EXPLICIT, MULTI-MAP SYMPLECTIC INTEGRATOR FOR THREE-BODY CLASSICAL TRAJECTORY STUDIES IN HYPERSPHERICAL COORDINATES

BY

PAUL BURKHARDT

B.S., Oregon State University, 1994

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Chemistry in the Graduate College of the University of Illinois at Urbana-Champaign, 2004

Urbana, Illinois
Abstract

Symplectic integrators are well known for preserving the phase space volume in Hamiltonian dynamics and are particularly suited for problems that require long integration times. There is a general operator splitting method for developing explicit symplectic integration algorithms to any arbitrary even order for separable Hamiltonians where the position and momentum coordinates are uncoupled. Explicit symplectic integrators for general Hamiltonians are more difficult to obtain, but can be developed by a composition of symplectic maps if the Hamiltonian can be split into exactly integrable parts. No general technique exists for splitting any Hamiltonian of general form. Many three body problems in classical mechanics can be effectively investigated in symmetrized, hyperspherical polar coordinates, but the Hamiltonian expressed in these coordinates is non-separable. In molecular dynamics, the hyperspherical coordinates facilitate the validation and visualization of potential energy surfaces and for quantum reactive scattering problems, the coordinates eliminate the need for adjusting the wavefunction between product and reactant channels. An explicit symplectic integrator for hyperspherical coordinates has not yet been devised. This dissertation presents an explicit, multi-map symmetrized composition method symplectic integrator for three-body Hamiltonians in symmetrized, hyperspherical polar coordinates, specifically for classical trajectory studies in the plane.
Acknowledgements

I would like to thank my committee members, Professor R. Linn Belford, Professor Todd J. Martinez, and Professor J. Douglas McDonald. I am grateful to Professor R. Linn Belford for his encouragement and providing me the opportunity to complete this dissertation. I am also grateful to Professor Todd J. Martinez for reviewing my dissertation and offering valuable feedback. I thank Professor Gregory S. Girolami, Head of the Chemistry Department, and Professor James M. Lisy, Associate Head of the Chemistry Department, for their patience and support. I thank and acknowledge Professor Don H. Secrest for introducing me to interesting topics in computers and science, and for challenging me to determine an explicit symplectic integrator for hyperspherical coordinates, the topic of this dissertation. I thank my colleague, Dr. Vance Wong, for his friendship throughout the years and for many meaningful conversations. His moral support and exchange of ideas made graduate life interesting and enjoyable.

Finally, I would like to thank my dear wife, Dr. Anne G. Copay. The completion of this dissertation is due in large part to her support and assistance. I thank her for the care of our children and household while she concurrently pursued her own doctoral degree and again during my leave from the University. I am grateful and appreciative for the many articles and books she graciously obtained at my request. Her encouragement throughout my doctoral candidacy provided inspiration and determination.
## Contents

1 Introduction ........................................ 1

2 Literature Review .................................. 4
   2.1 Historical Background .......................... 4
   2.2 Summary of Literary Body ....................... 6

3 Mathematical and Theoretical Background ....... 11
   3.1 Canonical Transformations and Symplectic Condition .......... 11
   3.2 Canonical Invariants .......................... 17
   3.3 Hamiltonian Mechanics ......................... 21
   3.4 Solutions To The Equations Of Motion .......... 27

4 Symplectic Integration Algorithms ............... 31
   4.1 Numerical Integration Overview ............... 31
   4.2 Symplectic Integration Algorithms .......... 33
   4.3 Explicit Symmetrized Symplectic Composition Methods .... 36
   4.4 Lie Group Approach To Symplectic Integrators ............ 42
   4.5 Explicit Multi-Map Symmetrized Symplectic Composition Method .... 46
   4.6 Demonstration of Symplectic Integration Algorithms .... 49
   4.7 Plots ........................................... 61

5 Hyperspherical Coordinates ....................... 63
   5.1 Overview ....................................... 63
   5.2 Jacobi Coordinates Transformation ............ 64
   5.3 Rotating Coordinates .......................... 69
   5.4 Symmetrized Hyperspherical Polar Coordinates ....... 76
   5.5 Hamiltonian And Equations Of Motion .......... 86
   5.6 Initial Phase State Coordinates ............... 91
   5.7 Principal Axes Transformation .................. 93

6 Explicit Symplectic Integrator For 3-Body Hyperspherical Hamiltonians 101
   6.1 Symmetrized Hyperspherical 3-Body Hamiltonian ........ 101
### Programs For 3-Body Classical Trajectories In Hyperspherical Coordinates

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Main Driver Using Burlirsch-Stoer Integrator</td>
</tr>
<tr>
<td>C2</td>
<td>Main Driver Using Second Order Symplectic Integrator</td>
</tr>
<tr>
<td>C3</td>
<td>Main Driver Using Fourth Order Runge-Kutta Integrator</td>
</tr>
</tbody>
</table>

### Programs For The F+H₂ Hamiltonian Dynamics

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Interface Programs For Stark-Werner F+H₂ Potential Energy Surface</td>
</tr>
<tr>
<td>D2</td>
<td>Include Files</td>
</tr>
<tr>
<td>D3</td>
<td>Input Data Files</td>
</tr>
</tbody>
</table>

### Programs For The H+H₂ Hamiltonian Dynamics

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>Interface Programs For BKMP2 H+H₂ Potential Energy Surface</td>
</tr>
</tbody>
</table>

---

vii
### E.2 Input Data Files

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>E.2.1 phasespace_01/dat_01</td>
<td>212</td>
</tr>
<tr>
<td>E.2.2 phasespace_01/dat_02</td>
<td>212</td>
</tr>
<tr>
<td>E.2.3 phasespace_02/dat_01</td>
<td>212</td>
</tr>
<tr>
<td>E.2.4 phasespace_02/dat_02</td>
<td>212</td>
</tr>
</tbody>
</table>

### F Classical Trajectory Programs For Henon-Heiles Hamiltonian

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>F.1 Main Driver Using Fourth Order Runge-Kutta Integrator</td>
<td>214</td>
</tr>
<tr>
<td>F.1.1 MAIN</td>
<td>214</td>
</tr>
<tr>
<td>F.2 Main Driver Using Second Order Symplectic Integrator</td>
<td>216</td>
</tr>
<tr>
<td>F.2.1 MAIN</td>
<td>216</td>
</tr>
<tr>
<td>F.3 Main Driver Using Fourth Order Symplectic Integrator</td>
<td>218</td>
</tr>
<tr>
<td>F.3.1 MAIN</td>
<td>218</td>
</tr>
<tr>
<td>F.4 Classical Trajectory Programs</td>
<td>220</td>
</tr>
<tr>
<td>F.4.1 EQNMOT</td>
<td>220</td>
</tr>
<tr>
<td>F.4.2 GETENERGY</td>
<td>220</td>
</tr>
</tbody>
</table>

### G Classical Trajectory Programs For Simple Harmonic Oscillator

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.1 Main Driver Using Fourth Order Runge-Kutta Integrator</td>
<td>222</td>
</tr>
<tr>
<td>G.1.1 MAIN</td>
<td>222</td>
</tr>
<tr>
<td>G.2 Main Driver Using Second Order Symplectic Integrator</td>
<td>224</td>
</tr>
<tr>
<td>G.2.1 MAIN</td>
<td>224</td>
</tr>
<tr>
<td>G.3 Main Driver Using Fourth Order Symplectic Integrator</td>
<td>226</td>
</tr>
<tr>
<td>G.3.1 MAIN</td>
<td>226</td>
</tr>
<tr>
<td>G.4 Classical Trajectory Programs</td>
<td>227</td>
</tr>
<tr>
<td>G.4.1 EQNMOT</td>
<td>227</td>
</tr>
<tr>
<td>G.4.2 GETENERGY</td>
<td>227</td>
</tr>
</tbody>
</table>

### H Classical Trajectory Programs For Simple Unperturbed Pendulum

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>H.1 Main Driver Using Fourth Order Runge-Kutta Integrator</td>
<td>228</td>
</tr>
<tr>
<td>H.1.1 MAIN</td>
<td>228</td>
</tr>
<tr>
<td>H.2 Main Driver Using Second Order Symplectic Integrator</td>
<td>230</td>
</tr>
<tr>
<td>H.2.1 MAIN</td>
<td>230</td>
</tr>
<tr>
<td>H.3 Main Driver Using Fourth Order Symplectic Integrator</td>
<td>231</td>
</tr>
<tr>
<td>H.3.1 MAIN</td>
<td>231</td>
</tr>
<tr>
<td>H.4 Classical Trajectory Programs</td>
<td>232</td>
</tr>
<tr>
<td>H.4.1 EQNMOT</td>
<td>232</td>
</tr>
<tr>
<td>H.4.2 GETENERGY</td>
<td>233</td>
</tr>
</tbody>
</table>

### I Appendix I: Utility Programs

<table>
<thead>
<tr>
<th>Subsection</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.1 Utility Programs</td>
<td>234</td>
</tr>
<tr>
<td>I.1.1 DERIV</td>
<td>234</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

The motion of three interacting bodies, known as the three-body problem, was proven by Poincare to have no analytic solution. It is the simplest system for which numerical treatment is required to solve the dynamics, making it an important benchmark for the study of more complex problems. It serves as a model for the dynamics of many natural systems of importance to science, from the macroscopic motion of celestial bodies to the interaction of particles at the quantum level. In the context of molecular dynamics, the three-body system is important in the study of atom-molecule collisions, bound states in triatomic molecules, and potential energy surface determination for three-body bimolecular reactions. It is simple enough that the solutions can be computed to numerically exact limits with full quantum mechanical treatment. Such systems have been investigated intensively using both classical and quantum mechanics, elucidating details of the processes behind chemical reactions and bonding.

Although molecular dynamics is more accurately described by quantum mechanics, there are problems that are quantum mechanically intractable or for which the classical Newtonian approach is more advantageous. Chaotic dynamics are less well-defined for quantum mechanics than for classical mechanics. Classical or quasi-classical trajectory calculations are a quick means for validating or calibrating potential energy surfaces for the study of molecular structure and interactions. The domain of long-lived states exhibit time scales that are many times the vibrational or rotational period of the system and may be too computationally prohibitive for full quantum treatment. Classical trajectories are also useful for states at near dissociation limits. Systems that exhibit weakly bound long-lived states at van der Waals minima in dissociation threshold channels can be effectively simulated by classical trajectory studies. One can use few-body classical dynamics to more easily study bifurcations in the classical phase space, such as periodic orbits, providing insight to quantum phenomena that would otherwise be too difficult or cumbersome in quantum methods. Studying these problems requires long simulation times, but rounding error from computer arithmetic becomes
dramatically pronounced. Special numerical treatment is required to minimize errors due to compounded roundoff error during integration over long time periods.

Solving the dynamics of a system usually involves integrating the Hamilton Equations of Motion. The Hamiltonian energy function has unique conservative and geometric properties. These properties can be degraded when the equations of motion are numerically integrated for long simulation times, degenerating conserved systems into dissipative ones. This problem can be solved by using an integrator that ensures a canonical transformation at each integration step. Such integrators are known as Symplectic Integration Algorithms (SIAs) and are well known to exhibit excellent conservation of the Poincare integral invariants at larger time steps than non-symplectic integrators. Although the SIA does not exactly conserve energy for autonomous systems, the energy is bounded and does not diverge, contrary to standard ordinary differential equation integrators at similar integration time scales and step size.

Symplectic integrators have been used in molecular dynamics for many years for their energy stability and low storage requirements. It is usually preferable to use explicit integrators for their performance than implicit methods. One of the most popular integrators in molecular dynamics, the Leapfrog integrator, is part of a class of multi-map, explicit symplectic integrators first developed in the particle accelerator physics community. These symplectic integrators were initially thought applicable only to separable Hamiltonians but were later shown to be general for any Hamiltonian that is reducible to integrable parts. A symplectic integrator can be developed for any general Hamiltonian using canonical generating functions, but these are necessarily implicit integrators except in the case of separable Hamiltonians. There is currently no general method for developing explicit symplectic integrators for any kind of Hamiltonian. Many of the symplectic integrators in the literature are for separable Hamiltonians, where the position and momentum coordinates are not coupled. There is a deficiency of explicit symplectic integrators in molecular dynamics for systems where the preferred coordinates result in a non-separable Hamiltonian.

The choice of coordinates is usually selected based on particular properties of the system that ease the computational work and expose desired properties of interest. One of the popular coordinates in molecular dynamics is relative, mass-scaled Jacobi coordinates. These are preferred in systems with two coupled bodies colliding with an incident particle. The kinetic energy operator can be expressed in a simple form in these coordinates. In quantum reactive scattering studies the interchange of the chemical species is particularly troublesome, where coordinates change for both reactants and products. Symmetrized hyperspherical polar coordinates may be employed to reduce the complexity in the interchange of the bodies and treat reactant and product channels without bias. These coordinates can also be easily derived by a transformation of the Jacobi coordinates. In hyperspherical polar coordinates, or often referred to as just hyperspherical coordinates, the system is reduced to effectively one particle in hyper-dimensional space. For three body systems, there are five hyper-angles
that parameterize one radial coordinate, the radius vector of a hypersphere. Hyperspherical coordinates provide convenient, abstract geometric perspectives on the internal configuration and motion of the particles, useful for visualization of the dynamics and potential energy surfaces.

A fast, efficient integrator is needed that can preserve the canonical properties and hence, the geometry of the phase space, to study long lifetime states in hyperspherical coordinates. Symplectic integrators can preserve the qualitative geometric behavior, but at the time of this dissertation, the author is not aware of any explicit symplectic integrators for Hamiltonians in hyperspherical coordinates. It is the purpose of this dissertation to present a class of explicit, second order Ruth-type, N-map symmetrized composition method, symplectic integrators for the classical study of three-body systems in symmetrized hyperspherical coordinates. The integrator is applied to solving the equations of motion for sample three-body problems in the context of molecular dynamics as demonstration of its efficacy in preserving global geometric properties. The investigation effectively demonstrates the usefulness of the symplectic integrator for preserving the canonical properties. The coordinate choice is important from both a classical and quantum perspective, and complemented with the symplectic integrator, existing and new studies can provide faster, more qualitatively correct results. The newly developed symplectic integrator is versatile and not restricted to problems in chemistry, but can be applied to any three-body system where the potential energy is a function of only the inter-particle distances. Computer programs have been developed to easily interface with different potential energy functions and integrators for comparison studies.

An attempt has been made to make this dissertation self-contained. Relevant equations and their derivations, along with the algorithms and source code for the computational work are provided. In Chapter 2, a review of the scientific body of literature is presented. The historical and seminal articles along with the significant current and recent work are summarized. The mathematical and theoretical background is provided in Chapter 3. Symplectic integration techniques relevant to the topic are described in Chapter 4 and sample demonstrations of symplectic integrators are given. The focus shifts to symmetrized hyperspherical coordinates in Chapter 5 where the mathematical background and derivation are provided for the desired hyperspherical coordinates. The new symplectic integrator of this dissertation is developed and described in Chapter 6. In Chapter 7, the methodology for the application of the symplectic integrator to a physical system in hyperspherical coordinates is given. The symplectic integrator is applied in Chapter 8 to two benchmark chemical systems. Finally, in Chapter 9, the results and conclusions are discussed.
Chapter 2

Literature Review

2.1 Historical Background

Symplectic integration is a relatively new field of study with much of the seminal work accomplished in the 1980s and early 1990s. The first study of symplectic integrators is attributed to DeVogelaere in 1956 [see 30], however symplectic integrators, such as the Leapfrog integrator, existed well before their symplectic nature was explicitly exposed. Symplectic integration algorithms were not further exploited until nearly thirty years later in the study of celestial mechanics and particle accelerators. In 1982, Wisdom introduced an explicit symplectic integrator based on an averaging principal for celestial mechanics [190]. Another method for explicit symplectic integration was introduced by Ruth in a classic 1983 paper [151] that brought the study of symplectic integrators to the forefront of many diverse areas of Hamiltonian mechanics. Other noteworthy pioneering work was done by Channell in 1983, Menyuk in 1984 [128], and Feng in 1986 [48].

Ruth relied on generating functions of canonical transformations to produce symplectic integration algorithms and gave explicit methods for special Hamiltonians up to third order and a prescription for higher order explicit integrators based on composing steps, similar to Runge-Kutta integrators, and solving for the constant coefficients to satisfy Hamilton’s equations of motion [151]. Central to the algorithms was the asymmetric method of advancing the canonical variables. Following this prescription, explicit fourth order algorithms were developed by Neri in 1988 [see 50] and independently by Candy and Rozmus [22]. Ruth had also arrived at an analytic solution for the fourth order integrator, but the work was unpublished [see 49]. Higher-order explicit integrators based on the early generating function method of Ruth would be very difficult to determine and prompted the development of better methods.
Although implicit symplectic integrators had been developed for general Hamiltonians [see 30 48], the early explicit symplectic integrators were considered valid only for Hamiltonians of the special form, $H(q, p) = T(p) + V(q)$, known as separable or potential form Hamiltonians. For generating functions based on terms of mixed variables the integration is necessarily implicit except in the special case where the Hamiltonian is separable. Implicit Runge-Kutta-Nystrom integrators were developed by Sanz-Serna [154]. Semi-explicit techniques for general Hamiltonians were later given by Channell and Scovel [30].

The Ruth integrators had been re-derived and greatly simplified using Lie methods by Forest [49 51] and Neri [see 50]. A composition method based on operator-splitting, consisting of a product of symplectic mappings, was introduced in the particle accelerator community but was not published until 1990 by Forest and Ruth [51]. About the same time Yoshida [194], Holman, and Wisdom [191] had also independently developed a two-map composition approach using operator-splitting. The formulation of the composition method requires splitting the Hamiltonian into a sum of exactly integrable Hamiltonians related to non-commutative operators, where the solutions for each can be integrated independently from the other. It was shown that only even ordered integrators were possible for symmetric mappings with this method. The Campbell-Baker-Hausdorff formula was employed to exploit Lie group properties by Forest and others. The association of Ruth type integrators with Lie groups was an insightful discovery that led to the important generality of the method for any number of splittings and general Hamiltonians. An explicit sixth order symplectic integrator was obtained by Forest [49] through the application of the Lie groups. In an acclaimed paper, Yoshida [194] generalized Ruth’s method and devised a scheme for constructing explicit, higher even-ordered symplectic integrators by the symmetric product of lower even-order symplectic mappings. Specifically, if one has a $2N$ order symmetric integrator, Yoshida developed a formula for obtaining a $2N + 2$ order symplectic integrator. In his paper, Yoshida derived fourth, sixth, and eighth order explicit symplectic integrators using the new technique, but assumed the method was valid only for separable Hamiltonians that could be split into two exactly solvable parts. It was proposed by Forest [50] that the technique devised by Yoshida for two-map compositions could be employed with the second order, symmetrized multi-map composition, derived by Forest and Ruth [51], to obtain higher ordered integrators for general Hamiltonians.

It was recognized by Ruth, Forest, and Neri [see 49 50 51] that the Ruth composition method could produce explicit symplectic integrators for any general Hamiltonian that can be split into $N$ exactly integrable parts, since the operator splitting imposed no special requirements on the nature of the Hamiltonian. Similarly for the approach given by Holman and Wisdom [191]. In 1996, Channell and Neri [29] indicated that the Ruth composition method could be extended to any polynomial Hamiltonian, based on monomial splitting, thereby greatly extending the applicability of the method.

Seemingly disconcerting limitations of symplectic integrators were manifested in the early
development period. It was proved by Ge and Marsden that only the true Hamiltonian can preserve all the Poincare integrals, therefore a symplectic integrator can only preserve a subset of the canonical properties and integral invariants of Hamiltonians \[54\]. Additionally, Suzuki proved in 1991 that it is not possible to obtain symplectic integrators with all positive stages or substeps for third order or greater \[181\].

A summary of symplectic numerical methods can be found in the literature \[30, 66, 124, 153, 155, 179, 195\]. Refer to McLachlan \[125\] for a historical background on symplectic integrators and an overview of splitting methods.

2.2 Summary of Literary Body

The field of astronomy has seen tremendous impact from symplectic integration algorithms. The study of planetary orbits in the solar system, particularly of the Jovian planets, using early integration techniques performed by Franklin et al. in 1990 \[53\] were later verified to be chaotic by Sussman and Wisdom using symplectic integration and published in the 1992 Science journal \[180\]. Following this work, chaos in the solar system was investigated by Saha and Tremaine in 1992 \[152\], Laskar in 1994 \[100\], and by Murray and Holman \[131\] in another Science article in 1999, demonstrating the applicability of symplectic integration. The success of early explicit symplectic methods and studies are due in large part to the pioneering work of Wisdom and Holman \[191\]. Other influential work on symplectic integrators for celestial mechanics includes the paper by Gladman, et al. \[56\], and more recently, the application of symplectic integrators to Kepler orbits \[65, 186\].

Notable early demonstrations of the superior geometry-preserving characteristics of symplectic integrators were published by Candy and Channell for Ruth-type explicit methods \[22\] and semi-explicit methods \[30\], respectively. The long-time trajectory stability of explicit Runge-Kutta Nystrom and Partitioned Runge-Kutta symplectic integrators were demonstrated by Okunbor and Skeel \[136\].

Symplectic integrators, like other numerical integrators, are subject to error due to discretization in floating-point arithmetic. The use of finite integer lattice maps and integer computation were shown to produce an exact symplectic map as the numerical approximation to the Hamiltonian flow without numerical error \[43\]. Lattice-map symplectic integrators using floating-point computation without rounding error were later introduced by Skeel \[165\].

In a 1994 molecular dynamics study of a 1000 unit polyethylene chain by Gray, et al., the superior energy conservation of symplectic integrators were verified for some popular second, fourth, and sixth order integrators. The symplectic integrators were verified to perform the best in regard to energy conservation, demonstrating small energy standard deviation, and
shown to be useful for large scale molecular dynamics studies\cite{60}. The equivalency of second-order implicit symplectic methods, often used in molecular dynamics, was demonstrated and the accuracy among the methods compared by Skeel, et al. in 1997 \cite{168}.

Existing explicit symplectic integrators up to fifth order could be made more accurate by minimizing an error function introduced by McLachlan and Atela \cite{126}. The error function is an nth-order coefficient in the Taylor expansion of the Hamiltonian truncation error. The accuracy was compared among popular explicit and implicit methods.

Symplectic integrators produce exact, or nearly exact, solutions to a nearby Hamiltonian that is a perturbation of the exact Hamiltonian. Non-symplectic integrators produce solutions that are for a nearby system but that system is not Hamiltonian \cite{109}. The excellent qualitative behavior of symplectic integrators extends to perturbed systems. The quality of symplectic integrators was correlated to perturbation theory by Benettin and Fasso \cite{12}. Symplectic Runge-Kutta methods for perturbed Hamiltonian systems were demonstrated to produce qualitative correct solutions, independent of the perturbation parameter, contrary to non-symplectic integrators \cite{176,177,178}.

Attempts to improve the accuracy of symplectic integrators at lower orders and reduce the magnitude of the energy oscillations attracted significant interest. Symplectic correctors based on the averaging principle were introduced by Wisdom, et al. \cite{192} in the context of perturbative systems and the delta function formalism for symplectic integrators. In this formalism, the difference between mapping and actual phase space variables have explicit form and are purported to be responsible for errors in the energy and state variables. Transformations between the mapping and actual variables are given and correctors devised with the motivation to eliminate the errors responsible for fluctuations in the energy and phase space. The approach is based on the averaging principle where fast phases or high frequency terms do not significantly affect the dynamics of the system, so these perturbations can be essentially ignored. The high frequency terms are added to the Hamiltonian as a series of periodic Dirac delta functions in the perturbation. The perturbation is eliminated for local time scales, allowing integration of an unperturbed Hamiltonian between the delta function periods, and then corrected outside of the integration time scale. As a consequence, the correctors do not need to be symplectic. The symplectic correctors are used to reduce the oscillatory nature of the energy and state variables without the extra cost of higher order or smaller time steps. The magnitude in the energy oscillation is dramatically reduced, but a gradual secular drift in the energy becomes evident, attributed to computational error. The accuracy of symplectic integrators and the relation of the computed solutions to the true Hamiltonian are described as being more than matching terms up to some order in the Taylor series, and this is evident through the delta function formalism. This work was followed by McLachlan in a separate paper \cite{123}. Later in 1999, Abdullaev \cite{2} developed a symmetric symplectic map where perturbations in the Hamiltonian are removed for an entire period, having similarities to the symplectic corrector method of Wisdom, et al. \cite{190,192}.
Although Suzuki proved that symplectic integrators with order three or greater could not have all positive stages, creative techniques were devised that could obtain all positive stages for integrators of effectively higher order. Eliminating the most dominant perturbation error terms in the asymptotic expansion of perturbed Hamiltonian systems of the form, $H = A + \varepsilon B$, where both $A$ and $B$ are Hamiltonian and explicitly integrable, can lead to effectively higher order integrators. A family of such integrators was first introduced by McLachlan in 1995 \[127\] and also independently derived by Chambers, et al. \[25\]. The approach is valid when the perturbation factor is much less than unity so terms multiplicative in higher powers of the factor are negligible and can be ignored. Higher-order integrators using this approach have all positive steps, unlike the symplectic integrators of the corresponding conventional order. Another advantage is the integrators have fewer sub steps so are more efficient and can be an order of magnitude faster than the conventional counterpart. In 2001, Laskar and Robutel \[101\] formally proved the existence, at all orders, of the family of symplectic integrators introduced by McLachlan \[122\] and added a corrector step, similar to the one introduced by Wisdom \[192\], for better accuracy when $A$ is Quadratic Form and $B$ depends only on the positions. The new family of integrators was demonstrated, up to tenth order, to be more accurate and stable with respect to the Leapfrog integrator, although no advantage is gained with order greater than two for small stepsizes \[101\].

Multiple time step methods have been developed for the Verlet/Leapfrog method. One such integrator, deemed Verlet-I, was constructed by Grubmuller, et al. for molecular dynamics \[63\]. Detailed error analysis on the Verlet-I integrator were performed by Littell, et al. \[107\]. The conservation of angular momentum was demonstrated for several explicit, multiple time step symplectic methods, including Verlet-I, by Zhang and Skeel \[197\].

Variable or adaptive step size methods are advantageous for problems where the solution curve may have flat areas so accuracy is not as crucial. By increasing the step size for these areas, performance can be enhanced. Unfortunately, variable step size breaks the structure preserving properties of symplectic integrators. The explanation given in the literature is the symplectic integrator solves the Hamiltonian flow for a nearby Hamiltonian, but changing the time step results in a solution map for a different Hamiltonian flow at each integration step, so the numerical solution for the original flow cannot be realized \[56, 60, 155, 166\]. It was suggested by Skeel and Gear that variable step-size symplectic integrators are possible but would not be obtained from varying the step size in a symplectic integrator constructed with fixed step size \[166\]. Alternatively, multiple time step methods can produce results and benefits equivalent to a variable time step method by partitioning the potential so the force is evaluated in parts at different time scales \[68\]. These can help reduce the number of force evaluations, typically the most performance critical computations in molecular dynamics.

In addition to Yoshida’s method, it is known that higher order symplectic methods can be constructed by combining a lower order method with its adjoint \[60, 137\]. Other methods for constructing higher ordered symplectic integrators are known and have been published
The adjoint of a symplectic integrator is essentially the opposite form, where the asymmetric update of the canonical variables is reversed. The adjoint is also symplectic but may not be computationally equivalent, and the error tolerance can be different. It has also been proved that explicit Runge-Kutta Nystrom methods are symplectic if and only if the adjoint is explicit.

Implicit symplectic integrators can be constructed for any general Hamiltonian, but the computational expense makes explicit methods more attractive. When numerical stability is more important and outweighs the performance cost, implicit methods are preferred. Meisbach developed implicit symplectic integrators, using generating functions, composed of steps similar to Runge-Kutta to avoid higher-order derivatives and demonstrated that implicit methods must be iterated to machine precision in order to maintain symplecticity. Improved implicit symplectic integrators were developed by Zhang and Skeel with the advantage of stability offered by implicit methods but with less computational cost. Other implicit methods can be found in the literature.

Symplectic integrators have been constructed for constrained Hamiltonians, an important class often implemented in rigid-body molecular dynamics. An example in the literature is a symplectic integrator developed for a separable and constrained Hamiltonian system as an alternative to quaternion approaches for solving the orientation degrees of freedom.

In classical trajectory calculations, common in molecular dynamics, the integration of the equations of motion is usually dominated by the force evaluations. By exploiting Hessian-vector products, Lopez-Marcos, et al. developed an efficient, effectively fourth order, explicit symplectic integrator for quadratic, separable Hamiltonians. The method relies on a Hessian-vector product with one force evaluation being less computationally expensive than two force evaluations. This follows earlier work exploiting lower order explicit, symplectic Runge-Kutta-Nystrom methods that effectively produce the same accuracy as a higher order method at less computational cost through pre- and post-processing of desired solution points. The preprocessing stage is similar to the symplectic correctors introduced in the literature.

The accuracy of a fourth order, force gradient symplectic integrator with positive time steps, a novelty that is contrary to the general proof of Suzuki, was investigated by Chin and Kidwell in 2000. The integrator requires three force and one gradient of the square force evaluations, and demonstrates far better accuracy than the standard fourth order symplectic integrator while being only ten percent slower in tests with the Kepler problem. In addition, they constructed and demonstrated sixth, eighth, tenth, and twelfth order integrators derived from the fourth order force gradient integrator of Chin, although not with all positive
time steps, possessing similarly better accuracy with respect to other comparable ordered symplectic integrators.

The phase space geometry preserving properties of symplectic integrators has been extensively demonstrated in the literature. It is well known that symplectic integrators are superior to other integrators for studies where the qualitative behavior of a system at long integration times is important. Generally, it is believed that conservation of energy is an indicator of accurate trajectory calculations, but Schlier and Seiter demonstrated in a 1998 paper that long-lived trajectories can be completely wrong even with energy conservation at pico or better magnitudes. Their study showed that energy conservation does not necessarily correlate to correct trajectories and symplectic integrators can provide correct trajectories when other integrators fail. Classical trajectories for the triatomic complex forming system, $\text{H}_3^+$ using the Diatomics-in-Molecules (DIM) potential energy surface, were studied with fourth and sixth order symplectic integrators. The sixth order symplectic integrator proved to be the best performer in terms of correctness versus computational expense.

The body of work on symplectic integrators is overwhelming for the special case where the Hamiltonian is separable. Symplectic integrators for non-separable Hamiltonians have been published for some problems. In molecular dynamics, the orientation coordinates can be expressed as quaternions, resulting in a non-separable Hamiltonian. In 1997, Dullweber, et al. devised a composition of symplectic maps, or splitting method, to obtain explicit, symplectic integrators for rigid body systems using a general, non-separable constrained Hamiltonian. This approach supplants the standard model of quaternion coordinates which couple momenta and position coordinates in the Hamiltonian. The approach uses planar rotations at each integration step for each rigid body. It is also a second-order method. This method is most efficient for symmetric rigid bodies with low degrees of freedom. In another quaternion-based, non-separable Hamiltonian system, Miller, et al. motivated by the availability of massively parallel computers for large biophysical molecular dynamics, devised a Hamiltonian and corresponding explicit, symplectic integrator by decomposing the fast degrees of freedom into rigid body rotations. In a recent publication, Tuwankotta and Quispel used a multi-map composition method to construct a symplectic integrator for the study of resonances in the elastic pendulum problem where the Hamiltonian is also non-separable.

The Hamiltonian is non-separable in hyperspherical coordinates, where the position and momentum coordinates are coupled. The first explicit symplectic integrator for three-body classical trajectory studies in hyperspherical coordinates will be presented in this dissertation and applied in the context of long lifetime states at van der Waals well region and near dissociation limits for some chemical systems.
Chapter 3

Mathematical and Theoretical Background

3.1 Canonical Transformations and Symplectic Condition

All symplectic integrators preserve the differential 2-form,

\[ w^2 = \sum_i dq'_i \wedge dp'_i = \sum_i dq_i \wedge dp_i. \]  

(3.1.1)

This is a statement of the conservation of phase space volume. Refer to other texts [6] for the derivation. In the one degree of freedom case, “symplecticness” is a conservation of the phase space area. Symplectic integrators evolve the Hamiltonian flow or time-map by a canonical transformation at each integration step. Canonical transformations preserve the phase space volume and satisfy the symplectic condition:

\[ M^T \cdot J \cdot M = J. \]  

(3.1.2)

The matrix \( M \) in the equation is the Jacobian matrix and \( J \) is an anti-symmetric square matrix,
where $0$ and $1$ are $n \times n$ null and identity matrices, respectively. A symplectic transformation is a canonical transformation. Canonical transformations can also be produced using generating functions. Symplectic methods provide an alternative approach to producing transformations that preserve canonical properties. A brief review of canonical transformations and the symplectic condition will be given.

The generalized position variables and their conjugate momenta, often denoted as $q$ and $p$ respectively, are the canonical variables familiar in Hamiltonian mechanics. The set of all generalized positions and momenta, $(q_i, \ldots, q_d, p_i, \ldots, p_d)$, make up the phase space with dimension $R^{2d}$. Given two functions, $u$ and $v$, the Poisson bracket with respect to the canonical variables is defined as,

$$\{u, v\}_{q,p} = \sum_i \left( \frac{\partial u}{\partial q_i} \frac{\partial v}{\partial p_i} - \frac{\partial v}{\partial q_i} \frac{\partial u}{\partial p_i} \right). \quad (3.1.4)$$

The summation is over the number of degrees of freedom. In matrix form, the Poisson bracket can be written as,

$$\{u, v\}_{q,p} = \left( \begin{array}{cc} \frac{\partial u}{\partial q} & \frac{\partial u}{\partial p} \\ \frac{\partial v}{\partial q} & \frac{\partial v}{\partial p} \end{array} \right) \left( \begin{array}{cc} 0 & 1 \\ -1 & 0 \end{array} \right) \left( \begin{array}{cc} \frac{\partial u}{\partial q} & \frac{\partial u}{\partial p} \\ \frac{\partial v}{\partial q} & \frac{\partial v}{\partial p} \end{array} \right). \quad (3.1.5)$$

One degree of freedom is assumed for simplicity, but the relation is easily generalized for higher dimensions by increasing the dimensionality of the matrices for each variable in the phase space. Recall the general form of the vector gradient operator:

$$\vec{\nabla}_s = \frac{\partial}{\partial s_1} \hat{e}_1 + \cdots + \frac{\partial}{\partial s_n} \hat{e}_n. \quad (3.1.6)$$

With respect to the canonical variables, the vector gradient operator is,
\[ \vec{q}_{q,p} = \frac{\partial}{\partial q_1} \hat{e}_1 + \cdots + \frac{\partial}{\partial q_d} \hat{e}_d + \frac{\partial}{\partial p_1} + \cdots + \frac{\partial}{\partial p_d} \hat{e}_2d \]

\[ = \left( \frac{\partial}{\partial q_1} \cdots \frac{\partial}{\partial q_d} \frac{\partial}{\partial p_1} \cdots \frac{\partial}{\partial p_d} \right)^T. \tag{3.1.7} \]

The general form of the Poisson bracket for one pair of functions can then be written as,

\[ \{ u_i, v_j \}_{q,p} = \left( \vec{q}_{q,p} \cdot u_i \right)^T \cdot J \cdot \left( \vec{q}_{q,p} \cdot v_j \right). \tag{3.1.8} \]

The matrix, \( J \), is an anti-symmetric square matrix,

\[ J = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \tag{3.1.9} \]

where 0 and 1 are \( n \times n \) zero and unit matrices, respectively. It has the following properties,

\[ J^T = J^{-1} = -J, \]
\[ J^2 = -I, \tag{3.1.10} \tag{3.1.11} \]

where, \( I \), is the identity matrix.

The Poisson bracket can also be written compactly for a set of functions. Consider there are \( 2d \) functions for both \( u \) and \( v \). It was shown for two functions, the Poisson bracket is,

\[ \{ u_i, v_j \} = \begin{pmatrix} \frac{\partial u_i}{\partial q_1} & \cdots & \frac{\partial u_i}{\partial q_d} & \frac{\partial u_i}{\partial p_1} & \cdots & \frac{\partial u_i}{\partial p_d} \end{pmatrix} \begin{pmatrix} d_{v_1} \\ \vdots \\ d_{v_d} \end{pmatrix} \begin{pmatrix} 0_d & 1_d \\ -1_d & 0_d \end{pmatrix} \begin{pmatrix} \frac{\partial v_j}{\partial q_1} \\ \vdots \\ \frac{\partial v_j}{\partial p_d} \end{pmatrix} \]

\[ = \begin{pmatrix} \frac{dv_1}{dq_1} \\ \vdots \\ \frac{dv_d}{dq_1} \\ \vdots \\ \frac{dv_1}{dp_d} \\ \vdots \end{pmatrix} \begin{pmatrix} \frac{\partial u_i}{\partial q_1} & \cdots & \frac{\partial u_i}{\partial q_d} & \frac{\partial u_i}{\partial p_1} & \cdots & \frac{\partial u_i}{\partial p_d} \end{pmatrix} \begin{pmatrix} dv_1 \\ \vdots \\ dv_d \end{pmatrix} \]

\[ \begin{pmatrix} 0_d & 1_d \\ -1_d & 0_d \end{pmatrix} \begin{pmatrix} \frac{\partial v_j}{\partial q_1} \\ \vdots \\ \frac{\partial v_j}{\partial p_d} \end{pmatrix} \]  

This can be extended for the \( 2d \) functions, \( u = u(u_1, \ldots, u_{2d}) \) and \( v = v(v_1, \ldots, v_{2d}) \), with the explicit matrix form being given by,
\[
\{u, v\} = \left( \begin{array}{cccc}
\frac{\partial u_1}{\partial q_1} & \cdots & \frac{\partial u_1}{\partial p_d} \\
\vdots & \ddots & \vdots \\
\frac{\partial u_d}{\partial q_1} & \cdots & \frac{\partial u_d}{\partial p_d}
\end{array} \right) \cdot \left( \begin{array}{cc}
0_d & 1_d \\
-1_d & 0_d \\
\end{array} \right) \cdot \left( \begin{array}{cccc}
\frac{\partial v_1}{\partial q_1} & \cdots & \frac{\partial v_1}{\partial p_d} \\
\vdots & \ddots & \vdots \\
\frac{\partial v_d}{\partial q_1} & \cdots & \frac{\partial v_d}{\partial p_d}
\end{array} \right).
\] (3.1.13)

Using matrix notation popular in literature [58], the Poisson bracket can be concisely written as,

\[
\{u, v\}_z = \left( \frac{\partial u}{\partial z} \right)^T \cdot J \cdot \left( \frac{\partial v}{\partial z} \right).
\] (3.1.14)

Here, \(u\) and \(v\) represent the complete set of functions with respect to the canonical variables. The symbol, \(z\), is a column matrix of the phase space variables such that,

\[
z = \left( q_1 \; \cdots \; q_d \; p_1 \; \cdots \; p_d \right)^T.
\] (3.1.15)

The antisymmetric matrix, \(J\), is a 2\(d\)x2\(d\) matrix.

The Poisson bracket has the following properties,

\[
\{u, u\}_{q,p} = \{v, v\}_{q,p} = 0, \quad (3.1.16)
\]
\[
\{u, v\}_{q,p} = -\{v, u\}_{q,p}, \quad (antisymmetry) \quad (3.1.17)
\]
\[
\{au + bv, w\} = a \{u, w\} + b \{v, w\}, \quad (linearity) \quad (3.1.18)
\]
\[
\{uv, w\} = u \{v, w\} + v \{u, w\}, \quad (LeibnizIdentity) \quad (3.1.19)
\]
\[
\{u, \{v, w\}\} + \{v, \{w, u\}\} + \{w, \{u, v\}\} = 0. \quad (JacobiIdentity) \quad (3.1.20)
\]

If the functions \(u, v\) are from the set of the canonical variables, then it is easily seen from the Poisson bracket definition and properties that,

\[
\{q_i, q_j\} = \{p_i, p_j\} = 0, \quad \text{and} \quad (3.1.21)
\]
\[
\{q_i, p_j\} = -\{p_i, q_j\} = \delta_{ij}.
\] (3.1.22)

The Kroenecker delta, \(\delta_{ij}\), equates to unity if the subscripts match, otherwise it is zero.
For all canonical conjugate pairs in phase space, these relations can be represented by the form,

\[ \{z, z\}_z = (\frac{\partial z}{\partial z})^T \cdot J \cdot (\frac{\partial z}{\partial z}) = J. \quad (3.1.23) \]

Suppose there are \( n \) functions of \( n \) variables, \( f_i (s_i, ..., s_n) \). The differential for the \( ith \) function is given by,

\[ df_i = \sum_j \frac{\partial f_i}{\partial s_j} ds_j, \quad (3.1.24) \]

or equivalently in matrix form,

\[ df_i = \left( \frac{\partial f_i}{\partial s_1} \cdots \frac{\partial f_i}{\partial s_n} \right) \cdot \begin{pmatrix} ds_1 \\ \vdots \\ ds_n \end{pmatrix}. \quad (3.1.25) \]

The derivative with respect to one of the dependent variables is then,

\[ \frac{df_i}{ds_1} = \left( \frac{\partial f_i}{\partial s_1} \cdots \frac{\partial f_i}{\partial s_n} \right) \cdot \begin{pmatrix} \frac{\partial s_1}{\partial s_1} \\ \vdots \\ \frac{\partial s_n}{\partial s_1} \end{pmatrix}. \quad (3.1.26) \]

This can be expanded for all \( n \) functions so that the transformation equation is,

\[
\begin{pmatrix}
\frac{df_1}{ds_1} & \frac{df_1}{ds_2} & \cdots & \frac{df_1}{ds_n} \\
\frac{df_2}{ds_1} & \frac{df_2}{ds_2} & \cdots & \frac{df_2}{ds_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{df_n}{ds_1} & \frac{df_n}{ds_2} & \cdots & \frac{df_n}{ds_n}
\end{pmatrix}
= \begin{pmatrix}
\frac{\partial f_1}{\partial s_1} & \frac{\partial f_1}{\partial s_2} & \cdots & \frac{\partial f_1}{\partial s_n} \\
\frac{\partial f_2}{\partial s_1} & \frac{\partial f_2}{\partial s_2} & \cdots & \frac{\partial f_2}{\partial s_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial f_n}{\partial s_1} & \frac{\partial f_n}{\partial s_2} & \cdots & \frac{\partial f_n}{\partial s_n}
\end{pmatrix}
\cdot
\begin{pmatrix}
\frac{\partial s_1}{\partial s_1} & \frac{\partial s_1}{\partial s_2} & \cdots & \frac{\partial s_1}{\partial s_n} \\
\frac{\partial s_2}{\partial s_1} & \frac{\partial s_2}{\partial s_2} & \cdots & \frac{\partial s_2}{\partial s_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial s_n}{\partial s_1} & \frac{\partial s_n}{\partial s_2} & \cdots & \frac{\partial s_n}{\partial s_n}
\end{pmatrix}.
\quad (3.1.27)
\]

The first matrix on the right hand side is the familiar Jacobian matrix,
The mapping of variables for a transformation can be given by the Jacobian matrix. The determinant of the Jacobian matrix, often called the functional determinant or “Jacobian”, is the volume element and is used in integral calculus to change the variables of integration. Introducing a transformation for the canonical variables, where \( z' = z' \left( q'_1, ..., q'_d, p'_1, ..., p'_d \right) \), the Poisson bracket can be written as,

\[
\{ z', z' \} \equiv \left( \frac{\partial z'}{\partial z} \right)^T \cdot J \cdot \left( \frac{\partial z'}{\partial z} \right).
\] (3.1.29)

The derivative terms are recognized as the Jacobian matrix. We will refer to the Jacobian matrix as, \( M \), with the elements,

\[
M_{ij} = \frac{\partial z'_i}{\partial z_j}.
\] (3.1.30)

The Poisson bracket definition in one dimension is easily seen as the determinant of the Jacobian matrix:

\[
|M| = \left| \begin{array}{cc} \frac{\partial u}{\partial q} & \frac{\partial u}{\partial p} \\ \frac{\partial v}{\partial q} & \frac{\partial v}{\partial p} \end{array} \right| = \{ u, v \}_{q, p}
\] (3.1.31)

The Poisson bracket for the transformed phase space variables in matrix notation is,

\[
\{ z', z' \}_z = M^T \cdot J \cdot M.
\] (3.1.32)

The right hand side of the equation is an important quantity. If the transformation is canonical, the phase space volume is conserved and the Poisson bracket satisfies the symplectic condition:

\[
M^T \cdot J \cdot M = J
\] (3.1.33)
In the one degree of freedom case, this reduces to the preservation of the phase space area where the determinant of the Jacobian matrix, and therefore the Poisson bracket, is unity. The symplectic condition is also equivalent to the form,

\[ M \cdot J \cdot M^T = J. \]  \hspace{1cm} (3.1.34)

This can be derived by multiplying the original form of the symplectic condition from the left by the inverse of \( M^T \), yielding,

\[ J \cdot M = (M^T)^{-1} J. \]  \hspace{1cm} (3.1.35)

Then multiply from the left by, \(-J\), and from the right by, \(J\), recalling the property, \(J^2 = -I\). The result is,

\[ M \cdot (J) = (J) \cdot (M^T)^{-1}. \]  \hspace{1cm} (3.1.36)

Finally, multiply from the right by, \(M^T\), to get the alternate form of the symplectic condition.

The symplectic condition is a sufficient and necessary condition for canonical transformations [58, 66]. The relation of the Poisson brackets with respect to the canonical variables leads to important conservative properties of canonical transformations.

### 3.2 Canonical Invariants

The Poisson brackets of the canonical variables are the Fundamental Poisson Brackets, and are invariant under canonical transformations. The invariance of the Fundamental Poisson Brackets is equivalent to the symplectic condition for canonical transformations [58, pg. 389]. Then for any canonical transformation, the Poisson bracket of the canonical variables leads to the following invariant property:

\[ \{z, z\}_z = \{z', z'\}_{z'} = \{z', z'\}_z = J. \]  \hspace{1cm} (3.2.1)

In fact, all Poisson brackets are invariant under a canonical transformation. By application of the symplectic condition, this can be shown.
Consider a transformation, $z' = \phi z$. The differential for the $ith$ function of $z$, with respect to the transformed canonical variables is,

$$dv_i = \sum \frac{\partial v_i}{\partial z'_j} \frac{\partial z'_j}{\partial z_i} dz_i.$$  \hspace{1cm} (3.2.2)

Explicitly written out for all $2d$ phase space variables, the differential is,

$$dv_i = \frac{\partial v_i}{\partial q'_1} \frac{\partial q'_1}{\partial q_i} dq_i + \cdots + \frac{\partial v_i}{\partial q'_d} \frac{\partial q'_d}{\partial q_i} dq_i + \frac{\partial v_i}{\partial p'_1} \frac{\partial p'_1}{\partial q_i} dq_i + \cdots + \frac{\partial v_i}{\partial p'_d} \frac{\partial p'_d}{\partial q_i} dq_i.$$  \hspace{1cm} (3.2.3)

The derivative of the $ith$ function with respect to one of the canonical variables is then,

$$\frac{dv_i}{dq_i} = \left( \frac{\partial q'_1}{\partial q_i} \ldots \frac{\partial q'_i}{\partial q_i} \ldots \frac{\partial q'_d}{\partial q_i} \right) \cdot \left( \begin{array}{c} \frac{\partial v_i}{\partial q'_1} \\ \vdots \\ \frac{\partial v_i}{\partial q'_d} \\ \frac{\partial v_i}{\partial p'_1} \\ \vdots \\ \frac{\partial v_i}{\partial p'_d} \end{array} \right).$$  \hspace{1cm} (3.2.4)

Expanded for all $2d$ functions, the equation in matrix form is,

$$\begin{pmatrix} \frac{\partial v_1}{\partial q_1} & \ldots & \frac{\partial v_{2d}}{\partial q_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial v_1}{\partial p_d} & \ldots & \frac{\partial v_{2d}}{\partial p_d} \end{pmatrix} = \begin{pmatrix} \frac{\partial q'_1}{\partial q_1} & \ldots & \frac{\partial q'_i}{\partial q_i} & \ldots & \frac{\partial q'_d}{\partial q_i} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial q'_1}{\partial p_d} & \ldots & \frac{\partial q'_i}{\partial p_d} & \ldots & \frac{\partial q'_d}{\partial p_d} \\ \frac{\partial p'_1}{\partial q_1} & \ldots & \frac{\partial p'_i}{\partial q_i} & \ldots & \frac{\partial p'_d}{\partial q_i} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\partial p'_1}{\partial p_d} & \ldots & \frac{\partial p'_i}{\partial p_d} & \ldots & \frac{\partial p'_d}{\partial p_d} \end{pmatrix} \cdot \begin{pmatrix} \frac{\partial v_1}{\partial q'_1} \\ \vdots \\ \frac{\partial v_1}{\partial q'_d} \\ \frac{\partial v_1}{\partial p'_1} \\ \vdots \\ \frac{\partial v_1}{\partial p'_d} \end{pmatrix}.$$  \hspace{1cm} (3.2.5)

This is equivalent to the matrix notation,

$$\frac{\partial v}{\partial z} = \frac{\partial z'}{\partial z} \frac{\partial v}{\partial z'}.$$  \hspace{1cm} (3.2.6)

The quantity,

$$\frac{\partial z'_i}{\partial z_i},$$

18
is the $M_{ji}$ component of the Jacobian matrix transpose, so that,

$$\frac{\partial v}{\partial z} = M^T \cdot \frac{\partial v}{\partial z'}.$$  \hfill (3.2.7)

Following this prescription, the derivative for the set of functions, $u$, is,

$$\left( \frac{\partial u}{\partial z} \right)^T = \left( M^T \frac{\partial u}{\partial z'} \right)^T = \left( \frac{\partial u}{\partial z'} \right)^T M.$$  \hfill (3.2.8)

The Poisson bracket for a transformation of the canonical variables is then,

$$\{ u, v \}_z = \left( \frac{\partial u}{\partial z} \right)^T \cdot J \cdot \left( \frac{\partial v}{\partial z} \right) = \left( \frac{\partial u}{\partial z'} \right)^T M \cdot J \cdot M^T \left( \frac{\partial v}{\partial z'} \right).$$  \hfill (3.2.9)

If the transformation is canonical, and therefore satisfies the symplectic condition, then

$$\{ u, v \}_z = \left( \frac{\partial u}{\partial z'} \right)^T \cdot J \cdot \left( \frac{\partial v}{\partial z'} \right),$$  \hfill (3.2.10)

or,

$$\{ u, v \}_z = \{ u, v \}_{z'}.$$

It is evident that all Poisson brackets are invariant under a canonical transformation, and that the symplectic condition is an important indicator of such a transformation.

The first integral invariant of Poincare is the conservation of volume:

$$V = \iiint ds_1 ds_2 \ldots ds_n = \iiint ds'_1 ds'_2 \ldots ds'_n.$$  \hfill (3.2.11)
This is equivalent to the tensor product two-form given earlier for the phase space volume. It can be shown that a transformation that satisfies the symplectic condition also preserves volume. The differential for a transformation in matrix notation can be written as,

\[ dz' = \frac{\partial z'}{\partial z} dz, \quad (3.2.12) \]

or in terms of the Jacobian matrix, \( M \),

\[ dz' = M dz. \quad (3.2.13) \]

The differential is the volume element for the phase space:

\[ dz' = dq'_1 \ldots dq'_n dp'_1 \ldots dp'_n. \quad (3.2.14) \]

It is well-known in integral calculus that the volume element can transformed by the absolute value of the functional determinant or Jacobian as shown,

\[ dz' = |M| dz. \quad (3.2.15) \]

The absolute value of the symplectic condition is,

\[ |M^T \cdot J \cdot M| = |M^T| \cdot |J| \cdot |M| = |J|. \quad (3.2.16) \]

This can be rearranged to,

\[ |M^T| \cdot |M| \cdot |J| = |J|. \]

Since the determinant is the same for any matrix and its transpose, \( |M^T| = |M| \), the relation becomes,

\[ |M|^2 \cdot |J| = |J|, \]

so that,
\[ |M| = \pm 1. \]

The volume element is then,

\[ dz' = |M| \, dz = dz. \quad (3.2.17) \]

It is obvious if a transformation satisfies the symplectic condition, it conserves volume.

### 3.3 Hamiltonian Mechanics

In dynamical systems, the Lagrangian or Hamiltonian is often used to express the total energy. The Lagrangian is a function of the position and velocity coordinates, and is the difference between the total kinetic and potential energy of a system, as denoted by \( L = T - V \). The Hamiltonian is a function of the position and momentum and is the sum of the total kinetic and potential energy of a system denoted by \( H = T + V \). The Hamiltonian can be derived from the Lagrangian using Lagrange’s Undetermined Multipliers. The Hamiltonian has certain unique properties that make it an attractive form for the study of dynamics. Since the momentum is an independent variable in the Hamiltonian, simpler and more abstract representations of the energy can be obtained. Cyclic coordinates, variables that do not appear in the energy function, can be ignored in both the Lagrangian and Hamiltonian, thereby reducing the number of equations of motion that need to be integrated. For the Lagrangian, each cyclic coordinate reduces the equations of motion to \( 2n - 1 \). For the Hamiltonian, any conjugate momentum of a cyclic coordinate is also a constant of the motion and can be ignored, reducing the equations of motion to \( 2n - 2 \). The reduction of integration equations is symmetric and two-fold in the Hamiltonian versus the Lagrangian. Once the Hamiltonian and all initial or instantaneous position and momentum coordinates are known, the equations of motion can be integrated over time and the complete dynamics of a classical system can be determined.

