{\partial \varphi}{\partial x_2} \right) = -2\lambda_s \omega \frac{(x_1 - x_2)^2}{|x_1 - x_2|^2} \Psi
\]

leaves a multiple of \(-2\lambda_s \omega\) in the two particle case.
The kinetic term for the two particle case becomes

\[ T = -2\lambda_s(\lambda_s - 1) \frac{1}{(x_1 - x_2)^2} + 2\omega(\lambda_s + 1) - \omega^2(x_1^2 + x_2^2) \]

which can be added to the potential term

\[ V = g \frac{1}{(x_1 - x_2)^2} + \omega^2(x_1^2 + x_2^2) \]

to produce

\[ [g - 2\lambda_s(\lambda_s - 1)] \frac{1}{(x_1 - x_2)^2} \Psi + 2\omega(\lambda_s + 1)\Psi = E\Psi. \]

There are two equations of interest here. One relates the interaction strength \( g \) with \( \lambda_s \), while the second yields the energies

\[ g = 2\lambda_s(\lambda_s - 1) \]
\[ E = 2\omega(\lambda_s + 1). \]

A.3.2 \( N = 3 \)

The three and four particle cases benefit from greater abstraction, so some additional notation is used while refining the notation for the two particle case. The steps taken by Sutherland differ slightly here: the notation is similar to that used in his more recent paper for the periodic model [10].

Begin again with the kinetic portion, expressed in terms of \( \phi \) and \( \varphi \), where \( N = 3 \)

\[ - \sum_{i=1}^{N} \frac{\partial^2}{\partial x_i^2} \Psi = - \sum_{i=1}^{3} \frac{\partial^2 \phi}{\partial x_i^2} \varphi - 2 \sum_{i=1}^{N} \frac{\partial \phi}{\partial x_i} \frac{\partial \varphi}{\partial x_i} - \sum_{i=1}^{N} \frac{\partial^2 \varphi}{\partial x_i^2} \phi, \]  

(A.29)
and express $\phi$ in terms of $\psi$

$$\phi = [\psi_{12}\psi_{13}\psi_{23}]^{\lambda_s}, \quad \psi_{ij} = |x_i - x_j|.$$  

Taking the first derivative of $\phi$ with respect to $x_1$ yields

$$\frac{\partial \phi}{\partial x_1} = \lambda_s \phi \frac{1}{\psi_{12}\psi_{13}\psi_{23}} \left[ \psi'_{12}\psi_{13}\psi_{23} + \psi_{12}\psi'_{13}\psi_{23} \right]$$

where further simplification leads to

$$\frac{\partial \phi}{\partial x_1} = \lambda_s \phi [\theta_{12} + \theta_{13}], \quad \theta_{ij} = \frac{\psi'_{ij}}{\psi_{ij}}.$$  

Consider the derivative of $\psi_{ij}$ with respect to $x_i$

$$\frac{\partial \psi_{ij}}{\partial x_i} = \frac{x_i - x_j}{|x_i - x_j|}$$

and then with respect to $x_j$

$$\frac{\partial \psi_{ij}}{\partial x_j} = -\frac{x_i - x_j}{|x_i - x_j|}$$

where the only difference is a negative sign. Applying this to $\theta_{ij}$ from Eq. (A.30) yields a convenient relationship

$$\theta_{ji} = \frac{\psi'_{ji}}{\psi_{ji}} = -\frac{\psi'_{ij}}{\psi_{ij}} = -\theta_{ij},$$

convenient because it will allow all derivatives to be taken with respect to the first
The remaining partials are thus
\[ \frac{\partial \phi}{\partial x_2} = \lambda_s \phi[-\theta_{12} + \theta_{23}] \]
\[ \frac{\partial \phi}{\partial x_3} = \lambda_s \phi[-\theta_{13} - \theta_{23}] . \]

The second derivatives further highlight the importance of these signs
\[ \frac{\partial^2 \phi}{\partial x_1^2} = \lambda_s^2 (\theta_{12}^2 + \theta_{13}^2 + 2\theta_{12} \theta_{13}) \phi + \lambda_s (\theta'_{12} + \theta'_{13}) \phi \]
\[ \frac{\partial^2 \phi}{\partial x_2^2} = \lambda_s^2 (\theta_{12}^2 + \theta_{23}^2 - 2\theta_{12} \theta_{23}) \phi + \lambda_s (\theta'_{12} + \theta'_{23}) \phi \]
\[ \frac{\partial^2 \phi}{\partial x_3^2} = \lambda_s^2 (\theta_{13}^2 + \theta_{23}^2 + 2\theta_{13} \theta_{23}) \phi + \lambda_s (\theta'_{13} + \theta'_{23}) \phi \]