The Hamiltonian is a function of the canonical variables, \( q \) and \( p \). It is can also be explicitly dependent upon time, so that \( H = H(p, q, t) \). Time-independent Hamiltonians, \( H = H(q, p) \), are also known as autonomous Hamiltonians. An autonomous Hamiltonian, and hence the energy, is a conserved quantity. We are interested in time-independent Hamiltonians and will focus our study on such systems.

The Hamilton canonical equations of motion are
\[ \dot{q}_i = \frac{\partial H}{\partial p_i}, \]
\[ \dot{p}_i = -\frac{\partial H}{\partial q_i}, \quad (3.3.1) \]

where \( q \) and \( p \) are the generalized position and momentum coordinates, respectively. There are then \( 2d \) equations of motion for \( d \) degrees of freedom. These equations are the simultaneous, first order differential equations that represent the time rate of change of the momentum exactly as Newton’s famous 2nd Law equation for the force,

\[ \vec{F} = \sum_i m_i \frac{d\vec{r}_i}{dt}. \quad (3.3.2) \]

This is a second order, ordinary differential equation \(^{69}\) which can be written for one particle in one dimension as,

\[ \ddot{x} - \frac{F}{m} = 0. \]

By defining \( y_1 = x \) and \( y_2 = \dot{x} \), the higher order differential equation is transformed to a system of two first order, ordinary differential equations,

\[ y'_1 = y_2, \]
\[ y'_2 = \frac{F}{m}. \]

The Hamilton equations of motion are then a system of autonomous differential equations of the form,

\[ y'_1 = f(y_1, y_2), \]
\[ y'_2 = g(y_1, y_2). \]

The right-hand side functions do not explicitly depend on the time variable. If the equations are linear and can be expressed in the matrix form,
\[ y' = Ay, \]

where the coefficient matrix, \( A \), is a constant, then it is possible to obtain analytical solutions using known methods for first-order, linear ordinary differential equations. One such method commonly found in texts on differential equations involves solving an equivalent eigenvalue problem [20]. Using a trial solution, \( y = xe^{\lambda x} \), differentiation and substitution produces,

\[ y' = \lambda xe^{\lambda x} \]
\[ = Axe^{\lambda x}. \]

This results in an eigenvalue equation,

\[ Ax = \lambda x, \]

and can be solved by obtaining the eigenvalues, roots to the secular determinant,

\[ \det(A - \lambda I) = 0, \]

and the eigenvectors, \( \vec{x}_i \). The general solution to the differential equations is then just a linear combination of the trial solution for each eigenvalue as given by,

\[ y = \sum_i c_i \vec{x}_i \exp(\lambda_i x). \]

For an example system of two linear differential equations,

\[
\begin{bmatrix}
  y'_1 \\
  y'_2
\end{bmatrix} =
\begin{bmatrix}
  a_{11} & a_{12} \\
  a_{21} & a_{22}
\end{bmatrix}
\begin{bmatrix}
  y_1 \\
  y_2
\end{bmatrix},
\]

the general solution would be,

\[
\begin{bmatrix}
  y_1 \\
  y_2
\end{bmatrix} = c_1 \begin{bmatrix}
  x_1 \\
  x_2
\end{bmatrix} \exp(\lambda_1 x) + c_2 \begin{bmatrix}
  x_1 \\
  x_2
\end{bmatrix} \exp(\lambda_2 x). \]
Since the equations of motion are partial derivatives of the Hamiltonian, each canonical variable is treated as an independent variable of the integration.

The equations of motion can be generated from the Poisson bracket of the canonical variables with the Hamiltonian. Using notation introduced earlier, the equations of motion can be written as,

\[ \frac{\dot{z}}{} = \{ z, H \} 
\tag{3.3.3} \]

In matrix form for one function, \( z_i \), this is equivalent to,

\[ \dot{z}_i = \left( \nabla_z \cdot z_i \right)^T \cdot J \cdot \left( \nabla_z \cdot H \right) \\
= J \cdot \left( \nabla_z \cdot H \right). \tag{3.3.4} \]

This essentially produces a set of simultaneous equations,

\[ \dot{q}_i = \{ q_i, H \}_{q,p} \\
\dot{p}_i = \{ p_i, H \}_{q,p} \tag{3.3.5} \]

The entire set for the equations of motion are then represented by,

\[ \dot{z} = J \cdot \frac{\partial H}{\partial z}. \tag{3.3.6} \]

For one degree of freedom, the equations of motion are easily reproduced from this relation,

\[ \dot{z} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{pmatrix} \\
= \begin{pmatrix} \dot{q} \\ \dot{p} \end{pmatrix} = \begin{pmatrix} \frac{\partial H}{\partial p} \\ -\frac{\partial H}{\partial q} \end{pmatrix}. \tag{3.3.7} \]

Given a transformation of the canonical variables, the equations of motion can be written as,
\[ \dot{z}' = M \dot{z}, \]

or,

\[ \dot{z}' = M \cdot J \cdot \frac{\partial H}{\partial z}. \]  

(3.3.8)

Recalling the procedure for transformations the matrix, \( \frac{\partial H}{\partial z} \), becomes

\[ \frac{\partial H}{\partial z} = \frac{\partial z'}{\partial z} \frac{\partial H}{\partial z} \]

\[ = M^T \frac{\partial H}{\partial z'}. \]  

(3.3.9)

The equations of motion for the transformation are then,

\[ \dot{z}' = M \cdot J \cdot M^T \cdot \frac{\partial H}{\partial z'}. \]

If the transformation is canonical, it will satisfy the symplectic condition so that,

\[ \dot{z}' = J \frac{\partial H}{\partial z'}. \]  

(3.3.10)

Hamiltonian systems have important conserved quantities, referred to as constants of the motion. Consider a function, \( u(q, p, t) \), of the canonical variables and time so that the time derivative for \( u \) is,

\[ \frac{du}{dt} = \frac{\partial u}{\partial q} \dot{q} + \frac{\partial u}{\partial p} \dot{p} + \frac{\partial u}{\partial t}. \]
Substituting for the equations of motion, this can be rewritten as,

\[
\frac{du}{dt} = \{u, H\}_{q,p} + \frac{\partial u}{\partial t}.
\]  

(3.3.11)

If,

\[
\{u, H\} = -\frac{\partial u}{\partial t} \quad \text{or} \quad \{H, u\} = \frac{\partial u}{\partial t},
\]

then the function, \(u\), is a constant of the motion and does not evolve with time. If \(u\) is not explicitly dependent upon time, the derivative reduces to,

\[
\frac{du}{dt} = \{u, H\}_{q,p}.
\]  

(3.3.12)

In this case, \(u\) is a constant of the motion if the Poisson bracket with the Hamiltonian is zero, \(\{u, H\} = 0\). This leads to an important realization that the Poisson bracket is a test for conserved quantities of the Hamiltonian. Indeed, it is trivially seen from the Poisson bracket properties that an autonomous Hamiltonian is a constant of the motion. Substituting the function, \(u\), with a time-independent Hamiltonian, it is obvious from the result that such a Hamiltonian is an invariant quantity as seen by,

\[
\frac{dH}{dt} = \{H, H\} = 0.
\]

Suppose there are two quantities that are constants of the motion. By application of the Jacobi Identity, a property of the Poisson bracket, it can be shown that the Poisson bracket of two constants of the motion is also a constant of the motion. This is known as Poisson’s Theorem. Let \(u\) and \(v\) be constants of the motion. The Jacobi Identity is then,

\[
\{u, \{v, H\}\} + \{v, \{u, H\}\} + \{H, \{u, v\}\} = 0,
\]  

(3.3.13)

reducing to,

\[
\{H, \{u, v\}\} = 0.
\]
Thus,

\[
\frac{d\{u, v\}}{dt} = 0,
\]

and so the Poisson bracket of \( u \) and \( v \) is a constant of the motion.

### 3.4 Solutions To The Equations Of Motion

The equations of motion have a general solution in closed-form. From infinitesimal canonical transformations (ICT), it is known for a function, \( u \), the differential can be given by the Poisson bracket such as,

\[
du = \varepsilon \{u, G\}.
\]

(3.4.1)

Assume \( \varepsilon \) is \( d\alpha \), so that,

\[
\frac{\partial}{\partial \alpha} u = \{u, G\}.
\]

(3.4.2)

Then the second derivative,

\[
\frac{\partial}{\partial \alpha} \left( \frac{\partial u}{\partial \alpha} \right),
\]

is,

\[
\frac{\partial^2 u}{\partial \alpha^2} = \{\{u, G\}, G\}.
\]

(3.4.3)

Higher derivatives are just a repetition of nesting the Poisson brackets.

The function, \( u \), can be expanded by a Taylor Series centered about \( \alpha_0 \) as given by,
\[ u(\alpha) = \sum_{n=0}^{\infty} \frac{(\alpha - \alpha_0)^n}{n!} \frac{\partial^n}{\partial \alpha^n} u(\alpha = \alpha_0). \] (3.4.4)

The derivatives of the function are evaluated at, \( \alpha = \alpha_0 \). The expansion can be written in terms of the Poisson brackets as,

\[ u(\alpha) = u_0 + \alpha \{u, G\}_0 + \frac{\alpha^2}{2!} \{\{u, G\}, G\}_0 + \frac{\alpha^3}{3!} \{\{\{u, G\}, G\}, G\}_0 + \ldots, \] (3.4.5)

where \( \alpha_0 = 0 \). The subscripts indicate the brackets are evaluated at zero. The series is similar to that for the exponential function,

\[ e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots. \] (3.4.6)

Allow \( \{\cdot, G\} \) to be an operator such that,

\[ \{\cdot, G\} f = \{f, G\}. \] (3.4.7)

The square of this operator, \( \{\cdot, G\} \cdot \{\cdot, G\} \), is then

\[ \{\cdot, G\}^2 = \{\{\cdot, G\}, G\}. \] (3.4.8)

Higher order brackets are obtained by succession of the operation, repeatedly nesting each bracket. Introducing an exponential function, \( e^{\alpha \{\cdot, G\}} \), where \( \alpha \) is a real scalar, the Taylor series for the exponent expanded about \( (\alpha \{\cdot, G\}) = 0 \) is given by,

\[ e^{\alpha \{\cdot, G\}} = \sum_{n=0}^{\infty} \frac{(\alpha \{\cdot, G\})^n}{n!} \frac{\partial^n}{\partial (\alpha \{\cdot, G\})^n} e^{\alpha \{\cdot, G\}} (\alpha \{\cdot, G\} = 0) \]

\[ = 1 + \alpha \{\cdot, G\} + \frac{\alpha^2}{2!} \{\cdot, G\}^2 + \frac{\alpha^3}{3!} \{\cdot, G\}^3 + \ldots. \] (3.4.9)

Comparing to the series for the function, \( u \), it is clear that the solution for \( u \) is exponential and can then be written as,
\[ u(\alpha) = e^{\alpha \{\cdot, G\}} u_0. \] (3.4.10)

It is evident that this is an evolution mapping of the function, \( u \), from some starting value, where the exponential, \( e^{\alpha \{\cdot, G\}} \), is the evolution operator. As noted in Goldstein \[58\], there is a correlation between the classical Poisson Bracket evolution operator and the quantum mechanical evolution operator. The forms are nearly identical.

The evolution operator can be expressed using Lie Algebra notation introduced by Dragt and Finn \[39\]. Let the Poisson bracket operator be written as,

\[ : f := \{\cdot, G\}. \] (3.4.11)

The Taylor series for the evolution operator expanded about, \( \alpha : f := 0 \), is given by,

\[ e^{\alpha : f} = \sum_{n=0}^{\infty} \frac{(\alpha : f)^n}{n!} \frac{\partial^n}{\partial (\alpha : f)^n} e^{\alpha : f} (\alpha : f := 0). \] (3.4.12)

The evolution mapping for the canonical transformation is then,

\[ u(\alpha) = e^{\alpha : f} u(\alpha_0). \] (3.4.13)

Substitute the time variable, \( t \), for \( \alpha \), the canonical variables function, \( z \), for \( u \), and the Hamiltonian, \( H \), for \( f \). The equations of motion in Lie algebra form are then,

\[ \frac{dz}{dt} =: H : z, \] (3.4.14)

and the exact phase flow generated by, \( : H : \), is

\[ z(t) = e^{t : H :} z(t_0). \] (3.4.15)

In expanded form, this equates to,

\[ z = z_0 + t \{z_0, H\} + \frac{t^2}{2!} \{z_0, H\}^2 + \frac{t^3}{3!} \{z_0, H\}^3 + \ldots. \] (3.4.16)
This is a time evolution of the phase space, where each canonical variable evolves in time as,

\[ q(t) = q_0 + t \{ q_0, H \} + \frac{t^2}{2!} \{ q_0, H \}^2 + \frac{t^3}{3!} \{ q_0, H \}^3 + \ldots \]  
\[ p(t) = p_0 + t \{ p_0, H \} + \frac{t^2}{2!} \{ p_0, H \}^2 + \frac{t^3}{3!} \{ p_0, H \}^3 + \ldots \]  

(3.4.17) 

(3.4.18)

The series given by the evolution or phase flow mapping is a natural integrator of the solution. However, this is an open-form solution requiring an infinite number of terms in the series to produce the exact solution to the equations of motion. Truncation of the series will be required resulting in an approximate solution.
Chapter 4

Symplectic Integration Algorithms

4.1 Numerical Integration Overview

An exactly integrable problem is one for which a solution can be determined analytically and expressed in closed form. Very few problems can be solved exactly, without any error. The majority of problems must be solved by many iterative steps to arrive at suitable approximations to the true solution, usually with the aid of a computer. The area of numerical integration encompasses many algorithmic approaches to solutions of a discrete nature. Symplectic integrators are one such approach for solving Hamiltonian dynamics.

Numerical integration techniques generate an approximation of the exact solution for a differential equation. For every ordinary or partial differential equation, there is a family of solution paths that comprise the general solution. Each solution path can not cross the path of any other. There exists a unique solution for given initial boundary values, where the solution curve can then be followed since its path will always be the unique path for the given starting point. Consider a first-order, ordinary differential equation (ODE) of the form,

$$\frac{dy}{dx} = f(x, y),$$

for which an initial value is known. A unique solution exists if both $y$ and its derivative are continuous in some region of space, by the Existence and Uniqueness Theorem \[20\] \[32\] \[37\]. Such an ODE can be rewritten in the form, $M(x, y) \, dx + N(x, y) \, dy = 0$. If the functions are in terms of just one independent variable so that the differential equation can be written as, $M(x) \, dx + N(y) \, dy = 0$, then the equation is separable and each function can
be integrated separately and simultaneously. A simple example would be, \( y' + y = 0 \), where \( N(x, y) \) is 1 and \( M(x, y) \) is just \( y \). This equation can be trivially and exactly integrated by separation of variables, resulting in the general solution, \( y(x) = e^{-x} \). The integrals for such differential equations can be determined analytically and are therefore, exactly integrable problems.

In the absence of an analytic solution, the integration of a problem can be performed iteratively using an integration algorithm. The basic premise of numerical integration is to follow the tangent line, or first derivative, starting from a known initial value to obtain the next value for the solution. This is continued from the newly obtained value, and the “steps” are repeated until the desired interval limit has been reached. This produces a succession of points, \((y_0, y_1, \ldots, y_n)\), obtained by the discrete steps, \((x_0, x_1, \ldots, x_n)\), over some interval. The closer the individual steps are to each other, the shorter the slopes, or derivatives of \( y(x) \), and hence a better approximation or fit to the solution. If an infinite number of steps with infinitesimal step size, \((x_{n+1} - x_n) \to 0\), could be taken within the desired interval, assuming convergence and infinite precision, the exact solution is obtained since the sum of infinitesimal areas under the slope is the act of the continuous integral.

The simplest method for numerically integrating an initial-value, first-order ODE is the Euler method. The general mapping is expressed by, \( y(x+1) = \phi(x) y(x) \). The Euler method can be derived by truncating the Taylor Series expansion for the solution after the linear term. The Taylor series expansion for the solution, centered about \( x_n \), is given by,

\[
y(x) = \sum_{i=0}^{\infty} \frac{(x - x_n)^i}{i!} \frac{\partial^i}{\partial x^i} y(x = x_n) \\
= y(x_n) + (x - x_n) y'(x_n) + \frac{(x - x_n)^2}{2!} y''(x_n) + \frac{(x - x_n)^3}{3!} y'''(x_n) + \ldots. \tag{4.1.1}
\]

Allowing, \( h = (x_{n+1} - x_n) \), the Euler method is then,

\[
y_{n+1} = y_n + hf(x_n, y_n), \tag{4.1.2}
\]

where \( f(x, y) \) is the derivative of \( y \). Since the infinite series is truncated, the method results in an approximate solution. In this case, the error is of order, \( O(h^2) \), so the solution is only accurate up to the quadratic term in the series. This is known as the local truncation error. A better approximation can be made by including the higher order terms in the Taylor series expansion, but at the price of determining higher-order derivatives. Other suitable methods exist, such as the fourth-order Runge-Kutta method which provides higher accuracy using first-order derivatives. Another source of error is due to the inability to express all decimal
numbers, also known as floating-point numbers, exactly. Computer precision is finite, leading to necessary truncation of floating point numbers. This leads to the infamous rounding or round-off error in computations. In addition, the representation of all floating point numbers on a computer is a discretization, and therefore only a subset, of the continuum of real numbers [57, 69].

The Euler method is an explicit, single-step integration technique of constant step size, $h$. In contrast to multi-step methods, the Euler method uses only one previous point or value to advance to the next point. It is an explicit method because the determination of a new point does not require knowledge of the new point. Conversely, implicit methods have the dependent variable on both sides of the integration equation and must use iteration at each step before advancement. These implicit integration techniques are generally much slower and more intensive to calculate. An example would be the implicit Euler method,

$$y_{n+1} = y_n + hf(x_n, y_{n+1}). \quad (4.1.3)$$

### 4.2 Symplectic Integration Algorithms

Integration methods that preserve the canonicity of Hamiltonian systems are known as Symplectic Integration Algorithms (SIAs). These are a special class of geometric integrators that approximate the exact phase flow of the Hamiltonian and produce a canonical, or symplectic, transformation at each integration step. This property ensures the preservation of global geometric properties, particularly the Poincare integral invariants such as the phase space volume. These properties are essential for studying the topology and long-term, qualitative behavior of a system.

An integration method is symplectic if, and only if, the symplectic condition holds for each integration step. If the method is symplectic, the time evolution is a canonical transformation, and the phase space volume will be conserved. This is indicated by the Poisson bracket for one integration step, which will be unity if the method is symplectic:

$$\{q_{n+1}, p_{n+1}\}_{q_n, p_n} = \frac{\partial q_{n+1}}{\partial q_n} \frac{\partial p_{n+1}}{\partial q_n} - \frac{\partial p_{n+1}}{\partial q_n} \frac{\partial q_{n+1}}{\partial p_n} = 1. \quad (4.2.1)$$

This follows from invariance of the Poisson brackets under a canonical transformation, such that $\{z', z'\}_z = J$. We will test the Euler method for the Simple Harmonic Oscillator (SHO), where the Hamiltonian is given by,
\[ H = \frac{p^2}{2} + \frac{q^2}{2}, \quad (4.2.2) \]

and the equations of motion are simply,

\[ \dot{q} = p \]
\[ \dot{p} = -q \]

The Euler method algorithm is then,

\[ q_{n+1} = q_n + t\dot{q}_n \]
\[ p_{n+1} = p_n + t\dot{p}_n \]

Substituting for the derivatives of the canonical variables with the equations of motion, produces,

\[ q_{n+1} = q_n + tp_n \]
\[ p_{n+1} = p_n - t\dot{q}_n \]

The Poisson bracket is then,

\[ \{q_{n+1}, p_{n+1}\}_{q_n, p_n} = (1) (1) - (-t) (t), \]

the result being,

\[ 1 + t^2 \neq 1. \]

It is obvious that the Euler method does not satisfy the symplectic condition and is therefore not an SIA. The phase space area will not be preserved for long-term integration. As the step size, \( t \), tends to zero, the integrator approaches symplecticity but will never be exactly symplectic. In most cases, smaller step sizes increase computational effort and are therefore
undesirable. Conversely, symplectic integrators preserve the canonical invariants at larger step sizes than similar ordered, conventional integrators. Studies in astronomy typically have very long time scales that are many times greater than the period of the dynamical bodies and even the age of the solar system or universe [190, 191, 56, 180, 152, 100, 131]. Qualitatively accurate results using relatively large step sizes have been reported for symplectic integrations [190, 191, 180]. The evolution of the outer planets was integrated by Wisdom and Holman in 1991 [191] for about one billion years with a step size of one year. In relation, the period of Jupiter is approximately twelve years. The accuracy at a step size of one year for the symplectic integration is comparable to step sizes of forty days or less with non-symplectic integration experiments [191]. The symplectic integrators allowed larger step sizes and therefore required less computational time without adversely undermining accuracy.

Suppose there is an integration method, similar to the Euler method, such that the solutions are given by,

\[
\begin{align*}
q_{n+1} &= q_n + t \frac{\partial H}{\partial p_n} (q_n, p_n) \\
p_{n+1} &= p_n - t \frac{\partial H}{\partial q_n} (q_{n+1}, p_n)
\end{align*}
\]

The distinction is in the evaluation of the derivatives at the coordinates in the parentheses. The position variable must be updated before the momentum variable, leading to an asymmetric time evolution of the variables. This bias is in contrast to the simultaneous transformation in the Euler method and other integration algorithms. Referring to the Simple Harmonic Oscillator again, the transformation steps are,

\[
\begin{align*}
q_{n+1} &= q_n + tp_n \\
p_{n+1} &= p_n - tq_{n+1} = (1 - t^2) p_n - tq_n
\end{align*}
\]

Evaluating the Poisson bracket for this integration method yields,

\[
\{q_{n+1}, p_{n+1}\}_{q_n, p_n} = \left(1 + t \frac{\partial^2 H}{\partial q_n \partial p_n} (q_n, p_n)\right) \left(1 - t \frac{\partial^2 H}{\partial p_n \partial q_n} (q_{n+1}, p_n)\right) \\
- \left(-t \frac{\partial^2 H}{\partial q_n^2} (q_{n+1}, p_n)\right) \left(t \frac{\partial^2 H}{\partial p_n^2} (q_n, p_n)\right) \\
= (1) (1 - t^2) - (-t) (t) = 1
\]

35
The Poisson bracket evaluates to unity so the integration method is symplectic. The adjoint of the method, obtained by reversing the advancement order of the canonical variables, also yields a symplectic integrator. This method is the symplectic Euler or Ruth explicit, first-order symplectic integrator.

4.3 Explicit Symmetrized Symplectic Composition Methods

The product of two canonical maps is also a canonical map \[58\]. Symplectic integrators can then be generated by a composition of symplectic integrators. Multi-map composition methods were introduced in 1990 by Forest and Ruth \[51\], Yoshida \[194\], and independently by Wisdom and Holman in 1991 \[191\]. These methods provide the basis for some of the popular explicit symplectic integrator algorithms. The basic premise will be described and used to construct a symplectic integration algorithm for this dissertation.

An exponential operator can be fashioned in the likeness of the time evolution operator for canonical transformations derived earlier. Consider two, non-commutative operators, A and B. These operators form a non-Abelian, symplectic Lie group where all elements of the group are non-commutative and can form a symplectic map as an exponential Poisson bracket operator. Allow \( \tau \) to denote the step size and be some small real number. A new exponential operator can be then be constructed as the Taylor series expansion,

\[
\exp[\tau (A + B)] = 1 + \tau (A + B) + \frac{\tau^2}{2!} (A + B)^2 + \frac{\tau^3}{3!} (A + B)^3 + \ldots + O (\tau^{n+1}).
\] (4.3.1)

This new exponential operator is symplectic if it produces the solutions to the Hamiltonian equations of motion. It is generally not possible to explicitly determine solutions generated by the exact form of this operator, which requires an infinite summation. The exponential operator can be approximated to any desired order in the time step, \( \tau \), by a product of exponential operators, known as the Lie-Trotter formula,

\[
\exp[\tau (A + B)] = \prod_{i=1}^{k} \exp (c_i \tau A) \cdot \exp (d_i \tau B) + O (\tau^{n+1}).
\] (4.3.2)

The constant coefficients, \((c_1, \ldots, c_k)\) and \((d_1, \ldots, d_k)\), are real numbers and the individual operators, \(\exp (c_i \tau A)\) and \(\exp (d_i \tau B)\), produce exactly solvable symplectic maps. This is
a symplectic mapping since it is a product of symplectic mappings. The approximation is carried out by multiplying the Taylor series expansion of each exponential in the product series and collecting the like-ordered terms up to some desired order, \(O(\tau^n)\), to produce a polynomial expression that compares with the exact operator expansion. Care should be taken to avoid commutative rearrangement since the operators do not commute. The constant coefficients, \((c_1, \ldots, c_k)\) and \((d_1, \ldots, d_k)\), are determined by equating each monomial term with the analog in the exact expansion to yield the appropriate set of non-linear, algebraic equations needed to determine the coefficients.

Recall the Hamiltonian Lie group operator,

\[
H := \{\cdot, H\},
\]

used to generate the exact phase flow, \(z(t) = e^{tH}z_0\). If the Hamiltonian can be written in the form, \(H = H_1 + H_2\), where the two sub-Hamiltonians relate to the non-commutative operators, \(A\) and \(B\), and can be solved analytically, then the exact solution for the equations of motion can be given by,

\[
z(\tau) = \exp \left[ \tau (A + B) \right] z_0. \tag{4.3.3}
\]

As stated earlier, the solution given above can not be solved analytically or computed exactly, since the series expansion requires an infinite number of terms. However, the exact solution can be approximated using Lie-Trotter formula,

\[
z'(\tau) = \left( \prod_{i=1}^{k} \exp (c_i \tau A) \cdot \exp (d_i \tau B) + O(\tau^{n+1}) \right) z_0. \tag{4.3.4}
\]

Essentially, the Hamiltonian time evolution operator for canonical transformations has been redefined using an operator splitting method, and approximated to some desired order, \(O(\tau^n)\). The expression is a general two-map, \(n\) ordered symplectic integrator. The approximation yields the procession of \(i = 1\) to \(i = k\) maps or sub steps,

\[
p_i = p_{i-1} - \tau c_i \frac{\partial H(q_{i-1}, p_{i-1})}{\partial q}
\]

\[
q_i = q_{i-1} + \tau d_i \frac{\partial H(q_{i-1}, p_{i})}{\partial p} \tag{4.3.5}
\]
This method constitutes a composition of maps based on operator splitting where, \( q_0 \) and \( p_0 \), are the known initial values. The composition is symplectic since symplectic diffeomorphisms form a canonical transformation group. The canonical variable transformation used in the composition is of the form derived by Ruth using generating functions. The momentum variable is advanced before the position variable. For a separable Hamiltonian, \( H = T(p) + V(q) \), the maps are,

\[
\begin{align*}
    p_i &= p_{i-1} - \tau c_i \frac{\partial V}{\partial q}(q_{i-1}) \\
    q_i &= q_{i-1} + \tau d_i \frac{\partial T}{\partial p}(p_{i-1})
\end{align*}
\]

(4.3.6)

In Yoshida’s 1990 derivation [194], the advancement is reversed and is given by,

\[
\begin{align*}
    q_i &= q_{i-1} + \tau c_i \frac{\partial T}{\partial p}(p_{i-1}) \\
    p_i &= p_{i-1} - \tau d_i \frac{\partial V}{\partial q}(q_{i-1})
\end{align*}
\]

(4.3.7)

To develop a specific \( nth \) ordered integrator from the two-map composition method, the coefficients, \((c_1, \ldots, c_k)\) and \((d_1, \ldots, d_k)\), will need to be determined for some \( k \) number of sub steps. The direct approach will be described first.

For a second-order approximation, truncating all higher-ordered terms, the product series in terms of the expanded operators is,

\[
\exp [\tau (A + B)] = \prod_{i=1}^{k} \left( 1 + \tau c_i A + \frac{\tau^2}{2!} c_i^2 A^2 + O(\tau^3) \right) \left( 1 + \tau d_i B + \frac{\tau^2}{2!} d_i^2 B^2 + O(\tau^3) \right) .
\]

(4.3.8)

A new and simple algebraic technique, based on repeated reduction of binary products, is introduced to solve for the constant coefficients. Allowing,

\[
    a_i = \frac{\tau}{2} c_i A \quad \text{and} \quad b_i = \frac{\tau}{2} d_i B,
\]

the series can be rewritten as,
\[
\exp[\tau (A + B)] = \prod_{i=1}^{k} \left( 1 + 2 \left( a_i + a_i^2 \right) \right) \left( 1 + 2 \left( b_i + b_i^2 \right) \right).
\] (4.3.9)

Starting with the first term in the product series, the multiplication can be further simplified with the substitution,

\[
A_i = 2 \left( a_i + a_i^2 \right), B_i = 2 \left( b_i + b_i^2 \right),
\] (4.3.10)

to give,

\[
(1 + A_1) (1 + B_1).
\] (4.3.11)

Multiplying with the second term of the product series and expanding produces,

\[
(1 + A_1) (1 + B_1) (1 + A_2) (1 + B_2) = (1 + A_1 + B_1 + A_1 B_1) (1 + A_2 + B_2 + A_2 B_2).
\] (4.3.12)

This product can be reduced by allowing another substitution, \( U_i = A_i + B_i + A_i B_i \), to yield,

\[
(1 + U_1) (1 + U_2).
\] (4.3.13)

Again, multiplying with the next term in the series and expanding yields,

\[
(1 + U_1) (1 + U_2) (1 + U_3) (1 + U_4) = (1 + U_1 + U_2 + U_1 U_2) (1 + U_3 + U_4 + U_3 U_4).
\] (4.3.14)

With each multiplicative term in the product series, the product can be simplified to an irreducible form, \((1 + N_i) (1 + N_j)\), using a substitution of the kind, \( S = X + Y + XY \). The procedure is repeated until the desired limit. Multiplying all factors and unraveling the substitutions produces the final result. Limiting the product series to \( k = 4 \) terms is sufficient for this exercise. Continuing with the second-order approximation, the product is
expanded for all factors. Since the order of the substitution form is always quadratic, all
cubic or higher terms in the expanded polynomial can be ignored. The result is,

\[
\exp(2a_1) \exp(2b_1) \exp(2a_2) \exp(2b_2) \exp(2a_3) \exp(2b_3) \exp(2a_4) \exp(2b_4)
= 1 + U_1 + U_2 + U_3 + U_4 + U_1U_2 + U_1U_3 + U_1U_4 + U_2U_3 + U_2U_4 + U_3U_4.
\] (4.3.15)

Regardless of how many terms are in the product series, the right hand side can always be
expressed in the form above for quadratic order. Unraveling the outer layer of substitution
and dropping all cubic and higher terms yields,

\[
1 + (A_1 + A_2 + A_3 + A_4) + (B_1 + B_2 + B_3 + B_4)
+ (A_1A_2 + A_1A_3 + A_1A_4 + A_2A_3 + A_2A_4 + A_3A_4)
+ (B_1B_2 + B_1B_3 + B_1B_4 + B_2B_3 + B_2B_4 + B_3B_4)
+ (A_1B_1 + A_1B_2 + A_1B_3 + A_1B_4 + A_2B_3 + A_2B_4 + A_3B_3 + A_3B_4 + A_4B_4)
+ (B_1A_2 + B_1A_3 + B_1A_4 + B_2A_3 + B_2A_4 + B_3A_4).
\] (4.3.16)

Repeating the process the product reduces to,

\[
1 + 2 \left[ (a_1 + a_1^2) + (a_2 + a_2^2) + (a_3 + a_3^2) + (a_4 + a_4^2) \right]
+ 2 \left[ (b_1 + b_1^2) + (b_2 + b_2^2) + (b_3 + b_3^2) + (b_4 + b_4^2) \right]
+ 4 \left[ a_1a_2 + a_1a_3 + a_1a_4 + a_2a_3 + a_2a_4 + a_3a_4 \right]
+ 4 \left[ b_1b_2 + b_1b_3 + b_1b_4 + b_2b_3b_2b_4 + b_3b_4 \right]
+ 4 \left[ a_1b_1 + a_1b_2 + a_1b_3 + a_1b_4 + a_2b_2 + a_2b_3 + a_2b_4 + a_3b_3 + a_3b_4 + a_4b_4 \right]
+ 4 \left[ b_1a_2 + b_1a_3 + b_1a_4 + b_2a_3 + b_2a_4 + b_3a_4 \right].
\] (4.3.17)

A pattern is evident, so for a second-order approximation, the product of \(k\) exponential
terms is,

\[
1 + \left[ 2 \sum_{i=1}^{k} a_i + 2 \sum_{i=1}^{k} b_i \right] + \left[ 2 \sum_{i=1}^{k} a_i^2 + 2 \sum_{i=1}^{k} b_i^2 + 4 \sum_{i=1}^{k-1} a_i \left( \sum_{j=i+1}^{k} a_j \right) + 4 \sum_{i=1}^{k-1} b_i \left( \sum_{j=i+1}^{k} b_j \right) \right]
\] (4.3.18)

\[ + \left[ 4 \sum_{i=1}^{k} a_i \left( \sum_{j=i}^{k} b_j \right) + 4 \sum_{i=1}^{k-1} b_i \left( \sum_{j=i}^{k} a_j \right) \right].\]
Grouping the like-ordered terms the second-order exponential operator approximation can be written as,

\[
\exp [\tau (A + B)] = \prod_{i=1}^{k} \exp (c_i \tau A) \cdot \exp (d_i \tau B) + O (\tau^3)
\]

\[
= 1 + \tau \left[ \left( \sum_{i=1}^{k} c_i \right) A + \left( \sum_{i=1}^{k} d_i \right) B \right] + \tau^2 \left[ \left( \frac{1}{2!} \sum_{i=1}^{k} c_i^2 + \sum_{i=1}^{k-1} c_i \cdot \left( \sum_{j=i+1}^{k} c_j \right) \right) A^2 + \left( \frac{1}{2!} \sum_{i=1}^{k} d_i^2 + \sum_{i=1}^{k-1} d_i \cdot \left( \sum_{j=i+1}^{k} d_j \right) \right) B^2 \right]
\]

\[
+ \left( \sum_{i=1}^{k} c_i \cdot \left( \sum_{j=i}^{k} d_j \right) \right) AB + \left( \sum_{i=1}^{k-1} d_i \cdot \left( \sum_{j=i+1}^{k} c_j \right) \right) BA \right] + O (\tau^3).
\]

(4.3.19)

This expression can then be equated to the exact Taylor series expansion for, \( \exp [\tau (A + B)] \) to second-order, to determine the constant coefficients. The procedure just described can be facilitated by symbolic computation to carry more terms from the product series.

The first order term requires the coefficients satisfy the equations,

\[
c_1 + c_2 + \cdots + c_k = 1,
\]

\[
d_1 + d_2 + \cdots + d_k = 1.
\]

(4.3.20)

For a first order approximation, \( n = 1 \), a symplectic mapping with the least number of sub steps can be obtained from the trivial scenario, \( k = c_1 = d_1 = 1 \), yielding a first-order symplectic integrator,

\[
\exp (\tau (A + B)) = \exp (\tau A) \cdot \exp (\tau B) + O (\tau^2).
\]

(4.3.21)

The second order term requires the coefficients satisfy,

\[
c_1 (d_1 + d_2 + \cdots + d_k) + c_2 (d_2 + d_3 + \cdots + d_k) + \cdots + c_k d_k = \frac{1}{2},
\]

(4.3.22)
as seen from the coefficient of the $AB$ product. The simplest second order symplectic mapping then derives from $k = 2$, $c_1 = c_2 = \frac{1}{2}$, $d_1 = 1$, and $d_2 = 0$ to yield,

$$\exp[\tau(A + B)] = \exp\left(\frac{1}{2}\tau A\right) \exp(\tau B) \exp\left(\frac{1}{2}\tau A\right) + O(\tau^3).$$

(4.3.23)

This is the familiar Leapfrog integrator, and matches the Ruth explicit, second-order symplectic integrator. Following this procedure, third and fourth order Ruth-type symplectic integrators were derived [22], with the fourth-order integrator coefficients given by,

\[
\begin{align*}
    c_1 &= c_4 = \frac{1}{2 \left(2 - \frac{1}{2}\right)}, \\
    d_1 &= d_3 = \frac{1}{2 - \frac{1}{2}}, \\
    c_2 &= c_3 = \frac{1 - \frac{3}{2}}{2 \left(2 - \frac{1}{2}\right)}, \\
    d_2 &= -\frac{\frac{1}{2}}{2 - \frac{3}{2}}, d_4 = 0,
\end{align*}
\]

(4.3.24)

for $k = 4$ sub steps. This method quickly becomes unwieldy for higher-ordered integrators, even with the aid of symbolic computation. In 1990, Yoshida developed a technique that was based on the two-map composition of lower-ordered integrators developed with the previous scheme, to produce $2^N + 2$ ordered symplectic integrators [194].

### 4.4 Lie Group Approach To Symplectic Integrators

The product of symplectic maps as exponential Poisson bracket operators can be facilitated by the Cambell-Baker-Hausdoff (CBH) formula. The formula is known for constructing a single Lie transformation from the product of two Lie transformations [39]. The Lie group properties of the CBH formula were exploited by Forest and Ruth [51] to simplify and generalize the composition method for Ruth type symplectic integrators. The general expression of the CBH formula for two non-commutative operators, $A$ and $B$, with coefficients, $\alpha$ and $\beta$, is given by,

$$\exp(\alpha A) \exp(\beta B) = \exp(C)$$

(4.4.1)

where,
\[ C = \alpha A + \beta B + \frac{1}{2} (\alpha \beta) [A, B] \]
\[ + \frac{1}{12} (\alpha^2 \beta) [A, A, B] + \frac{1}{12} (\alpha \beta^2) [B, B, A] \]
\[ + \frac{1}{24} (\alpha^2 \beta^2) [A, B, B, A] + \cdots . \]  
\[ (4.4.2) \]

The brackets in the expression are commutators where,

\[ [X, Y] = XY - YX, \]
\[ [X, X, Y] = [X, [X, Y]]. \]  
\[ (4.4.3) \]

The usual commutation rules and nesting apply. With the exception of the linear term, the new Lie group operator, \( C \), is expressed only in terms of the commutation relations of \( A \) and \( B \), which are the Lie products. Expanding \( \exp(C) \) in a series, it is obvious that a first-order symplectic mapping can be obtained from the CBH formula with \( \alpha = \beta = t \), exactly matching the Lie-Trotter formula for one sub step. The second order symplectic integrator can also be generated from the CBH formula. Following an approach employed by Yoshida, the CBH formula is reapplied in a symmetric fashion, with \( C \) being one of the operators, to obtain a symmetric triple product as follows,

\[ \exp (\alpha A) \exp (\beta B) \exp (\alpha A) = \exp (W) . \]  
\[ (4.4.4) \]

where,

\[ W = 2\alpha A + \beta B + \frac{1}{6} (\alpha \beta^2) [B, B, A] - \frac{1}{6} (\alpha^2 \beta) [A, A, B] \]
\[ + \frac{7}{360} (\alpha^4 \beta) [A, A, A, B] - \frac{1}{360} (\alpha \beta^4) [B, B, B, A] \]
\[ + \frac{1}{90} (\alpha^2 \beta^3) [A, B, B, A] + \frac{1}{45} (\alpha^3 \beta^2) [B, A, A, B] \]
\[ - \frac{1}{60} (\alpha^3 \beta^2) [A, A, B, A] + \frac{1}{30} (\alpha^2 \beta^3) [B, B, A, B] + \cdots . \]  
\[ (4.4.5) \]

Squaring, \( W \), and ignoring any cubic or higher terms, the CBH formula produces a second-order mapping identical to the Leapfrog integrator when \( \alpha = \frac{1}{2} \tau, \beta = \tau \). The conclusion is
more elegantly expressed by Forest and Ruth using Lie operators. It should be noted that the order of the terms in the symmetrized triple product obtained from the CBH formula is always odd. This property holds for all symmetric operators that obey time reversal so that,

\[ S(\tau) S(-\tau) = S(-\tau) S(\tau) = I, \tag{4.4.6} \]

where \( I \) is the identity. A general proof was provided by Yoshida as follows. The operator can be written generally for forward and backward time as,

\[
S(\tau) = \exp(\tau \gamma_1 + \tau^2 \gamma_2 + \tau^3 \gamma_3 + \tau^4 \gamma_5 + \cdots)
\]

\[
S(-\tau) = \exp(-\tau \gamma_1 + \tau^2 \gamma_2 - \tau^3 \gamma_3 + \tau^4 \gamma_4 - \tau^5 \gamma_5 + \cdots), \tag{4.4.7}
\]

where the parameters, \((\alpha, \beta)\), are each proportional to \(\tau\). All the odd powered terms cancel in the product of the two operators so that,

\[ S(\tau) S(-\tau) = \exp(2\tau^2 \gamma_2 + 2\tau^4 \gamma_4 + \cdots). \tag{4.4.8} \]

To satisfy the identity property, the terms of the exponent must vanish. Hence, all even powered terms are necessarily zero.

A fourth order symplectic map was derived from the CBH formula by Forest and Ruth with far greater simplicity than had previously been done. Similarly, a sixth order symplectic integrator was later developed by Forest using the same Lie group approach. The method can be extended to \(N\) maps for general Hamiltonians that can be decomposed into \(N\) exactly solvable parts, and would always be of second order \(51\). An example three-map operator was provided in their article. The important aspect of the composition method for higher orders is the symmetric product of exponential operators. The explanation for this characteristic was provided by Forest \(49\). It was about the same time that Yoshida published his method for generating higher even-order integrators using a simple technique of symmetrizing lower-ordered maps with the CBH formula. The technique is easily demonstrated for the fourth order integrator. Borrowing notation from Yoshida, the fourth order map is given by,

\[ S_{4th}(\tau) = S_{2nd}(x_1 \tau) S_{2nd}(x_0 \tau) S_{2nd}(x_1 \tau), \tag{4.4.9} \]

where,
\[ S_{2\text{nd}} = \exp \left( \tau \gamma_1 + \tau^3 \gamma_3 + \tau^5 \gamma_5 + \cdots \right). \]  

(4.4.10)

All quadratic and higher monomials are in terms of commutators with,

\[
\begin{align*}
\gamma_1 &= A + B \\
\gamma_3 &= \frac{1}{12} [B, B, A] - \frac{1}{24} [A, A, B] \\
\gamma_5 &= \frac{7}{5760} [A, A, A, A, B] + \cdots.
\end{align*}
\]

(4.4.11)

Adding the powers for the product of exponentials, the fourth order map becomes,

\[ S_{4\text{th}} (\tau) = \exp \left( \tau (x_0 + 2x_1) \gamma_1 + \tau^3 (x_0^3 + 2x_1^3) \gamma_3 + \tau^5 (x_0^5 + 2x_1^5) \gamma_5 + \cdots \right). \]  

(4.4.12)

The exact solution to fourth order requires,

\[ x_0 + 2x_1 = 1, x_0^3 + 2x_1^3 = 0, \]

eliminating any commutator terms. The real solution for the two equations is,

\[ x_0 = -\frac{2^{\frac{1}{3}}}{2 - 2^{\frac{1}{3}}}, \quad x_1 = \frac{1}{2 - 2^{\frac{1}{3}}}. \]

(4.4.13)

The coefficients from the operator approximation are then,

\[
\begin{align*}
\alpha_1 &= \alpha_4 = \frac{1}{2} x_1, & \alpha_2 &= \alpha_3 = \frac{1}{2} (x_0 + x_1) \\
\beta_1 &= \beta_3 = x_1, & \beta_2 &= x_0
\end{align*}
\]

(4.4.14)

This exactly matches the Ruth type fourth order integrator for a two-map splitting. A six order integrator was then determined using the symmetrized product of fourth order...
integrators. A general method for determining symplectic integrators of arbitrary even order was provided by Yoshida. Given a symmetric $2n$ order map, the procedure can be generalized for a $2n + 2$ order map by,

$$S_{2n+2}(\tau) = S_{2n}(z_1\tau) S_{2n}(z_0\tau) S_{2n}(z_1\tau),$$  \hspace{1cm} (4.4.15)

where

$$z_0 + 2z_1 = 1,$$  \hspace{1cm} (4.4.16)

The general solution being,

$$z_0 = -\frac{1}{2 - 2^{1/(2n+1)}}, \hspace{1cm} z_1 = \frac{1}{2 - 2^{1/(2n+1)}}.$$  \hspace{1cm} (4.4.17)

This general method allows one to construct even ordered symplectic integrators with analytic coefficients. The method was applied only for separable Hamiltonians that could be reduced into two exactly integrable parts in the article. Forest recognized that this method, based on the generality of the Ruth composition method, could be extended and used for the Ruth multi-map, symmetrized composition method to generate arbitrarily high, even ordered integrators \[50\]. A review of the symplectic methods at this time was published by Yoshida in 1993 \[195\].

### 4.5 Explicit Multi-Map Symmetrized Symplectic Composition Method

The multi-map, symmetrized composition method was described in brief by Forest and Ruth \[51\]. The general requirement being the Hamiltonian can be split into a sum of $N$ individually, exactly solvable parts, such that the equations of motion for each sub Hamiltonian could be solved analytically. The symmetrized product of maps, formed from exponential Poisson operators of a symplectic Lie group related to the individual sub Hamiltonians, comprise the $N$-map integrator. The general form can be written,
\[ H = \sum_{i=1}^{N} H_i, \]
\[ M = N_n \left( \frac{T}{2} \right) N_{n-1} \left( \frac{T}{2} \right) \cdots N_2 \left( \frac{T}{2} \right) N_1 (\tau) N_2 \left( \frac{T}{2} \right) \cdots N_{n-1} \left( \frac{T}{2} \right) N_n \left( \frac{T}{2} \right), \]  \hspace{1cm} (4.5.1)

where each map is an exponential Poisson bracket operator given by,

\[ N_i = \exp \left[ \alpha \tau \left( H_i : \right) \right]. \]  \hspace{1cm} (4.5.2)

The adjoint is then,

\[ M = N_1 \left( \frac{T}{2} \right) N_2 \left( \frac{T}{2} \right) \cdots N_{n-1} \left( \frac{T}{2} \right) N_n (\tau) N_{n-1} \left( \frac{T}{2} \right) \cdots N_2 \left( \frac{T}{2} \right) N_1 \left( \frac{T}{2} \right), \]  \hspace{1cm} (4.5.3)

Notice that the center map only occurs once. This is sometimes referred to as a midpoint method. The integrator can then be constructed so the most computationally intensive part of the Hamiltonian, usually force evaluations, is the center map and computed only once per step. As indicated by Channell and Neri \[29\], the derivation is easily accomplished by induction. Consider the two-map splitting, \( H = H_1 + H_2 \). A symplectic map for the splitting is approximated to second order by,

\[ \exp \left[ \tau \left( H_1 : + : H_2 : \right) \right] = \exp \left( \frac{T}{2} : H_2 : \right) \exp \left( \tau : H_1 : \right) \exp \left( \frac{T}{2} : H_2 : \right) + O (\tau^3). \]  \hspace{1cm} (4.5.4)

This is the celebrated Leapfrog integrator. If \( H_1 \) were only in terms of the position and \( H_2 \) only in terms of the momenta, then “kicks” refer to force evaluations from the transformation of \( H_1 \) and “drifts” refer to the velocity evaluations from the \( H_2 \) solutions. It can be stated that the integration is a succession of “half drift”, “full kick”, “half drift” operations. Pictorially, this represents the leaping action of a frog, hence the namesake. A three-map splitting is then,

\[ H = H_1 + H_2 + H_3 = H_\alpha + H_3, \]  \hspace{1cm} (4.5.5)

where, \( H_\alpha \), is the two-map splitting. It can then be written similarly to the two-map operator as,
exp [τ (: Hα : + : H3 :) ] = \exp \left( \frac{\tau}{2} : H3 : \right) \exp (\tau : Hα :) \exp \left( \frac{\tau}{2} : H3 : \right) + O (\tau^3)  \\
= \exp \left( \frac{\tau}{2} : H3 : \right) \exp [\tau (: H1 : + : H2 :) ] \exp \left( \frac{\tau}{2} : H3 : \right) + O (\tau^3) . \\
(4.5.6)

The center map is just the two-map given earlier. After substitution, the three-map is then,

\[
\exp [\tau (: H1 : + : H2 : + : H3 :) ] \\
= \exp \left( \frac{\tau}{2} : H3 : \right) \exp \left( \frac{\tau}{2} : H2 : \right) \exp (\tau : H1 :) \exp \left( \frac{\tau}{2} : H3 : \right) + O (\tau^3) . \\
(4.5.7)
\]

By repeated application, the \( N \)-map is generalized as,

\[
H = H1 + H2 + \cdots + HN , \\
(4.5.8)
\]

\[
\exp [\tau (: H1 : + : H2 : + \cdots + : HN :) ] \\
= \exp \left( \frac{\tau}{2} : HN : \right) \exp \left( \frac{\tau}{2} : HN-1 : \right) \cdots \exp (\tau : H1 :) \cdots \\
\times \cdots \exp \left( \frac{\tau}{2} : HN-1 : \right) \exp \left( \frac{\tau}{2} : HN : \right) + O (\tau^3) . \\
(4.5.9)
\]

The symmetrized \( N \)-map integrator is a second order composition. However, it can be used with the Yoshida technique to generate higher order integrators [50].

Any even-ordered, \( 2N + 2 \), explicit Ruth type symplectic integrators can be developed by application of Yoshida’s formula and the symmetrized Ruth composition method. The method is applicable to general Hamiltonians that can be decomposed into integrable parts, each exactly solvable if solved independently from the other parts. This greatly increases the number of Hamiltonian systems conducive to treatment by symplectic integration. One special class of Hamiltonians that explicit symplectic integrators can be easily generated is the class of polynomial Hamiltonians. A publication by Gjaja in 1994 [55] demonstrated that any monomial Hamiltonian of the form,

\[
H = q_1^{m_1} \cdots q_N^{m_N} p_1^{n_1} \cdots p_N^{n_N} , \\
(4.5.10)
\]
is exactly integrable. Thus, any Hamiltonian of the form,

\[ H = H(q^m_1 p^n_1, \ldots, q^m_N p^n_N), \]  

(4.5.11)
can be split into \( N \) exactly solvable maps. It was recognized by Channell and Neri [29] that an explicit symplectic integrator of arbitrary even order can be constructed for any polynomial Hamiltonian, greatly extending the applicability of the Yoshida-Ruth, symmetrized composition method. This approach was used by Blanes in 2002 to successfully construct explicit symplectic integrators by splitting polynomial Hamiltonians into exactly solvable maps [16].

The two-map composition method was originally believed to be only applicable to separable Hamiltonians for determining explicit symplectic integrators, particularly since it was built on the original foundation of generating functions employed by Ruth. It was realized later, by association with Lie groups, that the method was valid for any Hamiltonian that could be reduced into two individually, exactly solvable parts [50, 191, 194]. Furthermore, the principle could be extended for any number of maps. The symmetrized composition method is general for any Lie group with elements that form exponential Poisson bracket operators. The prescription can then be used to produce Ruth-type explicit symplectic integrators to an arbitrary order, especially aided by the Yoshida technique of generating \( 2N + 2 \) order maps. The development of explicit symplectic integration algorithms reduces to a matter of determining an appropriate splitting for a given Hamiltonian that would produce a sum of exactly integrable parts that can be independently solved or approximated by a symplectic map. This is true even if a symplectic approximation can not be found for the full Hamiltonian. One only needs to decompose the Hamiltonian into analytically solvable parts from which mappings can be constructed to approximate the exact phase flow map. This follows from the property of the symplectic diffeomorphism, or commonly called symplectomorphism group. The efficacy of SIAs will be demonstrated in the next section.

### 4.6 Demonstration of Symplectic Integration Algorithms

The mapping that generates the exact solution to the equations of motion is given by the full evolution operator,

\[ :H := \{ \cdot, H \}. \]

The equations of motion and the exact flow are then,
\[ \dot{z} = : H : = \{ z, H \}_z \]
\[ z = \exp ( t : H : ) z_0 \]

Splitting the operator to correspond to a pair of exactly solvable sub Hamiltonians, the operator can be written,

\[ : H := ( A + B ), \]

where the non-commutative operators, \( A \) and \( B \), are Poisson bracket operators,

\[ A = \{ \cdot, H_1 \} =: H_1 : \]
\[ B = \{ \cdot, H_2 \} =: H_2 : \]

The exact Hamiltonian flow is then,

\[ z = \exp ( t : H : ) z_0 = \exp [ t ( A + B ) ] z_0. \]

The solutions to the equations of motion can be approximated by the Lie-Trotter formula, so that the general \( n \) order symplectic integrator is given by,

\[ z ( t ) = \left( \prod_{i=1}^{k} \exp ( t c_i A ) \cdot \exp ( t d_i B ) + O \left( t^{n+1} \right) \right) z_0. \]

This is a product of symplectic mappings.

The simplest explicit, two-map symplectic integrator is the first order composition method:

\[ z = ( \exp ( t : H_1 : ) \exp ( t : H_2 : ) ) z_0 + O \left( t^2 \right). \]

Recall from the Lie-Trotter approximation, the procession of mappings is then,
\[ p_{i+1} = p_i - t \frac{\partial H_2}{\partial q}(q_i, p_i) \]
\[ q_{i+1} = q_i + t \frac{\partial H_1}{\partial p}(q_i, p_i) \]

where the derivatives are evaluated at the appropriate coordinates in the parentheses. For an autonomous, separable Hamiltonian, \( H(q, p) = T(p) + V(q) \), the mapping is then,

\[ p_{i+1} = p_i - t \frac{\partial V}{\partial q}(q_i) \]
\[ q_{i+1} = q_i + t \frac{\partial T}{\partial p}(p_{i+1}) \]

This result is easily derived for the Simple Harmonic Oscillator. Let

\[ H_1 = \frac{q^2}{2} \quad \text{and} \quad H_2 = \frac{p^2}{2}. \]

The dynamics for each sub Hamiltonian will be solved independently of the other and the solutions combined to produce the composition method. The first map, \( N_1(t) = \exp(t : H_1:) \), corresponds to the transformation,

\[ p_{i+1} = p_i - t \frac{\partial H_1}{\partial q}(q_i) \]
\[ q_{i+1} = q_i + t \frac{\partial H_1}{\partial p}(q_i) \]

The equations of motion for this first map are,

\[ \dot{p} = - \frac{\partial H_1}{\partial q} = -q \]
\[ \dot{q} = \frac{\partial H_1}{\partial p} = 0 \]
The position is a constant of the motion in this stage, so the exact solution to the integrals is obvious and is given by,

\[ p_1 = p_0 - tq_0 \]
\[ q_1 = q_0 \]

The dynamics for the second map, \( N_2 (t) = \exp (t : H_2 :) \), are now solved. The transformation is

\[ p_{i+1} = p_i - t \frac{\partial H_2}{\partial q} (p_i) \]
\[ q_{i+1} = q_i + t \frac{\partial H_2}{\partial p} (p_i) \]

The equations of motion for the second map are then,

\[ \dot{p} = - \frac{\partial H_2}{\partial q} = 0 \]
\[ \dot{q} = \frac{\partial H_2}{\partial p} = p \]

Again, the integrals are exactly solvable with the solutions,

\[ p_2 = p_1 \]
\[ q_2 = q_1 + tp_1 \]

The composition of the two mappings for the first integration step, leaving out the constant sub steps and adjusting the subscript labels accordingly, is

\[ p_1 = p_0 - tq_0 \]
\[ q_1 = q_0 + tp_1 \]
The explicit first order, two-map composition method, symplectic integration algorithm for
the Simple Harmonic Oscillator is simply,

\[ p_{i+1} = p_i - tq_i \]
\[ q_{i+1} = q_i + tp_{i+1} \]

This agrees with the integrator given earlier and is the action of a “kick” followed by a “drift”.
It can be shown that the mapping satisfies the symplectic condition at each integration step.
Making the appropriate substitution for the position step yields,

\[ p_{i+1} = p_i - tq_i \]
\[ q_{i+1} = q_i + tp_{i+1} = (1 - t^2) q_i + tp_i \]

The transformation mapping, \( z = \exp (t : H :) z_0 \), is simply,

\[
\begin{pmatrix}
  p_{i+1} \\
  q_{i+1}
\end{pmatrix} = \begin{pmatrix}
  1 & -t \\
  t & 1 - t^2
\end{pmatrix} \begin{pmatrix}
  p_i \\
  q_i
\end{pmatrix},
\]

and the Jacobian matrix is

\[
\begin{pmatrix}
  \frac{\partial q_{i+1}}{\partial q} & \frac{\partial q_{i+1}}{\partial p}
\end{pmatrix} = \begin{pmatrix}
  1 - t^2 & t \\
  -t & 1
\end{pmatrix}.
\]

The determinant of the Jacobian matrix is clearly unity, as expected for a symplectic transforma-
tion.

The explicit second order, Ruth-type two-map symmetrized composition method, symplectic
integrator, is famously known as the Leapfrog integrator or generalized midpoint method.
The form of the symplectic integrator is given by,

\[ z = \left( \exp \left( \frac{t}{2} : H_2 : \right) \exp (t : H_1 :) \exp \left( \frac{t}{2} : H_2 : \right) \right) z_0 + O (\tau^3) . \]

The general algorithm for a Hamiltonian, \( H (q, p) = H_1 (q, p) + H_2 (q, p) \), is then,
\[
q_{n+\frac{1}{2}} = q_n + \frac{t}{2} \frac{\partial H_2}{\partial p} (q_n, p_n)
\]
\[
p_{n+1} = p_n - t \frac{\partial H_1}{\partial q} (q_{n+\frac{1}{2}}, p_n)
\]
\[
q_{n+1} = q_{n+\frac{1}{2}} + \frac{t}{2} \frac{\partial H_2}{\partial p} (q_{n+\frac{1}{2}}, p_{n+1})
\]

Applied to a separable Hamiltonian of the form, \( H = H_1(q) + H_2(p) \), the integrator would have the action of a “half-drift” followed by a “kick” and then another “half-drift”, similar to the leaping action of a frog. Referring to the SHO Hamiltonian once again where

\[
H_1 = \frac{p^2}{2} \quad \text{and} \quad H_2 = \frac{q^2}{2}.
\]

The symmetric product of mappings is,

\[
N_1 \left( \frac{1}{2} t \right) N_2 (t) N_1 \left( \frac{1}{2} t \right).
\]

The dynamics are solved for each map independently and successively, transforming the canonical variables according to,

\[
p_{n+1} = p_n - \tau \frac{\partial H_i}{\partial q}
\]
\[
q_{n+1} = q_n + \tau \frac{\partial H_i}{\partial p}
\]

The procession of mappings given by this composition for the first integration step is as follows:

\( N_1 \):

\[
p_{\frac{1}{2}} = p_0 - \frac{1}{2} t \frac{\partial H_1}{\partial q} (q_0) = p_0
\]
\[
q_{\frac{1}{2}} = q_0 + \frac{1}{2} t \frac{\partial H_1}{\partial p} (p_0) = q_0 + \frac{1}{2} tp_0
\]
\[ N_2 : \]
\[
p_2 = p_2 - t \frac{\partial H_2}{\partial q} \left( q_2 \right) = p_2 - t q_2
\]
\[
q_2 = q_2 + t \frac{\partial H_2}{\partial p} \left( p_2 \right) = q_2
\]

\[ N_1 : \]
\[
p_2 = p_2 - \frac{1}{2} t \frac{\partial H_1}{\partial q} \left( q_2 \right) = p_2
\]
\[
q_2 = q_2 + \frac{1}{2} t \frac{\partial H_1}{\partial p} \left( p_2 \right) = q_2 + \frac{1}{2} t p_2
\]

Substituting as necessary to represent the transformations from the initial values, the set of sub steps can be written,

\[ N_1 : \]
\[
p_2 = p_0
\]
\[
q_2 = q_0 + \frac{1}{2} t p_0
\]

\[ N_2 : \]
\[
p_2 = \left( 1 - \frac{1}{2} t^2 \right) p_0 - t q_0
\]
\[
q_2 = q_0 + \frac{1}{2} t p_0
\]

\[ N_1 : \]
\[
p_2 = \left( 1 - \frac{1}{2} t^2 \right) p_0 - t q_0
\]
\[
q_2 = \left( 1 - \frac{1}{2} t^2 \right) q_0 + \left( t - \frac{1}{4} t^3 \right) p_0
\]

It is easy to determine that each map satisfies the symplectic condition by evaluating the Poisson bracket,

\[
\{ q', p' \}_{q_0, p_0} = \frac{\partial q'}{\partial q_0} \frac{\partial p'}{\partial p_0} - \frac{\partial p'}{\partial q_0} \frac{\partial q'}{\partial p_0}
\]

55
The results should be unity, as shown by,

\[
\begin{align*}
\{q_1^2, p_1^2\}_{q_0, p_0} &= (1) (1) - (0) = 1 \\
\{q_1^2, p_1^2\}_{q_0, p_0} &= (1) \left( 1 - \frac{1}{2} t^2 \right) - (-t) \left( \frac{1}{2} t \right) = 1 \\
\{q_2, p_2\}_{q_0, p_0} &= \left( 1 - \frac{1}{2} t^2 \right) \left( 1 - \frac{1}{2} t^2 \right) - (-t) \left( t - \frac{1}{4} t^3 \right) = 1
\end{align*}
\]

The transformation for each integration step is canonical and therefore the method is symplectic. The composition reduces to the following algorithm,

\[
\begin{align*}
q_{n+\frac{1}{2}} &= q_n + \frac{t}{2}p_n \\
p_{n+1} &= p_n - tq_{n+\frac{1}{2}} \\
q_{n+1} &= q_{n+\frac{1}{2}} + \frac{t}{2}p_{n+1}
\end{align*}
\]

In this particular scheme, the force evaluation, or “kick”, is performed once per integration step. The Leapfrog integrator is popular in molecular dynamics for its long-term energy stability and speed, although only the energy on average is preserved.

It was shown earlier that the second order SIA, or Leapfrog integrator, could be promoted using the Yoshida method to fourth order. The fourth order SIA integrator is written,

\[
z = \left( \prod_{i=1}^{4} \exp \left( tc_i A \right) \exp \left( td_i B \right) \right) z_0.
\]

The procession of mappings for \( i = 1, \ldots, 4 \) is then,

\[
\begin{align*}
p_{i+1} &= p_i - tc_i \frac{\partial H_2}{\partial q} (q_i, p_i) \\
q_{i+1} &= q_i + td_i \frac{\partial H_1}{\partial p} (q_i, p_{i+1})
\end{align*}
\]
This leads to seven substeps \((d_4 = 0)\) per integration step. The values for the \((c_i, d_i)\) constants were previously given and are found in the literature.

The second and fourth order symplectic integrators can be compared with the popular ODE integrator, the explicit fourth order, single-step Runge-Kutta method. The fourth order Runge-Kutta algorithm, RK4, can be found in many numerical and differential equations books and is given by,

\[
\begin{align*}
  k_1 &= hf(x_n, y_n) \\
  k_2 &= hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1) \\
  k_3 &= hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2) \\
  k_4 &= hf(x_n + h, y_n + k_3)
\end{align*}
\]

(4.6.1)

The derivative function, \(f(x, y) = y'\), is evaluated at the points in the parentheses and \(h\) is a constant step size. The variables are then incremented as follows,

\[
\begin{align*}
  x &= x_n + h \\
  y_{n+1} &= y_n + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4)
\end{align*}
\]

(4.6.2)

Computer programs have been written for these integration algorithms and are provided in the appendix. The integrators were applied to several simple Hamiltonian systems as demonstration of the efficacy of the symplectic integrators in preserving the phase space geometry.

The simplest Hamiltonian system tested was the Simple Harmonic Oscillator, described earlier as,

\[
H = \frac{p^2}{2} + \frac{q^2}{2},
\]

with the equations of motion being \(\dot{q} = p\) and \(\dot{p} = -q\). The exact solution is easily determined. The equations of motion are autonomous linear differential equations of the form:
\[ y'_1 = y_2 \]
\[ y'_2 = -y_1, \]

where \( y_1 = q \) and \( y_2 = p \). These can be written in matrix form as,

\[
\begin{bmatrix}
  y'_1 \\
  y'_2
\end{bmatrix} = 
\begin{bmatrix}
  0 & 1 \\
  -1 & 0
\end{bmatrix}
\begin{bmatrix}
  y_1 \\
  y_2
\end{bmatrix},
\]

Recall that solutions for differential equations of this form can be obtained by solving the corresponding eigenvalue problem. In this problem, the solutions can be determined by inspection and are simply,

\[
y_1 = A \cos x + B \sin x \\
y_2 = B \cos x - A \sin x
\]

The coefficients are constant and can be resolved by boundary conditions. Starting with the initial phase state of \( q = 1, p = 0 \), the coefficients of the exact solution are then \( A = 1 \) and \( B = 0 \). The exact, analytical solutions for the initial boundary values become,

\[
q = \cos t \\
p = -\sin t
\]

Clearly, the \( q, p \) phase plane is a circle. This result can be also be recognized from the form of the Hamiltonian,

\[
q^2 + p^2 = 2H,
\]

where the radius is \( \sqrt{2H} \) or unity since the energy is one half. The period is given by,

\[
\tau = \frac{2\pi}{\omega}.
\]
Recall that for a harmonic oscillator, the frequency is,

\[ \omega = \sqrt{\frac{k}{m}}. \]

For the Simple Harmonic Oscillator the force constant, \( k \), and mass, \( m \), are both unity. The period is then, \( \tau = 2\pi \), and the solution path will trace itself indefinitely. Numerical computation should produce a similar phase plane.

The system was integrated for ten million steps using a step size of one tenth accounting for a total simulation time of one million time units, approximately one hundred and fifty-nine thousand \((1\times10^6/2\pi)\) times longer than the period. The phase plane portraits, plots of the position versus conjugate momentum coordinates, are given in Figure 4.7.1. The non-symplectic integrator, RK4, does not preserve the phase space area, as evident by the degradation in the geometry. The size of the area gradually decreases as the integration time increases. The second order symplectic integrator, SIA2, better preserves the true geometry despite having larger local truncation error. In addition, the RK4 integrator does not conserve energy, resulting in a systematic drift in the energy as seen in the plot. The total energy decreases with time for the non-symplectic integrator, suggesting a dissipative force is acting on the system. The dissipation is an undesirable artifact of the numerical integration. The symplectic integrators also do not exactly conserve the energy but the energy does not diverge, instead oscillating with bounded amplitude, so on average the energy is conserved. Refer to other literature sources for similar comparative study of the Simple Harmonic Oscillator \( ^{155} \).

An excellent example of symplectic integration superiority in preserving geometry is the generation of the Poincaré section for the celebrated Hamiltonian introduced by Henon and Heiles \( ^{71} \). The system has two degrees of freedom and was used as a model for stellar motion in a galaxy with cylindrical symmetry. It was investigated for the existence of a third integral of the motion. The system is near integrable only at low energy. At certain energies, the system is chaotic and displays invariant tori and quasi-periodic trajectories. Unlike the Simple Harmonic Oscillator, there is no exact solution to conveniently verify the accuracy of the numerical computations.

The Henon-Heiles (HH) Hamiltonian is time-independent and separable, with the form,

\[
H = \frac{1}{2} (p_1^2 + p_2^2) + \frac{1}{2} (q_1^2 + q_2^2) + q_1^2 q_2 - \frac{1}{3} q_2^3. \tag{4.6.3}
\]

The equations of motion are then,
\[ \begin{align*}
\dot{q}_1 &= p_1 \\
\dot{q}_2 &= p_2 \\
\dot{p}_1 &= -(q_1 + 2q_1q_2) \\
\dot{p}_2 &= -(q_1^2 + q_2 - q_2^2)
\end{align*} \] (4.6.4)

The coordinates are usually referred to as \((q_x, q_y)\) and \((p_x, p_y)\). Following the conditions used in the journal article by Channell and Scovel [30], the system was integrated for 1, 200, 000 steps with a step size, \(h = \frac{1}{4}\). The initial conditions were, \(q_1 = 0.12, q_2 = 0.12, p_1 = 0.12, p_2 = 0.12\) with an energy of \(E = 0.029952\). The \(p_1\) coordinate is dependent and was determined from the other coordinates and energy from the following,

\[ p_1 = \left(2(E - V) - p_2^2\right)^{\frac{1}{2}}, \] (4.6.5)

where \(E\) is the total energy and \(V\) is the potential function,

\[ V = \frac{1}{2} (q_1^2 + q_2^2) + q_1^2q_2 - \frac{1}{3}q_2^3. \] (4.6.6)

Since the system is chaotic at higher energies, the normal phase portrait would appear as a clouded or shaded region since the trajectories fill up, unpredictably, the area of the phase space. Chaotic trajectories never pass through the same point more than once and have no period, but are quasi-periodic since they randomly pass through all of the phase space within a confined region [58]. An instrument to better visualize the dynamics was invented by Poincare and is known as the Poincare section or “surface of section”. In the Henon-Heiles problem, a recognizable Poincare surface of section is the projection of the \((q_2, p_2)\) points in the \((q_1 = 0, \dot{q}_1 > 0)\) plane. Essentially, each time the \(q_1\) coordinate changes sign, signifying the crossing of the \(q_1 = 0\) axis, the points, \((q_2, p_2)\), are scaled so that all points lie in the \((q_1 = 0, \dot{q}_1 > 0)\) plane, and then plotted. Refer to the 1982 paper by Henon [70] on computing Poincare maps.

The plots Figure [47.2] illustrate the phase portrait for the surface of section. The effects of dissipation of the geometry are evident in the section generated by the RK4 integrator, whereas the symplectic integrators clearly produce trajectories that follow the well-defined outline of the sub-manifold surface sections, and is assumed to maintain geometric integrity for all time. This is a strong indicator of the geometry-preserving properties of symplectic integrators. The energy clearly diverges for the RK4 integrator but remains stable for the SIA integrators. Other symplectic treatment of the Henon-Heiles can be found in the literature [17, 30, 129, 130, 155, 165, 179].
4.7 Plots

Figure 4.7.1: Simple Harmonic Oscillator phase space plots and corresponding relative energy plots, generated by the fourth-order Runge-Kutta integrator (a, b), the second-order symplectic integrator, SIA2 or leapfrog (c, d), and fourth-order symplectic integrator, SIA4 (e, f). Starting with the initial phasepoint, (1.0, 0.0), the SHO was integrated for $1 \times 10^7$ steps with a time step of 0.10 (simulation time = $1 \times 10^6$ time units).
Figure 4.7.2: Henon Heiles (HH) $q_2$ versus $p_2$ phase plots and corresponding relative energy plots, generated by the fourth-order Runge-Kutta integrator (a, b), the second-order symplectic integrator, SIA2 or leapfrog (c, d), and the fourth-order symplectic integrator, SIA4 (e, f). Starting with an energy of 0.029952 and initial phasepoint, (0.12, 0.12, 0.12, 0.12), the HH was integrated for $1.2 \times 10^6$ steps with a time step of $1/6$ (simulation time = $2 \times 10^5$ time units).
Chapter 5

Hyperspherical Coordinates

5.1 Overview

All measurable quantities are determined with respect to some coordinate system or reference. The measured quantities depend on the reference frame and coordinates of the observer. Although one can solve a problem with any number of coordinate systems, a judicious choice would simplify the work and possibly provide greater insight to the mechanics. Such a choice would be governed by the information desired from the problem and the constraints or obstacles involved in obtaining that information. In the study of two interacting bodies in a conservative system, relative coordinates would be an obvious choice. The potential depends only upon the innate properties of the bodies and their separation distance. It would not be necessary to track the trajectory of each body, but only the reduced mass which can be treated as a single body. The kinetic energy can be separated into the motion of a reduced mass relative to the center-of-mass.. By choosing to place the frame of reference at the center-of-mass, its motion can then be ignored, thus reducing the number of coordinates necessary to ascertain useful information. This same approach is used in the study of a three-body chemical reaction. In classical mechanics, Cartesian coordinates may be used without much trouble, but become cumbersome in a quantum study due to the interchange of the atoms and corresponding wave function changes. The problem can be solved through brute force with Cartesian coordinates, but a hyperspherical coordinate system would be more suitable since it treats the system as the motion of a single body in six-dimensional space and is indiscriminate to reactant and product channels.

In quantum reactive scattering studies, particle rearrangement is computationally expensive, particularly for heterogeneous reactions where all the chemical species differ. The usual method is to match the solutions for each arrangement channel in some internal configur-
tion space to account for the change in coordinates for the particles. Hyperspherical polar coordinates eliminate the need for matching solutions across every arrangement channel. The relationship between different arrangements of the particles in hyperspherical coordinates is a simple orthogonal transformation, a kinematic rotation, about a single angle. Transforming from one arrangement to another is accomplished by adding a constant angle to one coordinate. The system is essentially treated as a reduced particle in normalized, relative coordinates. The first hyperspherical coordinates were developed by Delves. Symmetrized hyperspherical coordinates were introduced by Smith in 1959 to treat both reactant and product channels symmetrically, without bias, and independent of the possible rearrangements of the particles. The first three-dimensional quantum reactive scattering calculation using symmetrized hyperspherical coordinates was reported by Kuppermann and Hipes in 1986, using a variation of the symmetrized hyperspherical coordinates developed by Whitten and Smith. Soon after, three-dimensional quantum reactive scattering using adiabatically adjusting, principal axes hyperspherical coordinates were introduced by Pack and Parker in 1987. Relationships between some popular principal axes of inertia hyperspherical coordinates were given by Kuppermann.

5.2 Jacobi Coordinates Transformation

Without any constraints, every free body has three degrees of freedom. For an N-body system, this corresponds to 3N coordinates. Within a rigid-body framework, where the distances between each body never changes, the degrees of freedom reduce to N(N-1). Only six coordinates are then necessary to describe the configuration of a rigid body, regardless of the number of bodies. Three determine the position and three more determine the orientation. The three-body system has a total of nine coordinates. By transforming to center-of-mass coordinates, the coordinates reduce to six independent variables. In hyperspherical coordinates, these six coordinates are the hyperradius and five angles.

The mass-normalized, relative Jacobi coordinates will first be derived independently and shown to be equivalent to those presented by Smith. From these starting coordinates, the hyperspherical coordinates will be introduced.

The first step is to separate the relative motion of the system from that of the center-of-mass. The derivation will be described in brief here. For three particles in one dimensional Cartesian coordinates, the kinetic energy and center-of-mass coordinates are given by,

\[ 2T = m_1 \dot{x}_1^2 + m_2 \dot{x}_2^2 + m_3 \dot{x}_3^2, \]
and,

\[ X_{CM} = m_1x_1 + m_2x_2 + \frac{m_3x_3}{M}, \tag{5.2.2} \]

where \( M = \sum_i m_i \) is the total mass. Let the relative coordinates be written as,

\[
\begin{align*}
    r_1 &= x_2 - x_1 \\
    r_2 &= x_3 - x_2 \\
    r_3 &= x_3 - x_1
\end{align*}
\tag{5.2.3}
\]

One dimension will be used for simplicity sake, the terms for the other two axes are easily obtained by following a similar procedure to the 1-D problem. Multiplying the kinetic energy by the total mass, \( M \), and substituting in the square of the time derivatives of the relative and center-of-mass coordinates, the kinetic energy can be reduced to

\[ 2T = M\dot{X}_{CM}^2 + \frac{m_1m_2}{M}\dot{r}_1^2 + \frac{m_2m_3}{M}\dot{r}_2^2 + \frac{m_1m_3}{M}\dot{r}_3^2 \tag{5.2.4} \]

and generalized as,

\[ 2T = M\dot{X}_{CM}^2 + \frac{m_km_i}{M}\dot{r}_k^2 + \frac{m_im_j}{M}\dot{r}_i^2 + \frac{m_km_j}{M}\dot{r}_j^2 \tag{5.2.5} \]

where \((kij) = (123)\). It is clear that the labels are arbitrary and there are three possible representations corresponding to the three cyclic permutations of \((kij)\). Dropping out the center-of-mass motion and using the reduced mass,

\[ \mu^2 = \frac{m_im_jm_k}{M}, \tag{5.2.6} \]

the general form of the kinetic energy can be written as

\[ 2T = \frac{\mu^2}{m_k}\dot{r}_k^2 + \mu^2 \left( \frac{\dot{r}_k^2}{m_j} + \frac{\dot{r}_j^2}{m_i} \right). \tag{5.2.7} \]
Permuting the three terms in the above equation produces three equivalent representations. It can be recognized that these representations of the energy equation correspond to the three possible Jacobi coordinate representations. Consider a system of three particles. Let \( \vec{r} \) be the separation vector between two of the particles and \( \vec{R} \) the separation vector between the third particle and the center-of-mass of the previous two. There are then three possible representations of these Jacobi vectors for each of the three separation distances.

Allow the following definition for one such representation.

\[
\begin{align*}
\vec{r}'_x &= \vec{r}'_2 = x_3 - x_2 \\
\vec{R}'_x &= x_1 - \frac{m_2 x_2 + m_3 x_3}{m_2 + m_3}.
\end{align*}
\tag{5.2.8}
\]

The kinetic energy can be derived in terms of these Jacobi vectors. Starting with the general form of the kinetic energy given earlier, let (kji) = (132) to give,

\[
2T = \frac{\mu^2}{m_1} \dot{r}'_2^2 + \mu^2 \left(\frac{\dot{r}'_1^2}{m_3} + \frac{\dot{r}'_3^2}{m_2}\right).
\tag{5.2.9}
\]

Multiplying by \((m_2 + m_3)\) and using the square of the time derivative of the second Jacobi vector, \(\vec{R}'_x\), the following equation can be derived,

\[
2T = \mu \left[ \frac{\mu}{m_1 (1 - \frac{m_1}{M})} \dot{r}'_x^2 + m_1 \left(1 - \frac{m_1}{M}\right) \dot{R}'_x^2 \right].
\tag{5.2.10}
\]

The general form is given by,

\[
2T = \mu \left[ \frac{\mu}{m_k (1 - \frac{m_k}{M})} \dot{r}'_k^2 + m_k \left(1 - \frac{m_k}{M}\right) \dot{R}'_k^2 \right].
\tag{5.2.11}
\]

This equation can be further simplified by symbolically defining the mass coefficients as

\[
d_k^2 = \frac{m_k}{\mu} \left(1 - \frac{m_k}{M}\right).
\tag{5.2.12}
\]
These are exactly the unitless mass coefficients given by Smith [172, Equation 1, pg. 736]. The final general form of the kinetic energy in terms of relative Jacobi coordinates is,

\[ T = \frac{1}{2\mu} \left( d_k^{-2} \ddot{r}_k^2 + d_k^{-2} \ddot{R}_k^2 \right). \]  

(5.2.13)

As previously mentioned, the choice of Jacobi representation is completely arbitrary, but in order for the mathematics to be consistent, one representation and labeling scheme must be chosen and adhered to. There is no loss of generality when doing this as the coordinates have no bias to the choice of representation and indeed, any labeling scheme is valid. In this study, \( \vec{r} \) is the separation vector for the coupled particle pair and \( \vec{R} \) is the distance between the lone particle and the center-of-mass of the coupled particle pair. Assigning symbolic labels allow the equations to be written in general form. Let \( k \) represent the lone particle, and \( ij \) the coupled particle pair, numbered respectively as particles 1, 2, and 3. The configuration, \((kij) = (132)\), is chosen to be similar to the labeling used by Smith and Johnson. The relative Jacobi vector equations can now be formulated as follows:

\[ \vec{r}_k = (\vec{x}_j - \vec{x}_i), \]  

(5.2.14)

\[ \vec{R}_k = \vec{x}_k - \frac{(m_j \vec{x}_j + m_i \vec{x}_i)}{(m_j + m_i)}. \]  

(5.2.15)

The \( \vec{x} \) vectors are position vectors in Cartesian coordinates. A simple cyclic permutation of the indices will render the equations for the other two Jacobi vector pairs. Changing either the symbolic or the numeric subscripts is equivalent. The mass-normalized, relative Jacobi coordinates along with the relevant parameters are defined as

\[ \vec{r}_k = d_k^{-1} \vec{r}_k, \]  

(5.2.16)

\[ \vec{R}_k = d_k \vec{R}_k, \]  

(5.2.17)

\[ d_k = \left[ \left( \frac{m_k}{\mu} \right) \left( 1 - \frac{m_k}{M} \right) \right]^{1/2}, \]  

(5.2.18)

\[ \mu = \left( \frac{m_1 m_2 m_3}{M} \right)^{1/2}, \]  

(5.2.19)

\[ M = m_1 + m_2 + m_3, \]  

(5.2.20)

where \( \mu \) is the three body reduced mass, \( M \) is the total mass, and \( d_k \) is the unitless normalization factor. The three Jacobi coordinate systems are related by an orthogonal transformation about a constant phase angle, \( \beta_{ij} \),
\[
\begin{pmatrix}
\vec{r}_j \\
\vec{R}_j
\end{pmatrix} = \begin{pmatrix}
\cos \beta_{ij} & \sin \beta_{ij} \\
-\sin \beta_{ij} & \cos \beta_{ij}
\end{pmatrix} \begin{pmatrix}
\vec{r}_i \\
\vec{R}_i
\end{pmatrix}.
\tag{5.2.21}
\]

This is a familiar rotation operation about an angle, but it is not a rotation in ordinary space, it is a kinematic rotation \[173\]. The kinematic rotation simply prescribes the transformation between the different Jacobi coordinates. Through cyclic permutations of (kji) in the above orthogonal transformation equation, all relations between the Jacobi coordinates can be obtained. The obtuse angle \(\beta_{ij}\) has a number of properties and identities:

\[
\beta_{ij} = -\beta_{ji}
\tag{5.2.22}
\]

The mass-normalized Jacobi coordinates expanded into Cartesian components in 3-dimensional space are:

\[
\vec{r}_k = \frac{1}{d_k} \left[ (x_j - x_i) \hat{i} + (y_j - y_i) \hat{j} + (z_j - z_i) \hat{k} \right]
\]

\[
\vec{R}_k = d_k \left[ \left( x_k - \frac{m_j x_j + m_i x_i}{m_j + m_i} \right) \hat{i} + \left( y_k - \frac{m_j y_j + m_i y_i}{m_j + m_i} \right) \hat{j} + \left( z_k - \frac{m_j z_j + m_i z_i}{m_j + m_i} \right) \hat{k} \right].
\tag{5.2.24}
\]

The subscript labels will be dropped at this point, assuming the (kji) = (132) configuration. The motion of the center-of-mass may be dropped from the kinetic energy by transforming to relative coordinates. The reduction in coordinates is possible since it was chosen to place the frame of reference at the center of mass and hence its motion is no longer observed. Using the relative displacements between the particles, the kinetic energy separates into the motion of the center-of-mass and a reduced mass moving relative to it. For a three-body problem, this is easily accomplished with the mass-normalized, relative Jacobi coordinates previously derived. The resulting form of the kinetic energy in terms of the normalized, relative Jacobi coordinates is then,

\[
T = \frac{1}{2} \mu \left( \dot{\vec{r}}^2 + \dot{\vec{R}}^2 \right).
\tag{5.2.25}
\]

Essentially, this has changed the perspective from observing three particles moving freely in three-dimensional space, to just one particle of reduced mass moving in six-dimensional space.
The normalized, center-of-mass Jacobi coordinates have the advantages that the kinetic energy can be expressed in diagonal form, and the transformation between different arrangements of the particles is a trivial orthogonal transformation about one angle. As stated by Smith [173], the transformation between the laboratory frame and these coordinates is linear with constant coefficients, preserving the volume element for all rotations in ordinary space. These coordinates form the basis for the derivation of the hyperspherical polar coordinates and ultimately the Hamiltonian and equations of motion.

5.3 Rotating Coordinates

A system of particles forms a moving body that can translate and rotate in space. A set of coordinates fixed in the body moving relative to a stationary set of coordinates, termed the body-fixed and space-fixed axes respectively, are introduced. Although treating the body-fixed coordinates as an absolute frame of reference is simpler, it would not provide an accurate description of the motion. Aside from the fact the rotating frame is not an inertial frame of reference, the rotation of the body would not be reflected in the body-fixed measurements. However, starting with initial body measurements is intuitive and can be easily resolved with respect to the space-fixed coordinates.

Six independent coordinates are required to describe the configuration of a body in space. Three of these coordinates specify the origin of the body-fixed axes and three more are needed to specify their orientation. The orientation between two Cartesian coordinate systems with the same origin can be evaluated by their nine direction cosines, which leads to a general linear transformation since the direction cosines are not all linearly independent. Choosing some set of three linearly independent functions of the cosines will provide the three necessary coordinates to specify the orientation, the most popular of these being the Euler angles. The six required coordinates are now determined, but a relation between the two different sets of coordinates would be valuable. We look again to the direction cosines. By requiring an orthogonality condition, the array of direction cosines can be rewritten and a matrix of transformation, which will preserve the magnitude of the vectors between the two coordinate systems, can then be obtained. We can now evaluate any coordinates between body-fixed and space-fixed axes that share the same origin. Using a right-handed y-convention of the Euler angles, the well-known transformation equation is as follows:

\[ \vec{r}_{\text{body}} = R\vec{r}_{\text{space}}, \]  

(5.3.1)

where \( R \) is the orthogonal Euler rotation matrix,
The rotation matrix is the product of three successive rotations, given by the following simple matrices.

\[
R = R_\gamma R_\beta R_\alpha \\
= \begin{pmatrix}
-\sin \gamma \sin \alpha + \cos \beta \cos \alpha \cos \gamma & \sin \gamma \cos \alpha + \cos \beta \sin \alpha \cos \gamma & -\cos \gamma \sin \beta \\
-\cos \gamma \sin \alpha - \cos \beta \cos \alpha \sin \gamma & \cos \gamma \cos \alpha - \cos \beta \sin \alpha \sin \gamma & \sin \gamma \sin \beta \\
\sin \beta \cos \alpha & \sin \beta \sin \alpha & \cos \beta \\
\end{pmatrix}.
\]

(5.3.2)

The rotation matrix is orthogonal, so its inverse is equivalent to its transpose,

\[
R^T = R^{-1}.
\]

(5.3.4)

Refer to texts such as Goldstein [58] for a full treatise on the Euler angles and their corresponding rotation matrices.

The transformation from body to space frame is a matter of multiplying the transpose of the rotation matrix to the body coordinates. We label the body-fixed axes \(X'Y'Z'\), and the space axes \(XYZ\). Primes denote the moving body-fixed axes.

\[
\vec{r} = R^{-1}\vec{r}', \\
\begin{pmatrix}
x' \\
y' \\
z'
\end{pmatrix} = \begin{pmatrix}
-\sin \gamma \sin \alpha + \cos \beta \cos \alpha \cos \gamma & -\cos \gamma \sin \alpha - \cos \beta \cos \alpha \sin \gamma & \sin \beta \cos \alpha \\
\sin \gamma \cos \alpha + \cos \beta \sin \alpha \cos \gamma & \cos \gamma \cos \alpha - \cos \beta \sin \alpha \sin \gamma & \sin \beta \sin \alpha \\
-\cos \gamma \sin \beta & \sin \gamma \sin \beta & \cos \beta
\end{pmatrix} \begin{pmatrix}
x \\
y \\
z
\end{pmatrix}.
\]

(5.3.5)
The time rate of a change for a vector differs between body-fixed and space-fixed axes. Since the body frame is rotating, an extra rotational term is observed in the space-fixed axes. Consider a point \( P \) moving in the body, letting \( \vec{r}_o \) designate the position vector for the origin of the body-axes relative to space. By vector addition, the space-fixed position vector for the point can be given as,

\[
\vec{r} = \vec{r}_o + \vec{r}'.
\]  

(5.3.6)

Based on physical arguments, the time derivative would be,

\[
\vec{v} = \vec{v}_o + \vec{v}' + \vec{v}_{rot}.
\]  

(5.3.7)

The first term on the right side is just the translational velocity of the origin of the body-fixed axes. If the origin of the two coordinate systems is the same, it can obviously be dropped. The second term is the translational velocity of the point measured in the body-fixed coordinates. The final term is the linear or tangential velocity of the point rotating with the body-frame relative to the space-fixed reference. This term is defined as \( \vec{\omega} \times \vec{r}' \), where \( \vec{\omega} \) is the angular velocity of the point with respect to the space-fixed coordinates. The time derivative of a vector in a rotating body can be evaluated and resolved in any desired coordinate system using the operator \([58, \text{pg. 195}]:\)

\[
\left( \frac{d}{dt} \right)_{\text{space}} = \left( \frac{d}{dt} \right)_{\text{body}} + \vec{\omega}_{\text{body}} \times .
\]  

(5.3.8)

The kinetic energy in terms of the relative Jacobi coordinates can now be evaluated using

\[
\dot{\vec{r}} = \dot{\vec{r}}' + \vec{\omega} \times \vec{r}'
\]  

(5.3.9)

\[
\dot{\vec{R}} = \dot{\vec{R}}' + \vec{\omega} \times \vec{R}',
\]  

(5.3.10)

where the dot products are

\[
\dot{\vec{r}}^2 = \dot{\vec{r}}' \cdot \dot{\vec{r}}' + (\vec{\omega} \times \vec{r}') \cdot (\vec{\omega} \times \vec{r}') + 2 \dot{\vec{r}}' \cdot (\vec{\omega} \times \vec{r}')
\]  

(5.3.11)

\[
\dot{\vec{R}}^2 = \dot{\vec{R}}' \cdot \dot{\vec{R}}' + (\vec{\omega} \times \vec{R}') \cdot (\vec{\omega} \times \vec{R}') + 2 \dot{\vec{R}}' \cdot (\vec{\omega} \times \vec{R}')
\]  

(5.3.12)
The space-fixed kinetic energy for a rotating body is now

$$
\frac{2T}{\mu} = \left( \dot{\vec{r}}^2 + \dot{\vec{R}}^2 \right) + \left[ (\vec{\omega} \times \vec{r}') \cdot (\vec{\omega} \times \vec{r}') + \left( \vec{\omega} \times \vec{R}' \right) \cdot \left( \vec{\omega} \times \vec{R}' \right) \right] + 2 \left[ \dot{\vec{r}}' \cdot (\vec{\omega} \times \vec{r}') + \dot{\vec{R}}' \cdot \left( \vec{\omega} \times \vec{R}' \right) \right].
$$

(5.3.13)

The middle term should be recognized as the addition of two inertia tensors in Quadratic Form, $\vec{\omega} \cdot I \cdot \vec{\omega}$, where

$$
I = \begin{bmatrix}
I_{xx} & I_{xy} & I_{xz} \\
I_{yx} & I_{yy} & I_{yz} \\
I_{zx} & I_{zy} & I_{zz}
\end{bmatrix},
$$

(5.3.14)

and $I_{ij} = I_{ji}$.

The diagonal terms are known as the moments of inertia and the off-diagonal terms are the products of inertia. Recall the tangential velocity is $v = (\omega \times r)$, so the middle term can be rewritten as

$$
v_r \cdot (\vec{\omega} \times \vec{r}') + v_R \cdot \left( \vec{\omega} \times \vec{R}' \right).
$$

(5.3.15)

The cyclic permutation property of a triple product as given by $a \cdot (b \times c) = c \cdot (a \times b) = b \cdot (c \times a)$, or in standard notation, $[abc] = [cab] = [bca]$, can be used to simplify the middle term. Note that changing the cyclic permutation set (i.e. $[acb]$) corresponds to the change in order of the cross-products and hence a change in sign. Using the $[bca]$ permutation, the angular velocity $\omega$, may be factored out to give

$$
\omega \cdot \left[ (r' \times v_r) + (R' \times v_R) \right].
$$

(5.3.16)

By recognizing the angular momentum in the form, $L = r \times mv$, this term becomes

$$
\frac{1}{\mu} \omega \cdot (L_r + L_R)
$$

(5.3.17)

The angular momentum can also be expressed as $L = I \cdot \omega$, so the middle term becomes

72
\[
\frac{1}{\mu} \omega \cdot (I_r + I_R) \cdot \omega = \frac{1}{\mu} \omega \cdot I \cdot \omega,
\]

(5.3.18)

where

\[
I_r = \mu \begin{bmatrix}
(r'_y^2 + r'_z^2) & -r'_x r'_y & -r'_x r'_z \\
-r'_x r'_y & (r'_x^2 + r'_z^2) & -r'_y r'_z \\
-r'_x r'_z & -r'_y r'_z & (r'_x^2 + r'_y^2)
\end{bmatrix},
\]

(5.3.19)

\[
I_R = \mu \begin{bmatrix}
(R'_y^2 + R'_z^2) & -R'_x R'_y & -R'_x R'_z \\
-R'_x R'_y & (R'_x^2 + R'_z^2) & -R'_y R'_z \\
-R'_x R'_z & -R'_y R'_z & (R'_x^2 + R'_y^2)
\end{bmatrix}.
\]

(5.3.20)

A similar simplification can be done for the kinetic energy cross-term. The kinetic energy can now be elegantly expressed as

\[
T = \frac{\mu}{2} \left( \ddot{r}^2 + \ddot{R}^2 \right) + \frac{1}{2} \omega \cdot I \cdot \omega + \mu \omega \cdot \left[ (r' \times \dot{r}') + (R' \times \dot{R}') \right].
\]

(5.3.21)

The elements of the inertia tensor are easily determined by adding \(I_r\) and \(I_R\) together, yielding

\[
I_{xx} = \mu \left\{ (r'_y^2 + R'_y^2) + (r'_z^2 + R'_z^2) \right\}
\]

(5.3.22)

\[
I_{yy} = \mu \left\{ (r'_x^2 + R'_x^2) + (r'_z^2 + R'_z^2) \right\}
\]

(5.3.23)

\[
I_{zz} = \mu \left\{ (r'_x^2 + R'_x^2) + (r'_y^2 + R'_y^2) \right\}
\]

(5.3.24)

\[
I_{xy} = -\mu \left\{ r'_x r'_y + R'_x R'_y \right\}
\]

(5.3.25)

\[
I_{xz} = -\mu \left\{ r'_x r'_z + R'_x R'_z \right\}
\]

(5.3.26)

\[
I_{yz} = -\mu \left\{ r'_y r'_z + R'_y R'_z \right\}
\]

(5.3.27)

A new quantity can be defined from the inertia elements and Jacobi coordinates. Recall the general form for the moments of inertia and the inertia products in familiar Cartesian coordinates as

73
\[ I_{xx} = \sum_{i} m_i (r_i^2 - x_i^2) \quad (5.3.28) \]
\[ I_{yy} = \sum_{i} m_i (r_i^2 - y_i^2) \quad (5.3.29) \]
\[ I_{zz} = \sum_{i} m_i (r_i^2 - z_i^2) \quad (5.3.30) \]
\[ I_{xy} = \sum_{i} -m_i x_i y_i \quad (5.3.31) \]
\[ I_{xz} = \sum_{i} -m_i x_i z_i \quad (5.3.32) \]
\[ I_{yz} = \sum_{i} -m_i y_i z_i \quad (5.3.33) \]

The moments of inertia terms are the result of a difference between a radial vector magnitude and the square of one of its Cartesian components. Analogously, we can determine a new quantity from the moments of inertia in Jacobi coordinates:

\[ \rho^2 = \left( r'_{x}^2 + R'_{x}^2 \right) + \left( r'_{y}^2 + R'_{y}^2 \right) + \left( r'_{z}^2 + R'_{z}^2 \right) \]
\[ = r'^2 + R'^2, \quad (5.3.34) \]

where,

\[ \rho_x^2 = r'_x^2 + R_x^2 \]
\[ \rho_y^2 = r'_y^2 + R_y^2 \]
\[ \rho_z^2 = r'_z^2 + R_z^2 \quad (5.3.35) \]

The sum of the Jacobi vector magnitudes define a new quantity, \( \rho^2 \), that acts as the radial length for the reduced mass in the new coordinate system. The square root of this quantity is the radius of a hypersphere, or hyperradius as it is known. This important quantity is present in all hyperspherical coordinates and will be visited later in more detail.

The products of inertia may be dropped in a coordinate system in which the inertia tensor is diagonal. Such a system can always be found \[58\], pg.198 and is known as the Principal
Axes coordinates. In geometric terms, this amounts to finding an axis of rotation in which the mass of the system is evenly distributed about this axis of rotation. It is well-known that quadratic forms, such as the kinetic energy and inertia tensor, may be transformed to Principal Axes by a similarity transformation. This is an orthogonal transformation that preserves the eigenvalues and the trace so the magnitudes of the vectors as well as the observable quantities are retained. The general transformation procedure will be described. Consider the equation,

\[ Q = \vec{x}^T A \vec{x}. \] (5.3.36)

This is the general equation for a Quadratic Form, where \( \vec{x} \) is a vector in generalized coordinates and \( A \) is a real symmetric matrix with \( n \) eigenvalues and corresponding orthonormal basis of \( n \) eigenvectors. These eigenvectors form a unitary matrix \( U \) which diagonalizes \( A \) into the similar matrix, \( D \):

\[ U^T A U = D, \] (5.3.37)

so,

\[ A = U D U^T, \] (5.3.38)

since \( U^T = U^{-1} \). Substituting back into the first equation, the Quadratic Form becomes:

\[ Q = \vec{x}^T U D U^T \vec{x}. \] (5.3.39)

The only non-zero elements of \( D \) are the eigenvalues of \( A \) along the diagonal. We define a new coordinate system, \( U^T \vec{x} = \vec{y} \), in which the Quadratic Form is diagonalized and so all cross-terms drop out. The Quadratic Form in these new coordinates is

\[ Q = \vec{y}^T D \vec{y}. \] (5.3.40)

The coordinates \( \vec{y} \), are the new Principal Axes coordinates. The Principal Axes Transformation for some real symmetric matrix is then,

\[ \vec{y} = U^T \vec{x}. \] (5.3.41)
The transformation then becomes the familiar problem of solving the characteristic equation to determine the eigenvectors and a unitary matrix. The direction of the eigenvectors actually point in the direction of the principal axes, and the eigenvalues determine the magnitude of the axes. The transformation can be accomplished simply by rotating the initial coordinate system to lie parallel to the eigenvectors. By transforming to Principal Axes coordinates, the inertia tensor products are zero and the kinetic energy reduces to,

\[ T = \frac{\mu}{2} \left( \dot{\mathbf{r}}^2 + \dot{\mathbf{R}}^2 \right) + \frac{1}{2} \left( I_{xx}\omega_x^2 + I_{yy}\omega_y^2 + I_{zz}\omega_z^2 \right) + \mu \mathbf{\tilde{w}} \cdot \left[ \left( \dot{\mathbf{r}} \times \dot{\mathbf{r}} \right) + \left( \dot{\mathbf{R}} \times \dot{\mathbf{R}} \right) \right]. \]  

(5.3.42)

Principal Axes are inherently tied to the body-fixed system, but as with any body-fixed coordinates, they may be expressed with respect to the space coordinates using the relation defined earlier.

### 5.4 Symmetrized Hyperspherical Polar Coordinates

The definitions for the Jacobi vectors in principal axes coordinates were implicitly defined by Smith \[174\] as:

\[
\begin{align*}
    r'_x &= \rho \cos \Theta \cos \Phi^k \\
    r'_y &= -\rho \sin \Theta \sin \Phi^k \\
    r'_z &= 0 \\
    R'_x &= \rho \cos \Theta \sin \Phi^k \\
    R'_y &= \rho \sin \Theta \cos \Phi^k \\
    R'_z &= 0
\end{align*}
\]

These definitions satisfy the principal axes condition, meaning the inertia products are zero. The principal moments of inertia are then:

\[
\begin{align*}
    I_{xx} &= \mu \left( r'_y^2 + R'_y^2 \right) = \mu \rho^2 \sin^2 \Theta \\
    I_{yy} &= \mu \left( r'_x^2 + R'_x^2 \right) = \mu \rho^2 \cos^2 \Theta \\
    I_{zz} &= \mu \left\{ \left( r'_x^2 + R'_x^2 \right) + \left( r'_y^2 + R'_y^2 \right) \right\} = \mu \rho^2
\end{align*}
\]

(5.4.1) (5.4.2) (5.4.3) (5.4.4) (5.4.5) (5.4.6) (5.4.7)
Three new variables, \((\rho \Theta \Phi)\), are now introduced. The hyperradius, \(\rho\), hinted at earlier from the sum of the Jacobi vector magnitudes, is the radial vector of the reduced mass moving in hyperspherical coordinates, or simply, the radius of a hypersphere. The product of the hyperradius magnitude and the reduced mass, \(\mu \rho^2\), defines the moment of inertia that is perpendicular to the plane of the particles. The hyperradius is independent of the choice of Jacobi coordinate arrangements and this invariance is prevalent in all hyperspherical coordinates. The two polar angles, \(\Theta\) and \(\Phi\), which will be collectively referred to as the hyperangles, are analogous to the azimuth and latitude spherical polar coordinates in 3D space, respectively. The azimuth angle, \(\Phi\), is used to transform between the different Jacobi arrangements. Recall for the normalized, relative Jacobi coordinates, the transformation was accomplished by a simple kinematic rotation. In hyperspherical coordinates, the transformation is even more trivial and simply amounts to shifting the azimuth angle by the constant kinematic rotation angle \([174\, 80]\) using the relation,

\[
\Phi^k = \Phi^j + \beta_{kj},
\]

or,

\[
\Phi^j = \Phi^i - \beta_{ij}.
\]

The angle equations can be obtained from the orthogonal transformation relation of the three different Jacobi coordinates and the Jacobi coordinates defined in hyperspherical coordinates by Smith. These two hyperangles and their periodic ranges were defined by Smith and will be given later.

The three coordinates, \((\rho \Theta \Phi)\), are known as the internal configuration coordinates. Three angles are needed to specify the orientation of the rotating body-frame with respect to stationary space-fixed frame. The Euler angles are the usual choice and will be referred to as the external configuration coordinates. The internal coordinates together with the Euler angles, \((\alpha \beta \gamma)\), form the six hyperspherical coordinates. The hyperradius is then parameterized by the five angles. A natural separation arises in these six new coordinates. The three internal coordinates determine the configuration or geometric shape of the system and the three external coordinates determine the orientation of the system. Since the potential energy depends only upon the configuration of the system and not the orientation, the number of variables required to map the system in configuration space is only three. Configuration space is the coordinate frame where the system is treated as a single particle.

The Smith-derived Jacobi definitions have the following useful relations,
\[ r'_x = \left( \frac{\partial R'_y}{\partial \Theta} \right) = \left( \frac{\partial R'_x}{\partial \Phi} \right) \]  \hspace{1cm} (5.4.8) \\
\[ r'_y = \left( \frac{\partial R'_x}{\partial \Theta} \right) = \left( \frac{\partial R'_y}{\partial \Phi} \right) \]  \hspace{1cm} (5.4.9) \\
\[ r'_z = 0 \]  \hspace{1cm} (5.4.10) \\
\[ R'_x = - \left( \frac{\partial r'_y}{\partial \Theta} \right) = - \left( \frac{\partial r'_z}{\partial \Phi} \right) \]  \hspace{1cm} (5.4.11) \\
\[ R'_y = - \left( \frac{\partial r'_x}{\partial \Theta} \right) = - \left( \frac{\partial r'_z}{\partial \Phi} \right) \]  \hspace{1cm} (5.4.12) \\
\[ R'_z = 0 \]  \hspace{1cm} (5.4.13) \\

Using the above relations, the derivatives of the principal axes, Jacobi coordinates can be written as,

\[ \dot{r}'_x = \left( \frac{\partial r'_x}{\partial \rho} \right) \dot{\rho} + \left( \frac{\partial r'_x}{\partial \Theta} \right) \dot{\Theta} + \left( \frac{\partial r'_x}{\partial \Phi} \right) \dot{\Phi} = \frac{r'_x}{\rho} \dot{\rho} - R'_y \dot{\Theta} - R'_x \dot{\Phi} \] \hspace{1cm} (5.4.14) \\
\[ \dot{r}'_y = \left( \frac{\partial r'_y}{\partial \rho} \right) \dot{\rho} + \left( \frac{\partial r'_y}{\partial \Theta} \right) \dot{\Theta} + \left( \frac{\partial r'_y}{\partial \Phi} \right) \dot{\Phi} = \frac{r'_y}{\rho} \dot{\rho} - R'_x \dot{\Theta} - R'_y \dot{\Phi} \] \hspace{1cm} (5.4.15) \\
\[ \dot{r}'_z = 0 \] \hspace{1cm} (5.4.16) \\
\[ \dot{R}'_x = \left( \frac{\partial R'_x}{\partial \rho} \right) \dot{\rho} + \left( \frac{\partial R'_x}{\partial \Theta} \right) \dot{\Theta} + \left( \frac{\partial R'_x}{\partial \Phi} \right) \dot{\Phi} = \frac{R'_x}{\rho} \dot{\rho} + r'_z \dot{\Theta} + r'_x \dot{\Phi} \] \hspace{1cm} (5.4.17) \\
\[ \dot{R}'_y = \left( \frac{\partial R'_y}{\partial \rho} \right) \dot{\rho} + \left( \frac{\partial R'_y}{\partial \Theta} \right) \dot{\Theta} + \left( \frac{\partial R'_y}{\partial \Phi} \right) \dot{\Phi} = \frac{R'_y}{\rho} \dot{\rho} + r'_z \dot{\Theta} + r'_y \dot{\Phi} \] \hspace{1cm} (5.4.18) \\
\[ \dot{R}'_z = 0 \] \hspace{1cm} (5.4.19) \\

The expanded derivatives are then,
\[ \dot{r}'_x = \left( \frac{\dot{b}}{\rho} - \tan \Theta \dot{\Theta} - \tan \Phi \dot{\Phi} \right) r'_x \]  
\[ \dot{r}'_y = \left( \frac{\dot{b}}{\rho} + \cot \Theta \dot{\Theta} + \cot \Phi \dot{\Phi} \right) r'_y \]  
\[ \dot{r}'_z = 0 \]  
\[ \dot{R}'_x = \left( \frac{\dot{b}}{\rho} - \tan \Theta \dot{\Theta} + \cot \Phi \dot{\Phi} \right) R'_x \]  
\[ \dot{R}'_y = \left( \frac{\dot{b}}{\rho} + \cot \Theta \dot{\Theta} - \tan \Phi \dot{\Phi} \right) R'_y \]  
\[ \dot{R}'_z = 0 \]  

The cross-term in the kinetic energy can be easily simplified using these time derivatives of the principal axes, Jacobi coordinates measured relative to the body-fixed system. Since the z-components are zero, the vector product in the cross-term reduces to

\[ \mu \omega_z \left\{ (r'_y R'_x - r'_x R'_y) \hat{\Theta} + (r'_x R'_x - r'_z R'_y) \hat{\Phi} \right\} + \left\{ - (r'_y R'_x - r'_x R'_y) \hat{\Theta} + (r'_x R'_x - r'_z R'_y) \hat{\Phi} \right\} \]

\[ = \mu \omega_z \left[ 2 \hat{\Phi} (r'_y R'_x - r'_x R'_y) \right]. \]  

Substituting the derivatives and simplifying will yield,

\[ \mu \omega_z \left[ \dot{\Theta} + (r'_x R'_x - r'_z R'_y) \right] = \mu \omega_z \left[ \dot{\Phi} (4A) \right]. \]  

In vector form, the term is just,

\[ - 4\mu \omega_z \hat{\Phi} (r' \times R') = -\mu \omega_z \dot{\Phi} (4A), \]

where A is the area vector for the two Jacobi vectors.