since a cross term is now negative. This will allow the cross terms to cancel, however first the above is simplified into a single sum while noting that
\[ \theta_{ij} = \frac{(x_i - x_j)}{|x_i - x_j|^2} \quad \theta'_{ij} = -\frac{1}{(x_i - x_j)^2} \]
thus arriving at
\[ \sum_{i=1}^{3} \frac{\partial^2 \phi}{\partial x_i^2} \varphi = 2\lambda_s (\lambda_s - 1) \sum_{i<j} \frac{1}{(x_i - x_j)^2} \varphi \phi \quad (A.31) \]
where the indices \( j \) of \( x_j \) have been chosen such that \( x_1 \geq x_2 \geq x_3 \), allowing the cross terms
\[ 2\lambda_s^2 \phi [\theta_{12} \theta_{13} - \theta_{12} \theta_{23} + \theta_{13} \theta_{23}] \]
to be simplified

\[
2\lambda^2 s \frac{\phi_{12} \theta_{13} \theta_{23}}{\theta_{12} \theta_{13} \theta_{23}} [\theta_{12} \theta_{13} - \theta_{12} \theta_{23} + \theta_{13} \theta_{23}]
\]

\[
2\lambda^2 s \phi_{12} \theta_{13} \theta_{23} \left[ \frac{|x_2 - x_3|^2}{(x_2 - x_3)} - \frac{|x_1 - x_3|^2}{(x_1 - x_3)} + \frac{|x_1 - x_2|^2}{(x_1 - x_2)} \right]
\]

\[
2\lambda^2 s \phi_{12} \theta_{13} \theta_{23} \left[ (x_2 - x_3) - (x_1 - x_3) + (x_1 - x_2) \right]
\]

and cancel.

The second term in Eq. (A.29) simplifies to

\[
2 \sum_{i=1}^{3} \frac{\partial \phi}{\partial x_i} \frac{\partial \varphi}{\partial x_i} = -6\lambda s \omega \phi \varphi,
\]

(A.32)

while the final term in Eq. (A.29) yields

\[
\sum_{i=1}^{3} \frac{\partial^2 \varphi}{\partial x_i^2} = \omega^2 \sum_{i=1}^{3} x_i^2 \phi \varphi - 3\omega \phi \varphi.
\]

(A.33)

Eq. (A.31) to (A.33) combined with the potential portion of the Hamiltonian, give

\[
[g - 2\lambda s (\lambda s - 1)] \sum_{i < j}^{4} \frac{1}{(x_1 - x_2)^2} \Psi + \omega (6\omega + 3) \Psi = E \Psi
\]

from which the energy is found to be

\[
E = 3\omega + 6\lambda s \omega.
\]
A.3.3  $N = 4$

The four particle case follows the three, however there are many more cross terms in the $\frac{\partial^2 \phi}{\partial x^2}$ portion. Focusing only on these $\theta_{ij}$ terms

$$\sum_{i=1}^{4} \frac{\partial^2 \phi}{\partial x_{i}^2} = \sum_{i<j} 2\lambda_s (\lambda_s - 1) \frac{1}{(x_i - x_j)^2} + 2\lambda_s^2 (\theta_{12}\theta_{13} + \theta_{12}\theta_{14} + \theta_{13}\theta_{14} + \theta_{12}\theta_{23} + \theta_{12}\theta_{24} + \theta_{23}\theta_{24} + \theta_{13}\theta_{34} + \theta_{23}\theta_{34} + \theta_{14}\theta_{24} + \theta_{14}\theta_{34} + \theta_{24}\theta_{34}).$$

each is multiplied by its remainder of missing terms and, after some simplification, it is found that

$$2\lambda_s^2 \theta_{12}\theta_{13}\theta_{23}\theta_{24}\theta_{34} [(x_1 - x_2)(x_1 - x_3)(x_1 - x_4)(x_2 - x_3 - x_2 + x_4 + x_3 - x_4)
 + (x_1 - x_4)(x_2 - x_4)(x_3 - x_4)(x_1 - x_3 - x_1 + x_2 + x_2 - x_3)
 + (x_1 - x_2)(x_2 - x_3)(x_2 - x_4)(x_1 - x_4 - x_1 + x_3 + x_3 - x_4)
 + (x_1 - x_3)(x_2 - x_3)(x_3 - x_4)(x_2 - x_4 - x_1 + x_4 + x_1 - x_2)] = 0.$$ 

The final expression is

$$[g - 2\lambda_s (\lambda_s - 1)] \sum_{i<j}^4 \frac{1}{(x_1 - x_2)^2} \Psi + \omega (12\lambda_s + 4) \Psi = E \Psi$$

from which the energy is found to be

$$E = 4\omega + 12\lambda_s \omega.$$
A.3.4 N PARTICLES

Following the derivation of the energy for $N = 2, 3$ and 4, the general solution is inferred as

$$ E = \omega N + \omega \lambda_\sigma N(N - 1). $$

(A.34)
Appendix B

Parallelization and GPUs

Section B.1 describes some high level differences between GPUs (Graphics Processing Units) and CPUs (Central Processing Units).