The kinetic energy can now be rewritten in terms of the new hyperspherical coordinates. Substituting for the Jacobi coordinates in the cross-term, the kinetic energy becomes,
\[ T = \frac{\mu}{2} \left[ \dot{\rho}^2 + \rho^2 \left( \dot{\Theta}^2 + \dot{\Phi}^2 \right) \right] + \frac{1}{2} \left( I_{xx} \omega_x^2 + I_{yy} \omega_y^2 + I_{zz} \omega_z^2 \right) - 2\mu \rho^2 \sin \Theta \cos \Phi \omega_z. \] (5.4.29)

This is the principal axes, kinetic energy equation for a rotating body in three dimensions, expressed in hyperspherical coordinates. We will call the \( \rho, \Theta, \) and \( \Phi \) coordinates the Smith-Whitten configuration. Johnson re-defined the \( \Theta \) and \( \Phi \) hyperangles to make the coordinates more suitable for configuration mapping. These angle changes make it possible to treat the three internal configuration coordinates, \( (\rho \Theta \Phi) \), as spherical polar coordinates in Johnson’s configuration mapping space. Not only was this an advantage in simplification, but it also matched Kuppermann’s widespread mapping space by a rotation of just ninety degrees, so transforming between the two is trivial. The angle modifications proposed by Johnson are as follows:

\[ \theta = \frac{\pi}{2} - 2\Theta, \] (5.4.30)
\[ \phi = \frac{\pi}{2} - 2\Phi. \] (5.4.31)

We will call the \( (\rho \theta \phi) \) coordinates the Johnson configuration. Changing to this configuration is accomplished simply by substituting the above angle changes and their derivatives into the kinetic energy equation derived so far. Using trigonometric identities, the result is

\[ T = \frac{\mu}{2} \left[ \dot{\rho}^2 + \frac{\rho^2}{4} \left( \dot{\Theta}^2 + \dot{\Phi}^2 \right) \right] + \frac{1}{2} \left( I_{xx} \omega_x^2 + I_{yy} \omega_y^2 + I_{zz} \omega_z^2 \right) + \frac{1}{2} \mu \rho^2 \cos \theta \dot{\phi} \omega_z. \] (5.4.32)

This matches the kinetic energy derived by Johnson \([82] \text{ Equation. 24}\). The equation is clearly a sum of three kinematic energies. The first term describes the kinetic energy of the system shape or configuration. The three particles always form a triangle and the energy of this triangle as it changes shape throughout the course of the motion is given by the first term. The second term is the rotational kinetic energy of the system as it rotates about some axis. The third term is a centrifugal energy due to the rotation of the body and is commonly known as the Coriolis energy.

Writing the kinetic energy in Quadratic Form is concise and lends itself to matrix treatment. The kinetic energy equation is then

\[ T = \frac{\mu}{2} q^T g \dot{q}. \] (5.4.33)
The six coordinates, \( (\dot{\rho}, \dot{\theta}, \dot{\phi}, \omega_x, \omega_y, \omega_z) \), make up the components of the column vector, \( \dot{q} \).

The real, symmetric matrix \( g \), is a metric tensor. The kinetic energy can be expanded in matrix notation by partitioning the six coordinates and the metric tensor. Using the same notation as Johnson in his second paper [81], the terms are

\[
\dot{q}^T = \left( \dot{Q}^T, \varpi^T \right) \tag{5.4.34}
\]

\[
g = \begin{pmatrix} G & C \\ C^T & K \end{pmatrix} \tag{5.4.35}
\]

where the matrices are

\[
\dot{Q}^T = \begin{pmatrix} \dot{\rho}, \dot{\theta}, \dot{\phi} \end{pmatrix} \tag{5.4.36}
\]

\[
\varpi^T = \begin{pmatrix} \omega_x, \omega_y, \omega_z \end{pmatrix} \tag{5.4.37}
\]

\[
G = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \rho^2 \frac{I}{4} & 0 \\ 0 & 0 & \rho^2 \frac{I}{4} \end{pmatrix} \tag{5.4.38}
\]

\[
K = \frac{1}{\mu} \begin{pmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{pmatrix} \tag{5.4.39}
\]

\[
C = \frac{1}{2} \rho^2 \cos \theta \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.4.40}
\]

The partitioned matrix equation for the kinetic energy is now

\[
T = \frac{\mu}{2} \left( \dot{Q}^T GQ + \varpi^T K \varpi + 2 \dot{Q}^T C \varpi \right). \tag{5.4.41}
\]

We recognize the three kinetic energy terms discussed previously. It would be more useful if this expression were in terms of the Euler angles rather than the angular velocity. This can be accomplished by a transformation of the Quadratic Form to produce a new metric tensor. A simple method for such a transformation has been developed will be described. In addition, an alternate tensor formulism approach will also be presented.

The angular velocity transformed to the Euler angles with respect to space-fixed coordinates is given by the known relation,
\[ \mathbf{\omega} = R \dot{\mathbf{\Omega}}, \]  

(5.4.42)

where

\[
R = \begin{pmatrix}
-\sin \beta \cos \gamma & \sin \gamma & 0 \\
\sin \beta \sin \gamma & \cos \gamma & 0 \\
\cos \beta & 0 & 1
\end{pmatrix}
\]  

(5.4.43)

\[ \dot{\mathbf{\Omega}}^T = \left( \dot{\alpha}, \dot{\beta}, \dot{\gamma} \right) \]  

(5.4.44)

This can be used to formulate a transformation relation between the \( \dot{q} \) coordinates in the Quadratic Form and the new desired coordinates as follows:

\[
E^T = \left( \dot{Q}^T, \dot{\Omega}^T \right)
\]  

(5.4.45)

\[ A = \begin{pmatrix}
I & 0 \\
0 & R
\end{pmatrix}
\]  

(5.4.46)

\[ \dot{q} = AE \]  

(5.4.47)

Here, \( I \) is the identity matrix and \( 0 \) is the null matrix. Substituting into the Quadratic Form of the kinetic energy produces,

\[
\dot{q}^T g \dot{q} = E^T A^T gA E.
\]  

(5.4.48)

The transformed, partitioned kinetic energy in Quadratic Form is now

\[
T = \frac{\mu}{2} \left( E^T g_{\Omega} E \right),
\]  

(5.4.49)

where

\[ g_{\Omega} = A^T gA. \]  

(5.4.50)

The matrix elements for the new metric tensor, \( g_{\Omega} \), are easily determined from the partitioned kinetic energy equation. The transformed kinetic energy in terms of the Euler angles is then,
\[
T = \frac{\mu}{2} \left( \dot{Q}^T G \dot{Q} + \dot{\Omega}^T R^T K R \dot{\Omega} + 2 \dot{Q}^T C \dot{R} \dot{\Omega} \right) \\
= \frac{\mu}{2} \left( \dot{Q}^T G \dot{Q} + \dot{\Omega}^T M \dot{\Omega} + 2 \dot{Q}^T H \dot{\Omega} \right),
\]

where

\[
\begin{align*}
 g_\Omega &= \begin{pmatrix} G & H \\ H^T & M \end{pmatrix} \\
 M &= R^T K R \\
 H &= C R
\end{align*}
\]

Only the matrix \( G \) from the first metric tensor remains unchanged. The inverse of the metric tensor can be obtained with little difficulty. The equation for the inverse starts out as,

\[
g_\Omega^{-1} = A^{-1} g^{-1} \left( A^{-1} \right)^T,
\]

since \((A^T)^{-1} = (A^{-1})^T\) for any matrix. A symbolic matrix form for the inverse metric tensor is given by Johnson as [82, Equation 37]

\[
g^{-1} = \begin{pmatrix} G^{-1} + G^{-1} C U C^T G^{-1} & -G^{-1} C U \\ -U C^T G^{-1} & U \end{pmatrix},
\]

where

\[
U = \left( K - C^T G^{-1} C \right)^{-1}.
\]

After some direct matrix multiplication, the inverse for the metric tensor is,

\[
g_\Omega^{-1} = \begin{pmatrix} G^{-1} + G^{-1} C U C^T G^{-1} & -G^{-1} C U \left( R^{-1} \right)^T \\ -R^{-1} U C^T G^{-1} & R^{-1} U \left( R^{-1} \right)^T \end{pmatrix},
\]

where
\[ R^{-1} = \begin{pmatrix} -\frac{\cos \gamma}{\sin \beta} & \frac{\sin \gamma}{\sin \beta} & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ \cot \beta & -\cot \beta & 1 \end{pmatrix} \]  

(5.4.59)

The formulas described above are valid for either the Smith-Whitten or Johnson configurations.

A general method, and perhaps more straightforward, for transforming the kinetic energy from one coordinate system to another can be accomplished using metric tensors. The derivation for the three-body kinetic energy using tensor formulation is presented.

The general form of the kinetic energy in terms of the metric tensor begins as

\[ T = \frac{1}{2} \sum_{ij} g_{ij} \dot{u}_i \dot{u}_j, \]  

(5.4.60)

where

\[ g_{ij} = \sum_k \mu \frac{\partial q_k}{\partial u_i} \frac{\partial q_k}{\partial u_j}, \]  

(5.4.61)

is the definition of the metric tensor. The \( g_{ij} \) are matrix elements of the metric tensor, which in this case is a covariant tensor of the second rank. The tensors may be transformed using the following relations,

\[ \bar{g}_{mn} = \sum_{ij} g_{ij} \frac{\partial u_i}{\partial q_m} \frac{\partial u_j}{\partial q_n}, \]  

(5.4.62)

\[ g_{ij} = \sum_{mn} \bar{g}_{mn} \frac{\partial q_m}{\partial u_i} \frac{\partial q_n}{\partial u_j}. \]  

(5.4.63)

Substituting into the kinetic energy produces
Changing the indices and applying the definition of a metric tensor, the expression is then,

\[ T = \frac{1}{2} \sum_{i,j} \left( \sum_{m} \bar{g}_{mn} \frac{\partial q_m}{\partial u_i} \frac{\partial q_n}{\partial u_j} \right) \dot{u}_i \dot{u}_j \]

(5.4.64)

\[ T = \frac{1}{2} \sum_{mn} \bar{g}_{mn} \dot{q}_m \dot{q}_n. \]

(5.4.65)

The kinetic energy is now expressed in the new coordinates, \( \dot{q} \). The quantity in the parenthesis is the metric tensor for these coordinates. The transformation between the two coordinate systems, \( u \) and \( q \), is evident in this expression and it is clear that the metric tensor plays a role in coordinate transformations. The transformation of the kinetic energy is then mostly a matter of differentiating \( u \) with respect to \( q \). This requires a relation between these two coordinate systems such that, \( u_i = u_i(q_1, ..., q_n) \). For a transformation between body and space-fixed axes, such a relation is obtained by the familiar Euler rotation matrix. The two coordinate systems, \( u \) and \( q \), represent the Jacobi and hyperspherical coordinates respectively. The summation may be separated into the same partitioned form of the kinetic energy as shown by

\[ T = \frac{\mu}{2} \left[ \sum_{i,j=1}^{3} \left( \sum_{k=1}^{6} \frac{\partial u_k}{\partial q_i} \frac{\partial u_k}{\partial q_j} \right) \dot{q}_i \dot{q}_j + \sum_{i,j=4}^{6} \left( \sum_{k=1}^{6} \frac{\partial u_k}{\partial q_i} \frac{\partial u_k}{\partial q_j} \right) \dot{q}_i \dot{q}_j + 2 \sum_{i=1}^{3} \sum_{j=4}^{6} \left( \sum_{k=1}^{6} \frac{\partial u_k}{\partial q_i} \frac{\partial u_k}{\partial q_j} \right) \dot{q}_i \dot{q}_j \right]. \]

(5.4.66)

In order to evaluate the derivatives, the Jacobi coordinates need to be expressed in terms of the hyperspherical coordinates. This is done by first transforming the Jacobi coordinates from body-fixed to space-fixed coordinates and expressing the result in hyperspherical coordinates. After carrying out the derivatives, the metric tensor and ultimately the kinetic energy can be evaluated. The procedure has the advantage of transforming the kinetic energy starting from any generalized coordinates. The derivatives may be tedious to evaluate but lend themselves to symbolic computation.

The metric tensor is a valuable entity as it holds the transformation coefficients in compact form. It is also instrumental in transforming the Hamiltonian from classical mechanics to
quantum mechanics. Merely substituting in the quantum operator for momenta in a classi-
cal Hamiltonian is incorrect for coordinate systems other than Cartesian. One could start
off in Cartesian coordinates and then transform to new coordinates to obtain a quantum-
mechanically correct Hamiltonian operator, but this procedure is not always feasible, par-
ticularly in the case where the classical Hamiltonian is not Cartesian. However, Podolsky
[144] formulized a general Hamiltonian operator for conservative and non-relativistic sys-
tems in terms of the metric tensor. Once the metric tensor from the classical Hamiltonian
is known, it can be used to determine the quantum mechanical Hamiltonian regardless of
what coordinate system was chosen.

The principal-axes, space-fixed kinetic energy in hyperspherical coordinates and Johnson
configuration is now,

\[
T = \frac{\mu}{2} \left[ \rho^2 + \frac{\rho^2}{4} \left( \dot{\theta}^2 + \dot{\phi}^2 \right) + \rho^2 \dot{\gamma}^2 + A \dot{\gamma}^2 + C \dot{\alpha}^2 + \rho^2 \cos \theta \dot{\gamma} \dot{\phi} + \rho^2 \cos \theta \cos \beta \dot{\alpha} \dot{\phi} + 2 \rho^2 \cos \beta \dot{\alpha} \dot{\gamma} + 2B \dot{\gamma} \dot{\alpha} \right],
\]

where

\[
A = \frac{1}{2} \rho^2 \left( 1 + \sin \theta \cos (2\gamma) \right) \tag{5.4.68}
\]
\[
B = \frac{1}{2} \rho^2 \sin \beta \sin \theta \sin (2\gamma) \tag{5.4.69}
\]
\[
C = \frac{1}{2} \rho^2 \sin^2 \beta \left( 1 - \sin \theta \cos (2\gamma) \right) + \rho^2 \cos^2 \beta \tag{5.4.70}
\]

The equations for the momentum can be determined by evaluating the derivative of the
kinetic energy function with respect to the velocity coordinates. These will be important
quantities when integrating the equations of motion.

5.5 Hamiltonian And Equations Of Motion

The Johnson Hamiltonian reduces to a simple form when expressed in terms of the body-fixed
total angular momentum components as given by,
\[ H = \frac{1}{2\mu} \left[ P_\rho^2 + \left( \frac{4}{\rho^2} \right) L^2(\theta, \phi) \right] + \frac{1}{\mu \rho^2} \left( \frac{J'_x^2}{1 - \sin \theta} + \frac{J'_y^2}{1 + \sin \theta} + \frac{J'_z^2}{2 \sin^2 \theta} \right) \]

\[ - \frac{4 \cos \theta J'_z P_\phi}{2\mu \rho^2 \sin^2 \theta} + V(\rho, \theta, \phi), \quad (5.5.1) \]

where

\[ L^2(\theta, \phi) = P_\rho^2 + \frac{P_\phi^2}{\sin^2 \theta}, \quad (5.5.2) \]

is the square of the angular momentum in classical mechanics, expressed in spherical coordinates. Since the Hamiltonian is a function of the position and momentum coordinates only, the angular momentum components need to be resolved in terms of associated rotation coordinates, these being the Euler angles and their conjugate momenta. Following the method by Johnson [82] this is accomplished using the vector equation,

\[
\begin{pmatrix}
J'_x \\
J'_y \\
J'_z
\end{pmatrix}
= \begin{pmatrix}
-\cos \gamma & \sin \gamma \cot \beta \cos \gamma \\
\sin \gamma \sin \beta & \cos \gamma \cos \gamma - \cot \beta \sin \gamma \\
0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
P_\alpha \\
P_\beta \\
P_\gamma
\end{pmatrix},
\]

\[ (5.5.3) \]

where

\[ J'_x = -\frac{\cos \gamma}{\sin \beta} P_\alpha + \sin \gamma P_\beta + \cot \beta \cos \gamma P_\gamma \quad (5.5.4) \]

\[ J'_y = \frac{\sin \gamma}{\sin \beta} P_\alpha + \cos \gamma P_\beta - \cot \beta \sin \gamma P_\gamma \quad (5.5.5) \]

\[ J'_z = P_\gamma \quad (5.5.6) \]

Substituting these new expressions for the body-fixed angular momentum coordinates results in a twelve dimensional phase space. The Hamiltonian, \( H = H(\rho \theta \phi \alpha \beta \gamma P_\rho P_\theta P_\phi P_\alpha P_\beta P_\gamma) \), is then a function of the six hyperspherical position coordinates and their conjugate momenta. The expression can be immediately reduced to four degrees of freedom in an 8-D phase space by fixing the space-frame \( Z \) axis in the direction of the total angular momentum vector, \( J \), which is a constant of the motion. The direction and magnitude of this vector is invariant. The \( J_x \) and \( J_y \) components are then zero and the Euler angle conjugate momenta can
be easily determined in terms of the space-fixed angular momentum components. These are given by

\[
\begin{pmatrix}
P_\alpha \\
P_\beta \\
P_\gamma \\
\end{pmatrix} = \begin{pmatrix}
0 & 0 & 1 \\
-\sin \alpha & \cos \alpha & 0 \\
\cos \alpha \sin \beta & \sin \alpha \sin \beta & \cos \beta \\
\end{pmatrix} \begin{pmatrix}
J_x \\
J_y \\
J_z \\
\end{pmatrix},
\] (5.5.7)

where

- \(P_\alpha = J_z\)  \\
- \(P_\beta = 0\)  \\
- \(P_\gamma = J_z \cos \beta\)  \\

The \(J_z\) component is just the magnitude of the total angular momentum, which is the same in either body-fixed or space-fixed coordinates, and is a constant. Substituting the above momenta equations into the body-fixed angular momentum components and expressing the result in terms of the Euler angles and their conjugate momenta, yields

- \(J'_x = -\sin \beta \cos \gamma P_\alpha\)  \\
- \(J'_y = \sin \beta \sin \gamma P_\alpha\)  \\
- \(J'_z = P_\gamma\)  \\

Unlike the space-fixed \(z\)-component of the angular momentum, the corresponding body-fixed component is not constant. This is because the direction of the angular momentum vector can change in the body frame, its direction being fixed only in the space frame. As the body rotates, the angular momentum vector with respect to space evolves with time.

Notice the similarity between the \(z\)-component of the angular momentum in the two reference frames, \(J'_z = P_\gamma\) and \(J_z = P_\alpha\). Both are equivalent to the conjugate momentum of an Euler angle responsible for the rotation about a \(z\)-axis. This can be related to the order of the three Euler angle rotations, which differs between the body and space-fixed coordinates. The first and last rotation is always about a \(z\)-axis. In space-fixed coordinates, the first rotation is about the \(z\)-axis by an angle \(\alpha\) and the last rotation being a rotation about the final \(z\)-axis by an angle \(\gamma\).

After substituting the expressions for the body-fixed, total angular momentum components
into the Hamiltonian and simplifying, the final three dimensional Hamiltonian in Johnson
hyperspherical coordinates is

\[
H = \frac{1}{2\mu} \left[ P_\rho^2 + \frac{4}{\rho^2} \left( P_\theta^2 + \frac{P_\phi^2}{\sin^2 \theta} \right) \right] + \frac{P_\gamma (P_\gamma - 4P_\phi \cos \theta)}{2\mu \rho^2 \sin^2 \theta} + \frac{(P_\alpha^2 - P_\gamma^2) (1 + \sin \theta \cos (2\gamma))}{\mu \rho^2 \cos^2 \theta} + V(\rho \theta \phi). \tag{5.5.14}
\]

The angles \( \alpha \) and \( \beta \) do not appear in the expression and are therefore cyclic coordinates.
Their conjugate momenta, \( P_\alpha \) and \( P_\beta \), are constants of the motion and can be separated
from the integration of the other coordinates. The eight coupled equations of motion are then

\[
\begin{align*}
\dot{\rho} &= \frac{\partial H}{\partial P_\rho} = \frac{P_\rho}{\mu} \tag{5.5.15} \\
\dot{\theta} &= \frac{\partial H}{\partial P_\theta} = \frac{4P_\theta}{\mu \rho^2} \tag{5.5.16} \\
\dot{\phi} &= \frac{\partial H}{\partial P_\phi} = \frac{4P_\phi - 2P_\gamma \cos \theta}{\mu \rho^2 \sin^2 \theta} \tag{5.5.17} \\
\dot{\gamma} &= \frac{\partial H}{\partial P_\gamma} = \frac{P_\gamma - 2P_\phi \cos \theta}{\mu \rho^2 \sin^2 \theta} - \left\{ \frac{2P_\gamma (1 + \sin \theta \cos (2\gamma))}{\mu \rho^2 \cos^2 \theta} \right\} \tag{5.5.18} \\
\dot{P}_\rho &= -\frac{\partial H}{\partial \rho} = \frac{4}{\mu \rho^3} \left( P_\theta^2 + \frac{P_\phi^2}{\sin^2 \theta} \right) + \frac{P_\gamma (P_\gamma - 4P_\phi \cos \theta)}{\mu \rho^3 \cos^2 \theta} \\
&+ \left\{ \frac{2 (P_\alpha^2 - P_\gamma^2) (1 + \sin \theta \cos (2\gamma))}{\mu \rho^3 \cos^2 \theta} \right\} - \frac{\partial V}{\partial \rho} \tag{5.5.19} \\
\dot{P}_\theta &= -\frac{\partial H}{\partial \theta} = \frac{\cos \theta}{\mu \rho^2 \sin^3 \theta} \left( 4P_\phi^2 + P_\gamma^2 - 4P_\gamma P_\phi \cos \theta \right) - \frac{P_\gamma P_\phi}{\mu \rho^2 \sin \theta} \\
&- \left\{ \frac{2 (P_\alpha^2 - P_\gamma^2) \sin \theta}{\mu \rho^2 \cos^3 \theta} (1 + \sin \theta \cos (2\gamma)) + \frac{(P_\alpha^2 - P_\gamma^2) \cos (2\gamma)}{\mu \rho^2 \cos \theta} \right\} - \frac{\partial V}{\partial \theta} \tag{5.5.20} \\
\dot{P}_\phi &= -\frac{\partial H}{\partial \phi} = -\frac{\partial V}{\partial \phi} \tag{5.5.21} \\
\dot{P}_\gamma &= -\frac{\partial H}{\partial \gamma} = \left\{ \frac{2 (P_\alpha^2 - P_\gamma^2) \sin \theta \sin (2\gamma)}{\mu \rho^2 \cos^2 \theta} \right\} \tag{5.5.22}
\end{align*}
\]

These are the complete spatial equations of motion for a three-body system in symmetrized,
hyperspherical polar coordinates using the Johnson configuration. Reduction to the planar

89
or two-dimensional case is easily carried out by making the appropriate coordinate changes. The quantities in curly brackets on the right-hand-side of the equations of motion can be dropped for planar motion.

The time derivatives for the two cyclic coordinates, $\alpha$ and $\beta$, can be calculated separately from the equations of the motion. The time derivative for $\alpha$ is determined from the canonical equation definition. Since the Hamiltonian does not contain any conjugate momentum for $\beta$, a similar treatment is not possible. However, the time derivative for $\beta$ can be determined by taking the derivative of the space-fixed $P_\gamma$ definition with respect to time and substituting in the expression for $\dot{P}_\gamma$ determined from the equations of motion. Given the following,

$$P_\alpha = J$$  \hspace{1cm} (5.5.23)
$$P_\beta = 0$$  \hspace{1cm} (5.5.24)
$$P_\gamma = J \cos \beta$$  \hspace{1cm} (5.5.25)
$$\dot{P}_\alpha = 0$$  \hspace{1cm} (5.5.26)
$$\dot{P}_\beta = 0$$  \hspace{1cm} (5.5.27)

The velocities for the $\alpha$ and $\beta$ coordinates can be determined and are,

$$\dot{\alpha} = 2P_\alpha \frac{1 + \sin \theta \cos(2\gamma)}{\mu \rho^2 \cos^2 \theta}$$  \hspace{1cm} (5.5.28)
$$\dot{\beta} = \frac{-2 \sin \theta \sin(2\gamma)}{\mu \rho^2 \cos^2 \theta} \sin \beta P_\alpha.$$  \hspace{1cm} (5.5.29)

The entire state of a classical system is completely determined by any instantaneous position and momenta coordinates in the phase space of the system, and the evolution of that state is determined by the canonical equations of the motion:

$$\frac{\partial p_i}{\partial t} = -f(p, q)$$  \hspace{1cm} (5.5.30)
$$\frac{\partial q_i}{\partial t} = g(p, q)$$  \hspace{1cm} (5.5.31)

What is left is to determine some initial coordinates from which the integration can be carried out.
5.6 Initial Phase State Coordinates

There are twelve hyperspherical coordinates, $(\rho \theta \phi \alpha \beta \gamma P_\rho P_\theta P_\phi P_\alpha P_\beta P_\gamma)$, for a three body system. The domain of all possible values of these coordinates constitutes the phase space continuum of the system. The total number of possible trajectories over this phase space volume is a constant. The state of any system can be described by the instantaneous set of generalized position and velocity coordinates. Similarly, the state of a three-particle system can be described by the twelve hyperspherical coordinates along with their velocities. The dynamics or evolution of the state is determined by integrating Hamilton’s Equations of Motion. The problem is then one of integration from some initial state. The initial state can be chosen arbitrarily, as any instantaneous set of position and momenta coordinates are sufficient to describe the state at some discrete time. Hence, it is an initial-value differential equations problem. The twelve initial coordinates and velocities must then be determined in order to begin the integration of the equations of motion.

The external orientation coordinates, the Euler angles, $(\alpha \beta \gamma)$, rotate the moving body frame, $X'Y'Z'$, denoted by primes, to the stationary space-fixed frame, $XYZ$. The external orientation angles can be determined from the principal axes of inertia of the particles. Recall the Principal Axes Transformation described earlier for Quadratic Forms, such as the kinetic energy and moment of inertia tensor. It is an orthogonal transformation that rotates any system to a principal axes system. The relation between the original coordinates and the principal axes coordinates can be determined from the direction cosines, or more concisely, from the Euler angles, as done for the body and space frame transformation. One can determine a unitary matrix that diagonalizes the moment of inertia tensor and equate it to the Euler rotation matrix to determine the Euler angles. In two dimensions, the process is trivial, since the transformation to principal axes is a simple rotation of one angle about the z-axis. The rotation matrix is given by,

$$ R = \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} $$

Equating any element of the unitary matrix to the corresponding element of the rotation matrix will yield an equation that can be solved for the rotation angle. The angle of rotation can also be determined from an eigenvector of the two-dimensional inertia tensor, since the eigenvectors point in the direction of the principal axes. The angle of the eigenvector is therefore the angle between the principal axes and the original axes. It can be determined using polar coordinates. Let,

$$ \vec{v}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} $$

91
be a normalized eigenvector of the 2-D inertia tensor. Then,

$$x_i^2 + y_i^2 = r^2 = 1$$

$$x_i = |\vec{r}| \cos \alpha$$

$$y_i = |\vec{r}| \sin \alpha$$

so,

$$\alpha = \tan^{-1}\left(\frac{y_i}{x_i}\right).$$

(5.6.1)

Another common method would be to perform the similarity transformation symbolically, and enforce the condition of diagonalization, where all off-diagonal elements are zero, to solve for the rotation angle. Once the angle has been determined, multiplying the rotation matrix with the original coordinates will produce the principal axes coordinates,

$$R \begin{pmatrix} x_1 & x_2 & \cdots \\ y_1 & y_2 & \cdots \end{pmatrix} = \begin{pmatrix} x_{1p} & x_{2p} & \cdots \\ y_{1p} & y_{2p} & \cdots \end{pmatrix}.$$  

(5.6.2)

In three dimensions, similar approaches can determine the orientation angles. Generally, one can diagonalize the inertia tensor and equate the unitary matrix to the Euler rotation matrix. The unitary matrix is composed of the normalized eigenvectors of the inertia tensor, determined by solving the secular determinant for the eigenvalues. Given the eigenvalue equation of the angular momentum, \( I \omega = \lambda \omega \), and dividing by the scalar of the angular velocity, \( \omega \), we obtain, \( I \dot{\hat{e}} = \dot{\lambda} \hat{e} \). The secular equation is then, \(DET(I - \lambda \hat{I}) = 0 \), where \( \hat{I} \) is the identity matrix. Solving the resultant polynomial yields three eigenvalues, each representing a moment of inertia in the new principal axes. The normalized eigenvectors then form the unitary matrix. Let \( U \) be the unitary matrix that diagonalizes the inertia tensor, \( I \), by a similarity transformation. The unitary matrix is orthogonal, its inverse being equal to its transpose. In general form, we have:

$$I_D = U^T IU$$  

(5.6.3)

$$U^T = U^{-1}$$  

(5.6.4)

$$I_D = \begin{pmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{pmatrix}.$$  

(5.6.5)
The process of determining the unitary matrix can be accomplished by any suitable linear algebra algorithm. Once the unitary matrix has been obtained, the three external orientation angles can be determined since the unitary matrix is equivalent to the Euler Rotation matrix, 

\[ R(\alpha \beta \gamma) = \begin{bmatrix} -\sin \gamma \sin \alpha + \cos \beta \cos \alpha \cos \gamma & \sin \gamma \cos \alpha + \cos \beta \sin \alpha \cos \gamma & -\cos \gamma \sin \beta \\ -\cos \gamma \sin \alpha - \cos \beta \cos \alpha \sin \gamma & \cos \gamma \cos \alpha - \cos \beta \sin \alpha \sin \gamma & \sin \gamma \sin \beta \\ \sin \beta \cos \alpha & -\sin \beta \sin \alpha & \cos \beta \end{bmatrix}. \] (5.6.6)

The angles can then be determined from the following relations,

\[ \tan \alpha = \frac{R_{32}}{R_{31}} \] (5.6.7)
\[ \cos \beta = R_{33} \] (5.6.8)
\[ \tan \gamma = -\frac{R_{23}}{R_{13}} \] (5.6.9)

Substituting the equivalent unitary matrix elements and solving the trigonometric equations, the external Euler angles are then,

\[ \alpha = \tan^{-1} \frac{U_{32}}{U_{31}} \] (5.6.10)
\[ \beta = \cos^{-1} U_{33} \] (5.6.11)
\[ \gamma = -\tan^{-1} \frac{U_{23}}{U_{13}} \] (5.6.12)

These angles provide the transformation relation between the space-fixed coordinates and body-fixed principal axes. Smith and Whitten described an alternative analytical approach to determining these angles. The full description follows.

### 5.7 Principal Axes Transformation

Starting in a center-of-mass system where the body-fixed and space-fixed coordinates are coincident, the components in both coordinate frames are identical. The Euler angles rotate
the original coordinates to principal axes. The area for a parallelogram is the magnitude of the normal vector formed from the vector cross-product of two adjacent sides. The area of a triangle is then half this magnitude. The area vector for the triangle formed from the three particles is half the cross-product of the two Jacobi vectors,

\[ \vec{A} = \frac{1}{2} \left( r'_x R'_y - r'_y R'_x \right) \hat{i} - \left( r'_x R'_z - r'_z R'_x \right) \hat{j} + \left( r'_y R'_z - r'_z R'_y \right) \hat{k} \]  

(5.7.1)

Here, the primes denote the original, non-principal axes coordinate system. The area vector, \( \vec{A} \), is taken to point in the positive z-axis of the principal axes. It can be treated in spherical polar coordinates where the angles, \( \alpha \) and \( \beta \), are equivalent to the spherical polar angles, \( \phi_A \) and \( \theta_A \), respectively. The area vector can then be described in spherical polar coordinates by the following equations,

\[ \vec{A} = A_x \hat{i} + A_y \hat{j} + A_z \hat{k} \]  

(5.7.2)

where,

\[ A_x = \frac{1}{2} \left( r'_y R'_z - r'_z R'_y \right) = |A| \sin \theta_A \cos \phi_A \]  

(5.7.3)

\[ A_y = \frac{1}{2} \left( r'_x R'_z - r'_z R'_x \right) = |A| \sin \theta_A \sin \phi_A \]  

(5.7.4)

\[ A_z = \frac{1}{2} \left( r'_y R'_z - r'_z R'_y \right) = |A| \cos \theta_A \]  

(5.7.5)

The first two Euler angles are determined from the spherical polar angles,

\[ \alpha = \phi_A = \tan^{-1} \left( \frac{A_x}{A_y} \right) \]  

(5.7.6)

\[ \beta = \theta_A = \cos^{-1} \left( \frac{A_z}{|A|} \right) \]  

(5.7.7)
The original coordinate system is then rotated by $\alpha$ and $\beta$ to an intermediate coordinate system, $\bar{X}\bar{Y}\bar{Z}$. The rotation is an orthogonal transformation in the positive, counter-clockwise direction about the $z'$-axis and $y'$-axis, respectively, as follows:

$$\bar{x} = R_{\beta}R_{\alpha}x'$$

where,

$$R_{\beta}R_{\alpha} = \begin{pmatrix} \cos \alpha \cos \beta & \sin \alpha \cos \beta & -\sin \beta \\ -\sin \alpha & \cos \alpha & 0 \\ \cos \alpha \sin \beta & \sin \alpha \sin \beta & \cos \beta \end{pmatrix}$$

$$R_{\alpha} = \begin{pmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R_{\beta} = \begin{pmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{pmatrix}$$

It is only a matter of rotating by a single angle, $\gamma$, about the new z-axis, also known as the figure axis, of the intermediate coordinate system to be parallel with the principal axes. The rotation is given by,

$$x^p = R_{\gamma}\bar{x}$$

where,

$$R_{\gamma} = \begin{pmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

A similarity transformation of the inertia tensor in the intermediate coordinate system, $\bar{X}\bar{Y}\bar{Z}$, can be performed using the above rotation matrix. Imposing a diagonalization rule such that the off-diagonal elements are zero, will provide the appropriate equations to solve for the final Euler angle, $\gamma$. Smith and Whitten introduced new relations to calculate this angle,
\[
\cos (2\gamma) = \frac{s}{Q} \tag{5.7.14}
\]
\[
\sin (2\gamma) = \frac{t}{Q} \tag{5.7.15}
\]

where,

\[
\bar{s} = \frac{(I_{yy} - I_{xx})}{\mu} = \bar{r}_x^2 + \bar{R}_y^2 - \left(\bar{r}_y^2 + \bar{R}_x^2\right) \tag{5.7.16}
\]
\[
\bar{t} = \frac{2I_{xy}}{\mu} = 2 \left(\bar{r}_x\bar{r}_y + \bar{R}_x\bar{R}_y\right) \tag{5.7.17}
\]
\[
Q = \left(s^2 + t^2\right)^{\frac{1}{2}} \tag{5.7.18}
\]

The angle can then be determined by evaluating an appropriate inverse trigonometric function \[\text{see } [80]\]. The angle is then given by,

\[
\gamma = \frac{1}{2} \tan^{-1} \left(\frac{t}{s}\right) \tag{5.7.19}
\]

The Euler angle, \(\gamma\), by definition can range over all four quadrants from 0 to \(2\pi\). There are two solutions for the angle in this range that satisfy the principal axes of inertia conditions:

\[
0 \leq \gamma_1 < \pi \\
\gamma_2 = \gamma_1 + \pi \tag{5.7.20}
\]

These angles, along with the angles, \(\alpha\) and \(\beta\), describe two principal axes systems that are rotated by \(\pi\) with respect to each other. It is arbitrarily chosen to use the first angle, \(\gamma_1\), to maintain consistency with Johnson’s choice.

The principal axes of inertia defined by Smith and Whitten rely on the conditions that the positive z-axis of the principal axes point in the same direction as the positive z-axis of the
area vector, and the moment of inertia are defined such that \( I_{yy} > I_{xx} \), where the largest moment of inertia is about the y-axis. The angle, \( \gamma \), is undefined for the symmetric top configuration, where \( I_{yy} = I_{xx} \). The nature of these conditions imposes constraints on the system that will be discussed later. We have so far determined the three external angles, \((\alpha\beta\gamma)\), that transform the system to principal axes of inertia coordinates. We must now solve for the other three coordinates.

The three internal coordinates, \((\rho\Theta\Phi)\), are easily determined from the Jacobi coordinates in principal axes of inertia coordinates as follows:

\[
\rho = \left( \vec{r}^2 + \vec{R}^2 \right)^{\frac{1}{2}} = \left( r_x^2 + r_y^2 + R_x^2 + R_y^2 \right)^{\frac{1}{2}}
\]

(5.7.21)

\[
\Theta = \tan^{-1} \frac{R_y}{r_x}
\]

(5.7.22)

\[
\Phi = \tan^{-1} \frac{R_x}{r_x}
\]

(5.7.23)

The two hyperangles are expressed in the Smith configuration and are easily modified for the Johnson configuration. The magnitude of the radial variable, \( \rho \), is the same for both configurations. We now have equations for all six of the independent hyperspherical coordinates. The corresponding velocities of the coordinates will be determined next.

A transformation relation between the velocities of the hyperspherical coordinates and the velocities of the relative Jacobi coordinates can be obtained, where \( \ddot{\vec{r}} = \ddot{\vec{r}} \left( \dot{\rho} \dot{\Theta} \dot{\Phi} \dot{\alpha} \dot{\beta} \dot{\gamma} \right) \). This relation provides the necessary equations to determine the velocities of the hyperspherical coordinates. The procedure for determining the transformation equation will be described next.

In space-fixed coordinates, the Jacobi velocity vectors are,

\[
\ddot{\vec{r}} = \dot{\vec{r}}' + \omega \times r',
\]

(5.7.24)

\[
\ddot{\vec{R}} = \dot{\vec{R}}' + \omega \times R',
\]

(5.7.25)

where the primed vectors are the principal axes Jacobi vectors implicitly defined by Smith and described earlier. The internal coordinate Jacobi velocities are easily separated out and the equations can be written in matrix form using the relations for the Jacobi velocities given previously. The rotational velocity, \( \omega \times r \), can be written in matrix form. In general, the rotational velocity vector product equation is,
\[
\omega \times r = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}.
\] (5.7.26)

Changing the sign is equivalent to interchanging two rows in the determinant form of the cross-product, so that \( r \times \omega = -\omega \times r \). Expanding this out yields,

\[
r \times \omega = \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}.
\] (5.7.27)

Using the expanded derivatives and vector products, the Jacobi velocity vector equations can be rewritten in matrix form as,

\[
\dot{\vec{r}} = \begin{pmatrix} \frac{r_y'}{\rho} & -R'_y - R'_z \\ -R'_x & \frac{r_z'}{\rho} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{\rho} \\ \dot{\Theta} \\ \dot{\Phi} \end{pmatrix} + \begin{pmatrix} 0 & -r'_z \\ -r'_x & 0 \end{pmatrix} \begin{pmatrix} r'_x \\ r'_y \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \end{pmatrix}.
\] (5.7.28)

\[
\dot{\vec{R}} = \begin{pmatrix} \frac{R'_x}{\rho} & R'_z & R'_y \\ R'_x & \frac{R'_y}{\rho} & R'_z \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{\rho} \\ \dot{\Theta} \\ \dot{\Phi} \end{pmatrix} + \begin{pmatrix} 0 & -R'_z \\ -R'_x & 0 \end{pmatrix} \begin{pmatrix} R'_x \\ R'_y \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \end{pmatrix}.
\] (5.7.29)

These two equations can be combined into one vector equation,

\[
\begin{pmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{r}_z \\ \dot{R}_x \\ \dot{R}_y \\ \dot{R}_z \end{pmatrix} = \begin{pmatrix} \frac{r'_x}{\rho} & -R'_y & -R'_z \\ \frac{r'_y}{\rho} & -R'_x & -R'_z \\ 0 & -R'_x & -R'_z \\ 0 & 0 & 0 -R'_x & R'_y \\ 0 & 0 & 0 -R'_y & R'_x \end{pmatrix} \begin{pmatrix} \dot{\rho} \\ \dot{\Theta} \\ \dot{\Phi} \end{pmatrix} + \begin{pmatrix} 0 & -r'_z \\ -r'_x & 0 \end{pmatrix} \begin{pmatrix} r'_x \\ r'_y \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \end{pmatrix}.
\] (5.7.30)

This transformation relation can be resolved in terms of the Euler angle velocities in the same manner as was done for the metric tensor. Recall that the right-hand vector was written earlier as, \( q^T = \left( \dot{Q}^T, \dot{\omega}^T \right) \). Using the same method and performing the necessary matrix multiplication, the time derivatives of the Jacobi coordinates in principal axes can be written as,
\[
\begin{pmatrix}
\dot{r}_x \\
\dot{r}_y \\
\dot{r}_z \\
\dot{\hat{R}}_x \\
\dot{\hat{R}}_y \\
\dot{\hat{R}}_z
\end{pmatrix}
= M
\begin{pmatrix}
\dot{\rho} \\
\dot{\Theta} \\
\dot{\Phi} \\
\dot{\alpha} \\
\dot{\beta} \\
\dot{\gamma}
\end{pmatrix},
\]

(5.7.31)

where,

\[
M = \begin{pmatrix}
\frac{r'}{\rho} & -R'_y & -R'_z & -r'_z \sin \beta \sin \gamma + r'_y \cos \beta & -r'_z \cos \gamma & r'_y \\
\frac{r'_y}{\rho} & -R'_z & -R'_y & -r'_z \sin \beta \cos \gamma - r'_y \cos \beta & r'_z \sin \gamma & -r'_x \\
0 & 0 & 0 & r'_y \sin \beta \cos \gamma + r'_x \sin \beta \sin \gamma & -r'_y \sin \gamma + r'_x \cos \gamma & 0 \\
\frac{R'_x}{\rho} & r'_y & r'_x & -R'_z \sin \beta \sin \gamma + R'_y \cos \beta & -R'_z \cos \gamma & R'_y \\
\frac{R'_y}{\rho} & r'_z & r'_y & -R'_z \sin \beta \cos \gamma - R'_x \cos \beta & R'_z \sin \gamma & -R'_x \\
0 & 0 & 0 & R'_y \sin \beta \cos \gamma + R'_x \sin \beta \sin \gamma & -R'_y \sin \gamma + R'_x \cos \gamma & 0
\end{pmatrix}
\]

(5.7.32)

This transformation provides the necessary equations to solve for the velocities of the hyperspherical coordinates.

All required initial hyperspherical phase space variables to carry out the integration of the equations of motion are now defined. Making the appropriate angle changes to transform to the Johnson configuration, the initial coordinates and their conjugate momenta are given by,
\[
\rho = \left( \vec{r}^2 + \vec{R}^2 \right)^{\frac{1}{2}} = \left( r_x^2 + r_y^2 + R_x^2 + R_y^2 \right)^{\frac{1}{2}} \quad (5.7.33)
\]
\[
\theta = \frac{\pi}{2} - 2 \left( \tan^{-1} \frac{R_y}{r_x} \right) \quad (5.7.34)
\]
\[
\phi = \frac{\pi}{2} - 2 \left( \tan^{-1} \frac{R_x}{r_x} \right) \quad (5.7.35)
\]
\[
\alpha = \tan^{-1} \frac{A_x}{A_y} \quad (5.7.36)
\]
\[
\beta = \cos^{-1} \left( \frac{A_x}{|A|} \right) \quad (5.7.37)
\]
\[
\gamma = \frac{1}{2} \tan^{-1} \frac{\vec{t}}{s} \quad (5.7.38)
\]
\[
P_\rho = \mu \dot{\rho} \quad (5.7.39)
\]
\[
P_\theta = \mu \rho^2 \dot{\theta} \quad (5.7.40)
\]
\[
P_\phi = \frac{1}{4} \mu \rho^2 \sin^2 \theta \dot{\phi} + \frac{1}{2} P\gamma \cos \theta = \frac{1}{4} \mu \rho^2 \sin^2 \theta \dot{\phi} + \frac{1}{2} J \cos \beta \cos \theta \quad (5.7.41)
\]
\[
P_\alpha = J \quad (5.7.42)
\]
\[
P_\beta = 0 \quad (5.7.43)
\]
\[
P_\gamma = J \cos \beta \quad (5.7.44)
\]

The momenta relations were obtained from the equations of motion. The dynamics of the system can now be studied in hyperspherical coordinates, given a suitable energy potential. This will be done by setting up computational algorithms to perform the calculations on a computer.
Chapter 6

Explicit Symplectic Integrator For 3-Body Hyperspherical Hamiltonians

6.1 Symmetrized Hyperspherical 3-Body Hamiltonian

Although it now known that symplectic integrators have general application for a wide variety of Hamiltonian systems, much recent work still focuses on separable Hamiltonians or Hamiltonians that are easily decomposed into few exactly solvable parts. Many of the symplectic integration algorithms are for separable, few-body problems for which explicit symplectic integrators are easily determined. Explicit symplectic algorithms for non-separable Hamiltonians are possible but to date, there is no general method. It is known that a composition of canonical transformation maps form a symplectic diffeomorphism group and the exact Hamiltonian itself, is canonical. Developing an explicit symplectic integrator for any kind of Hamiltonian can be accomplished by splitting the Hamiltonian into exactly solvable maps and composing a Forest-Ruth-type, multi-map symmetrized composition. The difficulty is judiciously splitting the Hamiltonian. It becomes somewhat of an art form that is specific for the kind of Hamiltonian encountered and it is not guaranteed that any general Hamiltonian can be split into exactly solvable maps.

The non-separable Hamiltonians employed in the study of three-body dynamics expressed in hyperspherical polar coordinates are of interest in chemistry for semi-classical trajectory and quantum reactive studies, but have not yet been treated symplectically. In addition to the Hamiltonian being non-separable, the degrees of freedom are often coupled in these systems. The Forest-Ruth-type, multi-map method can be used to construct an explicit integration scheme for hyperspherical Hamiltonians. An explicit second-order, multi-map symmetrized composition method, symplectic integration algorithm for a class of three-body Hamiltonians
expressed in symmetrized, hyperspherical polar coordinates is developed and presented.

The Smith-Whitten symmetrized, hyperspherical polar coordinates for three bodies [188] described earlier form the coordinate basis for a class of hyperspherical Hamiltonians. Other hyperspherical coordinates based on the Smith-Whitten coordinates were developed by Kuppermann, Schatz, and Johnson [80,82]. These other coordinates differed only in the hyperangles. The Johnson hyperspherical coordinates were selected for explicit symplectic treatment, however, the symplectic integration approach developed in this dissertation can be extended to all Hamiltonians in this family of hyperspherical coordinates. The Hamiltonian system has six degrees of freedom and the solutions must be solved numerically. The procedure to construct an explicit second-order, Forest-Ruth type multi-map symmetrized composition method, symplectic integrator for the Johnson hyperspherical Hamiltonian will be described. The hyperspherical coordinate system and Hamiltonian derivation were described previously.

6.2 Multi-Map Splitting, Planar Case

The Johnson symmetrized hyperspherical coordinates has been popular. Since the first publications in the early 1980s, there have been four hundred and thirty-nine citings [11]. The Johnson, three-dimensional hyperspherical Hamiltonian is a function of four hyperspherical coordinates and their conjugate momenta,

\[ H = H(\rho, \theta, \phi, \gamma, P_\rho, P_\theta, P_\phi, P_\gamma). \]  
(6.2.1)

The Hamiltonian equation is,

\[
H = \frac{1}{2\mu} \left[ \rho^2 + \frac{4}{\rho^2} \left( P_\theta^2 + \frac{P_\phi^2}{\sin^2 \theta} \right) \right] + \frac{P_\gamma (P_\gamma - 4 P_\phi \cos \theta)}{2\mu \rho^2 \sin^2 \theta} + \frac{(P_\alpha^2 - P_\beta^2)}{\mu \rho^2 \cos^2 \theta} (1 + \sin \theta \cos (2\gamma)) + V(\rho \theta \phi) \]  
(6.2.2)

The position coordinates, \( \alpha \) and \( \beta \), do not appear in the equation and are therefore ignorable, reducing the degrees of freedom from six to four. The third term in the kinetic energy, corresponding to the Coriolis effect of the rotating frame, can be dropped for the planar motion study resulting in the planar Hamiltonian,

\[
H = \frac{1}{2\mu} \left[ \rho^2 + \frac{4}{\rho^2} \left( P_\theta^2 + \frac{P_\phi^2}{\sin^2 \theta} \right) \right] + \frac{P_\gamma (P_\gamma - 4 P_\phi \cos \theta)}{2\mu \rho^2 \sin^2 \theta} + V(\rho \theta \phi) \]  
(6.2.3)

102
In this scenario the external angle, $\gamma$, is also ignorable so its conjugate momentum, $P_\gamma$, is a constant of the motion. The angle can then be dropped from the integration, reducing the degrees of freedom from four to three. The phase space variables are then,

$$(q_1, q_2, q_3, q_4, p_1, p_2, p_3, p_4) = (\rho, \theta, \phi, \gamma, P_\rho, P_\theta, P_\phi, P_\gamma)$$

(6.2.4)

Given a suitable potential energy function in terms of only the position variables, the Hamiltonian in this planar form can be split into exactly solvable maps so that the integrals for the equations of motion for each sub Hamiltonian can be determined analytically when solved independently of the other sub Hamiltonians. Since the full Hamiltonian is autonomous, each sub Hamiltonian derived from the splitting is also a constant of the motion. An appropriate splitting would be to separate the Hamiltonian into parts so no conjugate pairs are present in the terms. Eliminating terms with coupled conjugate variables ensures that one of the conjugate variables will be a constant of the motion for the dynamics of that particular splitting, resulting in trivial exact solutions. In general, such a splitting can be represented as,

$$H = \sum_{j=1}^{n-1} H_j (q_{r\neq s}, p_s) + H_n (q_s),$$

(6.2.5)

for,

$$s = 1, \ldots, d.$$

The $H_n$ splitting is the potential energy function and the sub Hamiltonians do not involve any conjugate pairs. Each momentum variable per degree of freedom is grouped into one splitting, separate from its conjugate position. There is then only one Hamiltonian splitting for each position evolution. The generalized transformation map for each degree of freedom per integration step is,

$$(^i)p_{k+1} = (^i)p_0 - \frac{1}{2} t \sum_{j=1}^{n-1} \left( \frac{\partial H_j}{\partial (^i)q} \left( ^{(r\neq s)}q_{j-1}, ^{(s)}p_{j-1} \right) + \frac{\partial H_j}{\partial (^i)p} \left( ^{(r\neq s)}q_{2n-j}, ^{(s)}p_{2n-j} \right) \right) - t \frac{\partial H_n}{\partial (^i)q} \left( ^{(s)}q_{n+1} \right)$$

$$(^i)q_{k+1} = (^i)q_0 + \frac{1}{2} t \left( \frac{\partial H_j}{\partial (^i)q} \left( ^{(r\neq s)}q_0, ^{(s)}p_0 \right) + \frac{\partial H_j}{\partial (^i)p} \left( ^{(r\neq s)}q_k, ^{(s)}p_{k+1} \right) \right),$$

(6.2.6)
where the subscript, \( n \), is the total number of maps corresponding to the splittings. The derivatives of the Hamiltonians are evaluated at the coordinates in the parentheses. This is a condensed representation of the integration sub-steps similar to those shown earlier for the second-order Leapfrog method. The derivatives of the final position update with respect to the initial position and momenta coordinates are,

\[
\frac{\partial^{(i)} q_{k+1}}{\partial^{(i)} q_0} = 1 + \frac{1}{2} t \frac{\partial}{\partial^{(i)} q_0} \frac{\partial H_j}{\partial^{(i)} p} \left( (r \neq s) q_k^{(s)} p_{k+1} \right) \\
\frac{\partial^{(i)} q_{k+1}}{\partial^{(i)} p_0} = \frac{1}{2} t \frac{\partial}{\partial^{(i)} p_0} \frac{\partial H_j}{\partial^{(i)} p} \left( (r \neq s) q_0^{(s)} p_0 + \frac{1}{2} \frac{\partial}{\partial^{(i)} p_0} \frac{\partial H_j}{\partial^{(i)} p} \left( (r \neq s) q_k^{(s)} p_{k+1} \right) \right). \tag{6.2.7}
\]

The equation of motion for the position update is linearly proportional in the conjugate momentum term so that,

\[
\frac{\partial H_j}{\partial^{(i)} p} = \alpha^{(i)} p, \tag{6.2.8}
\]

where \( \alpha \) is the proportionality constant. The position updates are then,

\[
\frac{\partial^{(i)} q_{k+1}}{\partial^{(i)} q_0} = 1 + \tau \frac{\partial^{(i)} p_{k+1}}{\partial^{(i)} q_0} \\
\frac{\partial^{(i)} q_{k+1}}{\partial^{(i)} p_0} = \tau + \tau \frac{\partial^{(i)} p_{k+1}}{\partial^{(i)} p_0}, \tag{6.2.9}
\]

where \( \tau \) is the consolidation of the constant terms from the partial derivatives. Substituting as appropriate to produce a one to one correspondence between the final update of the coordinates and the initial values, \( p \mapsto p_0, q \mapsto q_0 \), the Poisson bracket is evaluated as,

\[
\{^{(i)} q, ^{(i)} p \}^{q_0, p_0} = \frac{\partial^{(i)} q}{\partial^{(i)} q_0} \frac{\partial^{(i)} p}{\partial^{(i)} p_0} - \frac{\partial^{(i)} p}{\partial^{(i)} q_0} \frac{\partial^{(i)} q}{\partial^{(i)} p_0} \\
= \left( 1 + \tau \frac{\partial^{(i)} p}{\partial^{(i)} q_0} \right) \left( \frac{\partial^{(i)} p}{\partial^{(i)} p_0} \right) - \left( \frac{\partial^{(i)} p}{\partial^{(i)} q_0} \right) \left( \tau + \tau \frac{\partial^{(i)} p}{\partial^{(i)} p_0} \right), \tag{6.2.10}
\]

reducing to,
\[
\{ (i)q_i, (i)p_i \}_{(i)q_0, (i)p_0} = \frac{\partial (i)p_i}{\partial (i)q_0} - \tau \frac{\partial (i)p_i}{\partial (i)q_0},
\]
(6.2.11)

The Poisson bracket can be shown to be unity for all degrees of freedom after the symplectic update of the canonical coordinates. By application of the full differential and the property,

\[
\frac{\partial (i)q_1}{\partial (i)p_0} = \frac{\partial (i)q_2}{\partial (i)p_0} = \tau \frac{\partial (i)q_1}{\partial (i)q_0},
\]
(6.2.12)

the following relationship is deduced,

\[
\frac{\partial (i)p_i}{\partial (i)p_0} = 1 + \tau \frac{\partial (i)p_i}{\partial (i)q_0}
\]
(6.2.13)

Substituting into the Poisson bracket results in unity, thereby showing the symplectic condition is satisfied by the described multi-map composition scheme.