Section B.2 discusses the general layout of the OpenCL language, introducing programming concepts such as contexts, kernels, event handling, work items and device buffers.

Section B.3 builds on the previous two sections, discussing what steps one must take to parallelize an algorithm for execution on a GPU.

Section B.4 addresses Path Integral Monte Carlo generally, citing which portions of the algorithm may be parallelized.

B.1 Differences between a CPU and GPU

This section will first introduce the reader to some general metrics such as frequency, core count, cache, bandwidth, and latency before comparing some modern GPUs (Graphics Processing Unit) and CPUs (Central Processing Unit).
B.1.1 Background

Until roughly 10 years ago, the main consideration when comparing consumer CPUs was frequency. The Intel 486 DX4 (Figure B.1) ran at 100 MHz in 1994 and contained 16kb of level 1 cache. The fastest next generation CPU was the Intel Pentium MMX, running at 233 MHz, introduced during the summer of 1997.

![Figure B.1: An Intel 486 DX4. Image courtesy of Henry Mühlpfordt, Creative Commons license.](image)

The first dual core CPU to market was the Intel Pentium D Extreme Edition, though AMD’s Athlon X2 came out a few weeks later, released in 2005. These two companies introducing their first (consumer) dual core processors within weeks of each other was no coincidence.

Intel’s previous processor’s frequency plateaued around 3.5GHz, because $TDP$ (Thermal Design Power) scales as $TDP \propto frequency^2$. Single cores were getting too hot, so multiple cores at lower frequencies were a natural evolution.

The other side, GPUs, had already crossed into this realm with 3dfx’s Voodoo 2 (Figure B.2): its board had three chips working in parallel. It also had significantly more bandwidth, roughly 2.16 GB s$^{-1}$ vs the 531 MB s$^{-1}$ of a then-current Pentium
MMX. Additionally, multiple boards can work in tandem utilizing a configuration called SLI (Scan-Line Interleave).

Considering the nature of GPUs is to work on many floating point numbers with little effective correlation (that is, work on pixels without worrying about neighbors), a parallel approach is only natural, and bandwidth is often the limiting factor in massively parallel computation so it too increases [38].

### B.1.2 Modern Hardware

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Product</th>
<th>Cores</th>
<th>Width</th>
<th>Frequency</th>
<th>Performance</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel</td>
<td>Core i7 6700</td>
<td>4</td>
<td>8</td>
<td>4.0 GHz</td>
<td>128 GFLOPS s$^{-1}$</td>
<td>25 GB s$^{-1}$</td>
</tr>
<tr>
<td>Intel</td>
<td>Xeon 2640v4</td>
<td>10</td>
<td>8</td>
<td>2.6 GHz</td>
<td>208 GFLOPS s$^{-1}$</td>
<td>50 GB s$^{-1}$</td>
</tr>
<tr>
<td>nVidia</td>
<td>GTX 860M</td>
<td>640</td>
<td>1</td>
<td>1.0 GHz</td>
<td>640 GFLOPS s$^{-1}$</td>
<td>80 GB s$^{-1}$</td>
</tr>
<tr>
<td>nVidia</td>
<td>Tesla M2900</td>
<td>512</td>
<td>1</td>
<td>1.3 GHz</td>
<td>665 GFLOPS s$^{-1}$</td>
<td>178 GB s$^{-1}$</td>
</tr>
<tr>
<td>nVidia</td>
<td>GTX 980ti</td>
<td>2816</td>
<td>1</td>
<td>1.2 GHz</td>
<td>3380 GFLOPS s$^{-1}$</td>
<td>336 GB s$^{-1}$</td>
</tr>
<tr>
<td>nVidia</td>
<td>GTX 1080</td>
<td>2560</td>
<td>1</td>
<td>1.7 GHz</td>
<td>4352 GFLOPS s$^{-1}$</td>
<td>320 GB s$^{-1}$</td>
</tr>
<tr>
<td>nVidia</td>
<td>Tesla P100</td>
<td>3584</td>
<td>1</td>
<td>1.5 GHz</td>
<td>5376 GFLOPS s$^{-1}$</td>
<td>740 GB s$^{-1}$</td>
</tr>
</tbody>
</table>

*Table B.1: Some basic performance metrics for modern hardware. Note that the width of a core refers to how many single precision floating point numbers it can process per hertz.*

Modern hardware has evolved significantly since the Voodoo 2. Specifically with
the introduction of unified shaders. Prior to this it was quite difficult to harness the power of GPUs due to their specialized nature: any problems had to be processed as images, as if rendered.

Unified shaders brought with them a high level shading language (HLSL) and, eventually, computational libraries such as OpenCL or language extensions such as CUDA. The following table (table B.1) lists some commonly available CPUs and GPUs as well as theoretical performance metrics such as gigaflops and bandwidth.

### B.2 OpenCL Core Concepts

The main difference between programming on a GPU and programming on a CPU is the concept of ordered execution. CPUs have nested loops (Figure B.3), the inner most loop being called upon many times depending on how deeply nested it is. GPUs also have loops, however you can take one, or many, of those loops and execute each portion of the loop in parallel.

```c
void sin(
    float ** x,
    float ** y)
{
    for (int i=0; i < 10; ++i)
        for (int j=0; j < 10; ++j)
            y[i][j] = sin(x[i][j]);
}
```

*Figure B.3: CPU code with the inner loop intact. Execution is sequential, filling matrix y, one element at a time.*

This may not seem too difficult an idea to grasp, however there is a fundamental consequence in recognizing this difference: CPUs execute this inner loop in order, while GPUs have no ordering (Figure B.4). The first portion of the loop may finish
before the last. In terms of OpenCL language, any iteration of the inner most loop is a *workitem* which is in turn part of a *workgroup*. There are many workgroups being executed at once, and so the total number of workgroups times the total number of workitems per group gives us the global size. There is an additional level: workgroups are processed in discrete chunks called *Wavefronts*.

```c
void sin(
    float * x,
    float * y)
{
    int j = get_global_id(0);
    for (int i = 0; i < 10; ++i)
        y[i*SIZE+j] = sin(x[i*SIZE+j]);
}
```

*Figure B.4: GPU code with the inner loop parallelized. Every element in a row of the matrix is queued for calculation simultaneously, however the order of execution is unspecified.*

Synchronization happens at the workitem level. Using *local memory*, one can share data between workitems in the same workgroup. Workitems in different workgroups cannot interact (easily) and, when they do, it is at a heavy performance penalty.

Memory is also segregated in this manner (Figure B.5): there is *global memory*, which is the slowest but most abundant and also most visible (all workitems can interact with global memory directly). There is *constant memory* which is read only by the GPU, cached and in general lower latency than global memory while also being accessible to all workitems. There is *local memory* which is shared by workitems and is significantly faster than global memory as it is much closer to the execution units. Finally there is private memory, visible only within each workitem and effectively the fastest memory available.

Reading from or writing to memory is done asynchronously at the workitem level,
though the GPU gathers and broadcasts these such that they happen in consecutive chunks. What this means is that while CPUs deal with latency by cache hierarchy, with all initial reads propagating through all cache levels and then subsequent reads checking cache first, GPUs deal with latency by having many threads and gathering reads such that the memory controller is always busy. This can cause a bit of trouble for concepts such as reduction, where one wants to sum an array, or any serial action. *Barriers* are introduced to deal with this: they order access to global or local memory at the workgroup level, ensuring that all workitems complete their read or write before continuing.

Generally speaking global memory is used to store data on the order of 2 MB or greater, cache memory is precious at 64 kB and local memory is around 48 kB per workgroup.

Finally on the host side there is the concept of kernels and event handling. Kernels
are code which is compiled at run time to be executed by our OpenCL device. They are written in C99 style syntax with some OpenCL specific functions that allow for the reading of global, group and local ids or synchronization through barriers and atomics or even mathematical functions for mixing vectors, which are native to the language as types. Event handling orders the execution of individual kernels, allowing us to queue up many executions at once when possible and build in the order dependence as necessary.

Taking all the above into consideration, note that every workitem has up to three dimensions. This N-Dimensional space represents a volume of computational work we call an $NDRange$ (Figure B.6). The takeaway is that if you have a problem which can be split into a grid, each element in the grid computationally separate from every other, you have a problem which may be well adapted to GPU computation.

**B.3 Amdahl’s Law and GPU Performance**

GPU’s generally offer significantly greater theoretical performance than similarly priced CPUs, consuming significantly less watts per gigaflop. They are however difficult to program to. Serial execution on a GPU will not provide any speedup and, in fact, is often quite slower than serial execution on the CPU because of branch prediction, caching and the disparity in frequencies.