Following the general splitting method described, a suitable splitting for the planar Johnson hyperspherical Hamiltonian is given below,

\[
\begin{align*}
H_1 &= \frac{1}{2\mu} P_\rho^2 \\
H_2 &= \frac{2P_\theta^2}{\mu \rho^2} \\
H_3 &= \frac{4P_\phi (P_\phi - P_\gamma \cos \theta) + P_\gamma^2}{2\mu \rho^2 \sin^2 \theta} \\
H_4 &= V(\rho \theta \phi)
\end{align*}
\]
(6.2.14)

The equations of motion for the first degree of freedom in \( H_1 \) are:

\[
\begin{align*}
\dot{\rho} &= \frac{\partial H_1}{\partial P_\rho} = \frac{P_\rho}{\mu} \\
\dot{P}_\rho &= -\frac{\partial H_1}{\partial \rho} = 0.
\end{align*}
\]
(6.2.15)
The momentum is a constant of the motion for the dynamics of this sub Hamiltonian, so the integral for the position is trivial and can be solved exactly. The solutions are then,

\[ \rho = \rho_0 + t \frac{P_\rho}{\mu} \]
\[ P_\rho = P_{\rho_0} \]

All other canonical variables are constants of the motion and can be left out of the integration for this mapping.

The equations of motion and corresponding solutions for non-static variables in \( H_2 \) are:

\[ \dot{\theta} = \frac{\partial H_2}{\partial P_\theta} = \frac{4}{\mu \rho^2} P_\theta = \frac{2H_2}{P_\theta}, \]
\[ \dot{P}_\rho = -\frac{\partial H_2}{\partial \rho} = -\left[ -\frac{4P_\theta^2}{\mu \rho^3} = -\frac{2H_2}{\rho} \right], \]
\[ \theta = \theta_0 + t \frac{2H_2}{P_\theta}, \]
\[ P_\rho = P_{\rho_0} + t \frac{2H_2}{\rho} \]

Similarly for \( H_3 \):

\[ \dot{\phi} = \frac{\partial H_3}{\partial P_\phi} = \frac{1}{\mu \rho^2 \sin^2 \theta} \left( 4P_\phi - 2P_\gamma \cos \theta \right) \]
\[ \dot{P}_\rho = -\frac{\partial H}{\partial \rho} = -\left[ -\frac{2H_3}{\rho} \right], \]
\[ \dot{P}_\theta = \frac{\partial H_3}{\partial \theta} = -\left[ -\frac{2}{\sin \theta} \left( H_3 \cos \theta - \frac{P_\phi P_\gamma}{\mu \rho^2} \right) \right] \]
\[ \phi = \phi_0 + t \left[ \frac{1}{\mu \rho^2 \sin^2 \theta} \left( 4P_\phi - 2P_\gamma \cos \theta \right) \right] \]
\[ P_\rho = P_{\rho_0} - t \left[ -\frac{2H_3}{\rho} \right] \]
\[ P_\theta = P_{\theta_0} - t \left[ -\frac{2}{\sin \theta} \left( H_3 \cos \theta - \frac{P_\phi P_\gamma}{\mu \rho^2} \right) \right] \]

\[ (6.2.16) \]
The equations of motion for $H_4$ are the gradient of the potential energy function or force evaluations, since the potential depends only upon the position variables. The general form for the equations of motion and solutions for $H_4$ are:

\[
\begin{align*}
\dot{P}_\rho &= -\frac{\partial H_4}{\partial \rho} = -\frac{\partial V}{\partial \rho} \\
\dot{P}_\theta &= -\frac{\partial H_4}{\partial \theta} = -\frac{\partial V}{\partial \theta} \\
\dot{P}_\phi &= -\frac{\partial H_4}{\partial \phi} = -\frac{\partial V}{\partial \phi} \\
P_\rho &= P_{\rho_0} - t \left[ \frac{\partial H_4}{\partial \rho} \right] \\
P_\theta &= P_{\theta_0} - t \left[ \frac{\partial H_4}{\partial \theta} \right] \\
P_\phi &= P_{\phi_0} - t \left[ \frac{\partial H_4}{\partial \phi} \right]
\end{align*}
\]

\((6.2.17)\)

The second order, symmetrized multi-map composition is then,

\[
\exp (t : H :) = N_1 \left( \frac{1}{2} t \right) N_2 \left( \frac{1}{2} t \right) N_3 (t) N_3 \left( \frac{1}{2} t \right) N_2 \left( \frac{1}{2} t \right) N_1 \left( \frac{1}{2} t \right) + O (t^3),
\]

\((6.2.18)\)

where,

\[N_i = \exp (t : H_i :).\]

The Hamiltonian involves coupled degrees of freedom where the derivative evaluations for the equations of motion are updated with more than one degree of freedom. The dynamics require the simultaneous update of coupled degrees of freedom. Each mapping must solve the dynamics for all coupled degrees of freedom simultaneously. In Hamiltonians where the degrees of freedom are uncoupled, the symplectic mappings can be solved independently for each degree of freedom. The sub steps for each mapping can be given by following the procedure described earlier, taking into account the coupled degrees of freedom.

Given the following naming convention,
\[
(1) q, (2) q, (3) q, (1) p, (2) p, (3) p = (\rho, \theta, \phi, P_{\rho}, P_{\theta}, P_{\phi})
\]

the successive mappings for the first integration step are as follows,

\[ N_1 : \]
\[
(1) p_1 = (1) p_0 - \frac{1}{2} t \frac{\partial H_1}{\partial (1) q} \left( (1) p_0 \right) = (1) p_0
\]
\[
(1) q_1 = (1) q_0 + \frac{1}{2} t \frac{\partial H_1}{\partial (1) p} \left( (1) p_0 \right) = (1) q_0 + \frac{1}{2} t \left[ \frac{(1) p_0}{\mu} \right]
\]
\[
(2) p_1 = (2) p_0 - \frac{1}{2} t \frac{\partial H_1}{\partial (2) q} \left( (2) p_0 \right) = (2) p_0
\]
\[
(2) q_1 = (2) q_0 + \frac{1}{2} t \frac{\partial H_1}{\partial (2) p} \left( (2) p_0 \right) = (2) q_0
\]
\[
(3) p_1 = (3) p_0 - \frac{1}{2} t \frac{\partial H_1}{\partial (3) q} \left( (3) p_0 \right) = (3) p_0
\]
\[
(3) q_1 = (3) q_0 + \frac{1}{2} t \frac{\partial H_1}{\partial (3) p} \left( (3) p_0 \right) = (3) q_0
\]

\[ N_2 : \]
\[
(1) p_1 = (1) p_0 - \frac{1}{2} \frac{t}{\partial (1) q} \left( (1) q_1, (1) p_1 \right) = (1) p_0 - \frac{1}{2} t \left[ \frac{2H_2}{(1) q_1} \right]
\]
\[
(1) q_1 = (1) q_0 + \frac{1}{2} \frac{t}{\partial (1) p} \left( (1) q_1, (1) p_1 \right) = (1) q_0
\]
\[
(2) p_1 = (2) p_0 - \frac{1}{2} \frac{t}{\partial (2) q} \left( (2) q_1, (2) p_1 \right) = (2) p_0
\]
\[
(2) q_1 = (2) q_0 + \frac{1}{2} \frac{t}{\partial (2) p} \left( (2) q_1, (2) p_1 \right) = (2) q_0 + \frac{1}{2} t \left[ \frac{2H_2}{(2) p_1} \right]
\]
\[
(3) p_1 = (3) p_0 - \frac{1}{2} \frac{t}{\partial (3) q} \left( (3) q_1, (3) p_1 \right) = (3) p_0
\]
\[
(3) q_1 = (3) q_0 + \frac{1}{2} \frac{t}{\partial (3) p} \left( (3) q_1, (3) p_1 \right) = (3) q_0
\]
\[ N_3 : \]

\[ (1) p_3 = (1) p_1 - \frac{1}{2} t \frac{\partial H_3}{\partial (1) q} (1) q_1, (1) p_1 = (1) p_1 - \frac{1}{2} t \left[ -2H_3 \right] \]

\[ (1) q_3 = (1) q_1 + \frac{1}{2} t \frac{\partial H_3}{\partial (1) p} (1) q_1, (1) p_1 = (1) q_1 \]

\[ (2) p_3 = (2) p_1 - \frac{1}{2} t \frac{\partial H_3}{\partial (2) q} (2) q_1, (2) p_1 = (2) p_1 - \frac{1}{2} t \left[ -2H_3 \cos \left( \frac{(2) q_1}{2} - \frac{(3) p_1 (4) p_0}{\mu(1) q_1^2} \right) \right] \]

\[ (2) q_3 = (2) q_1 + \frac{1}{2} t \frac{\partial H_3}{\partial (2) p} (2) q_1, (2) q_1 = (2) q_1 \]

\[ (3) p_3 = (3) p_1 - \frac{1}{2} t \frac{\partial H_3}{\partial (3) q} (3) q_1, (3) p_1 = (3) p_1 \]

\[ (3) q_3 = (3) q_1 + \frac{1}{2} t \frac{\partial H_3}{\partial (3) p} (3) q_1, (3) q_1 = (3) q_1 + \frac{1}{2} t \left[ \frac{(4) p_1 - \frac{2}{(2) q_1^2} \cos \left( \frac{(2) q_1}{2} \right)}{\mu(1) q_1^2 \sin^2 \left( \frac{(2) q_1}{2} \right)} \right] \]

\[ N_4 : \]

\[ (1) p_5 = (1) p_3 - \frac{1}{2} t \frac{\partial H_4}{\partial (1) q} (1) q_3, (1) p_3 = (1) p_3 - \frac{1}{2} t \left[ \frac{\partial V}{\partial (1) q} \right] \]

\[ (1) q_5 = (1) q_3 + \frac{1}{2} t \frac{\partial H_4}{\partial (1) p} (1) q_3, (1) q_3 = (1) q_3 \]

\[ (2) p_5 = (2) p_3 - \frac{1}{2} t \frac{\partial H_4}{\partial (2) q} (2) q_3, (2) p_3 = (2) p_3 - \frac{1}{2} t \left[ \frac{\partial V}{\partial (2) q} \right] \]

\[ (2) q_5 = (2) q_3 + \frac{1}{2} t \frac{\partial H_4}{\partial (2) p} (2) q_3, (2) q_3 = (2) q_3 \]

\[ (3) p_5 = (3) p_3 - \frac{1}{2} t \frac{\partial H_4}{\partial (3) q} (3) q_3, (3) p_3 = (3) p_3 - \frac{1}{2} t \left[ \frac{\partial V}{\partial (3) q} \right] \]

\[ (3) q_5 = (3) q_3 + \frac{1}{2} t \frac{\partial H_4}{\partial (3) p} (3) q_3, (3) q_3 = (3) q_3 \]
N\textsubscript{3}:

\begin{align*}
(1)p_3 &= (1)p_3 - \frac{1}{2} t \frac{\partial H_3}{\partial (1)q} \left( (1)q_3^{(1)}p_3^{(1)} \right) = (1)p_3 - \frac{1}{2} t \left[ -2H_3' \right] \\
(1)q_3 &= (1)q_3 + \frac{1}{2} t \frac{\partial H_3}{\partial (1)p} \left( (1)q_3^{(1)}p_3^{(1)} \right) = (1)q_3 \\
(2)p_3 &= (2)p_3 - \frac{1}{2} t \frac{\partial H_3}{\partial (2)q} \left( (2)q_3^{(2)}p_3^{(2)} \right) \\
&= (2)p_3 - \frac{1}{2} t \left[ -\frac{2}{\sin (2)q_3} \left( H_3' \cos (2)q_3 - \frac{(3)p_3^{(4)}p_0}{\mu (1)q_3} \right) \right] \\
(2)q_3 &= (2)q_3 + \frac{1}{2} t \frac{\partial H_3}{\partial (2)p} \left( (2)q_3^{(2)}p_3^{(2)} \right) = (2)q_3 \\
(3)p_3 &= (3)p_3 - \frac{1}{2} t \frac{\partial H_3}{\partial (3)q} \left( (3)q_3^{(3)}p_3^{(3)} \right) = (3)p_3 \\
(3)q_3 &= (3)q_3 + \frac{1}{2} t \frac{\partial H_3}{\partial (3)p} \left( (3)q_3^{(3)}p_3^{(3)} \right) = (3)q_3 + \frac{1}{2} t \left[ \frac{(4)p_3^{(2)} - 2(4)p_0 \cos (2)q_3}{\mu (1)q_3^{(2)}} \right] \\
&+ \frac{1}{2} t \left[ \frac{(4)p_3^{(2)} - 2(4)p_0 \cos (2)q_3}{\mu (1)q_3^{(2)}} \right] \\
\end{align*}

N\textsubscript{2}:

\begin{align*}
(1)p_2 &= (1)p_3 - \frac{1}{2} t \frac{\partial H_2}{\partial (1)q} \left( (1)q_3^{(1)}p_3^{(1)} \right) = (1)p_3 - \frac{1}{2} t \left[ -2H_2' \right] \\
(1)q_2 &= (1)q_3 + \frac{1}{2} t \frac{\partial H_2}{\partial (1)p} \left( (1)q_3^{(1)}p_3^{(1)} \right) = (1)q_3 \\
(2)p_2 &= (2)p_3 - \frac{1}{2} t \frac{\partial H_2}{\partial (2)q} \left( (2)q_3^{(2)}p_3^{(2)} \right) = (2)p_3 \\
(2)q_2 &= (2)q_3 + \frac{1}{2} t \frac{\partial H_2}{\partial (2)p} \left( (2)q_3^{(2)}p_3^{(2)} \right) = (2)q_3 + \frac{1}{2} t \left[ 2H_2' \right] \\
(3)p_2 &= (3)p_3 - \frac{1}{2} t \frac{\partial H_2}{\partial (3)q} \left( (3)q_3^{(3)}p_3^{(3)} \right) = (3)p_3 \\
(3)q_2 &= (3)q_3 + \frac{1}{2} t \frac{\partial H_2}{\partial (3)p} \left( (3)q_3^{(3)}p_3^{(3)} \right) = (3)q_3 \\
\end{align*}
\( N_1 : \)

\[
(1) p_4 = (1) p_7^2 - \frac{1}{2} t \frac{\partial H_1}{\partial (1) q} (1) p_7^2 = (1) p_7^2 \\
(1) q_4 = (1) q_7^2 + \frac{1}{2} t \frac{\partial H_1}{\partial (1) p} (1) p_7^2 = (1) q_7^2 + \frac{1}{2} t \left[ (1) p_7^2 \right] \\
(2) p_4 = (2) p_7^2 - \frac{1}{2} t \frac{\partial H_1}{\partial (2) q} (2) p_7^2 = (2) p_7^2 \\
(2) q_4 = (2) q_7^2 + \frac{1}{2} t \frac{\partial H_1}{\partial (2) p} (2) p_7^2 = (2) q_7^2 \\
(3) p_4 = (3) p_7^2 - \frac{1}{2} t \frac{\partial H_1}{\partial (3) q} (3) p_7^2 = (3) p_7^2 \\
(3) q_4 = (3) q_7^2 + \frac{1}{2} t \frac{\partial H_1}{\partial (3) p} (3) p_7^2 = (3) q_7^2 
\]

The primes on the Hamiltonians denote the evaluation is made with updated variables from previous mappings. The total number of sub steps is forty-two, but most of the integrals are for constants of the motion. Developing a general computer program would require unnecessary evaluations of derivatives that equate to zero for this type of system.

The transformation can be shown to satisfy the symplectic condition by the generalized treatment shown earlier. The last mapping represents the total transformation from the initial values. For the first degree of freedom it can be written as,

\[
(1) p_3 = (1) p_0 - \frac{1}{2} t \left[ -\frac{2H_2}{(1) q_1^2} \right] - \frac{1}{2} t \left[ -\frac{2H_3}{(1) q_1^2} \right] - t \left[ \frac{\partial V}{\partial (1) q_1^2} \right] - \frac{1}{2} t \left[ -\frac{2H_2'}{(1) q_1^2} \right] - \frac{1}{2} t \left[ -\frac{2H_3'}{(1) q_1^2} \right] \\
(1) q_1 = (1) q_0 + \frac{1}{2} t \left[ \frac{(1)p_0}{\mu} \right] + \frac{1}{2} t \left[ \frac{(1)p_3}{\mu} \right] 
\]

where,

\[
(1) q_7^2 = (1) q_0 + \frac{1}{2} t \left[ \frac{(1)p_0}{\mu} \right].
\]

The Poisson bracket is then,
\[
\{q_1, p_3\}_{q_0, p_0} = \frac{\partial q_1}{\partial q_0} \frac{\partial p_3}{\partial q_0} - \frac{\partial p_3}{\partial q_0} \frac{\partial q_1}{\partial p_0} \\
= \left( 1 + \tau \frac{\partial p_3}{\partial q_0} \right) \left( \frac{\partial p_3}{\partial p_0} \right) - \left( \frac{\partial p_3}{\partial q_0} \right) \left( \tau + \frac{\partial p_3}{\partial p_0} \right) 
\] (6.2.20)

where,

\[
\tau = \frac{t}{2 \mu}. 
\] (6.2.21)

This reduces to,

\[
\{q_1, p_3\}_{q_0, p_0} = \frac{\partial p_3}{\partial q_0} - \tau \frac{\partial p_3}{\partial q_0} 
\] (6.2.22)

The full differential for the momentum, \((1) p_3\), given by the Chain Rule is,

\[
d(1) p_3 = \frac{\partial p_3}{\partial p_0} d(1) p_0 + \frac{\partial p_3}{\partial q_2} d(1) q_2 + \frac{\partial p_3}{\partial H_2} dH_2 \\
+ \frac{\partial p_3}{\partial H_3} dH_3 + \frac{\partial p_3}{\partial V} dV + \frac{\partial p_3}{\partial H_3'} dH_3' + \frac{\partial p_3}{\partial H_2'} dH_2' 
\] (6.2.23)

Expanding the differentials and using the property,

\[
\frac{\partial q_1}{\partial p_0} = \tau \frac{\partial q_2}{\partial q_0}, 
\] (6.2.24)

it is clear the following relationship is deduced,

\[
\frac{\partial p_3}{\partial p_0} = 1 + \tau \frac{\partial p_3}{\partial q_0}. 
\] (6.2.25)

Substituting this relationship into the Poisson bracket results in unity. Similarly for the other degrees of freedom, proving the method is symplectic.
The multi-map composition constructed for the Johnson hyperspherical Hamiltonian in the planar case has now been established as symplectic. The sub steps for each mapping can be grouped to simplify the composition, adjusting the subscripts labels for each successive update of the canonical variables as previously prescribed. The explicit, second order symplectic integration algorithm, composed of a total of fifteen sub steps per each integration step, is then given as:

\[ N_1:\]
\[
(1) q_{\frac{1}{2}} = (1) q_0 + \frac{1}{2} t \frac{\partial H_1}{\partial (1) p} (1) p_0 = (1) q_0 + \frac{1}{2} t \left[ \frac{(1) p_0}{\mu} \right]
\]

\[ N_2:\]
\[
(1) p_{\frac{1}{2}} = (1) p_0 - \frac{1}{2} t \frac{\partial H_2}{\partial (1) q} (1) q_{\frac{1}{2}}, (1) p_0 = (1) p_0 - \frac{1}{2} t \left[ \frac{-2H_2}{(1) q_{\frac{1}{2}}} \right]
\]
\[
(2) q_{\frac{1}{2}} = (2) q_0 + \frac{1}{2} t \frac{\partial H_2}{\partial (2) p} (2) q_0, (2) p_0 = (2) q_0 + \frac{1}{2} t \left[ \frac{2H_2}{(2) p_0} \right]
\]

\[ N_3:\]
\[
(1) p_1 = (1) p_{\frac{1}{2}} - \frac{1}{2} t \frac{\partial H_3}{\partial (1) q} (1) q_{\frac{1}{2}}, (1) p_{\frac{1}{2}} = (1) p_{\frac{1}{2}} - \frac{1}{2} t \left[ \frac{-2H_3}{(1) q_{\frac{1}{2}}} \right]
\]
\[
(2) p_{\frac{1}{2}} = (2) p_0 - \frac{1}{2} t \frac{\partial H_3}{\partial (2) q} (2) q_{\frac{1}{2}}, (2) p_0
\]
\[
= (2) p_0 - \frac{1}{2} t \left[ \frac{-2}{\sin((2) q_{\frac{1}{2}})} \right] \left( H_3 \cos((2) q_{\frac{1}{2}}) - \frac{(3) p_0 (4) p_0}{\mu(1) q_{\frac{1}{2}}^2} \right)
\]
\[
(3) q_{\frac{1}{2}} = (3) q_0 + \frac{1}{2} t \frac{\partial H_3}{\partial (3) p} (3) q_0, (3) p_0 = (3) q_0 + \frac{1}{2} t \left[ \frac{4(3) p_0 - 2(4) p_0 \cos((2) q_{\frac{1}{2}})}{\mu(1) q_{\frac{1}{2}}^2 \sin^2((2) q_{\frac{1}{2}})} \right]
\]

\[ N_4:\]
\[
(1) p_2 = (1) p_1 - t \frac{\partial H_4}{\partial (1) q} (1) q_{\frac{1}{2}} = (1) p_1 - t \left[ \frac{\partial V}{\partial (1) q} \right]
\]
\[
(2) p_{\frac{1}{2}} = (2) p_{\frac{1}{2}} - t \frac{\partial H_4}{\partial (2) q} (2) q_{\frac{1}{2}} = (2) p_{\frac{1}{2}} - t \left[ \frac{\partial V}{\partial (2) q} \right]
\]
\[
(3) p_1 = (3) p_0 - t \frac{\partial H_4}{\partial (3) q} (3) q_{\frac{1}{2}} = (3) p_0 - t \left[ \frac{\partial V}{\partial (3) q} \right]
\]
The algorithm can be implemented in a computer software program and applied to any three-body Hamiltonian expressed in Johnson hyperspherical polar coordinates. The symplectic nature of the integration should better preserve the qualitative, geometric properties of the system than a non-symplectic integrator.

6.3 Force Evaluations

The symplectic integration algorithm developed for the family of three-body hyperspherical Hamiltonians does not rely on any specific characteristics of the potential energy term, except that the potential energy must be a function of only the position coordinates. One is free to use any conservative three-body potential that satisfies this requirement. The force
evaluations are then a matter of calculating the gradient of the potential in hyperspherical coordinates.

The general first derivatives for a conservative potential in Johnson hyperspherical coordinates can be written as,

\[
\frac{\partial V}{\partial \eta} = \sum \frac{1}{2R_i} \left( \frac{\partial}{\partial \eta R_i^2} \right) \frac{\partial V}{\partial R_i},
\]

(6.3.1)

where \( \eta \) are the internal position variables and the \( R_i \) are the inter-particle distances in hyperspherical coordinates. The equations of motion and solutions for \( H_4 \) are:

\[
\begin{align*}
\dot{P}_\rho &= -\frac{\partial H_4}{\partial \rho} = -\frac{\partial V}{\partial \rho} = -\frac{1}{\rho} \left( R_1 \frac{\partial V_1}{\partial R_1} + R_2 \frac{\partial V_2}{\partial R_2} + R_3 \frac{\partial V_3}{\partial R_3} \right), \\
\dot{P}_\theta &= -\frac{\partial H_4}{\partial \theta} = -\frac{\partial V}{\partial \theta} = -\frac{\rho^2}{4} \cos \theta \left( \frac{d_1}{R_1} \sin \phi_1 \frac{\partial V_1}{\partial R_1} + \frac{d_2}{R_2} \sin \phi_2 \frac{\partial V_2}{\partial R_2} + \frac{d_3}{R_3} \sin \phi_3 \frac{\partial V_3}{\partial R_3} \right), \\
\dot{P}_\phi &= -\frac{\partial H_4}{\partial \phi} = -\frac{\partial V}{\partial \phi} = -\frac{\rho^2}{4} \sin \theta \left( \frac{d_1}{R_1} \cos \phi_1 \frac{\partial V_1}{\partial R_1} + \frac{d_2}{R_2} \cos \phi_2 \frac{\partial V_2}{\partial R_2} + \frac{d_3}{R_3} \cos \phi_3 \frac{\partial V_3}{\partial R_3} \right), \\
P_\rho &= P_{\rho_0} - t \left[ \frac{1}{\rho} \left( R_1 \frac{\partial V_1}{\partial R_1} + R_2 \frac{\partial V_2}{\partial R_2} + R_3 \frac{\partial V_3}{\partial R_3} \right) \right], \\
P_\theta &= P_{\theta_0} - t \left[ \frac{\rho^2}{4} \cos \theta \left( \frac{d_1}{R_1} \sin \phi_1 \frac{\partial V_1}{\partial R_1} + \frac{d_2}{R_2} \sin \phi_2 \frac{\partial V_2}{\partial R_2} + \frac{d_3}{R_3} \sin \phi_3 \frac{\partial V_3}{\partial R_3} \right) \right], \\
P_\phi &= P_{\phi_0} - t \left[ \frac{\rho^2}{4} \sin \theta \left( \frac{d_1}{R_1} \cos \phi_1 \frac{\partial V_1}{\partial R_1} + \frac{d_2}{R_2} \cos \phi_2 \frac{\partial V_2}{\partial R_2} + \frac{d_3}{R_3} \cos \phi_3 \frac{\partial V_3}{\partial R_3} \right) \right].
\end{align*}
\]

The derivative formulas assume the potential energy has an analytical form. This may not be the case with many ab initio potential energy surfaces. The derivatives for such potentials can be obtained by some form of polynomial interpolation where the lattice of potential energy points are fitted to a curve or surface. A common numerical technique is to use a cubic spline algorithm that exactly fits the data points to a third-order, piece-wise polynomial. The derivatives can be obtained from the spline algorithm \[\text{http://www.netlib.org/pppack}\].

The symplectic algorithm requires only the values of the separation distances and first derivatives and has no knowledge or preference on how those are determined.
Chapter 7

Methodology

7.1 Initial Values

The equations of motion are autonomous differential equations that can be solved numerically. Solving the equations of motion is an initial-value problem. Initial position and velocity values for the hyperspherical coordinates are needed. Within appropriate ranges for each coordinate, the initial values are mostly arbitrary. Initial conditions are often chosen to mimic empirical observation or to best produce interesting results.

Determination of the initial values for this study will be described. We will begin with an arbitrary set of initial conditions from which we can acquire initial coordinates and velocities in a Cartesian system. It can stated that the body and space frames are coincident at this stage, where the transformation to principal axes will separate the two coordinate systems. The system will first be transformed to the Center-of-Mass coordinates. Using these coordinates, the Jacobi vectors along with the inertia tensor, can be calculated. Following the method described to determine the Smith and Whitten principal axes, we will calculate the initial external orientation angles and transform to principal axes of inertia coordinates. The initial Jacobi coordinates and velocities in principal axes of inertia can then be solved and subsequently, all initial hyperspherical coordinates and velocities can be calculated, providing the required values to start the integration.

Two of the three particles will be coupled together, with their center-of-mass initially at rest. The third particle will act as the incident particle and can possess a collision velocity. The coupled particles form the diatomic molecule in this study and will be allowed to rotate and vibrate. We begin first by arbitrarily setting particles 2 and 3, the diatomic, on the x-axis in Cartesian coordinates, with the center-of-mass at the origin and particle 3 lying in the
positive quadrant. The y and z components are therefore zero. The general center-of-mass equation for the x component at the origin is,

\[ X_{CM} = \frac{1}{M} \sum_i m_i x_i = 0. \]  

(7.1.1)

This formula can be used to relate the x coordinate of the two particles as follows,

\[ m_2 x_2 + m_3 x_3 = 0 \]

\[ x_3 = -\frac{m_2}{m_3} x_2 \]  

(7.1.2)

An initial separation distance for particles 2 and 3 is needed. Let \( X = x_3 - x_2 \) be the separation variable, so that \( x_3 = x_2 + X \). Substituting into the center-of-mass relation above yields,

\[ x_2 = \left( -\frac{m_3}{m_2 + m_3} \right) X. \]  

(7.1.3)

We wish to allow any initial rotation in three dimensions for the coupled particles. This will be accomplished by using the Euler rotation matrix, \( R \), with the rotation angles provided as input parameters. We use the space-to-body fixed coordinate transformation form of the Euler rotation equation to orient the diatomic. The initial coordinates for the coupled particles are then given by the transformation,

\[
\begin{pmatrix}
  x_{body} \\
  y_{body} \\
  z_{body}
\end{pmatrix} = R_{\gamma\beta\alpha}
\begin{pmatrix}
  x_{space} \\
  y_{space} \\
  z_{space}
\end{pmatrix}.
\]  

(7.1.4)

If the initial rotation angles are zero, the transformation is an identity and the Cartesian position coordinates for particles 2 and 3 would be:

\[
\begin{align*}
  x_2 &= \left( -\frac{m_3}{m_2 + m_3} \right) X, \\
  y_2 &= 0, \\
  z_2 &= 0,
\end{align*}
\]

\[
\begin{align*}
  x_3 &= -\frac{m_2}{m_3} x_2, \\
  y_3 &= 0, \\
  z_3 &= 0.
\end{align*}
\]  

(7.1.5)
The initial Cartesian coordinates for the coupled particles are now defined with only the separation variable yet to be determined. The position coordinates of the impact particle, in this case the lone atom, are arbitrarily chosen and provided as run-time input parameters. In this study, the direction of translation for the impact particle is along the positive y-axis, with the initial position starting in the negative region. The starting y-component and initial separation distance of the coupled particles can be input parameters, a simple approach to the computer programs, but places the burden on the user. Alternatively, the initial separation distance can be calculated to resemble desired vibrational states of diatomics. This approach will be described.

At time, $t = 0$, we arbitrarily set the initial separation of the coupled particles to be equivalent to the maximum stretch of a diatomic molecule in the ground-state energy level. The maximum stretch occurs when the kinetic energy is zero, so all contributions to the total energy are potential. This occurs at the turning points for a harmonic oscillator, at either the maximum or minimum displacements. Equating the total energy, which is a conserved quantity and therefore a constant, to the potential energy, the maximum and minimum displacements can be solved. This is easily demonstrated using the harmonic oscillator model in one dimension,

$$E = V(x)$$

where,

$$E = h\nu \left( n + \frac{1}{2} \right),$$ \hfill (7.1.6)

$$V(x) = \frac{1}{2} kx^2$$ \hfill (7.1.7)

Solving the equality for $x$ results in,

$$x^2 = \frac{1}{k} 2h\nu \left( n + \frac{1}{2} \right).$$ \hfill (7.1.8)

The larger of the two roots would be the maximum stretch. The concept is the same for other analytical potential energy functions. The maximum stretch corresponds to the maximum separation distance between the two masses of the coupled particles.

A suitable method for arbitrarily defining the starting position coordinates for all three particles has now been defined. The initial velocities must be determined in order to finish
the first step in the computation. The rotational velocities of the two coupled particles are always in a plane and this fact simplifies the determination of their velocities. The linear velocity in terms of the angular velocity is known by the simple formula, \( \vec{V} = \vec{\omega} \times \vec{r} \). The diatomic rotates about one axis, therefore the planar rotation can be described by just one angular velocity component. If the z-axis is the axis of rotation, only the z-component of the angular velocity is non-zero. Recall the transformation between the angular velocity and Euler angles. In terms of the Euler angle derivatives where, \( \dot{\alpha} = \dot{\beta} = 0 \), the angular velocity in the body frame are,

\[
\begin{align*}
\omega_x &= 0, \\
\omega_y &= 0, \\
\omega_z &= \dot{\gamma}.
\end{align*}
\]

(7.1.9)

The velocities for the coupled particles are then,

\[
\begin{align*}
\dot{x}_i &= -y_i \omega_z, & (7.1.10) \\
\dot{y}_i &= x_i \omega_z, & (7.1.11) \\
\dot{z}_i &= 0. & (7.1.12)
\end{align*}
\]

where, \( \omega_z = \dot{\gamma} \). The angular velocity can be related to the angular momentum by the definition, \( J = I \cdot \omega \). Assuming planar rotation about the z-axis, the x and y components of the angular momentum would be zero so that

\[
J_z = I_{zz} \omega_z,
\]

(7.1.13)

where,

\[
I_{zz} = \sum_i m_i (r_i^2 - z_i^2).
\]

(7.1.14)

The magnitude of the angular momentum is equal to its z-component, \( J_z = |\vec{J}| \). We can determine the length of the angular momentum vector using the eigenvalue of the quantum mechanical total angular momentum operator, \( |\vec{J}| = \hbar \sqrt{j (j+1)} \), where, \( j \), is the rotational
quantum number. The angular velocity $z$-component can then be defined in terms of the angular momentum magnitude and moment of inertia about the $z$-axis as,

$$
\omega_z = \frac{J_z}{I_{zz}} = \frac{\hbar \sqrt{j (j + 1)}}{\sum_i m_i (x_i^2 + y_i^2)}.
$$  
(7.1.15)

Returning to our study where the coupled particles lie on the $x$-axis, the velocities are then,

$$
\begin{align*}
\dot{x}_2 &= 0, & \dot{x}_3 &= 0, \\
\dot{y}_2 &= x_2\omega_z, & \dot{y}_3 &= x_3\omega_z, \\
\dot{z}_2 &= 0, & \dot{z}_3 &= 0.
\end{align*}
$$  
(7.1.16)

where,

$$
\omega_z = \frac{\hbar \sqrt{j (j + 1)}}{m_2 (x_2^2 + y_2^2) + m_3 (x_3^2 + y_3^2)}.
$$  
(7.1.17)

The rotational quantum numbers can be read in as run-time parameters.

The initial velocity of the impact particle can be determined from the kinetic energies of the particles, equated to an experimental collision energy. Here, we are concerned only with the translational motion of the two colliding bodies. The total energy will be the sum of the center-of-mass kinetic energy of the coupled particles with the kinetic energy of the impact particle. Motion is allowed only along the $y$-axis and the coupled particles will initially be at rest. Let the impact particle be body, $A$, and the center-of-mass of the coupled particles be body, $B$. We then have,

$$
\begin{align*}
T_A &= \frac{1}{2} m_A \left( x_A^2 + y_A^2 + z_A^2 \right), \\
T_B &= \frac{1}{2} m_B \left( x_B^2 + y_B^2 + z_B^2 \right).
\end{align*}
$$  
(7.1.18)  
(7.1.19)

where,
\[ \begin{align*}
\dot{x}_A' &= \dot{x}_B' = 0 \\
\dot{y}_B' &= 0 \\
\dot{z}_A' &= \dot{z}_B' = 0
\end{align*} \]

The motion is only along the y-axis, so the system center-of-mass velocity is,

\[ \dot{Y}_{CM} = \frac{m_A\dot{y}_A' + m_B\dot{y}_B'}{M}, \quad \text{where} \quad M = m_A + m_B \quad (7.1.20) \]

The y-components of the bodies in center-of-mass coordinates is then,

\[ \begin{align*}
\dot{y}_A &= \dot{y}_A' - \dot{Y}_{CM} \\
\dot{y}_B &= \dot{y}_B' - \dot{Y}_{CM}
\end{align*} \quad (7.1.21) \]

(7.1.22)

Since the body, B, is at rest, the equations reduce to,

\[ \begin{align*}
\dot{Y}_{CM} &= \frac{m_A\dot{y}_A'}{M} \\
\dot{y}_A &= \left(1 - \frac{m_A}{M}\right) \frac{m_B\dot{y}_A'}{M} = \frac{m_B\dot{y}_A'}{M} \\
\dot{y}_B &= -\dot{Y}_{CM} = -\frac{m_A}{M} \dot{y}_A'
\end{align*} \quad (7.1.23) \]

(7.1.24) \]

(7.1.25)

The total kinetic energy of the system in center-of-mass coordinates is the sum of the kinetic energies of the two bodies in the y direction,

\[ T = \frac{1}{2} m_A\dot{y}_A^2 + \frac{1}{2} m_B\dot{y}_B^2. \quad (7.1.26) \]

Substituting in for the velocities determined earlier yields,

\[ T = \frac{1}{2} \frac{m_A m_B^2}{M^2} y_A^2 + \frac{1}{2} \frac{m_B m_A^2}{M^2} y_A'^2 \quad (7.1.27) \]
which reduces to,

\[ T = \frac{1}{2} \frac{m_A m_B}{M} \dot{y}_A^2. \]  

(7.1.28)

The initial velocity for the impact particle is now determined as,

\[ \dot{y}_A' = \left( \frac{2MT}{m_A m_B} \right)^{\frac{1}{2}}, \]  

(7.1.29)

where,

\[
\begin{align*}
m_A &= m_1 \\
m_B &= m_2 + m_3
\end{align*}
\]  

(7.1.30)

The energy in the velocity equation can be set to an experimental collision energy, read in as a parameter.

All initial position and velocity coordinates can now be defined, completing the first step in the dynamics study. We are interested in center-of-mass coordinates which reduce the number of coordinates in the problem by ignoring the three coordinates for the center of mass. The position and velocity coordinates are easily transformed to center-of-mass coordinates by shifting the original coordinates to be relative to the system’s center of mass. The center-of-mass is the origin in these new coordinates and all dynamics are relative to the center-of-mass. The center-of-mass equations are given by,

\[
\begin{align*}
X_{CM} &= \frac{1}{M} \sum_i m_i x_i \\
Y_{CM} &= \frac{1}{M} \sum_i m_i y_i \\
Z_{CM} &= \frac{1}{M} \sum_i m_i z_i
\end{align*}
\]  

(7.1.31)

The new center-of-mass position and velocity coordinates are then,
\[ q_i = q'_i - Q_{CM} \quad (7.1.32) \]
\[ \dot{q}_i = \dot{q}'_i - \dot{Q}_{CM} \quad (7.1.33) \]

where \( q = x, y, z \) and the primes denote the original coordinates prior to the center-of-mass transformation. The second step of the study is complete.

The entire dynamical study can be carried forth in center-of-mass Cartesian coordinates at this point if desired. The next step in our study is to transform the coordinates to body-fixed Principal Axes of Inertia coordinates. There are many methods that can be used, but we will use the method developed by Smith-Whitten described earlier.

### 7.2 Computer Programs

The symplectic integration algorithm developed in this dissertation was implemented in a computer program and compared with two popular non-symplectic ODE integrators, the fourth order Runge-Kutta algorithm and the adaptive step-size Bulirsch-Stoer integrator from Numerical Recipes in Fortran 77 [145]. The computer programs were developed for single trajectory studies and designed to be agnostic to the form of the potential and integrator. The potential energy surface interface is generic and only requires inter-particle separation distances and first derivatives. Similarly, the integration interface is easily modified for different integrators. The symplectic integrator was named, HS2D_SIA2, to signify it was a second order symplectic integration algorithm (SIA2) for planar hyperspherical (HS2D) coordinates. The fourth-order Runge-Kutta and Bulirsch-Stoer integrators were named RK4 and BS, respectively.

The following describes the driver program flow for the integration studies.

I. Set initial conditions.

II. Initialize potential energy surface.

III. Calculate initial potential energy.

IV. Print initial phase space, potential energy, hyperspherical coordinates.

V. Integrate the equations of motion; print intermediate coordinate values.

The initial conditions provided in the first step can be arbitrarily obtained but in this study
the values were determined using the methodology provided in the previous chapter. The initial values for the hyperspherical coordinates and their velocities will be determined first in Cartesian coordinates and transformed to Jacobi coordinates. The procedure for calculating the initial hyperspherical coordinates can be outlined as follows:

I. Define arbitrary, initial Cartesian coordinates.

II. Transform to Center-Of-Mass coordinates.

III. Transform to Principal Axes of Inertia coordinates.

IV. Transform to Hyperspherical Coordinates.

The units are in atomic units and angles are computed in radians. The values for physical quantities in atomic units were obtained from the National Institute of Standards and Technology (NIST) Reference for Constants, Units, and Uncertainty [http://physics.nist.gov] based on the 2002 Committee on Data for Science and Technology (CODATA) recommended values for international use [133]. The following table summarizes the conversion of atomic units with the International System (SI) units used in the computer programs.

<table>
<thead>
<tr>
<th>Physical Quantity</th>
<th>Atomic Unit</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy</td>
<td>$E_h$ (Hartree)</td>
<td>$4.35974417 \times 10^{-18}$ J</td>
</tr>
<tr>
<td>Mass</td>
<td>$m_e$ (electron mass)</td>
<td>$9.1093826 \times 10^{-31}$ kg</td>
</tr>
<tr>
<td>Length</td>
<td>$a_0$ (bohr)</td>
<td>$0.5291772108 \times 10^{-10}$ m</td>
</tr>
<tr>
<td>Time</td>
<td>$\hbar/E_h$</td>
<td>$2.418884326505 \times 10^{-17}$ s</td>
</tr>
<tr>
<td>Velocity</td>
<td>$a_0 \dot{E}_h/\hbar$</td>
<td>$2.1876912633 \times 10^6$ m s$^{-1}$</td>
</tr>
<tr>
<td>Momentum</td>
<td>$\hbar/a_0$</td>
<td>$1.99285166 \times 10^{-24}$ kg m s$^{-1}$</td>
</tr>
<tr>
<td>Force</td>
<td>$E_h/a_0$</td>
<td>$8.2387225 \times 10^{-8}$ N</td>
</tr>
</tbody>
</table>

Table 7.2.1: Atomic Units

The computer programs are intended to be compliant with the Fortran 1977 ISO/ANSI standard [75] with the exception of non-standard but ubiquitous constructs such as the “IMPLICIT NONE” and “INCLUDE” statements. Some variable names may also be longer than the allowable six characters, but most compilers support longer names.

All calculations were performed in platform-specific double precision for the real floating-point numbers. The typical word size for most microprocessor architectures is four bytes with eight bits to one byte. According to the IEEE 754-1985 standard [41] for binary floating-point numbers, single precision storage occupies one 32 bit word, and double precision occupies two consecutive 32 bit words. An IEEE 754 double precision floating-point number on such
a machine would have sixty-four bits to represent the number where one bit is used for the sign, ten for the exponent, and fifty-three bits for storing the mantissa, also known as the significand. The number of precise significant base-10 decimal digits converted from IEEE 754 binary floating-point is typically six to nine digits and fifteen to seventeen for double precision [69, 85]. Refer to the 1991 publication by Goldberg [57] on the IEEE standard and floating-point arithmetic.

The programs are relatively fast where most single trajectory calculations can be computed in less than a few minutes on a modern personal desktop computer. The evaluation of the force, first derivatives of the potential, dominates the computational time. No explicit optimizations such as loop unrolling or parallelization have been done. The symplectic integrator is written without any loop constructs to avoid unnecessary calculations of phase space variables that are constants of the dynamics for a particular symplectic map. The comparison integrators were not specialized for the particular Hamiltonian, treating it as any other ordinary differential equation. The symplectic integrator required only one force evaluation per integration step, accounting for considerable speed advantage over the fourth order Runge-Kutta and the Bulirsch-Stoer integrators.
Chapter 8

Modeling and Simulations

8.1 The H + H$_2$ System

The simplest chemical system for modeling molecular reaction dynamics is the atom-diatom interchange between a hydrogen atom and a hydrogen molecule, $\text{H} + \text{H}_2 \rightarrow \text{H}_2 + \text{H}$. The total interaction species consists of just three nuclei and three electrons. The reaction is important not only as a benchmark for more complicated studies, but also for elucidating important processes of hydrogen gas in interstellar space and celestial bodies. The first accurate, three-dimensional quantum mechanical scattering calculations for this reaction were reported by Kuppermann and Schatz in 1975 and 1976 for both differential and integral cross sections [97 157].

The ab initio potential energy surface used in this simulation is the “BKMP2” H$_3$ surface published in 1996 by Boothroyd, Keogh, Martin, and Peterson [19]. The classical barrier height for the surface is 9.61 kcal/mol. This surface was selected based on the availability of computer programs for calculating the potential energy and first derivatives, as well as its accuracy in the van der Waals well and asymptotic region. The initial energies and separation distances, $R_{\text{H-H}_2}$, between the ballistic atom and the center of the target molecule were chosen to be within these regions of long vibrational or rotational periods. The van der Waals region is reported to be at a separation distance approximately between six and ten bohr ($a_0$) ($6a_0 < R_{\text{H-H}_2} < 10a_0$).

To clearly demonstrate the efficacy of the symplectic integrator, the center-of-mass collision energy was set to an initial value of zero. Although this is an unrealistic energy for a reaction, closed orbits are easily obtained and serve as a test for long-term, qualitative behavior. At certain configurations, the system oscillates indefinitely and the phase plane should display
a stable periodic trajectory. The impact particle, the lone hydrogen atom, is aligned with the center-of-mass of the coupled particles, the hydrogen molecule, corresponding to the $\chi = 90^\circ$ angle in the BKMP2 article. The initial configuration shape is that of an isosceles triangle where the separation distance between the impact particle and coupled particles is -6.58 bohr and the initial separation distance of the coupled particles is 1.449 bohr. These distances correspond to the van der Waals minimum of the potential and are considered to be competitively accurate with other surfaces. The following table lists the input data for the numerical calculations. All units are in atomic units and angles are initially in degrees but converted to radians in the computations.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1837.15258</td>
<td>MASS OF PARTICLE A (ELECTRON MASS)</td>
</tr>
<tr>
<td>1837.15258</td>
<td>MASS OF PARTICLE B (ELECTRON MASS)</td>
</tr>
<tr>
<td>1837.15258</td>
<td>MASS OF PARTICLE C (ELECTRON MASS)</td>
</tr>
<tr>
<td>0</td>
<td>ROTATIONAL QUANTUM NUMBER</td>
</tr>
<tr>
<td>0.0</td>
<td>CENTER OF MASS COLLISION ENERGY (HARTREE)</td>
</tr>
<tr>
<td>1.449</td>
<td>DISTANCE BETWEEN PARTICLES B AND C (BOHR)</td>
</tr>
<tr>
<td>0.</td>
<td>X-COORDINATE FOR IMPACT PARTICLE A (BOHR)</td>
</tr>
<tr>
<td>-6.58</td>
<td>Y-COORDINATE FOR IMPACT PARTICLE A (BOHR)</td>
</tr>
<tr>
<td>0.</td>
<td>Z-COORDINATE FOR IMPACT PARTICLE A (BOHR)</td>
</tr>
<tr>
<td>0.</td>
<td>EULER ANGLE ABOUT Z-AXIS OF B-C COUPLE (DEGREES)</td>
</tr>
<tr>
<td>0.</td>
<td>EULER ANGLE ABOUT Y-AXIS OF B-C COUPLE (DEGREES)</td>
</tr>
<tr>
<td>0.</td>
<td>EULER ANGLE ABOUT Z-AXIS OF B-C COUPLE (DEGREES)</td>
</tr>
<tr>
<td>1000000</td>
<td>MAXIMUM NUMBER OF INTEGRATION STEPS</td>
</tr>
<tr>
<td>1.0</td>
<td>TIME STEP SIZE</td>
</tr>
<tr>
<td>100</td>
<td>INTEGRATION PRINTOUT INTERVAL</td>
</tr>
</tbody>
</table>

Table 8.1.1: Input parameters for H + H$_2$ reaction.

Phase plane portraits for the hyperradius versus its conjugate momentum and plots of the relative energy versus time for the initial conditions given in Table 8.1.1 are provided in Figure 8.3.1.

At the smaller step size all the integrators clearly depict a closed orbit and the energy remains relatively constant as expected for a conservative system. Recall the hyperradius represents the size of the triangle formed by the three particles. The range of motion for the hyperradius in the phase plane indicates that the system is undergoing small amplitude vibration. At this center-of-mass collision energy and position on the potential surface, the system can be interpreted as a weakly bound, intermediate triatomic state formed by the weak van der Waals forces. The separation distance between the lone particle and the coupled particle pair oscillate as the attractive and repulsive forces of the potential pull and push the particles in unison. The oscillation is periodic and can be surmised to be indefinite and constant.
The motion is analogous to a marble confined to a bowl, rolling up and down the sides of the bowl. The marble retraces its path after each period and never stops moving, seemingly obligated to a fate of eternal unrest. The question remains if this story holds at larger step sizes or longer simulation times. In Figure 8.3.2, the same initial conditions from Table 8.1.1 with the step size increased by an order of magnitude are used for the integration.

When the step size was increased by an order of magnitude, the story fails for the non-symplectic integrators. The phase plane generated by the symplectic integrator at the larger step sizes preserves the geometric integrity, but the non-symplectic integrators clearly obliterate the phase space structure at the larger step sizes. In addition, there is a gradual but secular drift in the energy. In contrast, the energy is bounded and stable for the symplectic integrator. Trajectories for the symplectic integrator are more qualitatively correct. The geometry preservation and energy stability are well-known properties of symplectic integrators. The new integrator developed in this study displays the same well-known attributes of symplectic integrators.

8.2 The F + H₂ System

The reaction between atomic fluorine and molecular hydrogen, $F + H₂ → HF + H$, is another benchmark bimolecular reaction. This reaction has a low activation energy barrier and is highly exoergic. It is one of the most intensely studied chemical reactions from both experimental and theoretical perspectives since the reaction is fast and can be carried out in the ground state with an experimentally accessible energy. The products are formed from the abstraction of hydrogen by the incident fluorine atom. The reaction has an early barrier, releasing the exoergicity in the entrance channel of the reactants where much of the released energy is pumped into the internal vibrational energy of the products [106, section 4.2]. In a bulk system, this can lead to a population inversion where there are more product molecules in an excited vibrational state than in the ground state, a key property necessary for light amplification by stimulated emission of radiation (LASER). The $F + H₂$ reaction is therefore of interest in the study of chemical lasers. The reaction is also special due to the manifestation of resonances, a quantum mechanical phenomenon of long-lived, quasibound states that play a significant role in the final distribution of product vibrational states [106].

The development of a highly accurate, ab initio potential energy surface introduced by Stark and Werner [175] in 1996 allowed the reconciliation of theoretical quantum calculations with empirical observation for the first time. Even before the formal publication of the surface, a preliminary version had excellent agreement with the transition state spectroscopy results from a combined theoretical and experimental study by Manolopoulos, et al. [116] for the photodetachment of the $FH₂^−$ anion. The first fully quantum mechanical calculations to resolve the asymptotic products of the $F+H₂$ reaction using the new Stark-Werner surface were
accomplished by Castillo, et al. [24] in 1996 using Smith-type hyperspherical coordinates. In this article, the theoretical quantum calculations produced higher than expected forward scattering for the third excited vibrational quantum state of the HF product, contrary to quasiclassical trajectory calculations and experimental results. It was originally conjectured to be attributed to the existence of quantum mechanical reactive scattering resonances, but determined to be quantum mechanical tunneling of the reaction barrier at high orbital angular momentum [23, 24, 149]. However, reactive resonances are predicted by theory and it still difficult for laboratory experiments to quantitatively resolve these quantum predictions [4, 23, 119, 149, 170]. A summary of the important early research and knowledge of the $F + H_2$ reaction can be found in a 1997 article by Manolopoulos [112]. Since that publication, the reaction and its isotopic variations has continued to receive attention and remains an active area of research [4, 9, 23, 108, 119, 170]. The leading quantum computations for the $F + H_2$ reaction are done in hyperspherical coordinates [171]. The classical trajectory analog studies may be useful when revisited with symplectic integration.

Classical trajectory calculations in Johnson hyperspherical coordinates for the $F + H_2$ system were performed to verify the canonical properties of the symplectic integrator of this dissertation. The Stark-Werner potential energy surface [175] for the FH$_2$ system was selected for its accuracy and popularity. The potential energy also has an accurate analytical form and computer programs for calculating the energy and derivatives are available from the authors.

Initial values for the phase space hyperspherical variables were arbitrarily chosen to produce closed trajectories so manifestation of the symplectic properties could be exaggerated. This serves as a benchmark test for the symplecticness of an integrator. The initial interparticle separation distance for the coupled particles was calculated to mimic the maximum stretch of a hydrogen molecule in the ground state using the two-body term of the Stark-Werner potential energy in analytical form. The initial configuration of the three particles is T-shaped geometry where the particle positions represent the vertices of an isosceles triangle with the altitude distance of 8.0 bohr.

129
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>34631.9703</td>
<td>MASS OF PARTICLE A (ELECTRON MASS)</td>
</tr>
<tr>
<td>1837.15258</td>
<td>MASS OF PARTICLE B (ELECTRON MASS)</td>
</tr>
<tr>
<td>1837.15258</td>
<td>MASS OF PARTICLE C (ELECTRON MASS)</td>
</tr>
<tr>
<td>0</td>
<td>ROTATIONAL QUANTUM NUMBER</td>
</tr>
<tr>
<td>0.0</td>
<td>CENTER OF MASS COLLISION ENERGY (HARTREE)</td>
</tr>
<tr>
<td>1.67263657</td>
<td>DISTANCE BETWEEN PARTICLES B AND C (BOHR)</td>
</tr>
<tr>
<td>0.0</td>
<td>X-COORDINATE FOR IMPACT PARTICLE A (BOHR)</td>
</tr>
<tr>
<td>-8.00</td>
<td>Y-COORDINATE FOR IMPACT PARTICLE A (BOHR)</td>
</tr>
<tr>
<td>0.0</td>
<td>Z-COORDINATE FOR IMPACT PARTICLE A (BOHR)</td>
</tr>
<tr>
<td>0.0</td>
<td>EULER ANGLE ABOUT Z-AXIS OF B-C COUPLE (DEGREES)</td>
</tr>
<tr>
<td>0.0</td>
<td>EULER ANGLE ABOUT Y-AXIS OF B-C COUPLE (DEGREES)</td>
</tr>
<tr>
<td>0.0</td>
<td>EULER ANGLE ABOUT Z-AXIS OF B-C COUPLE (DEGREES)</td>
</tr>
<tr>
<td>1000000</td>
<td>MAXIMUM NUMBER OF INTEGRATION STEPS</td>
</tr>
<tr>
<td>1.0</td>
<td>TIME STEP SIZE</td>
</tr>
<tr>
<td>100</td>
<td>INTEGRATION PRINTOUT INTERVAL</td>
</tr>
</tbody>
</table>

Table 8.2.1: Input parameters for F + H$_2$ reaction.

Phase plane portraits for the hyperradius versus its conjugate momentum and plots of the relative energy versus time for the initial conditions given in Table 8.2.1 are provided in Figure 8.3.3. All the integrators produce similar phase portraits of the triatomic system with stable conservation of the total energy. The large range of the hyperradius suggests the system is undergoing large amplitude vibration. The step size is increased by an order of magnitude and the results are displayed in Figure 8.3.4. At larger step sizes, the phase space structure becomes diffused for all but the symplectic integrator, another testament to the superior volume-preserving quality of symplectic integrators. Again, the degradation of the energy parallels that of the phase space geometry, falsely implying that system is dissipative when in fact, it is conservative.

Another set of initial conditions produce an interesting phase plane portrait. The difference in these initial conditions is a simple change of the separation distance of the lone particle from the center of mass of the coupled particle pair. The total input parameters are listed in Table 8.2.2 and the corresponding phase space plots are in Figure 8.3.5.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3463.9703</td>
<td>MASS OF PARTICLE A (ELECTRON MASS)</td>
</tr>
<tr>
<td>1837.15258</td>
<td>MASS OF PARTICLE B (ELECTRON MASS)</td>
</tr>
<tr>
<td>1837.15258</td>
<td>MASS OF PARTICLE C (ELECTRON MASS)</td>
</tr>
<tr>
<td>0</td>
<td>ROTATIONAL QUANTUM NUMBER</td>
</tr>
<tr>
<td>0.0</td>
<td>CENTER OF MASS COLLISION ENERGY (HARTREE)</td>
</tr>
<tr>
<td>1.67263657</td>
<td>DISTANCE BETWEEN PARTICLES B AND C (BOHR)</td>
</tr>
<tr>
<td>-5.00</td>
<td>X-COORDINATE FOR IMPACT PARTICLE A (BOHR)</td>
</tr>
<tr>
<td>0.0</td>
<td>Y-COORDINATE FOR IMPACT PARTICLE A (BOHR)</td>
</tr>
<tr>
<td>0.0</td>
<td>Z-COORDINATE FOR IMPACT PARTICLE A (BOHR)</td>
</tr>
<tr>
<td>0.0</td>
<td>EULER ANGLE ABOUT Z-AXIS OF B-C COUPLE (DEGREES)</td>
</tr>
<tr>
<td>0.0</td>
<td>EULER ANGLE ABOUT Y-AXIS OF B-C COUPLE (DEGREES)</td>
</tr>
<tr>
<td>1000000</td>
<td>MAXIMUM NUMBER OF INTEGRATION STEPS</td>
</tr>
<tr>
<td>10.0</td>
<td>TIME STEP SIZE</td>
</tr>
<tr>
<td>100</td>
<td>INTEGRATION PRINTOUT INTERVAL</td>
</tr>
</tbody>
</table>

Table 8.2.2: Input parameters for F + H₂ reaction.

The small range of the hyperradius suggests the system is undergoing small amplitude vibration, similar to the first integration test of the H₃ system in the previous section.

The symplectic integrator developed in this dissertation once again demonstrates excellent phase space topology integrity and stable energy conservation in comparison to the non-symplectic integrators.
Figure 8.3.1: Phase plane portraits (left column) and relative energy versus time plots (right column) for the H + H₂ reaction. Comparison of integrators for the Hyperspherical, Three-body Hamiltonian using the BKMP2 H₃ Potential Energy Surface. Bulirsch-Stoer (a-b), Runge-Kutta 4th order (c-d), Symplectic 2nd order (e-f) integrators. Integrated for $1 \times 10^6$ steps with a step size of 1.0.
Figure 8.3.2: Phase plane portraits (left column) and relative energy versus time plots (right column) for the H + H$_2$ reaction. Comparison of integrators for the Hyperspherical, Three-body Hamiltonian using the BKMP2 H$_3$ Potential Energy Surface. Bulirsch-Stoer (a-b), Runge-Kutta 4th order (c-d), Symplectic 2nd order (e-f) integrators. Integrated for $1 \times 10^6$ steps with a step size of 10.0.
Figure 8.3.3: Phase plane portraits (left column) and relative energy versus time plots (right column) for the F + H$_2$ reaction. Comparison of integrators for the Hyperspherical, Three-body Hamiltonian using the Stark-Werner FH$_2$ Potential Energy Surface. Bulirsch-Stoer (a-b), Runge-Kutta 4th order (c-d), Symplectic 2nd order (e-f) integrators. Integrated for $1 \times 10^6$ steps with a step size of 1.0.
Figure 8.3.4: Phase plane portraits (left column) and relative energy versus time plots (right column) for the F + H$_2$ reaction. Comparison of integrators for the Hyperspherical, Three-body Hamiltonian using the Stark-Werner FH$_2$ Potential Energy Surface. Bulirsch-Stoer (a-b), Runge-Kutta 4th order (c-d), Symplectic 2nd order (e-f) integrators. Integrated for $1 \times 10^6$ steps with a step size of 10.0.
Figure 8.3.5: Phase plane portraits (left column) and relative energy versus time plots (right column) for the F + H₂ reaction. Comparison of integrators for the Hyperspherical, Three-body Hamiltonian using the Stark-Werner FH₂ Potential Energy Surface. Bulirsch-Stoer (a-b), Runge-Kutta 4th order (c-d), Symplectic 2nd order (e-f) integrators. Integrated for 1x10⁶ steps with a step size of 10.0.
Chapter 9

Summary and Conclusions

The canonical attribute of symplectic integrators ensure the nearly exact solution to a nearby Hamiltonian for each integration step. Symplectic integrators produce the time evolution map, or phase flow, for a Hamiltonian system. Although only the true Hamiltonian can preserve all the integral invariants, all symplectic integration methods preserve the phase space volume. The phase space volume preserving property of symplectic integrators is most apparent for periodic or quasi-periodic trajectories after long integration times. The symplectic integrators exhibit less error than non-symplectic integrators in computed trajectories at larger step sizes and longer simulations. Symplectic integrators are therefore preferred for long term qualitative dynamical studies.

An explicit, multi-map Ruth-type symplectic integrator for a class of three-body Hamiltonians in hyperspherical polar coordinates has been developed and presented in this dissertation, specifically for planar motion. The integrator was applied to two prototypical three-body chemical systems to evaluate its symplectic properties. The study successfully demonstrated that the integrator possessed the characteristic, superior geometry-preserving properties of symplectic integrators for long term qualitative dynamics. The symplectic integrator better conserved the phase space volume and total energy than comparative, non-symplectic integrators at larger step sizes and longer integration times. In addition, it was computationally faster requiring only one force evaluation per step. It is the first explicit symplectic integrator for Hamiltonians of three interacting bodies in hyperspherical coordinates, specifically for the class of Smith-type symmetrized, hyperspherical polar coordinates.

Explicit symplectic integrators can be constructed from a composition of exactly integrable canonical maps but no general solution exists for non-separable or general Hamiltonians. Although implicit or semi-explicit symplectic integrators can be developed for any Hamiltonian, the added cost of iterations per step required by implicit integrators will negatively impact performance. The presented integrator, by nature of being explicit, is more efficient.
and algorithmically faster than non-explicit methods.

Classical trajectories of chemical systems with bound or long-lived states are prospective candidates for symplectic integration. Chemical reactions can also be considered for symplectic treatment despite the short timescales. We suggest symplectic integrators would be preferable for investigation of long lifetime states near the asymptotic limit of dissociation for chemical reactions. Quantum mechanical predictions are equivalent to classical mechanics at the dissociation limit where the quantization of energy levels decays into a continuum, also known as the quantum-classical limit, and can be effectively investigated using classical trajectory studies. Hyperspherical coordinates can be employed to simplify transformation between different particle arrangements and the mapping of the configuration space. The hyperspherical coordinates also provide unique and abstract visualization of the phase space and potential energy surfaces. The presented integrator is well-suited for such studies and should be preferred over other integrators for Hamiltonian dynamics using Smith-type class of hyperspherical coordinates.

The symplectic integrator of this dissertation can aid in the investigation of classical trajectories of three-body chemical systems in symmetrized, hyperspherical coordinates. It is purported to be particularly useful in the calibration of three-body potential energy surfaces and probing long lifetime states at van der Waals minima in the dissociation channels of three-body, $A + BC$, chemical reaction dynamics. Investigation of weakly bound van der Waals states at near threshold limits for triatomic systems has been an area of recent interest. Reactions such as, $O + HCl \rightarrow OH + Cl$, have accurate potential energy surfaces exhibiting shallow van der Waals minima at the entrance and exit channels. In 2002, resonances were calculated for this reaction at low energies by Xie, et al. [193] using hyperspherical coordinates. Effects of weakly bound van der Waals states at dissociation thresholds in ozone have received increased attention due to a new accurate potential energy surface. [8, 62, 84, 162, 163]. The $F + H_2$ reaction also exhibits van der Waals minima and periodic orbits for the collinear geometries have been studied [52]. Periodic orbits of bound states in the classical phase space for three body systems are suited for classical trajectory analysis, especially at the dissociation limit where it becomes more difficult to assign vibrational states to quantum states [47, 52, 83]. Periodic orbits of bound states in hypochlorous, HOCl, and hypobromous, HOBr, acids have recently been investigated [7, 187] using classical mechanics to elucidate quantum spectra otherwise computationally prohibitive or cumbersome. Such systems are prospective candidates for symplectic treatment of the classical trajectories with the integrator of this dissertation. These trajectories serve as a diagnostic of the chemical dynamics and verification of molecular structure in the form of potential energy surfaces, important properties for the continued pursuit of understanding chemical processes.

The integrator is generally applicable to any three-body system with a conservative potential energy in classical mechanics. It may serve well in studies of three body motion in astronomy, particularly for Kepler orbits which are usually performed in Cartesian or Jacobi coordinates.
Application of hyperspherical coordinates may provide interesting insight to the dynamics of motion under gravitational forces. Other areas in physics and chemistry may find the use beneficial.

The symplectic integrator developed in this dissertation may admit improvements both in theoretical and computational respects. The choice of the Hamiltonian maps and implementation of the stages for the integrator may be improved and the number of maps reduced. Higher order accuracy for the symplectic integrator could be obtained using the Yoshida symmetrized, even-order method but the advantage of smaller truncation error is undermined by the increase of stages and introduction of negative steps, thereby affecting performance and increasing the possibility of roundoff error. Techniques for avoiding the backward stepping have been discussed in the literature review and may prove useful.

The presented integrator is applicable for planar motion only, neglecting the cross-term for the centrifugal energy of the rotating body. The plane of the particles can be allowed to rotate in the three-dimensional space frame where the final Euler angle, $\gamma$, is not a constant as it is for planar motion. However, the fully spatial Hamiltonian in hyperspherical coordinates does not easily decompose into exactly integrable maps, a requirement for the development of explicit symplectic integrators. The splitting of the spatial Hamiltonian can be done as in the planar case, but with the addition of a new sub Hamiltonian as a function of the coupled variables in the Euler angle and its conjugate momentum. The equations of motion for this sub Hamiltonian are nonlinear and no analytical solution is obvious. A semi-explicit approach, where the added Hamiltonian map is integrated by an implicit method based on a canonical generating function, may offer the symplectic advantages at a cost to computation. Finally, extension to quantum mechanical studies is possible and left for future pursuit.
Bibliography


140


[80] B. Johnson. On hyperspherical coordinates and mapping the internal configurations of a three body system. *Journal of Chemical Physics*, 73(10):5051–5058, 1980. \[5.7\] \[5.1\] \[5.4\]


Appendix A

Integrator Programs

A.1 Symplectic Integrator Programs

A.1.1 HS2D_SIA2

Explicit, multi-map symplectic integrator for three-body classical trajectory studies in hyperspherical coordinates. Developed specifically for Hamiltonians in Smith-Whitten-Johnson symmetrized, hyperspherical polar coordinates for planar motion. Refer to for algorithm theoretical basis and construction.

```fortran
SUBROUTINE HS2D_SIA2(TIMESTEP, PHASEPT)
IMPLICIT NONE
INCLUDE 'NUMCON.FOR'
INCLUDE 'PHASESPACE.FOR'
INCLUDE 'MASS.FOR'

DOUBLE PRECISION $ TIMESTEP, HALFSTEP,
$ DVDN(3)
DOUBLE PRECISION $ INVQ1, INVIZ, DENOM,
```

157
C

******************************************************************************
C  Statement Functions
C
C  NOTE: These functions should be inlined.
C  Refer to compiler options and documentation.
C
******************************************************************************

C Statement function declarations.
DOUBLE PRECISION
$ DH1DP1, $ DH2DQ1, DH2DP2, $ DH3DQ1, DH3DQ2, DH3DP3

C Statement function definitions.
DH1DP1(P) = P/MU
DH2DQ1(P) = -FOUR*P*INVIZ*INVQ1
DH2DP2(P) = FOUR*P*INVIZ
DH3DQ1() = DH3*INVQ1
DH3DQ2(P) = ( DH3*COSQ2 + TWO*PHASEPT(P4)+INVIZ )/SINQ2
DH3DP3(P) = ( FOUR*P - TWO*JZ )/DENOM

C

******************************************************************************
C q1 = rho , p1 = prho
C q2 = theta , p2 = ptheta
C q3 = phi , p3 = pphi
C p4 = pgamma = constant
C
C  Symmetrised mapping: N1(1/2 t )N2(1/2 t )N3(1/2 t )N4( t )N3(1/2 t )N2(1/2 t )N1(1/2 t )
C
******************************************************************************

HALFSTEP = HALF*TIMESTEP
PEXT2 = PHASEPT(P4)+PHASEPT(P4)

C MAP1
PHASEPT(Q1) = PHASEPT(Q1) + HALFSTEP*DH1DP1( PHASEPT(P1) )
INVQ1 = ONE/PHASEPT(Q1)
INVIZ = ONE/( MU*PHASEPT(Q1)*PHASEPT(Q1) )

C MAP2
PHASEPT(P1) = PHASEPT(P1) - HALFSTEP*DH2DQ1( PHASEPT(P2) )
PHASEPT(Q2) = PHASEPT(Q2) + HALFSTEP*DH2DP2( PHASEPT(P2) )
COSQ2 = DCOS( PHASEPT(Q2) )
SINQ2 = DSN( PHASEPT(Q2) )
DENOM = INVIZ/( SINQ2*SINQ2 )
JZ = PHASEPT(P4)+COSQ2
DH3 = ( FOUR+PHASEPT(P3)+PHASEPT(P3))

C MAP3
PHASEPT(P1) = PHASEPT(P1) - HALFSTEP*DH3DQ1( )
PHASEPT(P2) = PHASEPT(P2) - HALFSTEP*DH3DP3( PHASEPT(P3) )
PHASEPT(Q3) = PHASEPT(Q3) + HALFSTEP*DH3DQ2( PHASEPT(P3) )

C MAP4
CALL FORCE(PHASEPT, DVDN)
PHASEPT(P1) = PHASEPT(P1) - TIMESTEP*DVIN(1)
PHASEPT(P2) = PHASEPT(P2) - TIMESTEP*DVIN(2)
PHASEPT(P3) = PHASEPT(P3) - TIMESTEP*DVIN(3)
DH3 = ( FOUR-PHASEPT(P3)+PHASEPT(P3))

C MAP5
PHASEPT(P1) = PHASEPT(P1) - HALFSTEP*DH3DQ1( )
PHASEPT(P2) = PHASEPT(P2) - HALFSTEP*DH3DP3( PHASEPT(P3) )
PHASEPT(Q3) = PHASEPT(Q3) + HALFSTEP*DH3DQ2( PHASEPT(P3) )

C MAP6
PHASEPT(P1) = PHASEPT(P1) - HALFSTEP*DH3DQ1( )
PHASEPT(P2) = PHASEPT(P2) - HALFSTEP*DH3DP3( PHASEPT(P3) )
PHASEPT(Q3) = PHASEPT(Q3) + HALFSTEP*DH3DQ2( PHASEPT(P3) )

158
A.1.2 SIA2

Explicit, second order Ruth-type symplectic integrator for Hamiltonian dynamics.

SUBROUTINE SIA2(TIMESTEP, QVAR, PVAR, DEGREES)
IMPLICIT NONE
INCLUDE 'NUMCONFOR'
INCLUDE 'INTG.FOR'
INTEGER I1
DOUBLE PRECISION $TIMESTEP, HALFSTEP,
$ VELOC(DEGREES),
$ FORCE(DEGREES)
HALFSTEP = HALF*TIMESTEP
CALL EQNMOT(QVAR, PVAR, DEGREES, VELOC, FORCE)
DO 10, I1 = 1, DEGREES
  QVAR(I1) = QVAR(I1) + HALFSTEP*VELOC(I1)
10 CONTINUE
CALL EQNMOT(QVAR, PVAR, DEGREES, VELOC, FORCE)
DO 11, I1 = 1, DEGREES
  PVAR(I1) = PVAR(I1) + TIMESTEP*FORCE(I1)
11 CONTINUE
CALL EQNMOT(QVAR, PVAR, DEGREES, VELOC, FORCE)
DO 12, I1 = 1, DEGREES
  QVAR(I1) = QVAR(I1) + HALFSTEP*VELOC(I1)
12 CONTINUE
RETURN
END

A.1.3 SIA4

Explicit, fourth order Ruth-type symplectic integrator for Hamiltonian dynamics.
Program: SIA4

Explicit 4th order symplectic integrator.
The coefficients for the mappings were first determined by Neri and Candy.

Method is by composition of exactly integrable Hamiltonians:
\[ H = H_1(q) + H_2(p) \]


********************************************************************************

SUBROUTINE SIA4(TIMESTEP, QVAR, PVAR, DEGREES)

IMPLICIT NONE

INCLUDE 'NUMCON.FOR'

INTEGER I1

DOUBLE PRECISION $ TIMESTEP,

$ VELOC(DEGREES),

$ FORCE(DEGREES)

DOUBLE PRECISION THIRD

PARAMETER (THIRD = ONE/THREE)

DOUBLE PRECISION $ C1, C2, D1, D2

DOUBLE PRECISION $ ASTEP, BSTEP, CSTEP, DSTEP

DOUBLE PRECISION $ CUBRT2

CUBRT2 = TWO**(THIRD)

D1 = ONE/( TWO - CUBRT2 )

D2 = -CUBRT2*D1

C1 = HALF*D1

C2 = ( ONE - CUBRT2 )*C1

ASTEP = C1*TIMESTEP

BSTEP = D1*TIMESTEP

CSTEP = C2*TIMESTEP

DSTEP = D2*TIMESTEP

CALL EQNMOT(QVAR, PVAR, DEGREES, VELOC, FORCE)

DO 10, I1 = 1, DEGREES

QVAR(I1) = QVAR(I1) + ASTEP*VELOC(I1)

10 CONTINUE

CALL EQNMOT(QVAR, PVAR, DEGREES, VELOC, FORCE)

DO 11, I1 = 1, DEGREES

PVAR(I1) = PVAR(I1) + BSTEP*FORCE(I1)

11 CONTINUE

CALL EQNMOT(QVAR, PVAR, DEGREES, VELOC, FORCE)

DO 12, I1 = 1, DEGREES

QVAR(I1) = QVAR(I1) + CSTEP*VELOC(I1)

12 CONTINUE

CALL EQNMOT(QVAR, PVAR, DEGREES, VELOC, FORCE)

DO 13, I1 = 1, DEGREES

PVAR(I1) = PVAR(I1) + DSTEP*FORCE(I1)

13 CONTINUE

CALL EQNMOT(QVAR, PVAR, DEGREES, VELOC, FORCE)

DO 14, I1 = 1, DEGREES

QVAR(I1) = QVAR(I1) + CSTEP*VELOC(I1)

14 CONTINUE

CALL EQNMOT(QVAR, PVAR, DEGREES, VELOC, FORCE)

DO 15, I1 = 1, DEGREES

PVAR(I1) = PVAR(I1) + BSTEP*FORCE(I1)

15 CONTINUE

CALL EQNMOT(QVAR, PVAR, DEGREES, VELOC, FORCE)

DO 16, I1 = 1, DEGREES

QVAR(I1) = QVAR(I1) + ASTEP*VELOC(I1)

16 CONTINUE

160
A.2 Ordinary Differential Equations Integrator Programs

A.2.1 RK4

Explicit, fourth order Runge-Kutta integrator for ordinary differential equations.

```
SUBROUTINE RK4(X, STEP, Y, NDE)
IMPLICIT NONE
INCLUDE 'NUMC0N.FOR'
INTEGER I1
DOUBLE PRECISION SIXTH
PARAMETER (SIXTH = ONE/SIX)
INTEGER NDE
DOUBLE PRECISION $ K1(NDE) , $ K2(NDE) , $ K3(NDE) , $ K4(NDE)
DOUBLE PRECISION $ STEP, HALFSTEP

HALFSTEP = HALF*STEP
X0 = X
DO 5, I1 = 1, NDE
  Y0(I1) = Y(I1)
  CONTINUE

C Integration
CALL DERIV(X0, Y0, DYDX)
DO 10, I1 = 1, NDE
  K1(I1) = STEP*DYDX(I1)
  Y(I1) = Y0(I1) + HALF*K1(I1)
  CONTINUE

X1 = X0 + HALFSTEP
CALL DERIV(X1, Y, DYDX)
DO 20, I1 = 1, NDE
  K2(I1) = STEP*DYDX(I1)
  Y(I1) = Y0(I1) + HALF*K2(I1)
  CONTINUE
```

161
CALL DERIV(X1, Y, DYDX)
DO 30, I1 = 1, NDE
   K3(I1) = STEP*DYDX(I1)
   Y(I1) = Y0(I1) + K3(I1)
30 CONTINUE

X0 = X0 + STEP
CALL DERIV(X0, Y0, DYDX)
DO 40, I1 = 1, NDE
   K4(I1) = STEP*DYDX(I1)
   Y(I1) = Y0(I1) + SIXTH*( K1(I1) + TWO*( K2(I1) + K3(I1) ) + K4(I1) )
40 CONTINUE

RETURN
END

A.2.2 RK4_QP

Explicit, fourth order Runge-Kutta integrator for ordinary differential equations. Interface for Hamiltonian phase space variables.

******************************************************************************
* Author: Paul Burkhardt
* Date: September 28, 2004
* Program: RK4
* Explicit 4th order Runge-Kutta integrator.
******************************************************************************
SUBROUTINE RK4_QP(X, STEP, QVAR, PVAR, DEGREES, NDE)
IMPLICIT NONE
INCLUDE 'NUMCON.FOR'
INCLUDE 'INTG.FOR'
INTEGER IDX
DOUBLE PRECISION SIXTH
PARAMETER (SIXTH = ONE/SIX)
INTEGER NDE
DOUBLE PRECISION $ K1(NDE) , $ K2(NDE) , $ K3(NDE) , $ K4(NDE)
DOUBLE PRECISION $ STEP, HALFSTEP
DOUBLE PRECISION $ X, X0, X1,
$ Y(NDE), Y0(NDE),
$ DYDX(NDE)
HALFSTEP = HALF*STEP
X0 = X
DO 5, IDX = 1, DEGREES
   Y0(IDX) = QVAR(IDX)
   Y0(IDX+DEGREES) = PVAR(IDX)
5 CONTINUE

CALL DERIV(X0, Y0, DYDX, DEGREES)
DO 10, IDX = 1, NDE
   K1(IDX) = STEP*DYDX(IDX)
   Y(IDX) = Y0(IDX) + HALF*K1(IDX)
10 CONTINUE

162
X1 = X0 + HALFSTEP
CALL DERIV(X1, Y, DYDX, DEGREES)
DO 20, IDX = 1, NDE
   K2(IDX) = STEP*DYDX(IDX)
   Y(IDX) = Y0(IDX) + HALF*K2(IDX)
20 CONTINUE
CALL DERIV(X1, Y, DYDX, DEGREES)
DO 30, IDX = 1, NDE
   K3(IDX) = STEP*DYDX(IDX)
   Y(IDX) = Y0(IDX) + K3(IDX)
30 CONTINUE
X1 = X0 + STEP
CALL DERIV(X1, Y, DYDX, DEGREES)
DO 40, IDX = 1, NDE
   K4(IDX) = STEP*DYDX(IDX)
   Y(IDX) = Y0(IDX) + SIXTH*(K1(IDX) + TWO*(K2(IDX) + K3(IDX)) + K4(IDX))
40 CONTINUE
DO 50, IDX = 1, DEGREES
   QVAR(IDX) = Y(IDX)
   PVAR(IDX) = Y(IDX+DEGREES)
50 CONTINUE
RETURN
END

A.3 Include Files for Integrator Programs

A.3.1 INTG.FOR

*****************************************************************************
* Author: Paul Burkhardt                                                *
* Date: September 28, 2004                                             *
* Include File                                                          *
* Purpose: Include file for integration variables.                     *
*****************************************************************************
INTEGER DEGREES
DOUBLE PRECISION
   $ QVAR(DEGREES)
   $ PVAR(DEGREES)

A.3.2 MASS.FOR

*****************************************************************************
* Author: Paul Burkhardt                                                *
* Date: September 28, 2004                                             *
* Include File                                                          *
* Purpose: Include file defining mass related constants.                *
*****************************************************************************
DOUBLE PRECISION
   $ JTOT
A.3.3 NUMCON.FOR

******************************************************************************
* Author: Paul Burkhardt
* Date: September 28, 2004
* Include File
* Purpose: Include file for double precision numerical constants.
******************************************************************************
C ******************************************************************************
C  * WHOLE CONSTANTS
C ******************************************************************************
DOUBLE PRECISION ZERO
PARAMETER (ZERO = 0.D0)
DOUBLE PRECISION ONE
PARAMETER (ONE = 1.D0)
DOUBLE PRECISION TWO
PARAMETER (TWO = 2.D0)
DOUBLE PRECISION THREE
PARAMETER (THREE = 3.D0)
DOUBLE PRECISION FOUR
PARAMETER (FOUR = 4.D0)
DOUBLE PRECISION FIVE
PARAMETER (FIVE = 5.D0)
DOUBLE PRECISION SIX
PARAMETER (SIX = 6.D0)

C ******************************************************************************
C  * FRACTION CONSTANTS
C ******************************************************************************
DOUBLE PRECISION QUARTER
PARAMETER (QUARTER = 0.25D0)
DOUBLE PRECISION HALF
PARAMETER (HALF = 0.5D0)

A.3.4 PHASESPACE.FOR

******************************************************************************
* Author: Paul Burkhardt
* Date: September 28, 2004
* Include File: PHASE.FOR
* Purpose: Include file for integration variables.
******************************************************************************
INTEGER DOF
PARAMETER (DOF = 4)
INTEGER PHASEDIM
PARAMETER (PHASEDIM = 2*DOF)
DOUBLE PRECISION
$PHASEPT(PHASEDIM),$
$PHASEFL(PHASEDIM)$

DOUBLE PRECISION
$QVAR(DOF),$
$PVAR(DOF)$

INTEGER
$Q1, Q2, Q3, Q4,$
$P1, P2, P3, P4$

PARAMETER (
$Q1 = 1,\$
$Q2 = 2,\$
$Q3 = 3,\$
$Q4 = 4,\$
$P1 = 5,\$
$P2 = 6,\$
$P3 = 7,\$
$P4 = 8 )$
Appendix B

Hyperspherical Coordinates Programs

B.1 Symmetrized Hyperspherical Coordinates Programs

B.1.1 COMT

Transform initial Cartesian coordinates to Center-of-Mass (COM) coordinates.

<table>
<thead>
<tr>
<th>ARGUMENTS</th>
<th>I/O</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MASS</td>
<td>IN</td>
<td>REAL8</td>
<td>Mass values of the particles. Array of dimension, NBODY.</td>
</tr>
<tr>
<td>MTOT</td>
<td>IN</td>
<td>REAL8</td>
<td>Total mass.</td>
</tr>
<tr>
<td>JZ</td>
<td>IN</td>
<td>REAL8</td>
<td>Total angular momentum Z-component.</td>
</tr>
<tr>
<td>ECOL</td>
<td>IN</td>
<td>REAL8</td>
<td>Mechanical energy of collision for center of masses.</td>
</tr>
<tr>
<td>RDIST</td>
<td>IN</td>
<td>REAL8</td>
<td>Initial separation distance between coupled particles.</td>
</tr>
<tr>
<td>ACOORD</td>
<td>IN</td>
<td>REAL8</td>
<td>Initial position vector for impact particle in Cartesian coordinates. Array of dimension, NSPACE.</td>
</tr>
<tr>
<td>ROTA</td>
<td>IN</td>
<td>REAL8</td>
<td>Initial orientation angles in radians, for coupled particles. Array of dimension, NSPACE.</td>
</tr>
<tr>
<td>X</td>
<td>OUT</td>
<td>REAL8</td>
<td>The Cartesian X-axis component of the position for each particle. Array of dimension, NBODY.</td>
</tr>
<tr>
<td>Y</td>
<td>OUT</td>
<td>REAL8</td>
<td>The Cartesian Y-axis component of the position for each particle. Array of dimension, NBODY.</td>
</tr>
</tbody>
</table>
SUBROUTINE COM(MASS, MTOT, JZ, ECOL, RDIST, ACOORD, ROTA, VX, VY, VZ)
IMPLICIT NONE
INCLUDE 'SPATIAL_DIM.FOR'
INCLUDE 'NUMCONF.FOR'
INCLUDE 'JACOBI_CONF.FOR'
INCLUDE 'CARTDEF.FOR'
INTEGER I1
DOUBLE PRECISION XCOM, YCOM, ZCOM,
DOUBLE PRECISION MASS(NBODY), MTOT,
DOUBLE PRECISION MRATIO
DOUBLE PRECISION ECOL
DOUBLE PRECISION ACOORD(NSPACE), ROTA(NSPACE),
DOUBLE PRECISION RDIST
DOUBLE PRECISION JZ, WZ, IZZ
DOUBLE PRECISION SINA, SINB, SINC,
DOUBLE PRECISION COSA, COSB, COSC
INTEGER MA, MB, NA, NB
PARAMETER (MA = 3, NA = 3, MB = 3, NB = 2)
DOUBLE PRECISION BROT(MA, NA), Q(MB, NB), C(MA, NB)
INTEGER TRANSAN, TRANSBN
PARAMETER (TRANSAN = 'N', TRANSBN = 'N')

C ***************************************************************
C Determine starting position and velocity coordinates.
C
C The center-of-mass of the coupled particles is arbitrarily set
C at the origin. The incident particle always travels in the
C positive y direction by convention.
C
C Let X = x3 - x2, so x2 = x3 - X, where X is the maximum
C ground-state stretch of the diatomic.
C
C Set the Center-of-Mass (COM) at the origin so m2x2 + m3x3 = 0.
C Substituting in x3 for the COM equation, yields:
C m2(x3 - X) + m3x3 = (m2 + m3)x3 - m2X = 0
C so, x3 = (m2/(m2 + m3))X
C
C Also from the COM equation, x2 = -(m3/m2)x3
C
C Let k = 1, j = 3, i = 2
C***************************************************************

C Initial positions.
C
RDIST is the inter-particle distance for the BC pair.
\[
X(J) = \left( \frac{\text{MASS}(I)}{\text{MASS}(I) + \text{MASS}(J)} \right) \times \text{RDIST}
\]

\[
Y(J) = 0
\]

\[
Z(J) = 0
\]

MRATIO = \(-\frac{\text{MASS}(J)}{\text{MASS}(I)}\)

\[
X(I) = X(J) \times \text{MRATIO}
\]

\[
Y(I) = 0
\]

\[
Z(I) = 0
\]

**Position vector for particle A read in as an input parameter.**

\[
X(K) = \text{ACOORD}(1)
\]

\[
Y(K) = \text{ACOORD}(2)
\]

\[
Z(K) = \text{ACOORD}(3)
\]

**Initial orientation angles read in as input parameters.**

\[A = \text{ROTA}(1) = \text{ALPHA}\]

\[B = \text{ROTA}(2) = \text{BETA}\]

\[C = \text{ROTA}(3) = \text{GAMMA}\]

**Compute trigonometric functions of the orientation angles once.**

\[
\sin A = \sin(\text{ROTA}(1))
\]

\[
\sin B = \sin(\text{ROTA}(2))
\]

\[
\sin C = \sin(\text{ROTA}(3))
\]

\[
\cos A = \cos(\text{ROTA}(1))
\]

\[
\cos B = \cos(\text{ROTA}(2))
\]

\[
\cos C = \cos(\text{ROTA}(3))
\]

**Y-convention Euler rotation matrix of transformation.**

\[
\text{EROT}(1, 1) = -\sin C \times \sin A + \cos B \times \cos C \times \cos A
\]

\[
\text{EROT}(2, 1) = -\cos C \times \sin A - \cos B \times \cos C \times \sin A
\]

\[
\text{EROT}(3, 1) = \sin B \times \cos A
\]

\[
\text{EROT}(1, 2) = \sin C \times \cos A + \cos B \times \cos C \times \sin A
\]

\[
\text{EROT}(2, 2) = \cos C \times \cos A
\]

\[
\text{EROT}(3, 2) = \sin B \times \sin A
\]

\[
\text{EROT}(1, 3) = \cos C \times \sin B
\]

\[
\text{EROT}(2, 3) = \cos C \times \cos B
\]

\[
\text{EROT}(3, 3) = \cos B
\]

**Store the position vectors of particles 2 and 3 in matrix form for more compact computation.**

\[
Q(1, 1) = X(I)
\]

\[
Q(2, 1) = Y(I)
\]

\[
Q(3, 1) = Z(I)
\]

\[
Q(1, 2) = X(J)
\]

\[
Q(2, 2) = Y(J)
\]

\[
Q(3, 2) = Z(J)
\]

**Transform the position coordinates of the particle pair to body-fixed coordinates using Euler rotation matrix.**

**Multiplies two matrices: A(MA,NA)xB(MB,NB) = C(MA,NB)**

**The resulting matrix is returned.**

CALL DGE MM(TRANSA, TRANSB, MA, NA, NB, NA, ONE, EROT, \$ MA, Q, MB, ZERO, C, MA)

**New body-fixed position coordinates.**

\[
X(I) = C(1, 1)
\]

\[
Y(I) = C(2, 1)
\]

\[
Z(I) = C(3, 1)
\]

\[
X(J) = C(1, 2)
\]

\[
Y(J) = C(2, 2)
\]

\[
Z(J) = C(3, 2)
\]

******************************************************************************

**Particles B and C have initial velocities determined from the**

* rotational angular momentum.

**Particle A needs initial velocity in y-direction only.**

**Read in center-of-mass collision kinetic energy, KE. This energy**

* is the sum of the kinetic energies of each collision body in

**center-of-mass coordinates. It does not depend on the**

**independent motion of each individual particle.**

******************************************************************************
Moment of Inertia around the z-axis for the particle pair.

\[ I_{ZZ} = \text{MASS}(I) \times (X(I) \times X(I) + Y(I) \times Y(I)) \]
\[ + \text{MASS}(J) \times (X(J) \times X(J) + Y(J) \times Y(J)) \]

Z-component of the angular velocity.

\[ W_Z = \frac{J_Z}{I_{ZZ}} \]

Initial velocities.

\[ V_X(J) = -Y(J) \times W_Z \]
\[ V_Y(J) = X(J) \times W_Z \]
\[ V_Z(J) = 0 \]
\[ V_X(I) = V_X(J) \times \text{MRATIO} \]
\[ V_Y(I) = V_Y(J) \times \text{MRATIO} \]
\[ V_Z(I) = 0 \]
\[ V_X(K) = \text{DSQRT} \left( \frac{2 \times \text{MTOT} \times \text{ECOL}}{\text{MASS}(K) \times (\text{MASS}(I) + \text{MASS}(J))} \right) \]
\[ V_Y(K) = 0 \]

Transform to center of mass by shifting the origin.

\[ X_{COM} = \frac{\text{MASS}(I) \times X(I) + \text{MASS}(J) \times X(J) + \text{MASS}(K) \times X(K)}{\text{MTOT}} \]
\[ Y_{COM} = \frac{\text{MASS}(I) \times Y(I) + \text{MASS}(J) \times Y(J) + \text{MASS}(K) \times Y(K)}{\text{MTOT}} \]
\[ Z_{COM} = \frac{\text{MASS}(I) \times Z(I) + \text{MASS}(J) \times Z(J) + \text{MASS}(K) \times Z(K)}{\text{MTOT}} \]
\[ V_{XCOM} = \frac{\text{MASS}(I) \times V_X(I) + \text{MASS}(J) \times V_X(J) + \text{MASS}(K) \times V_X(K)}{\text{MTOT}} \]
\[ V_{YCOM} = \frac{\text{MASS}(I) \times V_Y(I) + \text{MASS}(J) \times V_Y(J) + \text{MASS}(K) \times V_Y(K)}{\text{MTOT}} \]
\[ V_{ZCOM} = \frac{\text{MASS}(I) \times V_Z(I) + \text{MASS}(J) \times V_Z(J) + \text{MASS}(K) \times V_Z(K)}{\text{MTOT}} \]

Shift all coordinates to be relative to the Center-of-Mass.

Do 10, 11 = 1, NICEY
\[ X(11) = X(11) - X_{COM} \]
\[ Y(11) = Y(11) - Y_{COM} \]
\[ Z(11) = Z(11) - Z_{COM} \]
\[ V_X(11) = V_X(11) - V_{XCOM} \]
\[ V_Y(11) = V_Y(11) - V_{YCOM} \]
\[ V_Z(11) = V_Z(11) - V_{ZCOM} \]
10 CONTINUE
RETURN
END

B.1.2 DERIV

Calculate the equations of motion for the planar Hamiltonian.
SUBROUTINE DERIV(T, PHASEPT, PHASEFL)

IMPLICIT NONE

INCLUDE 'NUMCON.FOR'
INCLUDE 'SIPREFIX.FOR'
INCLUDE 'MASS.FOR'
INCLUDE 'HYPDEF.FOR'
INCLUDE 'ENUMHYP.FOR'
INCLUDE 'PHASESPACE.FOR'

INTEGER NUM
PARAMETER (NUM = 3)

DOUBLE PRECISION T

DOUBLE PRECISION $ C1, C2, C3, C4, C5$

DOUBLE PRECISION $ SINTHETA, COSTHETA, COTTHETA, IZ$

DOUBLE PRECISION $ DVDN(NUM)$

Integration variable assignment.
The variables could be aliased to avoid run-time assignment.

RHO = PHASEPT(Q1)
THETA = PHASEPT(Q2)
PHI = PHASEPT(Q3)

PRHO = PHASEPT(P1)
PTHETA = PHASEPT(P2)
PPHI = PHASEPT(P3)
PGAMMA = PHASEPT(P4)

This momentum is a constant of the motion and is equivalent to
the total angular momentum: PGAMMA = JTOT

Masses, MTOT, MU are in mass units and E2, E3 are unitless.
DE'S are in energy units.
Angular momentum is in MASS*DISTANCE^2/TIME

CALL FORCE(PHASEPT, DVDN)

***************************************************************

Hamiltonian Equations of Motion

There are twelve position and momentum variables that make up
H(1, ..., 12) = H(rho, theta, phi, alpha, beta, gamma)

Their conjugate momentum, Palpha, Pbeta, are constants of the
motion and do not need to be integrated.

Only eight of the twelve Hyperspherical position and momentum
variables need be solved for the integration:
H(rho, theta, phi, gamma, Prho, Ptheta, Pphi, PGamma)

The alpha angle is inherently coupled with the gamma angle and
can be integrated separately, but will be included in the
integration with the rest of the variables for convenience.

In the planar case the external angle, gamma, is cyclic so the
phase space reduces to six dimensions:
H(rho, theta, phi, Prho, Ptheta, Pphi)

Coefficients and constants for substitution into equations
of motion.

***************************************************************
SINTHETA = DSIN(THETA)
COSTHETA = DCOS(THETA)
COTTHETA = COSTHETA/SINTHETA
IZ = MU/RHO/RHO

C1 = ONE/( 1Z+SINTHETA+SINTHETA )
C2 = ONE/( 1Z+COSTHETA+COSTHETA )
C3 = FOUR*PPHI+C1
C4 = FOUR*PTHETA*PTHETA/( RHO*IZ ) + PPHI+C3/RHO

C Calculate equations of motion for planar Hamiltonian.
C
RIKDT = PHRHO/MU;
THETADT = FOUR*PTHETA/IZ
PHRHD = -DVDN(PHII)

C Check for special case of zero angular momentum.
IF (P GAMMA .LT. PICO) THEN
   PHIDT = C3
   GAMMADT = -HALF*C3*COSTHETA
   PTHETADT = C3*COSTHETA + C1
   PHRHD = C4 + C3/RHO - DVDN(RHOO)
ENDIF

C Integration variable assignment.
The variables could be aliased to avoid run-time assignment.
PHASEFL(Q1_) = RHODT
PHASEFL(Q2_) = THETADT
PHASEFL(Q3_) = PHIDT
PHASEFL(Q4_) = GAMMADT

PHASEFL(P1_) = PHRHD
PHASEFL(P2_) = PTHETADT
PHASEFL(P3_) = PPHIDT
PHASEFL(P4_) = GAMMADT

RETURN

END

B.1.3 FORCE

Evaluate the force. Determine first derivatives of the potential energy.
PHASEPT IN REAL8 Independent variables of integration. Hyperspherical phase space coordinates. Array of dimension, 2xDOF.

DVIN OUT REAL8 Force gradients. Derivatives of the potential with respect to the internal position coordinates. Array of dimension, NUM.

SUBROUTINE FORCE(PHASEPT, DVIN)
IMPLICIT NONE
INCLUDE 'NUMCON.FOR'
INCLUDE 'MASS.FOR'
INCLUDE 'JACOBI_CONF.FOR'
INCLUDE 'ENUM_HYP.FOR'
INCLUDE 'PHASESPACE.FOR'

INTEGER I1, NUM
PARAMETER (NUM = 3)

DOUBLE PRECISION $ SINTHETA, COSTHETA,
$ SINPHI(NUM), COSPHI(NUM)

DOUBLE PRECISION $ RDIST(NUM), DVDR(NUM)

DOUBLE PRECISION $ DVDN(NUM)

DOUBLE PRECISION $ RHO, RHO2, INVRHO, COEFF

C Care must to taken in labelling the interatomic distances and corresponding potential parameters.
C R(1) = R(BC) = H-H distance
C R(2) = R(AC) = F-H2 distance
C R(3) = R(AB) = F-H1 distance

SINTHETA = DSIN( PHASEPT(THETA)
COSTHETA = DCOS( PHASEPT(THETA)

SINPHI(K) = DSIN( PHASEPT(PHI(K))
SINPHI(I) = DSIN( PHASEPT(PHI(I)) - E2 )
SINPHI(J) = DSIN( PHASEPT(PHI(J)) + E3 )

COSPHI(K) = DCOS( PHASEPT(PHI(K))
COSPHI(I) = DCOS( PHASEPT(PHI(I)) - E2 )
COSPHI(J) = DCOS( PHASEPT(PHI(J)) + E3 )

C RDIST(1) = D(K)*RHO*DSQRT( HALF*( ONE + SINTHETA*SINPHI(K) ) )
C RDIST(2) = D(I)*RHO*DSQRT( HALF*( ONE + SINTHETA*SINPHI(I) ) )
C RDIST(3) = D(J)*RHO*DSQRT( HALF*( ONE + SINTHETA*SINPHI(J) ) )

RHO = PHASEPT(RHO)
DO 10, I1 = 1, NUM
RDIST(I1) = D(I1)*RHO
$ * DSQRT( HALF*( ONE + SINTHETA*SINPHI(I1) ) )
10 CONTINUE

C Get the derivatives of the potential with respect to the three interatomic distances: RAB, RBC, RAC
C Return dvdr in units of MASS*DISTANCE^2/TIME^2
CALL PESDERIV(RDIST, DVDR)

C Calculate derivatives of the potential in hyperspherical coordinates with respect to the three internal coordinates:
C rho, theta, phi.

INVRHO = ONE/RHO
DVIN(RHO) = ZERO
DVIN(THETA) = ZERO
DVIN(PHI) = ZERO

DO 20, I1 = 1, NUM
\[
\text{COEFF} = \text{DVDR}(I1) \times \text{D}(I1) \times \text{D}(I1) \times \frac{\text{RHO2}}{\text{FOUR} \times \text{RDIST}(I1)}
\]

\[
\text{DVDN(RHO)} = \text{DVDN(RHO)} + \text{DVDR}(I1) \times \text{RDIST}(I1) \times \text{INVRHO}
\]

\[
\text{DVDN(THETA)} = \text{DVDN(THETA)} + \text{COSTHETA} \times \text{SINPHI}(I1) \times \text{COEFF}
\]

\[
\text{DVDN(PHI)} = \text{DVDN(PHI)} + \text{SINTHETA} \times \text{COSPHI}(I1) \times \text{COEFF}
\]

\[
B.1.4 \text{ GETENERGY}
\]

Calculate the total energy from the Hamiltonian.

---

**Author:** Paul Burkhardt  
**Date:** September 28, 2004  
**Program:** GETENERGY  
**Calculate the Hamiltonian energy function for three particles in  
Johnson symmetrized, hyperspherical coordinates.**  
Energy = \( H = KE + PE \)

**ARGUMENTS**  
I / O  
TYPE  
DESCRIPTION

<table>
<thead>
<tr>
<th>PHASEPT</th>
<th>IN</th>
<th>REAL8</th>
<th>Independent variables of integration. Hyperspherical phase space coordinates. Array of dimension, 2xDOF.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENERGY</td>
<td>OUT</td>
<td>REAL8</td>
<td>Hamiltonian total system energy.</td>
</tr>
</tbody>
</table>

**SUBROUTINE** GETENERGY(PHASEPT, ENERGY)  
**IMPLICIT** NONE

```
INCLUDE 'NUMCON.FOR'  
INCLUDE 'PHASESPACE.FOR'  
INCLUDE 'MASS.FOR'  
INCLUDE 'PESCOMMON.FOR'
```

**DOUBLE PRECISION**  
$\$ ENERGY, KE

**DOUBLE PRECISION**  
$\$ SINANG, COSANG, INVIZ

**DOUBLE PRECISION**  
$\$ C1, C2

```
C \text{ Calculate kinetic energy.}  
\text{SINANG} = \text{DSIN(PHASEPT(2))}  
\text{COSANG} = \text{DCOS(PHASEPT(2))}  
\text{C1} = \text{PHASEPT(8)+PHASEPT(8)}  
\text{C2} = \text{ONE/(SINANG+SINANG)}  
\text{INVIZ} = \text{ONE/(MU+PHASEPT(1)+PHASEPT(1))}
```

\[
\text{KE} = \frac{1}{2} \times \frac{1}{\text{MU}} \times \text{PHASEPT(5)+PHASEPT(5)}
\]

\[
\$ + \frac{1}{\text{INVIZ}}
\]

\[
\$ + \left( \frac{1}{\text{PHASEPT(6)}+\text{PHASEPT(6)}} + \frac{\text{PHASEPT(7)+PHASEPT(7)}}{\text{C2}} \right)
\]

\[
\$ + \left( \frac{\text{HALF}+\text{C1} - \text{TWO}+\text{PHASEPT(7)+PHASEPT(8)}+\text{COSANG}+\text{INVIZ}+\text{C2}}{\text{C1}} \right)
\]

\[
\$ + \left( \frac{\text{ONE}+\text{SINANG}+\text{DCOS(TWO)+PHASEPT(4)}}{\text{INVIZ}+(\text{COSANG+COSANG})} \right)
\]

\[
\text{ENERGY} = \text{KE} + \text{PE}
\]
Transform principal axes, center-of-mass Cartesian coordinates to Smith-Whitten-Johnson symmetrized, hyperspherical polar coordinates.

---

**Author**: Paul Burkhardt  
**Date**: September 28, 2004  
**Program**: HYPT  
**Hyperspherical coordinates transformation.**  
Transform Principal Axes of Inertia, Center-of-Mass Cartesian coordinates to Johnson symmetrized, hyperspherical coordinates.  
Returns an array of the hyperspherical coordinates and momenta independent variables of integration used to solve the canonical equations of motion.

**ARGUMENTS**  
**I/O** | **TYPE** | **DESCRIPTION**
--- | --- | ---
MU | IN REAL8 | Reduced mass for three particles.
JTOT | IN REAL8 | Total angular momentum.
EULER | IN REAL8 | External orientation angles. Describes the relation between the stationary and the rotating frames. Equivalent to Euler angles for Space-to-Body transformations. Array of dimension, NSPACE.
RJ1 | IN REAL8 | Jacobi position vector for coupled particles. Describes the "small" inter-particle distance. Array of dimension, NSPACE.
RJ2 | IN REAL8 | Jacobi position vector for impact particle. Describes the "big" inter-particle distance. Array of dimension, NSPACE.
RJ1DT | IN REAL8 | Jacobi velocity vector for coupled particles. Describes the "small" inter-particle separation. Array of dimension, NSPACE.
RJ2DT | IN REAL8 | Jacobi velocity vector for impact particle. Describes the "big" inter-particle separation. Array of dimension, NSPACE.
YN | OUT REAL8 | Phase space variables in Hyperspherical coordinates. Position and momenta independent variables of integration. Array of dimension, NVAR.

---

**SUBROUTINE** HYPT(MU, JTOT, EULER, RJ1, RJ2, RJ1DT, RJ2DT, YN)

**IMPLICIT** NONE

**INCLUDE**
- 'SPATIAL_DIM.FOR'
- 'NUMCON.FOR'
- 'PHYSCON.FOR'
- 'JACOBI_CONF.FOR'
- 'JACDEF.FOR'
- 'ENUMXYZ.FOR'
- 'HYPERF.FOR'
- 'ENUMHYP.FOR'

**CHARACTER** APPNAM(*)

174
PARAMETER (APPNAM = 'h hyp t . f ' )

INTEGER $ I 1 , S T A T U S$

INTEGER DF

PARAMETER (DF = 6)

DOUBLE PRECISION $ I , S T A T U S$

DOUBLE PRECISION $ M U , J T O T , E U L E R ( N S P A C E )$

DOUBLE PRECISION $ Q D T ( D F ) , Q P$

DOUBLE PRECISION $ R H O , I N V R H O$

DOUBLE PRECISION $ S I N A , S I N B , S I N C$

DOUBLE PRECISION $ C O S A , C O S B , C O S C$

DOUBLE PRECISION $ S I N T H E T A , C O S T H E T A , T W O T H E T A$

DOUBLE PRECISION $ S S , S C$

INTEGER $ N L E , N R H S , L D A , L D B$

PARAMETER (NLE = 6, NRHS = 1, LDA = 6, LDB = 6)

INTEGER $ I P I V ( N L E )$

DOUBLE PRECISION $ A A ( L D A , N L E ) , B B ( L D B )$

DOUBLE PRECISION $ H A L F P I , T W O P I$

PARAMETER (HALFP = HALF*PI, TWOPI = TWO*PI)

***************************************************************************************************************
Hyperspherical Position Coordinates
Coordinates determined from the Smith-Whitten Jacobi vector definitions in principal axes with the Johnson modifications, and the external rotation angles that relate the stationary frame to the rotating principal axes of inertia, frame.