GPUs process many threads in discrete chunks, typically we are looking at wavefronts of 32 or 64 threads at any given time. CPUs conversely process threads serially, assigning each thread to a single core. The key to efficient execution on a GPU is having many threads in flight, each accessing memory contiguously.

With this in mind, the first step in adapting an algorithm for execution on a GPU
Figure B.6: The entire workspace of a problem can be thought of as an N-Dimensional volume comprised or workgroups, each containing a set of workitems. Image taken from AMD OpenCL User Guide 2015.

is to consider what part of the code is most computationally intensive and then how exactly one can parallelize that part of the code. This is summarized best in Amdahl’s law

\[ S = \frac{1}{(1 - P) + P/N}, \]  

which gives a general expectation of the maximum performance one may achieve by parallelizing some portion of their code. Here \( P \) is the fraction of code which has been parallelized, \( N \) is the number of cores which will execute that fraction and \( S \) is
the speedup factor.

Assuming a large value of $N$, lets say only a quarter of the code is serial in nature. This would leave us with a maximum speedup factor of 4. Going to a tenth, we get a speedup factor of 10.

\section*{B.4 Path Integral Monte Carlo on a GPU}

Path Integral Monte Carlo (PIMC) can be broken down into many components, as covered in its respective section. A brief overview of the jargon is provided here as well.

A state configuration, as shown in Figure B.7, is comprised of $M$ slices and the number of slices increases as the temperature is lowered. Each slice has $N$ beads on it. This forms a position matrix of size $N \times M$. Every particle on a slice is labeled by an index. These shared indices across slices form worldlines. A worldline can be thought of as a single particle existing across an imaginary time dimension, potentially in a different position at each time step.

Updates are performed at a worldline level, either serially modifying a single world line across the imaginary time dimension or mixing one worldline with another via swaps. Once a worldline has been modified, the potential calculation occurs on a per-slice basis: only particles sharing a slice may interact.

The size of an update is primarily determined by $\bar{M}$. Once an update has been prepared, the change in action is calculated by taking the difference in potential across the $\bar{M}$ affected slices.

Taking a broad view there is the generation of states, performed by updates, and the evaluation of these states via an action calculation. A fixed number of updates
Figure B.7: Here we can see the configuration of some state. There are three worldlines and seven slices, corresponding to a total of three particles. The area between \((\alpha + \bar{m})\tau\) and \(\alpha \tau\) is the area worked on by an update.

are performed per \textit{MCStep} and every MCStep is accompanied by a measurement of relevant estimators.

Identifying which of these components can be parallelized is important, however first consider which of these components accounts for the most computational time. Estimators happen at a very low frequency, less frequent as the size of the state increases with decreasing temperature or increasing numbers of particles. Most estimators can be parallelized, however very little of the total computational time is spent on them.

Each MCStep is performed by preparing states via updates and accepting or them rejecting them via a metropolis sampling stage. Updates happen on a per-bead level meaning they are often serial in nature. There are a few exceptions such as
the Remove, Open and Center of Mass updates. Open happens quite often and is
good candidates for parallelization. Center of mass is trivial to parallelize and also
computationally intensive due to the action calculation in the metropolis sampling
stage being an entire world line. The other moves, Staging, Advance, Recede, Insert,
Close and Swap all have serial components and are not good candidates.

Something all these moves have in common is that they metropolis sample, cal-
culating a reduced action on perturbed beads. This turns out to be where the bulk
of the computational time is spent when interactions are turned on. That is to say
if there is some potential between beads on a single slice, the majority of the time is
spent calculating this potential.

This suggests that a good place to start is the external and interaction potentials:
everything else can remain on the host (CPU). There will be additional overhead due
to this sharing of the computation between multiple devices. The state has to be
synchronized between the two, and so data is constantly flowing from the host to the
GPU and back.

Synchronization may have a heavy, vendor specific, cost. Every time data is
transferred between there is a delay of two parts: some constant we will call kernel
latency and then a variable time related to the size of the transfer. AMD GPUs
were found to have a higher kernel latency than nVidia GPUs, roughly an order of
magnitude higher.

Kernel latency can be completely neutralized by moving all computation to the
GPU and, even in the cases of serial execution for some portions of the code, would
result in a significant improvement of the current code. The speedup would be drastic
for low particle counts, diminishing as particle count increases. Generally speaking
AMD offers significantly more performance per dollar; additionally their GPUs are designed to handle more complicated kernels. Moving all computation to the GPU would allow the effective utilization of AMD GPUs.