Q(1, 2, 3, 4, 5, 6) = Q( rho, theta, phi, alpha, beta, gamma)

***************************************************************************************************************

Hyperradius vector magnitude.
The hyerradius defines the size of the triangle formed from the three particles. The hyerradius magnitude is determined from the square root of the sum of the squares of the Jacobi vector components.

RHO = ZERO
DO 15, I1 = 1, NSPACE
RHO2 = RHO2 + RJ1(I1)*RJ1(I1) + RJ2(I1)*RJ2(I1)
CONTINUE
RHO = DSQRT(RHO2)
INVRHO = ONE/RHO

C Hyerradius independent variable of integration.
YN[RHO2] = RHO

C Hyperangle, "theta".
The hyperangle as modified by Johnson: 0 <= theta <= PI/2
The sine and cosine of this angle will always produce a positive result.
The arctangent has a unique solution in the -PI/2 to PI/2 range.
The Principal Axes of Inertia as defined by Smith and Whitten, require a positive hyperangle, theta, in the first quadrant with the range, 0 to PI/4.
Twice the hyperangle, \( \theta \), has a unique solution between 0 and \( PI / 2 \):

\[ 0 < \theta < PI / 2 \]

The result of the \( \text{ATAN}() \) function produces an angle in the range \(-PI/2 \) to \( PI/2 \).

\[ \text{TWOTheta} = 2 \times \text{DABS}( \text{DATAN2}(X, Y) ) \]

\[ \text{QP} = X1 \times X1 + X2 \times X2 - Y1 \times Y1 - Y2 \times Y2 \]

The result of the \( \text{ACOS}() \) function produces an angle in the range 0 to \( PI \).

\[ \text{TWOTheta} = \text{DACOS}( \text{QP/RHO2} ) \]

Ensure result lies in proper quadrant: 0 to \( PI / 2 \)

\[ \text{IF} (\text{TWOTheta} > \text{HALFPI}) \text{ THEN} \]

\[ \text{TWOTheta} = \text{PI} - \text{TWOTheta} \]

\[ \text{END IF} \]

Johnson modified hyperangle, \( \theta \)

\[ \text{YN} (\text{TWOTheta}) = \text{HALFPI} - \text{TWOTheta} \]

Hyperangle, "phi".

The arctangent has a unique solution in the \(-PI/2 \) to \( PI/2 \) range.

The Principal Axes of Inertia as defined by Smith and Whitten,

require a positive hyperangle, \( \phi \), with the range, 0 to 2\( PI \).

There are then two solutions for the hyperangle, \( \phi \), one in the 0 to \( PI \) range, and the other from \( PI \) to 2\( PI \). This correlates with the solutions for the Euler angle, \( \gamma \). It was arbitrarily chosen to work with the principal axes of inertia associated with the gamma angle whose solution lies in the 0 to \( PI \) range.

Similarly, the hyperangle, \( \phi \), must also be associated with this same principal axes. Substituting the hyperangles and hyperradius into the Jacobi vector definitions will determine the appropriate hyperangle, \( \phi \).

NOTE:

The \( \text{ATAN2}() \) intrinsic function returns a result in the range \(-PI \) to \( PI \). Shift the \( \text{ATAN2}() \) result and use the modulus to normalize the solution.

Require positive angle, all quadrants.

Full range: 0 <= \( \phi \) <= 2\( PI \)

\[ \text{PHI} = \text{DMOD}( \text{DATAN2}(X, Y) + \text{TWOPI}, \text{TWOPI} ) \]

Johnson modified hyperangle, \( \phi \)

\[ \text{YN} (\text{PHI}) = \text{HALFPI} - \text{TWO} \times \text{PHI} \]

Hyperspherical external angles, "alpha", "beta", "gamma".

The three angles are identical to the external Euler angles of the transformation that describe the orientation of the stationary frame to the rotating, principal axes of inertia frame.

Alpha angle.

\[ \text{YN} (\text{ALPHA}) = \text{EULER}(1) \]

Beta angle.

\[ \text{YN} (\text{BETA}) = \text{EULER}(2) \]

Gamma angle.

\[ \text{YN} (\text{GAMMA}) = \text{EULER}(3) \]

**************************************************************************************************

**Hyperspherical Position Velocities**

The coordinate time derivatives are determined from the linear transformation relation between the Jacobi vector component.

time derivatives with the hyperspherical coordinate velocities.

\[ \text{QDT}(1, 2, 3, 4, 5, 6) = \]

\[ \text{QDT}(\text{RHODT}, \text{THETADT}, \text{PHIDT}, \text{ALPHADT}, \text{BETADT}, \text{GAMMADT}) \]

**************************************************************************************************

Euler rotation angles used for principal axes of inertia
Transformation

\[ C = \text{EULER}(1) = \alpha \]
\[ B = \text{EULER}(2) = \beta \]
\[ \gamma = \text{EULER}(3) = \gamma \]

Compute trigonometric functions of the orientation angles once.

\[ \sin \alpha = \sin(\text{EULER}(1)) \]
\[ \sin \beta = \sin(\text{EULER}(2)) \]
\[ \sin \gamma = \sin(\text{EULER}(3)) \]
\[ \cos \alpha = \cos(\text{EULER}(1)) \]
\[ \cos \beta = \cos(\text{EULER}(2)) \]
\[ \cos \gamma = \cos(\text{EULER}(3)) \]

\[ SS = \sin \beta \times \sin \gamma \]
\[ SC = \sin \beta \times \cos \gamma \]

Determine the hyperspherical coordinate time derivatives by a linear transformation. There are six linearly independent equations with variables in \(\rho dt, \theta dt, \phi dt, \alpha dt, \beta dt,\) and \(\gamma dt\).

Transformation relation for Jacobi velocities and hyperspherical velocities. Transformation matrix is a system of simultaneous linear algebraic equations (SLAEs).

\[
\begin{align*}
AA(1,1) &= RJ1(X) \times \text{INVRHO} \\
AA(2,1) &= RJ1(Y) \times \text{INVRHO} \\
AA(3,1) &= \text{ZERO} \\
AA(4,1) &= RJ2(X) \times \text{INVRHO} \\
AA(5,1) &= RJ2(Y) \times \text{INVRHO} \\
AA(6,1) &= \text{ZERO} \\

AA(1,2) &= -RJ2(Y) \\
AA(2,2) &= -RJ2(X) \\
AA(3,2) &= \text{ZERO} \\
AA(4,2) &= RJ1(Y) \\
AA(5,2) &= RJ1(X) \\
AA(6,2) &= \text{ZERO} \\

AA(1,3) &= -RJ2(Z) \\
AA(2,3) &= -RJ2(Y) \\
AA(3,3) &= \text{ZERO} \\
AA(4,3) &= RJ1(Z) \\
AA(5,3) &= RJ1(X) \\
AA(6,3) &= \text{ZERO} \\

AA(1,4) &= RJ1(Y) \times \cos \beta - RJ1(Z) \times \sin \beta \\
AA(2,4) &= -(RJ1(Z) \times \sin \gamma + RJ1(X) \times \cos \gamma ) \\
AA(3,4) &= RJ1(Y) \times \cos \gamma + RJ1(X) \times \sin \gamma \\
AA(4,4) &= RJ2(Y) \times \cos \beta - RJ2(Z) \times \sin \beta \\
AA(5,4) &= -(RJ2(Z) \times \sin \gamma + RJ2(X) \times \cos \gamma ) \\
AA(6,4) &= RJ2(Y) \times \cos \gamma + RJ2(X) \times \sin \gamma \\

AA(1,5) &= -RJ1(Z) \times \cos \gamma \\
AA(2,5) &= RJ1(Z) \times \sin \gamma \\
AA(3,5) &= RJ1(X) \times \cos \gamma - RJ1(Y) \times \sin \gamma \\
AA(4,5) &= RJ2(Z) \times \cos \gamma \\
AA(5,5) &= RJ2(Z) \times \sin \gamma \\
AA(6,5) &= RJ2(X) \times \cos \gamma - RJ2(Y) \times \sin \gamma \\

AA(1,6) &= RJ1(Y) \\
AA(2,6) &= -RJ1(X) \\
AA(3,6) &= \text{ZERO} \\
AA(4,6) &= RJ2(Y) \\
AA(5,6) &= -RJ2(X) \\
AA(6,6) &= \text{ZERO} \\
\end{align*}
\]

Jacobi coordinate velocities as the linearly dependent variables.

DO 20, I1 = 1, NSPACE
BB(I1) = RJ1DT(I1)
BB(I1+NSPACE) = RJ2DT(I1)
20 CONTINUE

Solve the linear system of equations: \(A\times X = B\).

Solutions for independent variables, \(X\), are returned in the matrix, \(B\).

CALL DGESV(NLE, NRHS, AA, LDA, IPIV, BB, LDB, STATUS)

IF (STATUS .NE. 0) THEN
WRITE(UNIT=AUNIT, FMT=('(/, A, /, A, 110)'))
$ 'UNEXPECTED RETURN FROM DGESV IN '//'APPNAM///' ',
$ 'Return status value: ',
177
```plaintext
$ STATUS
  IF (STATUS .LT. 0) THEN
    WRITE (UNIT=*, FMT='(X,A,110,A)')
    'The ',
    STATUS
  ELSE IF (STATUS .GT. 0) THEN
    WRITE (UNIT=*, FMT='(2(X,A/ ) )')
    'The factor, U, is exactly zero ,'
    'The factorization has been completed , but the '//'
  'solution can not be computed due to a singularity .'
END IF
END IF

C Hyperspherical velocity coordinates in Smith configuration.
DO 30 , I1 = 1 , DF
  QDT(I1) = BB(I1)
CONTINUE
C Convert the hyperangle velocities to Johnson configuration.
QDT(THETA) = -TWO*QDT(THETA)
QDT(PHI) = -TWO*QDT(PHI)

C Hyperspherical Momenta
C
C The momenta are determined from the equations of motion.
C The independent position and velocity variables are required.
C Use Johnson-modified hyperangles.
C
C P(1,2,3,4,5,6) = P(PRHO,PTHETA,PPHI,PALPHA,PBETA,PGAMMA)

PALPHA = JTOT
SINTHETA = DSIN( YN(THETA) )
COSTHETA = DCOS( YN(THETA) )
C Hyperradius momentum.
YN(PRHO) = M*QDT(RHO)

C Hyperspherical angle, "theta", momentum.
YN(PTHETA) = M*RHO2*QDT(THETA)+QUARTER
C Constant of the motion, equivalent to the total angular momentum.
YN(PALPHA) = PALPHA
C
C Momentum of beta angle.
C Constant of the motion
YN(PBETA) = ZERO
C
C Momentum of gamma angle.
YN(PGAMMA) = PALPHA*COSTHETA
C
C Hyperspherical angle, "phi", momentum.
YN(PPHI) = HALF+YN(PGAMMA)*COSTHETA
  + QUARTER*M*RHO2*SINTHETA*SINTHETA*QDT(PHI)
RETURN
END

B.1.6 INIT

Get the initial state variables to start the integration for solving the dynamics.

******************************************************************************
* Author : Paul Burkhardt
* Date: September 28, 2004
* Program: INIT
******************************************************************************

178
Gets the values for the initial state of the system.

<table>
<thead>
<tr>
<th>ARGUMENTS</th>
<th>I/O</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSTEP</td>
<td>OUT</td>
<td>REAL8</td>
<td>Integration step size.</td>
</tr>
<tr>
<td>MSTEP</td>
<td>OUT</td>
<td>INT4</td>
<td>Maximum number of integration steps.</td>
</tr>
<tr>
<td>PSTEP</td>
<td>OUT</td>
<td>INT4</td>
<td>Step interval for printing.</td>
</tr>
<tr>
<td>YN</td>
<td>OUT</td>
<td>REAL8</td>
<td>Phase space variables in Hyperspherical coordinates. Position and momenta independent variables of integration. Array of dimension, NVAR.</td>
</tr>
</tbody>
</table>

SUBROUTINE INIT(TSTEP, MSTEP, PSTEP, YN)

IMPLICIT NONE

INCLUDE 'SPATIAL_DIM.FOR'
INCLUDE 'NUMON.FOR'
INCLUDE 'SREP.FOR'
INCLUDE 'PHYSFOR.FOR'
INCLUDE 'JACOBICONF.FOR'
INCLUDE 'JACDEF.FOR'
INCLUDE 'CARTDEF.FOR'
INCLUDE 'HYPDEF.FOR'
INCLUDE 'MASS.FOR'

INTEGER $I1$
INTEGER $JQ$

DOUBLE PRECISION $JZ$, $ECOL$, $INVMU$, $INVMT$

DOUBLE PRECISION $ACORD(NSPACE)$, $AROTN(NSPACE)$, $RDIST$

INTEGER $MSTEP$, $PSTEP$

DOUBLE PRECISION $TSTEP$

PARAMETER (RADDEG = PI / 180.0D0)

INTEGER $IUNIT$

PARAMETER ($IUNIT = 10$)

CHARACTER $INFILE$(*)

PARAMETER ($INFILE = 'input.in')

OPEN(UNIT=$IUNIT$, FILE=$INFILE$)

READ(UNIT=$IUNIT$, FMT=1000) MASS
READ(UNIT=$IUNIT$, FMT=1100) ECOL, RDIST, ACOORD, AROTN
READ(UNIT=$IUNIT$, FMT=1200) MSTEP
READ(UNIT=$IUNIT$, FMT=1100) PSTEP
READ(UNIT=$IUNIT$, FMT=1000) TSTEP

179
**ATOMIC UNITS**

**MASS:** ELECTRON MASS (me)

**LENGTH:** BOHR

**ENERGY:** HARTREE

**TIME:** HBAR/HARTREE

---

MTOT = 0

DO 10 11 = 1, NBODY
  MTOT = MTOT + MASS(I1)
10 CONTINUE

INVMT = ONE/MTOT

DO 11 11 = 1, NSPACE
  AROTN(I1) = AROTN(I1) / RADDEG
11 CONTINUE

MU = DSQRT(MASS(I1) * MASS(J1) * MASS(K1) * INVMT)

INVMU = ONE/MU

E2 = TWO * DATAN(MASS(J1) * INVMU)

E3 = TWO * DATAN(MASS(I1) * INVMU)

JZ = DSQRT(JQ * (JQ + ONE)) * FEMTO/(EMS*A0*A0)

---

CALL COMT(MASS, MTOT, JZ, ECOL, RDIST, ACOORD, AROTN, X, Y, Z, VX, VY, VZ)

WRITE(UNIT=*, FMT='(3(3E24.16/))') X, Y, Z

WRITE(UNIT=*, FMT='(3(3E24.16/))') VX, VY, VZ

---

**Convert to Jacobi relative coordinates.**

Jacobi coordinates from Center-Of-Mass Cartesian coordinates, vectors have no z-components.

Following the Smith and Whitten labeling scheme, the Jacobi vectors are defined as kji = 132 configuration, so k = i, j = 3, i = 2.
\[ r_1 = (d_k - 1)(X_j - X_i) \]
\[ r_2 = d_k \left[ X_k - \frac{m_j X_j + m_i X_i}{m_j + m_i} \right] \]

---

CALL JACOBI(MASS, D(K), X, Y, Z, VX, VY, VZ, RJ1, RJ2, RJ1DT, RJ2DT)

CALL PAT(MU, RJ1, RJ2, RJ1DT, RJ2DT, X, Y, Z, VX, VY, VZ, EULER)

CALL JACOBI(MASS, D(K), X, Y, Z, VX, VY, VZ, RJ1, RJ2, RJ1DT, RJ2DT)

---

Total Angular Momentum

The total angular momentum vector is a constant of the motion with an invariant direction and magnitude. Arbitrarily align the space-fixed (stationary) Z-axis in the direction of the total angular momentum vector. Angular momentum; \( J = m(r \times v) \)

Total angular momentum; \( J_{\text{tot}} = \sum J_i \)

**JTOT = ZERO**

DO 20, I1 = 1, NBODY
  JTOT = JTOT + MASS(I1) * (X(I1) * VY(I1) - Y(I1) * VX(I1))
20 CONTINUE

IF (DABS(JTOT) .LT. PICO) THEN
  JTOT = ZERO
END IF

---

Calculate the instantaneous, symmetrized hyperspherical coordinates and velocities.

CALL HYPT(MU, JTOT, EULER, RJ1, RJ2, RJ1DT, RJ2DT, YN)

WRITE(UNIT=*, FMT='(4(3E24.16))') YN

RETURN
Transform Cartesian coordinates to mass-weighted Jacobi coordinates.

**ARGUMENTS**

<table>
<thead>
<tr>
<th>I/O</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MASS</td>
<td>IN</td>
<td>REAL8</td>
</tr>
<tr>
<td>DK</td>
<td>IN</td>
<td>REAL8</td>
</tr>
<tr>
<td>X</td>
<td>IN</td>
<td>REAL8</td>
</tr>
<tr>
<td>Y</td>
<td>IN</td>
<td>REAL8</td>
</tr>
<tr>
<td>Z</td>
<td>IN</td>
<td>REAL8</td>
</tr>
<tr>
<td>VX</td>
<td>IN</td>
<td>REAL8</td>
</tr>
<tr>
<td>VY</td>
<td>IN</td>
<td>REAL8</td>
</tr>
<tr>
<td>VZ</td>
<td>IN</td>
<td>REAL8</td>
</tr>
<tr>
<td>RJ1</td>
<td>OUT</td>
<td>REAL8</td>
</tr>
<tr>
<td>RJ2</td>
<td>OUT</td>
<td>REAL8</td>
</tr>
<tr>
<td>RJ1DT</td>
<td>OUT</td>
<td>REAL8</td>
</tr>
<tr>
<td>RJ2DT</td>
<td>OUT</td>
<td>REAL8</td>
</tr>
</tbody>
</table>

**SUBROUTINE** JACOBI(MASS, DK, X, Y, Z, VX, VY, VZ, RJ1, RJ2, RJ1DT, RJ2DT)

**IMPLICIT** NONE

**DOUBLE PRECISION**

include 'NUMCON.FOR'
include 'SPATIAL_DIM.FOR'
include 'JACOBI_CONF.FOR'
include 'ENUM_XYZ.FOR'
include 'CARTDEF.FOR'
include 'JACDEF.FOR'

C

182
The Jacobi vectors are defined from the relative separation of the three particles. One Jacobi vector is defined as the separation between two particles. The other vector is the separation from the center-of-mass of the coupled particle pair to the lone particle. There are then three possible representations of Jacobi vectors based on the choice of the particle coupling. One representation will be chosen and the labeling scheme will be adhered to.

The unitless mass coefficient is obtained from the relative coordinate transformation.

Cartesian coordinates are already in principal axes of inertia.

Jacobi position coordinates.

\[
\begin{align*}
RJ_1(X) &= (X(J) - X(I)) \cdot \text{INVDK} \\
RJ_1(Y) &= (Y(J) - Y(I)) \cdot \text{INVDK} \\
RJ_1(Z) &= (Z(J) - Z(I)) \cdot \text{INVDK} \\
RJ_2(X) &= DK \cdot (X(K) - (\text{MASS}(I) \cdot X(I) + \text{MASS}(J) \cdot X(J)) \cdot \text{INVM}) \\
RJ_2(Y) &= DK \cdot (Y(K) - (\text{MASS}(I) \cdot Y(I) + \text{MASS}(J) \cdot Y(J)) \cdot \text{INVM}) \\
RJ_2(Z) &= DK \cdot (Z(K) - (\text{MASS}(I) \cdot Z(I) + \text{MASS}(J) \cdot Z(J)) \cdot \text{INVM})
\end{align*}
\]

Jacobi velocity coordinates.

\[
\begin{align*}
RJ_{1DT}(X) &= (VX(J) - VX(I)) \cdot \text{INVDK} \\
RJ_{1DT}(Y) &= (VY(J) - VY(I)) \cdot \text{INVDK} \\
RJ_{1DT}(Z) &= (VZ(J) - VZ(I)) \cdot \text{INVDK} \\
RJ_{2DT}(X) &= DK \cdot (VX(K) - (\text{MASS}(I) \cdot VX(I) + \text{MASS}(J) \cdot VX(J)) \cdot \text{INVM}) \\
RJ_{2DT}(Y) &= DK \cdot (VY(K) - (\text{MASS}(I) \cdot VY(I) + \text{MASS}(J) \cdot VY(J)) \cdot \text{INVM}) \\
RJ_{2DT}(Z) &= DK \cdot (VZ(K) - (\text{MASS}(I) \cdot VZ(I) + \text{MASS}(J) \cdot VZ(J)) \cdot \text{INVM})
\end{align*}
\]

\[\text{RETURN}\]

\[\text{END}\]

B.1.8 PAT

Transform center-of-mass Cartesian coordinates to principal axes of inertia coordinates.
and phi.

Returns the Principal Axes of Inertia Cartesian coordinates and
the Euler rotation angles used to transform the original
coordinates.

ARGUMENTS I/O TYPE DESCRIPTION

MU IN REAL8 Reduced mass for three particles.

RJ1 IN/OUT REAL8 Jacobi position vector for coupled
particles. Describes the "small"
inter-particle distance.
Array of dimension, NSPACE.

RJ2 IN/OUT REAL8 Jacobi position vector for impact
particle. Describes the "big"
inter-particle distance.
Array of dimension, NSPACE.

RJ1DT IN/OUT REAL8 Jacobi velocity vector for coupled
particles. Describes the "small"
inter-particle separation.
Array of dimension, NSPACE.

RJ2DT IN/OUT REAL8 Jacobi velocity vector for impact
particle. Describes the "big"
inter-particle separation.
Array of dimension, NSPACE.

X IN/OUT REAL8 The Cartesian X-axis component of the
position for each particle.
Array of dimension, NBODY.

Y IN/OUT REAL8 The Cartesian Y-axis component of the
position for each particle.
Array of dimension, NBODY.

Z IN/OUT REAL8 The Cartesian Z-axis component of the
position for each particle.
Array of dimension, NBODY.

VX IN/OUT REAL8 The Cartesian X-axis component of the
velocity for each particle.
Array of dimension, NBODY.

VY IN/OUT REAL8 The Cartesian Y-axis component of the
velocity for each particle.
Array of dimension, NBODY.

VZ IN/OUT REAL8 The Cartesian Z-axis component of the
velocity for each particle.
Array of dimension, NBODY.

EULER OUT REAL8 External orientation angles.
Describes the relation between the
stationary and the rotating frames.
Equivalent to Euler angles for
Space-to-Body transformations.
Array of dimension, NSPACE.

SUBROUTINE PAT(MU, RJ1, RJ2, RJ1DT, RJ2DT, X, Y, Z, VX, VY, VZ,
$ EULER)

IMPLICIT NONE

INCLUDE 'SPATIAL_DIM.FOR'
INCLUDE 'NUMCON.FOR'
INCLUDE 'PHYSCON.FOR'
INCLUDE 'SIPREFIX.FOR'
INCLUDE 'ENUMXYZ.FOR'
INCLUDE 'CARTDEF.FOR'
INCLUDE 'JACDEF.FOR'

INTEGER I1, J1

DOUBLE PRECISION
$ MU

DOUBLE PRECISION
$ IX, IY, IZ, IXX, IXY, IZZ,
$ IXY, IXZ, IVZ

DOUBLE PRECISION
Components of area vector generated from the cross-product of the Jacobi vectors. The area vector is the normal vector to the triangle formed by the three particles. The definition of the area is derived from the well-known vector equation for the area of a parallelogram.

\[
\begin{align*}
AX &= \text{HALF} \cdot (RJ_1(Y) \cdot RJ_2(Z) - RJ_1(Z) \cdot RJ_2(Y)) \\
AY &= \text{HALF} \cdot (RJ_1(X) \cdot RJ_2(Z) - RJ_1(Z) \cdot RJ_2(X)) \\
AZ &= \text{HALF} \cdot (RJ_1(X) \cdot RJ_2(Y) - RJ_1(Y) \cdot RJ_2(X)) \\
\text{AREA} &= \text{DSQRT} (AX^2 + AY^2 + AZ^2)
\end{align*}
\]

WRITE( ' (X,A/4 E24.16/)' )
$AX,AY,AZ,AREA$

NOTE:
The ATAN(y/x) intrinsic function for the arctangent produces a result in only the 1st and 4th quadrants (-PI/2 TO PI/2).
The ATAN2(y, x) intrinsic function will return a solution in the -PI to PI range. Shift the ATAN2() result and use the modulus to normalize the solution over the desired 0 to 2PI range.
The ATAN() function also prevents divide by zero errors.
IF ( (AX .EQ. ZERO) .AND. (AY .EQ. ZERO) ) THEN
   ALPHA = ZERO
ELSE
   ALPHA = D3ED( DATAN2( AY, AX ) + TWOPi, TWOPi )
ENDIF

C The ACOS() intrinsic function returns the result of the
C arccosine in the desired range of 0 to PI.
BETA = DACOS( AZ/AREA )

C Transform to an intermediate coordinate system by rotating through
C alpha and beta to determine gamma.
SINALPHA = DSIN(ALPHA)
COSALPHA = DCOS(ALPHA)
SINBETA = DSIN(BETA)
COSBETA = DCOS(BETA)

C Rot (beta) * Rot (alpha)
ROT(1, 1) = COSALPHA*COSBETA
ROT(2, 1) = SINALPHA*COSBETA
ROT(3, 1) = COSALPHA*SINBETA
ROT(1, 2) = SINALPHA*COSBETA
ROT(2, 2) = COSALPHA*COSBETA
ROT(3, 2) = SINALPHA*SINBETA
ROT(1, 3) = -SINBETA
ROT(2, 3) = ZERO
ROT(3, 3) = COSBETA

C The particles lie in the xy plane with the z-axis pointing in the
C direction of the area vector. The z-axis components should be zero
C and could be ignored in the calculations that follow.
B(1, 1) = RJ1(X+)
B(2, 1) = RJ1(Y+)
B(3, 1) = RJ1(Z+)
B(1, 2) = RJ2(X+)
B(2, 2) = RJ2(Y+)
B(3, 2) = RJ2(Z+)
B(1, 3) = RJ1DT(X+)
B(2, 3) = RJ1DT(Y+)
B(3, 3) = RJ1DT(Z+)
B(1, 4) = RJ2DT(X+)
B(2, 4) = RJ2DT(Y+)
B(3, 4) = RJ2DT(Z+)

C Transform to the intermediate coordinate system.
C Matrix-matrix multiplication routine.
CALL DQSM(M1) TRAENA, TRANSB, MA, NB, NA, ONE, ROT,
$ MA, B, NB, ZERO, C, MA)

C Intermediate coordinate system after rotating by alpha and beta.
RJ1(X+)=C(1, 1)
RJ1(Y+) = C(2, 1)
RJ1(Z+) = C(3, 1)
RJ2(X+) = C(1, 2)
RJ2(Y+) = C(2, 2)
RJ2(Z+) = C(3, 2)
RJ1DT(X+) = C(1, 3)
RJ1DT(Y+) = C(2, 3)
RJ1DT(Z+) = C(3, 3)
RJ2DT(X+) = C(1, 4)
RJ2DT(Y+) = C(2, 4)
RJ2DT(Z+) = C(3, 4)

C WRITE(UNIT=,FMT=*)'initial r,R in intermediate from pat_sw'
C WRITE(UNIT=,FMT=('(2(3/E24.16/)'))')RJ1,RJ2
C WRITE(UNIT=,FMT=*)'initial rdt,Rdt in intermediate from pat_sw'
C WRITE(UNIT=,FMT=('(2(3/E24.16/)'))')RJ1DT,RJ2DT

C ***************************************************************************
C C * DETERMINING THE EXTERNAL ROTATION ANGLE, GAMMA
C C * The final rotation angle, gamma, describes the rotation about
C * the final z-axis, also known as the figure-axis, of the
C * stationary coordinate system in relation to the rotating
C * coordinate system.
C *
The principal axes of inertia system is the desired rotating coordinate system. Applying the condition of principal axes, where all cross-terms of the inertia tensor are zero, one can determine an appropriate angle, \( \gamma \).

Smith and Whitten devised an approach to solve for the angle \( \gamma \). This method and its necessary constraints will be applied.

Special constants defined by Smith–Whitten in order to determine the final external orientation angle, \( \gamma \), in principal axes of inertia coordinates. These principal axes require the area vector for the triangle formed by the three particles be aligned in the positive z-axis direction and that \( I_{yy} > I_{xx} \).

\[
S = \frac{(I_{yy} - I_{xx})}{mu} \\
T = 2I_{xy}/mu \\
Q = \sqrt{(SP^2 + TP^2)} \\
SP = RJ1(X_1) + RJ1(X_2) + RJ1(X_3) \\
TP = 2(RJ1(Y_1) + RJ1(Y_2) + RJ1(Y_3)) \\
QP = \sqrt{SP^2 + TP^2} \\
\]

The ATAN2() intrinsic function for the arctangent returns a result in the \(-PI to PI\) range. Shift the ATAN2() result and use the modulus to normalize the solution over the desired 0 to 2PI range. The Euler angle, \( \gamma \), can range between 0 and 2PI. Therefore, two solutions for \( \gamma \) can be obtained using this method. One for the 0 to PI range, and another for the PI to 2PI range, arrived by adding PI to the solution.

Following the approach taken by Johnson, the solution that falls in the 0 to PI range for \( \gamma \) will be used.

Following the approach taken by Johnson, the solution that falls in the 0 to PI range for \( \gamma \) will be used.

\[
\text{GAMMA} = \text{DMOD}(\text{HALF} \times \text{DATAN2} (\text{TP}, \text{SP}) + \text{TWOPI}, \text{TWOPI}) \\
\]

Using \( \gamma \), transform the intermediate coordinates to the final principal axes of inertia as defined by Smith–Whitten.

\[
\sin \gamma = \text{DSIN}(\text{GAMMA}) \\
\cos \gamma = \text{DCOS}(\text{GAMMA}) \\
\]

Rot(\( \gamma \))

\[
\text{ROTT}(1,1) = \cos \gamma \\
\text{ROTT}(2,1) = -\sin \gamma \\
\text{ROTT}(3,1) = 0 \\
\text{ROTT}(1,2) = \sin \gamma \\
\text{ROTT}(2,2) = \cos \gamma \\
\text{ROTT}(3,2) = 0 \\
\text{ROTT}(1,3) = 0 \\
\text{ROTT}(2,3) = 0 \\
\text{ROTT}(3,3) = 1 \\
\]

\[
\text{B}(1,1) = \text{RJ1}(X_1) \\
\text{B}(2,1) = \text{RJ1}(Y_1) \\
\text{B}(3,1) = \text{RJ1}(Z_1) \\
\text{B}(1,2) = \text{RJ1DT}(X_1) \\
\text{B}(2,2) = \text{RJ1DT}(Y_1) \\
\text{B}(3,2) = \text{RJ1DT}(Z_1) \\
\text{B}(1,3) = \text{RJ1DT}(X_1) \\
\text{B}(2,3) = \text{RJ1DT}(Y_1) \\
\text{B}(3,3) = \text{RJ1DT}(Z_1) \\
\]

Transform to Smith–Whitten principal axes of inertia coordinates.

\[
\text{CALL} \text{DGEMM}(\text{TRANSA}, \text{TRANSB}, \text{MA}, \text{NB}, \text{NA}, \text{ONE}, \text{ROT}, \text{B}, \text{C}) \\
\]

187
New body-fixed, principal axes Jacobi coordinates.

By construction, the position vectors should not have z-components.

\[
\begin{align*}
R_{J1}(X) &= C(1, 1) \\
R_{J1}(Y) &= C(2, 1) \\
R_{J1}(Z) &= C(3, 1) \\
R_{J2}(X) &= C(1, 2) \\
R_{J2}(Y) &= C(2, 2) \\
R_{J2}(Z) &= C(3, 2) \\
R_{J1DT}(X) &= C(1, 3) \\
R_{J1DT}(Y) &= C(2, 3) \\
R_{J1DT}(Z) &= C(3, 3) \\
R_{J2DT}(X) &= C(1, 4) \\
R_{J2DT}(Y) &= C(2, 4) \\
R_{J2DT}(Z) &= C(3, 4)
\end{align*}
\]

C WRITE(UNIT=∗,FMT=∗) 'initial r, R in principal axes from pat_sw'
C WRITE(UNIT=∗,FMT='(2(3 E24 . 16 / )) ') RJ1, RJ2
C WRITE(UNIT=∗,FMT=∗) 'initial rdt, Rdt in principal axes from pat_sw'
C WRITE(UNIT=∗,FMT='(2(3 E24 . 16 / )) ') RJ1DT, RJ2DT

Check products of inertia for zero closeness.

DO 5 I1 = 1, 2
   DO 6 J1 = I1 + 1, NSPACE
      ICROSS = MU*( C(I1, 1)*C(J1, 1) + C(I1, 2)*C(J1, 2) )
      IF ( DABS(ICROSS) .GT. MILLI ) THEN
         WRITE(∗, '(/A, E24 . 16 / )') 'ICROSS = ', ICROSS
         STOP
      ENDIF
   6 CONTINUE
5 CONTINUE
C Determine moment of inertia matrix using Jacobi coordinates.

\[
\begin{align*}
IXX &= MU*( R_{J1}(Y)*R_{J1}(Y) + R_{J2}(Y)*R_{J2}(Y) ) \\
IYY &= MU*( R_{J1}(X)*R_{J1}(X) + R_{J2}(X)*R_{J2}(X) ) \\
IZZ &= IXX + IYY \\
IXY &= -MU*( R_{J1}(X)*R_{J1}(Y) + R_{J2}(X)*R_{J2}(Y) ) \\
IXZ &= -MU*( R_{J1}(X)*R_{J1}(Z) + R_{J2}(X)*R_{J2}(Z) ) \\
IYZ &= -MU*( R_{J1}(Y)*R_{J1}(Z) + R_{J2}(Y)*R_{J2}(Z) )
\end{align*}
\]

IM(1, 1) = IXX
IM(1, 2) = IXY
IM(1, 3) = IXZ
IM(2, 1) = IYY
IM(2, 2) = IYZ
IM(2, 3) = IZZ
IM(3, 1) = IXZ
IM(3, 2) = IZZ
IM(3, 3) = IZZ

IF ( IYY .LT. IXX ) THEN
   GAMMA = GAMMA + HALFPI
END IF

************************************************************
C * TRANSFORM TO PRINCIPAL AXES
C * Euler rotation matrix using external orientation angles.
C ************************************************************

The orientation angles that relate the original, stationary coordinate system to the rotating, Principal Axes of Inertia coordinates, were determined using the Smith-Whitten method where the area vector of the triangle formed by the three particles points in the positive z-axis.

Orientation angles are identical to Euler angles:
C rotation about initial z-axis: 0 <= alpha <= 2PI
C rotation about intermediate y-axis: 0 <= beta <= PI
C rotation about final z-axis: 0 <= gamma <= PI (not 2PI)
C A = EULER(1) = ALPHA
C B = EULER(2) = BETA
C C = EULER(3) = GAMMA

EULER(1) = ALPHA
EULER(2) = BETA
EULER(3) = GAMMA

C Compute trigonometric functions of the orientation angles once.
SINA = DSIN( EULER(1) )
SINB = DSIN( EULER(2) )
SINC = DSIN( EULER(3) )
COSA = DCOS( EULER(1) )
COSB = DCOS( EULER(2) )
COSC = DCOS( EULER(3) )

C Y-convention Euler rotation matrix of transformation.
C Space to Body transformation: rbody = A * rspace
EROT(1, 1) = COSB * COSA + COSC * SINA * SINC
EROT(1, 2) = SINC * COSA + COSB * SINA * COSC
EROT(1, 3) = SINB * COSA
EROT(2, 1) = ( COSC * SINA + COSB * COSA * SINC )
EROT(2, 2) = COSC * COSA - COSB * SINA * SINC
EROT(2, 3) = SINB * SINA
EROT(3, 1) = - COSC * SINB
EROT(3, 2) = SINC
EROT(3, 3) = COSB

C Matrix multiply the orthogonal Euler rotation matrix with the original coordinates to rotate to principal axes.
C Store the position and velocity vectors in matrix form for more compact computation.
C B is a MxN array with position vectors in first D columns and velocity vectors in the remaining columns.
DO 20 , I1 = 1 , NBODY
BB(1 , I1) = X( I1 )
BB(2 , I1) = Y( I1 )
BB(3 , I1) = Z( I1 )
BB(1 , I1+ NBODY) = VX( I1 )
BB(2 , I1+ NBODY) = VY( I1 )
BB(3 , I1+ NBODY) = VZ( I1 )
20 CONTINUE

C Multiply unitary transformation matrix with original coordinates to transform to new principal axes of inertia coordinates.
CALL DGERM(TRANSA, TRANSB, MA, NN, NA, ONE, EROT, $ MA, BB, MB, ZERO, CC, MA)

C New body-fixed, principal axes of inertia coordinates as defined by Smith and Whitten.
DO 30 , I1 = 1 , NBODY
X( I1 ) = CC(1 , I1 )
Y( I1 ) = CC(2 , I1 )
Z( I1 ) = CC(3 , I1 )
VX( I1 ) = CC(1 , I1+NBODY)
VY( I1 ) = CC(2 , I1+NBODY)
VZ( I1 ) = CC(3 , I1+NBODY)
30 CONTINUE

RETURN
END

B.1.9 PRINTENERGY

Write energy data to a file.
Write the energy data to a file for plotting.

ARGUMENTS  I/O   TYPE   DESCRIPTION
TIME        IN    REAL8  Elapsed Simulation time.
ENERGY      IN    REAL8  Hamiltonian total system energy.
FUNIT       IN    INT4   Output file unit number.

SUBROUTINE PRINTENERGY(TIME, ENERGY, FUNIT)
IMPLICIT NONE

INTEGER FUNIT
CHARACTER* FMTOUT
PARAMETER (FMTOUT = '(3(E24.16, 5X))')

DOUBLE PRECISION $ TIME, E0, ENERGY
LOGICAL FIRST
SAVE FIRST, E0
DATA FIRST / .TRUE. /

IF (FIRST) THEN
  E0 = ENERGY
  FIRST = .FALSE.
ENDIF

WRITE(UNIT=FUNIT,FMT=FMTOUT)TIME, ENERGY, ENERGY/E0
RETURN
END

B.1.10 PRINTHYP

Write integrated hyperspherical coordinates to a file.

SUBROUTINE PRINTHYP
IMPLICIT NONE

INCLUDE 'HYDEF.FOR'
INCLUDE 'ENUMHYP.FOR'
INCLUDE 'PHASESPACE.FOR'
INCLUDE 'PRINTHYP.FOR'

CHARACTER* FMTOUT
PARAMETER (FMTOUT = '(12 E24.16)')

WRITE(UNIT=HSUNIT,FMT=FMTOUT)
$ PHASEPT(Q1_)
$ PHASEPT(Q2_)
$ PHASEPT(Q3_)
$ YN(ALPHA_)
$ YN(BETA_)
$ YN(GAMMA_)
$ PHASEPT(P1_)
$ PHASEPT(P2_)
$ PHASEPT(P3_)
$ YN(PALPHA_)
$ YN(PBETA_)
$ YN(PGAMMA_)

190
B.1.11 PRINTPHASE

Write the integrated phase space coordinates for one degree of freedom to a file.

```
AUTHOR: Paul Burkhardt
Date: September 28, 2004

ARGUMENTS

I/O TYPE DESCRIPTION
X IN REAL8 Position coordinate.
Y IN REAL8 Momentum coordinate.
FUNIT IN INT4 Output file unit number.

SUBROUTINE PRINTPHASE(X, Y, FUNIT)
IMPLICIT NONE

PARAMETER (FMTOUT = ' ( 2 ( E24 . 16 , 5X) ) ')

DOUBLE PRECISION X, Y
WRITE(UNIT=FUNIT,FMT=FMTOUT) X, Y
RETURN
END
```

B.2 Include Files For Hyperspherical Coordinates Programs

B.2.1 CARTDEF.FOR

```
AUTHOR: Paul Burkhardt
Date: September 28, 2004

INCLUDE File

Purpose: Include file that contains the declarations, defines, and constants for Cartesian coordinates.

INTEGER

PARAMETER (NBODY = 3)

DOUBLE PRECISION
```
B.2.2 ENUM_DIST.FOR

*******************************************************************************
  Author: Paul Burkhardt  
  Date: September 28, 2004  

  Include File  

  Purpose: Include file that enumerates the three inter-particle 
  distances and maps to the potential parameters.  

  R(BC) = R(1) = H-H distance  
  R(AC) = R(2) = F-H2 distance  
  R(AB) = R(3) = F-H1 distance  

*******************************************************************************

INTEGER AB, BC, AC
PARAMETER (AB = 3, BC = 1, AC = 2)

B.2.3 ENUM_HYP.FOR

*******************************************************************************
  Author: Paul Burkhardt  
  Date: September 28, 2004  

  Include File  

  Purpose: Include file that enumerates the twelve hyperspherical 
  coordinates and momenta.  

  Q(1,2,3,4,5,6) = Q(rho, theta, phi, alpha, beta, gamma)  
  P(1,2,3,4,5,6) = P(prho, ptheta, pphi, palpha, pbeta, pgamma)  
  H(Q,P)  

*******************************************************************************

INTEGER $RHO, $THETA, $PHI, $ALPHA, $BETA, $GAMMA

INTEGER $PRHO, $PTHETA, $PPHI, $PALPHA, $PBETA, $PGAMMA

PARAMETER ($RHO = 1, $THETA = 2, $PHI = 3, $ALPHA = 4, $BETA = 5, $GAMMA = 6)

PARAMETER ($PRHO = 7, $PTHETA = 8, $PPHI = 9, $PALPHA = 10, $PBETA = 11, $PGAMMA = 12)
B.2.4 ENUM_XYZ.FOR

*****************************************************************************
  Author: Paul Burkhardt
  Date: September 28, 2004
  Include File
  Purpose: Include file that enumerates the three Cartesian coordinate components.
  \n  \texttt{xyz = (1,2,3)}

INTEGER \texttt{X}, \texttt{Y}, \texttt{Z}
PARAMETER (\texttt{X} = 1, \texttt{Y} = 2, \texttt{Z} = 3)

*****************************************************************************

B.2.5 HYPDEF.FOR

*****************************************************************************
  Author: Paul Burkhardt
  Date: September 28, 2004
  Include File
  Purpose: Include file that contains the declarations, defines, and constants for the
  hyperspherical coordinates.

INTEGER \texttt{NVAR}
PARAMETER (\texttt{NVAR} = 12)

DOUBLE PRECISION \texttt{YN(NVAR)}, \texttt{DYDT(NVAR)}

DOUBLE PRECISION
  \$ \texttt{RHO},
  \$ \texttt{THETA},
  \$ \texttt{PHI},
  \$ \texttt{ALPHA},
  \$ \texttt{BETA},
  \$ \texttt{GAMMA},
  \$ \texttt{PRHO},
  \$ \texttt{PTHETA},
  \$ \texttt{PPHI},
  \$ \texttt{PALPHA},
  \$ \texttt{PBETA},
  \$ \texttt{PGAMMA}

DOUBLE PRECISION
  \$ \texttt{RHODT},
  \$ \texttt{THETADT},
  \$ \texttt{PHIDT},
  \$ \texttt{ALPHADT},
  \$ \texttt{BETADT},
  \$ \texttt{GAMMADT},
  \$ \texttt{PRHODT},
  \$ \texttt{PTHETADT},
  \$ \texttt{PPHIDT},
  \$ \texttt{PALPHADT},
  \$ \texttt{PBETADT},
  \$ \texttt{PGAMMADT}

*****************************************************************************

B.2.6 JACDEF.FOR

*****************************************************************************
  Author: Paul Burkhardt
  Date: September 28, 2004
  Include File
  Purpose: Include file for Jacobi relative coordinate variables.

*****************************************************************************
B.2.7 JACOBI_CONF.FOR

******************************************************************************
DOUBLE PRECISION
$ RJ1(NSPACE) , RJ2(NSPACE) ,
$ RJ1DT(NSPACE) , RJ2DT(NSPACE)
******************************************************************************

B.2.7 JACOBI_CONF.FOR

******************************************************************************
Author: Paul Burkhardt
Date: September 28, 2004
Include File
Purpose: Include file that defines one of the three possible
Jacobi vector representations or configuration.

INTEGER I, J, K
PARAMETER (K = 1, J = 3, I = 2)
******************************************************************************

B.2.8 MASS.FOR

******************************************************************************
Author: Paul Burkhardt
Date: September 28, 2004
Include File
Purpose: Include file defining mass related constants.
******************************************************************************
DOUBLE PRECISION
$ JTOT
DOUBLE PRECISION
$ MASS(3) , MTOT, MU,
$ E2, E3, D(3)
COMMON /ANGM/ JTOT
COMMON /MASS/ MASS, MTOT, MU, E2, E3, D
******************************************************************************

B.2.9 NUMCON.FOR

******************************************************************************
Author: Paul Burkhardt
Date: September 28, 2004
Include File
Purpose: Include file for double precision numerical constants.
******************************************************************************
DOUBLE PRECISION ZERO
PARAMETER (ZERO = 0.D0)
DOUBLE PRECISION ONE
PARAMETER (ONE = 1.D0)
******************************************************************************

194
B.2.10 ODEINT.FOR

*******************************************************************************
* Author: Paul Burkhardt
* Date: September 28, 2004
* Include File
* Purpose: Include file for integration parameters.
* H1 Initial step guess
* EPS Error tolerance
* HSTEP Integration step size
* LS Last time step
* T2 Time step size
*******************************************************************************

INTEGER
$ NOK, NBAD,
$ KMAX, KOUNT
DOUBLE PRECISION
$ EPS, H1, HMIN,
$ DXSAV, XP(200), YP(10,200)
COMMON /PATH/ KMAX, KOUNT, DXSAV, XP, YP

B.2.11 PES_COMMON.FOR

*******************************************************************************
* Author: Paul Burkhardt
* Date: September 28, 2004
* Include File
* Purpose: Include file for potential parameters.
* Definition of inter-particle distances.
* R1 = F-H1 distance
* R2 = H1-H2 distance
* R3 = F-H2 distance
*******************************************************************************

DOUBLE PRECISION
$ R(3), PE, DPE(3)

INTEGER R1, R2, R3
PARAMETER (R1 = 1, R2 = 2, R3 = 3)
B.2.12 PHASESPACE.FOR

INCOMPLETE

INTEGER DOF
PARAMETER (DOF = 4)

INTEGER PHASEDIM
PARAMETER (PHASEDIM = 2*DOF)

DOUB URE PRECISION
$ PHASEPT(PHASEDIM),
$ PHASEFL(PHASEDIM)

DOUB URE PRECISION
$ QVAR(DOF),
$ PVAR(DOF)

INTEGER $ Q1, Q2, Q3, Q4,
$ P1, P2, P3, P4

PARAMETER ( $ Q1 = 1,
$ Q2 = 2,
$ Q3 = 3,
$ Q4 = 4,
$ P1 = 5,
$ P2 = 6,
$ P3 = 7,
$ P4 = 8 )

B.2.13 PHYSCON.FOR

INCOMPLETE

INCLUDE FILE

Purpose: Include file for integration variables.

INCLUDE FILE

Purpose: Include file for double-precision physical constants.

Constants from the National Institute of Standards and Technology (NIST) Reference for Constants, Units, and Uncertainty. 
[http://physics.nist.gov]

**QUANTITIES IN ATOMIC UNITS**

**Hartree in Joules**

DOUB URE PRECISION HJ
PARAMETER (HJ = 4.35974417D−18)

**Electron mass in kilograms**

DOUB URE PRECISION EMS
PARAMETER (EMS = 9.1093826D−31)

**Bohr radius in meters**

DOUB URE PRECISION A0
PARAMETER (A0 = 0.5291772108D-10)
C Time in seconds
DOUBLE PRECISION AUT
PARAMETER (AUT = 2.41884326505D-17)
C Velocity in meters per second
DOUBLE PRECISION AUV
PARAMETER (AUV = 2.1876912633D6)
C Momentum in kilogram meter per second
DOUBLE PRECISION AUM
PARAMETER (AUM = 1.99285166D-24)
C Force in Newtons
DOUBLE PRECISION AUF
PARAMETER (AUF = 8.2387225D-8)

******************************************************************************************
C COMMON PHYSICAL CONSTANTS
******************************************************************************************
C Planck constant in Joules
DOUBLE PRECISION PLANCK
PARAMETER (PLANCK = 6.6260693D-34)
C Planck constant over 2 pi in Joules
DOUBLE PRECISION HBAR
PARAMETER (HBAR = 1.05457168D-34)
C Electron mass in atomic mass units
DOUBLE PRECISION EMAU
PARAMETER (EMAU = 5.4857990945D-4)
C Speed of light in vacuum in meters per second
DOUBLE PRECISION CLGHT
PARAMETER (CLGHT = 2.99792458D8)
C Electron volt in joules
DOUBLE PRECISION EVJ
PARAMETER (EVJ = 1.60217653D-19)
C Atomic mass constant in kilograms
DOUBLE PRECISION AMU
PARAMETER (AMU = 1.66053886D-27)
C Avogadro constant
DOUBLE PRECISION AV0
PARAMETER (AV0 = 6.0221415D23)
C Hartree energy in electron volts
DOUBLE PRECISION HEV
PARAMETER (HEV = 27.2113845D0)
Cật*
C Ratio of circumference to diameter of a circle
C ************************************************************
C Calculated numerically on PC using 4*atan2(1.0,1.0)
DOUBLE PRECISION PI
PARAMETER (PI = 3.141592653589793D0)

B.2.14 PRINTHYP.FOR

******************************************************************************
* Author: Paul Burkhardt
* Date: September 28, 2004
* Include File
* Purpose: Include file with variables for printing the coordinates.
******************************************************************************
INTEGER HSUNIT
PARAMETER (HSUNIT = 11)
B.2.15 SIPREFIX.FOR

*******************************************************************************
* Author: Paul Burkhardt
* Date: September 28, 2004
* Include File
* Purpose: Include file that defines SI metric prefix constants.
*******************************************************************************

DOUBLE PRECISION DEKA
PARAMETER (DEKA = 1.D1)
DOUBLE PRECISION HECTO
PARAMETER (HECTO = 1.D2)
DOUBLE PRECISION KILO
PARAMETER (KILO = 1.D3)
DOUBLE PRECISION MEGA
PARAMETER (MEGA = 1.D6)
DOUBLE PRECISION GIGA
PARAMETER (GIGA = 1.D9)
DOUBLE PRECISION TERA
PARAMETER (TERA = 1.D12)
DOUBLE PRECISION PETA
PARAMETER (PETA = 1.D15)
DOUBLE PRECISION EXA
PARAMETER (EXA = 1.D18)
DOUBLE PRECISION ZETTA
PARAMETER (ZETTA = 1.D21)
DOUBLE PRECISION YOTTA
PARAMETER (YOTTA = 1.D24)

DOUBLE PRECISION CENTI
PARAMETER (CENTI = 1.D0/HECTO)
DOUBLE PRECISION MILLI
PARAMETER (MILLI = 1.D0/KILO)
DOUBLE PRECISION MICRO
PARAMETER (MICRO = 1.D0/MEGA)
DOUBLE PRECISION NANO
PARAMETER (NANO = 1.D0/GIGA)
DOUBLE PRECISION PICO
PARAMETER (PICO = 1.D0/TERA)
DOUBLE PRECISION FEMTO
PARAMETER (FEMTO = 1.D0/PETA)
DOUBLE PRECISION ATTO
PARAMETER (ATTO = 1.D0/EXA)
DOUBLE PRECISION ZEPTO
PARAMETER (ZEPTO = 1.D0/ZETTA)
DOUBLE PRECISION YOCTO
PARAMETER (YOCTO = 1.D0/YOTTA)

DOUBLE PRECISION FEMTO2
PARAMETER (FEMTO2 = 1.D0/PETA**2)

198
B.2.16 SPATIAL_DIM.FOR

******************************************************************************
 * Author: Paul Burkhardt
 * Date: September 28, 2004
 * Include File
 * Purpose: Include file for parameterizing spatial dimension.
******************************************************************************
PARAMETER(NSPACE = 3)
Appendix C

Programs For 3-Body Classical Trajectories In Hyperspherical Coordinates

C.1 Main Driver Using Burlirsch-Stoer Integrator

C.1.1 HS2D

Main driver for integrating the equations of motion using a Burlirsch-Stoer integrator.

```
AUTHOR:  Paul Burkhardt
DATE:    September 28, 2004

PROGRAM: HS2D

IMPLICIT NONE

EXTERNAL DERIV, BSSTEPD

INCLUDE 'NUMCON.FOR'
INCLUDE 'PESCOMMON.FOR'
INCLUDE 'HYPERDEF.FOR'
INCLUDE 'ENUMQHYP.FOR'
INCLUDE 'PHASESPACE.FOR'
INCLUDE 'PRINTDYP.FOR'
INCLUDE 'ODERTE.FOR'

INTEGER $ 11

INTEGER $ MSTEP, PSTEP

DOUBLE PRECISION
```
$ \text{TIME, TSTEP}$

DOUBLE PRECISION $\text{ENERGY}$

INTEGER $\text{ENUNIT, PSUNIT}$

PARAMETER (ENUNIT = 31)
PARAMETER (PSUNIT = 2)

CHARACTER(*) $\text{ENFILE, PSFILE}$

PARAMETER (ENFILE = 'energy bs.dat')
PARAMETER (PSFILE = 'phasespace bs.dat')

DATA TIME, ZERO/
DATA EPS, HMIN /1.0, 0.0/
DATA KMAX, DXSAV, XP, YP /0, 0.0, 200.0, 2000.0/

C * GET INITIAL STATE VALUES
CALL INIT(TSTEP, MSTEP, PSTEP, YN)

C Open output files.
OPEN(UNIT = ENUNIT, FILE = ENFILE)
OPEN(UNIT = PSUNIT, FILE = PSFILE)
OPEN(UNIT = HSUNIT, FILE = HSFILE)

C Independent variables of integration used to solve the
C canonical equations of motion.
C $H(Q, P) = H(RHO, THETA, PH, GAMMA, PRHO, PTHETA, PPHI, PGAMMA)$
PHASEPT(Q1) = YN(RHO)
PHASEPT(Q2) = YN(THETA)
PHASEPT(Q3) = YN(PH)
PHASEPT(Q4) = YN(GAMMA)
PHASEPT(P1) = YN(PRHO)
PHASEPT(P2) = YN(PTHETA)
PHASEPT(P3) = YN(PPHI)
PHASEPT(P4) = YN(PGAMMA)

C Initialize potential energy surface (PES) and determine initial
C potential energy.
CALL PESINIT(PHASEPT)
CALL GETENERGY(PHASEPT, ENERGY)
CALL PRINTENERGY(TIME, ENERGY, ENUNIT)
CALL PRINTPHASE(PHASEPT(Q1), PHASEPT(P1), PSUNIT)
CALL PRINTHYP

C *****************************************************************
C INTEGRATE HAMILTONIAN EQUATIONS OF MOTION
C *****************************************************************
DO 50 I1 = 1, MSTEP
   TIME = TIME + TSTEP
   C Call the integrator routine.
   CALL ODEINT(PHASEPT, PHASEDIM, TIME, (TIME+TSTEP),
      EPS, TSTEP, HMIN, NOK, NBAD, DERIV, BSSTEPD)
   IF ( MOD(I1, PSTEP) .EQ. 0 ) THEN
      CALL GETENERGY(PHASEPT, ENERGY)
      CALL PRINTENERGY(TIME, ENERGY, ENUNIT)
      CALL PRINTPHASE(PHASEPT(Q1), PHASEPT(P1), PSUNIT)
      CALL PRINTHYP
   ENDIF
50 CONTINUE

C Close all files.
CLOSE(ENUNIT)
CLOSE(PSUNIT)
CLOSE(HSUNIT)
C.2 Main Driver Using Second Order Symplectic Integrator

C.2.1 HS2D

Main driver for integrating the equations of motion using explicit, second order multi-map symmetrized composition method, symplectic integrator.

```fortran
PROGRAM HS2D
IMPLICIT NONE
INCLUDE 'NUMCON.FOR'
INCLUDE 'PESCOMMON.FOR'
INCLUDE 'HYPDEF.FOR'
INCLUDE 'ENUMHYP.FOR'
INCLUDE 'PHASESPACE.FOR'
INCLUDE 'PRINTHYP.FOR'
INTEGER $ I1
INTEGER $ MSTEP, PSTEP
DOUBLE PRECISION $ TIME, TSTEP
DOUBLE PRECISION $ ENERGY
INTEGER $ ENUNIT, $ PSUNIT
PARAMETER (ENUNIT = 31)
PARAMETER (PSUNIT = 2)
CHARACTER*(*) $ ENFILE,
$ PSFILE
PARAMETER (ENFILE = 'energy_sia2.dat')
PARAMETER (PSFILE = 'phasespace_sia2.dat')
DATA TIME / ZERO /

C *******************************************************
C * GET INITIAL STATE VALUES
C *******************************************************
CALL INIT(TSTEP, MSTEP, PSTEP, YN)
C Open output files.
OPEN(UNIT = ENUNIT, FILE = ENFILE)
```

Author: Paul Burkhardt
Date: September 28, 2004
Program: HS2D

Classical trajectory calculation for three particles in Smith-Whitten-Johnson symmetrized hyperspherical coordinates.
C Independent variables of integration used to solve the canonical equations of motion:
C \( H(Q,P) = H(\text{RHO}, \text{THETA}, \text{PHI}, \text{GAMMA}, \text{PRHO}, \text{PTHETA}, \text{PPHI}, \text{PGAMMA}) \)

\[
\begin{align*}
\text{PHASEPT}(Q_1) &= YN(\text{RHO}) \\
\text{PHASEPT}(Q_2) &= YN(\text{THETA}) \\
\text{PHASEPT}(Q_3) &= YN(\text{PHI}) \\
\text{PHASEPT}(Q_4) &= YN(\text{GAMMA}) \\
\text{PHASEPT}(P_1) &= YN(\text{PRHO}) \\
\text{PHASEPT}(P_2) &= YN(\text{PTHETA}) \\
\text{PHASEPT}(P_3) &= YN(\text{PPHI}) \\
\text{PHASEPT}(P_4) &= YN(\text{PGAMMA})
\end{align*}
\]

C Initialize potential energy surface (PES) and determine initial potential energy.
CALL PESINIT(PHASEPT)
CALL GETENERGY(PHASEPT, ENERGY)
CALL PRINTENERGY(TIME, ENERGY, ENUNIT)
CALL PRINTPHASE(PHASEPT(Q_1), PHASEPT(P_1), PSUNIT)
CALL PRINTHYP

C *********************************************************************
C INTEGRATE HAMILTONIAN EQUATIONS OF MOTION
C *********************************************************************
DO 50 I1 = 1, MSTEP
    TIME = TIME + TSTEP
    CALL HS2D_J1A2(TSTEP, PHASEPT)
    IF ( MOD(I1, PSTEP) .EQ. 0 ) THEN
        CALL GETENERGY(PHASEPT, ENERGY)
        CALL PRINTENERGY(TIME, ENERGY, ENUNIT)
        CALL PRINTPHASE(PHASEPT(Q_1), PHASEPT(P_1), PSUNIT)
        CALL PRINTHYP
    ENDIF
50 CONTINUE
C Close all files.
CLOSE(ENUNIT)
CLOSE(PSUNIT)
CLOSE(HSUNIT)
END

C.3 Main Driver Using Fourth Order Runge-Kutta Integrator

C.3.1 HS2D

Main driver for integrating the equations of motion using a fourth order Runge-Kutta integrator.
**Classical trajectory calculation for three particles in Smith–Whitten–Johnson symmetrized hyperspherical coordinates.**

******************************************************************************PROGRAM HS2D******************************************************************************

**IMPLICIT** NONE

**INCLUDE 'NUMCON.FOR'**

**INCLUDE 'PESCOMMON.FOR'**

**INCLUDE 'HYPERF.FOR'**

**INCLUDE 'ENUMHYP.FOR'**

**INCLUDE 'PHASESPACE.FOR'**

**INCLUDE 'PRINTHYP.FOR'**

**INTEGER** $I1$

**INTEGER** $MSTEP, PSTEP$

**DOUBLE PRECISION** $TIME, TSTEP$

**DOUBLE PRECISION** $ENERGY$

**INTEGER** $ENUNIT,$ $PSUNIT$

**PARAMETER** (ENUNIT = 31)

**PARAMETER** (PSUNIT = 2)

**CHARACTER** ($**$)

$ENFILE,$ $PSFILE$

**PARAMETER** (ENFILE = 'energy_rk4.dat')

**PARAMETER** (PSFILE = 'phasespace_rk4.dat')

**DATA** $TIME$ = /ZERO/

******************************************************************************C******************************************************************************

**C GET INITIAL STATE VALUES**

**C******************************************************************************CALL INIT(TSTEP, MSTEP, PSTEP, YN)**

**C Open output files.**

**OPEN** (UNIT = ENUNIT, **FILE** = ENFILE)

**OPEN** (UNIT = PSUNIT, **FILE** = PSFILE)

**OPEN** (UNIT = HSUNIT, **FILE** = HSFILE)

**C Independent variables of integration used to solve the**

**C canonical equations of motion.**

**C**

$H(Q,P) = H(RHO, THETA, PHI, GAMMA, PRHO, PTHETA, PPHI, PGAMMA)$

**PHASEPT** ($Q1$) = $YN (RHO)$

**PHASEPT** ($Q2$) = $YN (THETA)$

**PHASEPT** ($Q3$) = $YN (PHI)$

**PHASEPT** ($Q4$) = $YN (GAMMA)$

**PHASEPT** ($P1$) = $YN (PRHO)$

**PHASEPT** ($P2$) = $YN (PTHETA)$

**PHASEPT** ($P3$) = $YN (PPHI)$

**PHASEPT** ($P4$) = $YN (PGAMMA)$

**C Initialize potential energy surface (PES) and determine initial**

**C potential energy.**

**CALL PESINIT(PHASEPT)**

**CALL GETENERGY(PHASEPT, ENERGY)**

**CALL PRINTENERGY(TIME, ENERGY, ENUNIT)**

**CALL PRINTPHASE(PHASEPT($Q1$), PHASEPT($P1$), PSUNIT)**

**CALL PRINTHYP**

******************************************************************************C******************************************************************************

**C INTEGRATE HAMILTONIAN EQUATIONS OF MOTION**

**C******************************************************************************DO 50 $I1 = 1, MSTEP$

$TIME = TIME + TSTEP$

50**CONTINUE**
CALL the integrator routine.
CALL RK4(TIME, TSTEP, PHASEPT, PHASEDIM)

IF ( MOD(I1, PSTEP) .EQ. 0 ) THEN
  CALL GETENERGY(PHASEPT, ENERGY)
  CALL PRINTENERGY(TIME, ENERGY, ENUNIT)
  CALL PRINTPHASE(PHASEPT(Q1), PHASEPT(P1), PSUNIT)
ENDIF

CONTINUE

CLOSE(ENUNIT)
CLOSE(PSUNIT)
CLOSE(HSUNIT)

END
Appendix D

Programs For The F+H₂ Hamiltonian Dynamics

D.1 Interface Programs For Stark-Werner F+H₂ Potential Energy Surface

D.1.1 PESDERIV

Interface for determining potential energy surface derivatives required for force evaluations.

```
C ***********************************************************************************************
C Author: Paul Burkhardt
Date: September 28, 2004
Program: PESDERIV
Calculate Potential Energy Surface (PES) derivatives.

ARGUMENTS I/O TYPE DESCRIPTION
RDIST IN REAL8 Inter-particle distances. Array of dimension, 3.
DVDR OUT REAL8 Derivatives of potential with respect to the inter-particle distances. Array of dimension, 3.

C ***********************************************************************************************
SUBROUTINE PESDERIV(RDIST, DVDR)
IMPLICIT NONE
INCLUDE 'ENUMDIST.FOR'
INCLUDE 'POTENTIAL.FOR'

DOUBLE PRECISION
$ RDIST(3), DVDR(3)
C ***********************************************************************************************
C * POTENTIAL ENERGY SURFACE DERIVATIVES
```

206
D.1.2 PESINIT

Interface to initialize potential energy surface.

SUBROUTINE PESINIT(PHASEPT)

IMPLICIT NONE

INCLUDE 'JACOBI_CONF.FOR'
INCLUDE 'POTENTIAL.FOR'
INCLUDE 'PHASESPACE.FOR'

DOUBLE PRECISION DVDN(3)

C ***********************************************************************
C POTENTIAL PARAMETERS
C ***********************************************************************
C *
C These parameters are specific to the energy potential.
C ***********************************************************************
C Initialize potential energy surface.
C Parameters are stored in a common block.
CALL PREPOT
C Determine initial potential energy value.
CALL FORCE(PHASEPT, DVDN)
RETURN
END
D.2 Include Files

D.2.1 FILEPATH.FOR

********************************************************************************
* Author: Paul Burkhardt
* Date: September 28, 2004
* Include File
* Purpose: Include file with variables for input file paths.
********************************************************************************

CHARACTER(∗) PARAM2
PARAMETER(PARAM2 = ‘././potential/two.param’)

CHARACTER(∗) PARAM3
PARAMETER(PARAM3 = ‘././potential/three.param’)

********************************************************************************

D.2.2 POTENTIAL.FOR

********************************************************************************
* Author: Paul Burkhardt
* Date: September 28, 2004
* Include File
* Purpose: Include file for potential parameters.
* Definition of inter-particle distances.
* R1 = F-H1 distance
* R2 = H1-H2 distance
* R3 = F-H2 distance
********************************************************************************

INCLUDE ’PESCOMMON.FOR’

INTEGER $ BSW(198) , CSW(198) , DSW(198)

DOUBLE PRECISION $ ASW(198) , PSW(12)

COMMON /CPARM/ ASW, PSW
COMMON /CINT/ BSW, CSW, DSW

D.3 Input Data Files

D.3.1 phasespace_01/dat_01

34631.9703 MASS(A) MASS OF PARTICLE A (ELECTRON MASS)
1837.15258 MASS(B) MASS OF PARTICLE B (ELECTRON MASS)
1837.15258 MASS(C) MASS OF PARTICLE C (ELECTRON MASS)
0 JQ ROTATIONAL QUANTUM NUMBER
1.67263657 ECOL CENTER OF MASS COLLISION ENERGY (HARTREE)
0. AX X-COORDINATE FOR IMPACT PARTICLE A (BOHR)
−8.0. AY Y-COORDINATE FOR IMPACT PARTICLE A (BOHR)
0. AZ Z-COORDINATE FOR IMPACT PARTICLE A (BOHR)
0. PHI EULER ANGLE ABOUT Z-AXIS OF B-C COUPLE (DEGREES)
0. THETA EULER ANGLE ABOUT Y-AXIS OF B-C COUPLE (DEGREES)
0. GAMMA EULER ANGLE ABOUT Z-AXIS OF B-C COUPLE (DEGREES)
### D.3.2 phasespace_01/dat_02

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass(A)</td>
<td>34631.9703</td>
</tr>
<tr>
<td>Mass(B)</td>
<td>1837.15258</td>
</tr>
<tr>
<td>Mass(C)</td>
<td>1837.15258</td>
</tr>
<tr>
<td>JQ</td>
<td>0.0</td>
</tr>
<tr>
<td>Ecoul</td>
<td>0.0</td>
</tr>
<tr>
<td>Rdist</td>
<td>1.67263657</td>
</tr>
<tr>
<td>Ax</td>
<td>-5.0</td>
</tr>
<tr>
<td>Ay</td>
<td>0.0</td>
</tr>
<tr>
<td>Az</td>
<td>0.0</td>
</tr>
<tr>
<td>Phi</td>
<td>0.0</td>
</tr>
<tr>
<td>Theta</td>
<td>0.0</td>
</tr>
<tr>
<td>Gamma</td>
<td>0.0</td>
</tr>
<tr>
<td>Mstep</td>
<td>1000000</td>
</tr>
<tr>
<td>Tstep</td>
<td>1.0</td>
</tr>
<tr>
<td>Pstep</td>
<td>100</td>
</tr>
</tbody>
</table>

### D.3.3 phasespace_02/dat_01

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass(A)</td>
<td>34631.9703</td>
</tr>
<tr>
<td>Mass(B)</td>
<td>1837.15258</td>
</tr>
<tr>
<td>Mass(C)</td>
<td>1837.15258</td>
</tr>
<tr>
<td>JQ</td>
<td>0.0</td>
</tr>
<tr>
<td>Ecoul</td>
<td>1.67263657</td>
</tr>
<tr>
<td>Rdist</td>
<td>0.0</td>
</tr>
<tr>
<td>Ax</td>
<td>-5.0</td>
</tr>
<tr>
<td>Ay</td>
<td>0.0</td>
</tr>
<tr>
<td>Az</td>
<td>0.0</td>
</tr>
<tr>
<td>Phi</td>
<td>0.0</td>
</tr>
<tr>
<td>Theta</td>
<td>0.0</td>
</tr>
<tr>
<td>Gamma</td>
<td>0.0</td>
</tr>
<tr>
<td>Mstep</td>
<td>1000000</td>
</tr>
<tr>
<td>Tstep</td>
<td>1.0</td>
</tr>
<tr>
<td>Pstep</td>
<td>100</td>
</tr>
</tbody>
</table>

### D.3.4 phasespace_02/dat_02

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass(A)</td>
<td>34631.9703</td>
</tr>
<tr>
<td>Mass(B)</td>
<td>1837.15258</td>
</tr>
<tr>
<td>Mass(C)</td>
<td>1837.15258</td>
</tr>
<tr>
<td>JQ</td>
<td>0.0</td>
</tr>
<tr>
<td>Ecoul</td>
<td>1.67263657</td>
</tr>
<tr>
<td>Rdist</td>
<td>0.0</td>
</tr>
<tr>
<td>Ax</td>
<td>-5.0</td>
</tr>
<tr>
<td>Ay</td>
<td>0.0</td>
</tr>
<tr>
<td>Az</td>
<td>0.0</td>
</tr>
<tr>
<td>Phi</td>
<td>0.0</td>
</tr>
<tr>
<td>Theta</td>
<td>0.0</td>
</tr>
<tr>
<td>Gamma</td>
<td>0.0</td>
</tr>
<tr>
<td>Mstep</td>
<td>1000000</td>
</tr>
<tr>
<td>Tstep</td>
<td>1.0</td>
</tr>
<tr>
<td>Pstep</td>
<td>100</td>
</tr>
</tbody>
</table>
Appendix E

Programs For The H+H₂ Hamiltonian Dynamics

E.1 Interface Programs For BKMP2 H+H₂ Potential Energy Surface

E.1.1 PESDERIV

Interface for determining potential energy surface derivatives required for force evaluations.

```
SUBROUTINE PESDERIV(RDIST, DVDR)

IMPLICIT NONE

INCLUDE 'ENUMDIST.FOR'
INCLUDE 'PESCOMMON.FOR'

DOUBLE PRECISION $ RDIST(3) , DVDR(3)

C *** potential energy surface derivatives
```

Author: Paul Burkhardt
Date: September 28, 2004

Program: PESDERIV

Calculate Potential Energy Surface (PES) derivatives.

Arguments I/O TYPE DESCRIPTION
RDIST IN REAL8 Inter-particle distances. Array of dimension, 3.
DVDR OUT REAL8 Derivatives of potential with respect to the inter-particle distances. Array of dimension, 3.
E.1.2 PESINIT

Interface to initialize potential energy surface.

---

**Author**: Paul Burkhardt

**Date**: September 28, 2004

**Program**: PESINIT

**Initialize Potential Energy Surface (PES)**.

**ARGUMENTS**

<table>
<thead>
<tr>
<th>I/O</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHASEPT</td>
<td>IN</td>
<td>REAL</td>
</tr>
</tbody>
</table>

**SUBROUTINE** PESINIT(PHASEPT)

**IMPLICIT** NONE

INCLUDE 'JACOBI_CONF.FOR'

INCLUDE 'PESCOMMON.FOR'

INCLUDE 'PHASESPACE.FOR'

**DOUBLE PRECISION**

$\text{DVDN}(3)$

---

**POTENTIAL PARAMETERS**

These parameters are specific to the energy potential.

Parameters are stored in a common block.

Determine initial potential energy value.

**CALL** FORCE(PHASEPT, DVDN)

**RETURN**

**END**
E.2 Input Data Files

E.2.1 phasespace_01/dat_01

| 1837.15258 | MASS(A) MASS OF PARTICLE A (ELECTRON MASS) |
| 1837.15258 | MASS(B) MASS OF PARTICLE B (ELECTRON MASS) |
| 1837.15258 | MASS(C) MASS OF PARTICLE C (ELECTRON MASS) |
| 0.0 | ECoul CENTER OF MASS COLLISION ENERGY (HARTREE) |
| 1.449 | RDIST DISTANCE BETWEEN PARTICLES B AND C (BOHR) |
| 0.0 | AX X-COORDINATE FOR IMPACT PARTICLE A (BOHR) |
| 0.0 | AZ Z-COORDINATE FOR IMPACT PARTICLE A (BOHR) |
| 0.0 | PHI EULER ANGLE ABOUT Z-AXIS OF B-C COUPLE (DEGREES) |
| 0.0 | THETA EULER ANGLE ABOUT Y-AXIS OF B-C COUPLE (DEGREES) |
| 0.0 | GAMMA EULER ANGLE ABOUT Z-AXIS OF B-C COUPLE (DEGREES) |
| 1000000 | MSTEP MAXIMUM NUMBER OF INTEGRATION STEPS |
| 1.0 | TSTEP TIME STEP SIZE |
| 100 | PSTEP INTEGRATION PRINTOUT INTERVAL |

E.2.2 phasespace_01/dat_02

| 1837.15258 | MASS(A) MASS OF PARTICLE A (ELECTRON MASS) |
| 1837.15258 | MASS(B) MASS OF PARTICLE B (ELECTRON MASS) |
| 1837.15258 | MASS(C) MASS OF PARTICLE C (ELECTRON MASS) |
| 0.0 | ECoul CENTER OF MASS COLLISION ENERGY (HARTREE) |
| 1.449 | RDIST DISTANCE BETWEEN PARTICLES B AND C (BOHR) |
| 0.0 | AX X-COORDINATE FOR IMPACT PARTICLE A (BOHR) |
| 0.0 | AZ Z-COORDINATE FOR IMPACT PARTICLE A (BOHR) |
| 0.0 | PHI EULER ANGLE ABOUT Z-AXIS OF B-C COUPLE (DEGREES) |
| 0.0 | THETA EULER ANGLE ABOUT Y-AXIS OF B-C COUPLE (DEGREES) |
| 0.0 | GAMMA EULER ANGLE ABOUT Z-AXIS OF B-C COUPLE (DEGREES) |
| 1000000 | MSTEP MAXIMUM NUMBER OF INTEGRATION STEPS |
| 1.0 | TSTEP TIME STEP SIZE |
| 100 | PSTEP INTEGRATION PRINTOUT INTERVAL |

E.2.3 phasespace_02/dat_01

| 1837.15258 | MASS(A) MASS OF PARTICLE A (ELECTRON MASS) |
| 1837.15258 | MASS(B) MASS OF PARTICLE B (ELECTRON MASS) |
| 1837.15258 | MASS(C) MASS OF PARTICLE C (ELECTRON MASS) |
| 0.0 | ECoul CENTER OF MASS COLLISION ENERGY (HARTREE) |
| 1.4 | RDIST INITIAL DISTANCE BETWEEN PARTICLE B AND C (BOHR) |
| 0.0 | AX X-COORD FOR IMPACT PARTICLE A (BOHR) |
| 0.0 | AZ Z-COORD FOR IMPACT PARTICLE A (BOHR) |
| 0.0 | PHI EULER ANGLE ABOUT Z-AXIS OF B-C COUPLE (DEGREES) |
| 0.0 | THETA ROTATION ABOUT INTERMEDIATE Y-AXIS OF B-C COUPLE (DEGREES) |
| 0.0 | GAMMA ROTATION ABOUT FINAL Z-AXIS OF B-C COUPLE (DEGREES) |
| 1000000 | MSTEP MAXIMUM NUMBER OF INTEGRATION STEPS |
| 1.0 | TSTEP TIME STEP SIZE |
| 100 | PSTEP INTEGRATION PRINTOUT INTERVAL |

E.2.4 phasespace_02/dat_02
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MASS(A)</td>
<td>MASS OF PARTICLE A (ELECTRON MASS)</td>
</tr>
<tr>
<td>MASS(B)</td>
<td>MASS OF PARTICLE B (ELECTRON MASS)</td>
</tr>
<tr>
<td>MASS(C)</td>
<td>MASS OF PARTICLE C (ELECTRON MASS)</td>
</tr>
<tr>
<td>JQ</td>
<td>ROTATIONAL QUANTUM NUMBER</td>
</tr>
<tr>
<td>ECOL</td>
<td>CENTER OF MASS COLLISION ENERGY (HARTREE)</td>
</tr>
<tr>
<td>RDIST</td>
<td>INITIAL DISTANCE BETWEEN PARTICLE B AND C (BOHR)</td>
</tr>
<tr>
<td>AX</td>
<td>X-COORD FOR IMPACT PARTICLE A (BOHR)</td>
</tr>
<tr>
<td>AY</td>
<td>Y-COORD FOR IMPACT PARTICLE A (BOHR)</td>
</tr>
<tr>
<td>AZ</td>
<td>Z-COORD FOR IMPACT PARTICLE A (BOHR)</td>
</tr>
<tr>
<td>PHI</td>
<td>ROTATION ABOUT Z-AXIS OF B-C COUPLE (DEGREES)</td>
</tr>
<tr>
<td>THETA</td>
<td>ROTATION ABOUT INTERMEDIATE Y-AXIS OF B-C COUPLE (DEGREES)</td>
</tr>
<tr>
<td>GAMMA</td>
<td>ROTATION ABOUT FINAL Z-AXIS OF B-C COUPLE (DEGREES)</td>
</tr>
<tr>
<td>MSTEP</td>
<td>MAXIMUM NUMBER OF INTEGRATION STEPS</td>
</tr>
<tr>
<td>TSTEP</td>
<td>TIME STEP SIZE</td>
</tr>
<tr>
<td>PSTEP</td>
<td>INTEGRATION PRINTOUT INTERVAL</td>
</tr>
</tbody>
</table>

213
Appendix F

Classical Trajectory Programs For Henon-Heiles Hamiltonian

F.1 Main Driver Using Fourth Order Runge-Kutta Integrator

F.1.1 MAIN

Main driver for integrating the equations of motion using a fourth order Runge-Kutta integrator.

```
PROGRAM MAIN
IMPLICIT NONE
INCLUDE 'NUMCON.FOR'

INTEGER $I1
INTEGER $MAXSTEP, $PRINTSTEP
INTEGER $ENUNIT, $PSUNIT
CHARACTER $ENFILE*(32), $PSFILE*(32)
```

---

Author: Paul Burkhardt
Date: September 28, 2004
Program: MAIN

Non-symplectic integration for separable (potential-form) Hamiltonians of the form: $H = T(p) + V(q)$
Calculates dynamics for the Henon-Heiles stellar motion problem.

---

214
DOUBLE PRECISION
$ TIME,
$ TIMESTEP,
$ ENERGY,
$ SIMTIME

INTEGER
$ DEGREES,
$ DIMENS

PARAMETER (DEGREES = 2, DIMENS = 2+DEGREES)

DOUBLE PRECISION
$ QVAR(DEGREES),
$ PVAR(DEGREES),
$ TEMP(2*DEGREES)

PARAMETER (TIMESTEP = 1.0D0/6.0D0)
PARAMETER (MAXSTEP = 1200000)
PARAMETER (PRINTSTEP = 1)
PARAMETER (ENUNIT = 1)
PARAMETER (PSUNIT = 10)
PARAMETER (ENFILE = 'energy_rk4.dat')
PARAMETER (PSFILE = 'phasespace_rk4.dat')

DOUBLE PRECISION
$ THIRD,
$ VE,
$ HE

PARAMETER (THIRD = ONE/THREE)
PARAMETER (HE = 0.029952D0)

DOUBLE PRECISION
$ CI, DI, FACTOR,
$ Q2, P2

C Open files.
OPEN(UNIT = ENUNIT, FILE = ENFILE)
OPEN(UNIT = PSUNIT, FILE = PSFILE)

C Initialization.
SIMTIME = MAXSTEP*TIMESTEP
TIME = ZERO

DO 5, 11 = 1, DEGREES
QVAR(11) = 0.12D0
PVAR(11) = 0.12D0
5 CONTINUE

C Determine the momentum in the x direction.
VE = QVAR(2)*QVAR(2)*(HALF - THIRD*QVAR(2))
$ + QVAR(1)*QVAR(1)*(QVAR(2) + HALF)

IF (VE .GT. HE) THEN
  WRITE(*,*)'Invalid energy!'
STOP
ENDIF

PVAR(1) = DSQRT( TWO*(HE - VE) - PVAR(2)*PVAR(2))

CALL GETENERGY(QVAR, PVAR, DEGREES, ENERGY)
CALL PRINTENERGY(TIME, ENERGY, ENUNIT)
CALL PRINTPHASE(QVAR(2), PVAR(2), PSUNIT)

C Integrate.
DO 10, 11 = 1, MAXSTEP
  TIME = TIME + TIMESTEP
  TEMP(1) = QVAR(1)
  TEMP(2) = QVAR(2)
  TEMP(3) = PVAR(1)
  TEMP(4) = PVAR(2)
  CALL RK4QP(TIME, TIMESTEP, QVAR, PVAR, DEGREES, DIMENS)
10 IF ( MOD(11, PRINTSTEP) .EQ. 0 ) THEN
    C Plot the Poincare Section for the trajectories that cross the
    C q1 = 0 axis while dq1/dt > 0.
    IF ( TEMP(1)*QVAR(1) .LT. ZERO ) THEN

FACTOR = ONE/( TEMP(1) – QVAR(1) )
CI = TEMP(1)*FACTOR
DI = -QVAR(1)*FACTOR

C Scale and plot the points (q2, p2).
Q2 = CI*QVAR(2) + DI*TEMP(2)
P2 = CI*PVAR(2) + DI*TEMP(4)
CALL GETENERGY(QVAR, PVAR, DEGREES, ENERGY)
CALL PRINTENERGY(TIME, ENERGY, ENUNIT)
CALL PRINTPHASE(Q2, P2, PSUNIT)
ENDIF

ENDIF
10 CONTINUE
CLOSE(ENUNIT)
CLOSE(PSUNIT)
END

F.2 Main Driver Using Second Order Symplectic Integrator

F.2.1 MAIN

Main driver for integrating the equations of motion using an explicit, second order Ruth-type symplectic integrator.

******************************************************************************
** Author: Paul Burkhardt
** Date: September 28, 2004
** Program: MAIN
** Symplectic integration for separable (potential-form)
** Hamiltonians of the form: H = T(p) + V(q)
** Calculates dynamics for the Henon-Heiles stellar motion problem.
******************************************************************************

PROGRAM MAIN
IMPLICIT NONE
INCLUDE 'NUMCON.FOR'
INTEGER $I1
INTEGER $MAXSTEP, $PRINTSTEP
INTEGER $ENUNIT, $PSUNIT
CHARACTER $ENFILE*(*) , $PSFILE*(*)
DOUBLE PRECISION $TIME, $TIMESTEP, $ENERGY, $SIMTIME
INTEGER $DEGREES

216
PARAMETER (DEGREES = 2)

DOUBLE PRECISION
$ QVAR(\text{DEGREES})$
$ PVAR(\text{DEGREES})$
$ \text{TEMP}(2 \times \text{DEGREES})$

PARAMETER (TIMESTEP = 1.0D0/6.0D0)
PARAMETER (MAXSTEP = 1200000)
PARAMETER (PRINTSTEP = 1)
PARAMETER (ENUNIT = 1)
PARAMETER (PSUNIT = 10)
PARAMETER (ENFILE = 'energy_sia2.dat')
PARAMETER (PSFILE = 'phasespace_sia2.dat')

DOUBLE PRECISION
$ \text{THIRD},$
$ \text{VE},$
$ \text{HE}$

PARAMETER (THIRD = ONE/THREE)
PARAMETER (HE = 0.029952D0)

DOUBLE PRECISION
$ \text{CI}, \text{DI}, \text{FACTOR},$
$ \text{Q2}, \text{P2}$

C Open files.
OPEN (UNIT = ENUNIT, FILE = ENFILE)
OPEN (UNIT = PSUNIT, FILE = PSFILE)

C Initialisation.
SIMTIME = MAXSTEP \times \text{TIMESTEP}
TIME = ZERO

DO 5, I1 = 1, DEGREES
  QVAR(I1) = 0.12D0
  PVAR(I1) = 0.12D0
5 CONTINUE

C Determine the momentum in the x direction.
VE = QVAR(2) \times QVAR(2) \times (HALF - \text{THIRD} \times QVAR(2))
  + QVAR(1) \times QVAR(1) \times (QVAR(2) + HALF)
IF (VE .GT. HE) THEN
  WRITE(*,*) 'Invalid energy!'
  STOP
ENDIF

PVAR(1) = DSQRT( TWO \times (HE - VE) - PVAR(2) \times PVAR(2))
CALL GETENERGY(QVAR, PVAR, DEGREES, ENERGY)
CALL PRINTENERGY(TIME, ENERGY, ENUNIT)
CALL PRINTPHASE(QVAR(2), PVAR(2), PSUNIT)

C Integrate.
DO 10, I1 = 1, MAXSTEP
  TIME = TIME + TIMESTEP
  \text{TEMP}(1) = QVAR(1)
  \text{TEMP}(2) = QVAR(2)
  \text{TEMP}(3) = PVAR(1)
  \text{TEMP}(4) = PVAR(2)
  CALL SIA2(TIMESTEP, QVAR, PVAR, DEGREES)
  IF (MOD(I1, PRNITSTEP) .EQ. 0) THEN
    C Plot the Poincare Section for the trajectories that cross the
    C \text{q1} = 0 axis while dq1/dt > 0.
    IF (\text{TEMP}(1) \times QVAR(1) \LT. ZERO) THEN
      FACTOR = ONE/(\text{TEMP}(1) - \text{QVAR}(1))
      CI = \text{TEMP}(1) \times \text{FACTOR}
      DI = -\text{QVAR}(1) \times \text{FACTOR}
    ENDIF

    Q2 = CI \times QVAR(2) + DI \times \text{TEMP}(2)
    P2 = CI \times PVAR(2) + DI \times \text{TEMP}(4)
    CALL GETENERGY(QVAR, PVAR, DEGREES, ENERGY)
    CALL PRINTENERGY(TIME, ENERGY, ENUNIT)
    CALL PRINTPHASE(Q2, P2, PSUNIT)
 ENDIF
10 CONTINUE
F.3 Main Driver Using Fourth Order Symplectic Integrator

F.3.1 MAIN

Main driver for integrating the equations of motion using an explicit, fourth order Ruth-type symplectic integrator.

```fortran
PROGRAM MAIN
IMPLICIT NONE
INCLUDE 'NUMCON.FOR'
INTEGER $ I1
INTEGER $ MAXSTEP, $ PRNTSTEP
INTEGER $ ENUNIT, $ PSUNIT
CHARACTER $ ENFILE(*), $ PSFILE(*)
DOUBLE PRECISION $ TIME, $ TIMESTEP, $ ENERGY, $ SIMTIME
INTEGER $ DEGREES
PARAMETER (DEGREES = 2)
DOUBLE PRECISION $ QVAR(DEGREES), $ PVAR(DEGREES), $ TEMP(2*DEGREES)
PARAMETER (TIMESTEP = 1.0D0/6.0D0)
PARAMETER (MAXSTEP = 1200000)
PARAMETER (PRNTSTEP = 1)
PARAMETER (ENUNIT = 1)
```
PARAMETER(PSUNIT = 10)
PARAMETER(ENFILE = 'energy_sia4.dat')
PARAMETER(PSFILE = 'phasespace_sia4.dat')

DOUBLE PRECISION
\$ THIRD,
\$ VE,
\$ HE

PARAMETER(THIRD = ONE/THREE)
PARAMETER(HE = 0.029952D0)

DOUBLE PRECISION
\$ THIRD,
\$ VE,
\$ HE

DOUBLE PRECISION
\$ CI , DI , FACTOR,
\$ Q2, P2

* Open files. *
OPEN(UNIT = ENUNIT, FILE = ENFILE)
OPEN(UNIT = PSUNIT, FILE = PSFILE)

* Initialization. *
SIMTIME = MAXSTEP*TIMESTEP
TIME = ZERO
DO 5 I1 = 1, DEGREES
QVAR(I1) = 0.12D0
PVAR(I1) = 0.12D0
5 CONTINUE

* Determine the momentum in the x direction. *
VE = QVAR(2)*QVAR(2)*((HALF - THIRD*QVAR(2)) + QVAR(1)*QVAR(1)*(QVAR(2) + HALF))
IF (VE .GT. HE) THEN
WRITE(*,*) 'Invalid energy!'
STOP
ENDIF
PVAR(1) = DSQRT( TWO*(HE - VE) - PVAR(2)*PVAR(2))
CALL GETENERGY(QVAR, PVAR, DEGREES, ENERGY)
CALL PRINTENERGY(TIME, ENERGY, ENUNIT)
CALL PRINTPHASE(QVAR(2), PVAR(2), PSUNIT)

* Integrate. *
DO 10 I1 = 1, MAXSTEP
TIME = TIME + TIMESTEP
TEMP(1) = QVAR(1)
TEMP(2) = QVAR(2)
TEMP(3) = PVAR(1)
TEMP(4) = PVAR(2)
CALL SIA4(TIMESTEP, QVAR, PVAR, DEGREES)
IF (MOD(I1, PRNTSTEP) .EQ. 0) THEN
C Plot the Poincare Section for the trajectories that cross the qi = 0 axis while dq1/dt > 0.
IF (TEMP(1)*QVAR(1) LT ZERO) THEN
FACTOR = ONE/((TEMP(1) - QVAR(1))
CI = TEMP(1)*FACTOR
DI = -QVAR(1)*FACTOR
ENDIF
Q2 = CI*QVAR(2) + DI*TEMP(2)
P2 = CI*PVAR(2) + DI*TEMP(4)
CALL GETENERGY(QVAR, PVAR, DEGREES, ENERGY)
CALL PRINTENERGY(TIME, ENERGY, ENUNIT)
CALL PRINTPHASE(Q2, P2, PSUNIT)
ENDIF
10 CONTINUE

CLOSE(ENUNIT)
CLOSE(PSUNIT)
END
F.4 Classical Trajectory Programs

F.4.1 EQNMOT

Calculate the equations of motion.

```
SUBROUTINE EQNMOT(QVAR, PVAR, DEGREES, VELOC, FORCE)
IMPLICIT NONE
INCLUDING 'INTG.FOR'
INCLUDING 'NUMCON.FOR'
INTEGER I1
DO 10, I1 = 1, DEGREES
VELOC(I1) = PVAR(I1)
10 CONTINUE

SUBROUTINE GETENERGY(QVAR, PVAR, DEGREES, ENERGY)
IMPLICIT NONE
INCLUDING 'NUMCON.FOR'
```

F.4.2 GETENERGY

Calculate the total energy from the Hamiltonian.

```
SUBROUTINE GETENERGY(QVAR, PVAR, DEGREES, ENERGY)
IMPLICIT NONE
INCLUDING 'NUMCON.FOR'
```

220
INCLUDE 'INTG.FOR'

INTEGER I1

DOUBLE PRECISION ENERGY

DOUBLE PRECISION THIRD
PARAMETER (THIRD = ONE/THREE)

ENERGY = ZERO

C Calculate Kinetic Energy.
DO 10, I1 = 1, DEGREES
  ENERGY = ENERGY + PVAR(I1)*PVAR(I1)
CONTINUE
  ENERGY = HALF*ENERGY

C Calculate Potential Energy.
ENERGY = ENERGY + QVAR(2)*QVAR(2)* ( HALF - THIRD*QVAR(2) )
  + QVAR(1)*QVAR(1)* ( QVAR(2) + HALF )
RETURN
END
Appendix G

Classical Trajectory Programs For Simple Harmonic Oscillator

G.1 Main Driver Using Fourth Order Runge-Kutta Integrator

G.1.1 MAIN

Main driver for integrating the equations of motion using a fourth order Runge-Kutta integrator.

******************************************************************************
* Author: Paul Burkhardt
* Date: September 28, 2004
* Program: MAIN
* Non-symplectic integration for separable (potential-form) Hamiltonians of the form: H = T(p) + V(q)
* Calculates dynamics for a Simple Harmonic Oscillator (S.H.O).
******************************************************************************

PROGRAM MAIN
IMPLICIT NONE

INCLUDE 'NUMCON.FOR'

INTEGER $ I1
INTEGER $ MAXSTEP,
$ PRNTSTEP
INTEGER $ ENUIT,
$ PSUNIT
CHARACTER $ ENFILE+(*)
$ PSFILE+(*)
DOUBLE PRECISION
$ TIME,
$ TIMESTEP,
$ ENERGY,
$ SIMTIME

INTEGER
$ DEGREES,
$ DIMENS

PARAMETER (DEGREES = 1, DIMENS = 2*DEGREES)

DOUBLE PRECISION
$ QVAR(DEGREES),
$ PVAR(DEGREES)

PARAMETER (TIMESTEP = 1.0D0/10.0D0)
PARAMETER (MAXSTEP = 1000000)
PARAMETER (PRINTSTEP = 1000)
PARAMETER (ENUNIT = 1)
PARAMETER (PSUNIT = 10)
PARAMETER (ENFILE = 'energy_rk4.dat')
PARAMETER (PSFILE = 'phasespace_rk4.dat')

C Open files.
OPEN (UNIT = ENUNIT, FILE = ENFILE)
OPEN (UNIT = PSUNIT, FILE = PSFILE)

C Initialization.
SIMTIME = MAXSTEP*TIMESTEP
TIME = ZERO

DO 5, 11 = 1, DEGREES
  QVAR(11) = ONE
  PVAR(11) = ZERO
5 CONTINUE

CALL GETENERGY(QVAR, PVAR, DEGREES, ENERGY)
CALL PRINTENERGY(TIME, ENERGY, ENUNIT)
CALL PRINTPHASE(QVAR, PVAR, PSUNIT)

C Integrate.
DO 10, 11 = 1, MAXSTEP
  TIME = TIME + TIMESTEP
  CALL RK4QP(TIME, TIMESTEP, QVAR, PVAR, DEGREES, DIMENS)
  IF ( MOD(11, PRINTSTEP) EQ 0 ) THEN
    CALL GETENERGY(QVAR, PVAR, DEGREES, ENERGY)
    CALL PRINTENERGY(TIME, ENERGY, ENUNIT)
    CALL PRINTPHASE(QVAR, PVAR, PSUNIT)
  ENDIF
10 CONTINUE

CLOSE(ENUNIT)
CLOSE(PSUNIT)
END
G.2 Main Driver Using Second Order Symplectic Integrator

G.2.1 MAIN

Main driver for integrating the equations of motion using an explicit, second order Ruth-type symplectic integrator.

```
PROGRAM MAIN
IMPLICIT NONE
INCLUDE 'NUMCON.FOR'

INTEGER $ I1$
INTEGER $ MAXSTEP$, $ PRNTSTEP$
INTEGER $ ENUNIT$, $ PSUNIT$
CHARACTER $ ENFILE(*), PSFILE(*)$
DOUBLE PRECISION $ TIME$, $ TIMESTEP$, $ ENERGY$, $ SIMTIME$
INTEGER $ DEGREES$
PARAMETER (DEGREES = 1)
DOUBLE PRECISION $ QVAR(DEGREES)$, $ PVAR(DEGREES)$
PARAMETER (TIMESTEP = 1.0D0/10.0D0)
PARAMETER (MAXSTEP = 10000000)
PARAMETER (PRINTSTEP = 1000)
PARAMETER (ENUNIT = 1)
PARAMETER (PSUNIT = 10)
PARAMETER (ENFILE = 'energy_sia2.dat')
PARAMETER (PSFILE = 'phasespace_sia2.dat')

C Open files.
OPEN(UNIT = ENUNIT, FILE = ENFILE)
OPEN(UNIT = PSUNIT, FILE = PSFILE)

C Initialization.
SIMTIME = MAXSTEP*TIMESTEP
TIME = ZERO

DO 5, I1 = 1, DEGREES
     QVAR(I1) = ONE
     PVAR(I1) = ZERO
     CONTINUE
```

224
CALL GETENERGY(QVAR, PVAR, DEGREES, ENERGY)
CALL PRINTENERGY(TIME, ENERGY, ENUNIT)
CALL PRINTPHASE(QVAR, PVAR, PSUNIT)

C Integrate.
DO 10, I1 = 1, MAXSTEP
   TIME = TIME + TIMESTEP
   CALL SIA2(TIMESTEP, QVAR, PVAR, DEGREES)
   IF ( MOD(I1, PRNTSTEP) .EQ. 0 ) THEN
      CALL GETENERGY(QVAR, PVAR, DEGREES, ENERGY)
      CALL PRINTENERGY(TIME, ENERGY, ENUNIT)
      CALL PRINTPHASE(QVAR, PVAR, PSUNIT)
   ENDIF
10 CONTINUE
CLOSE(ENUNIT)
CLOSE(PSUNIT)
END

G.3 Main Driver Using Fourth Order Symplectic Integrator

G.3.1 MAIN

Main driver for integrating the equations of motion using an explicit, fourth order Ruth-type symplectic integrator.

******************************************************************************
* Author: Paul Burkhardt
* Date: September 28, 2004
* Program: MAIN
* Symplectic integration for separable (potential-form) Hamiltonians of the form: H = T(p) + V(q)
* Calculates dynamics for a Simple Harmonic Oscillator (S.H.O).
******************************************************************************

PROGRAM MAIN
IMPLICIT NONE
INCLUDE 'NUMCON.FOR'

INTEGER $ I1
INTEGER $ MAXSTEP,
$ PRNTSTEP
INTEGER $ ENUNIT,
$ PSUNIT
CHARACTER $ ENFILE*(*)
$ PSFILE*(*)
DOUBLE PRECISION $ TIME,
$ TIMESTEP,
$ ENERGY,

225
G.4 Classical Trajectory Programs

G.4.1 EQNMOT

Calculate the equations of motion.
G.4.2 GETENERGY

Calculate the total energy from the Hamiltonian.

```fortran
SUBROUTINE GETENERGY(QVAR, PVAR, DEGREES, ENERGY)
IMPLICIT NONE
INCLUDE 'NUMCON.FOR'
INCLUDE 'INTG.FOR'
INTEGER I1
DOUBLE PRECISION ENERGY
ENERGY = ZERO
C Calculate total energy.
DO 10, I1 = 1, DEGREES
ENERGY = ENERGY + ( QVAR(I1)+QVAR(I1) + PVAR(I1)+PVAR(I1) )
10 CONTINUE
ENERGY = HALF*ENERGY
RETURN
END
```

227
Appendix H

Classical Trajectory Programs For Simple Unperturbed Pendulum

H.1 Main Driver Using Fourth Order Runge-Kutta Integrator

H.1.1 MAIN

Main driver for integrating the equations of motion using a fourth order Runge-Kutta integrator.

```fortran
PROGRAM MAIN
IMPLICIT NONE
INCLUDE 'NUMCON.FOR'

INTEGER $I1
INTEGER $MAXSTEP,$ PRINTSTEP
INTEGER $ENUNIT,$ PSUNIT
CHARACTER $ENFILE*(*),$ PSFILE*(*)
```

228
DOUBLE PRECISION
$ TIME,$
$ TIMESTEP,$
$ ENERGY,$
$ SIMTIME$

INTEGER
$ DEGREES,$
$ DIMENS$

PARAMETER (DEGREES = 1, DIMENS = 2*DEGREES)

DOUBLE PRECISION $ QVAR(DEGREES),$ $ PVAR(DEGREES)$

PARAMETER (TIMESTEP = 1.0D0/10.0D0)
PARAMETER (MAXSTEP = 1000000)
PARAMETER (PRINTSTEP = 1000)
PARAMETER (ENUNIT = 1)
PARAMETER (PSUNIT = 10)
PARAMETER (ENFILE = 'energy_rk4.dat')
PARAMETER (PSFILE = 'phasespace_rk4.dat')

C Open files.
OPEN (UNIT = ENUNIT, FILE = ENFILE)
OPEN (UNIT = PSUNIT, FILE = PSFILE)

C Initialization.
SIMTIME = MAXSTEP*TIMESTEP
TIME = ZERO

DO 5, I1 = 1, DEGREES
QVAR(I1) = ONE
PVAR(I1) = ZERO
5 CONTINUE

CALL GETENERGY(QVAR, PVAR, DEGREES, ENERGY)
CALL PRINTENERGY(TIME, ENERGY, ENUNIT)
CALL PRINTPHASE(QVAR, PVAR, PSUNIT)

C Integrate.
DO 10, I1 = 1, MAXSTEP
TIME = TIME + TIMESTEP
CALL RK4QP(TIME, TIMESTEP, QVAR, PVAR, DEGREES, DIMENS)

IF ( MOD(I1, PRINTSTEP) .EQ. 0 ) THEN
CALL GETENERGY(QVAR, PVAR, DEGREES, ENERGY)
CALL PRINTENERGY(TIME, ENERGY, ENUNIT)
CALL PRINTPHASE(QVAR, PVAR, PSUNIT)
ENDIF
10 CONTINUE

CLOSE(ENUNIT)
CLOSE(PSUNIT)
END
H.2 Main Driver Using Second Order Symplectic Integrator

H.2.1 MAIN

Main driver for integrating the equations of motion using an explicit, second order Ruth-type symplectic integrator.

```fortran
PROGRAM MAIN
IMPLICIT NONE
INCLUDE 'NUMCON.FOR'

INTEGER $ I1
INTEGER $ MAXSTEP, $ PRINTSTEP
INTEGER $ ENUNIT, $ PSUNIT
CHARACTER $ ENFILE(*), $ PSFILE(*)

DOUBLE PRECISION $ TIME, $ TIMESTEP, $ ENERGY, $ SIMTIME
INTEGER $ DEGREES

PARAMETER (DEGREES = 1)

DOUBLE PRECISION $ QVAR(DEGREES), $ PVAR(DEGREES)

PARAMETER (TIMESTEP = 1.0D0/10.0D0)
PARAMETER (MAXSTEP = 10000000)
PARAMETER (PRINTSTEP = 1000)
PARAMETER (ENUNIT = 1)
PARAMETER (PSUNIT = 10)
PARAMETER (ENFILE = 'energy_sia2.dat')
PARAMETER (PSFILE = 'phasespace_sia2.dat')

OPEN files.
OPEN (UNIT = ENUNIT, FILE = ENFILE)
OPEN (UNIT = PSUNIT, FILE = PSFILE)

C Initialization
SIMTIME = MAXSTEP*TIMESTEP
TIME = ZERO

DO 5, I1 = 1, DEGREES
QVAR(I1) = ONE
PVAR(I1) = ZERO
5 CONTINUE
```

230
CALL GETENERGY(QVAR, PVAR, DEGREES, ENERGY)
CALL PRINTENERGY(TIME, ENERGY, ENUNIT)
CALL PRINTPHASE(QVAR, PVAR, PSUNIT)

CALL SIA2(TIMESTEP, QVAR, PVAR, DEGREES)

IF ( MOD(I1, PRNTSTEP) .EQ. 0 ) THEN
    CALL GETENERGY(QVAR, PVAR, DEGREES, ENERGY)
    CALL PRINTENERGY(TIME, ENERGY, ENUNIT)
    CALL PRINTPHASE(QVAR, PVAR, PSUNIT)
ENDIF

CONTINUE

CLOSE(ENUNIT)
CLOSE(PSUNIT)
END

H.3 Main Driver Using Fourth Order Symplectic Integrator

H.3.1 MAIN

Main driver for integrating the equations of motion using an explicit, fourth order Ruth-type symplectic integrator.

*******************************************************************************
* Author: Paul Burkhardt
* Date: September 28, 2004
* Program: MAIN
* Symplectic integration for separable (potential-form) Hamiltonians of the form: H = T(p) + V(q)
* Calculates dynamics for a Simple Unperturbed Pendulum (S.U.P).
*******************************************************************************
H.4 Classical Trajectory Programs

H.4.1 EQNMOT

Calculate the equations of motion.

******************************************************************************
* Author: Paul Burkhardt
* Date: September 28, 2004
* Program: EQNMOT
* Purpose: Calculate canonical equations of motion for the Simple Unperturbed Pendulum.
******************************************************************************
H = 1/2(p^2) + (1 - cosq)

 dq/dt = dH/dp = velocity
dp/dt = -dH/dq = force

 SUBROUTINE EQNMOT(QVAR, PVAR, DEGREES, VELOC, FORCE)
 IMPLICIT NONE
 INCLUDE 'INTG.FOR'
 INCLUDE 'NUMCON.FOR'
 INTEGER  I1
 DOUBLE PRECISION $ VELOC(DEGREES) ,
 $ FORCE(DEGREES)

 C Calculate position rate of change.
 DO 10, I1 = 1 , DEGREES
 VELOC(I1) = PVAR(I1)
 10 CONTINUE

 C Calculate momentum rate of range.
 FORCE(1) = -DSIN(QVAR(1))
 RETURN
 END

H.4.2 GETENERGY

Calculate the total energy from the Hamiltonian.

 SUBROUTINE GETENERGY(QVAR, PVAR, DEGREES, ENERGY)
 IMPLICIT NONE
 INCLUDE 'NUMCON.FOR'
 INCLUDE 'INTG.FOR'
 INTEGER  I1
 DOUBLE PRECISION ENERGY
 DOUBLE PRECISION THIRD
 PARAMETER(THIRD = ONE/THREE)

 ENERGY = ZERO

 C Calculate Kinetic Energy.
 DO 10, I1 = 1 , DEGREES
 ENERGY = ENERGY + PVAR(I1)*PVAR(I1)
 10 CONTINUE

 ENERGY = HALF*ENERGY

 C Calculate Potential Energy.
 ENERGY = ENERGY + ( ONE - DCOS(QVAR(1)) )
 RETURN
 END
Appendix I

Appendix I: Utility Programs

I.1 Utility Programs

I.1.1 DERIV

Calculate the equations of motion. Used by the Fourth-Order Runge-Kutta integrator routine.

```fortran
SUBROUTINE DERIV(X, Y, DYDX, DEGREES)
IMPLICIT NONE

INCLUDE 'INTG.FOR'

INTEGER I1

DOUBLE PRECISION $ X, $ DYDX(*), $ Y(*)

DOUBLE PRECISION $ VELOC(DEGREES) , $ FORCE(DEGREES)

DO 5, I1 = 1, DEGREES
QVAR( I1 ) = Y( I1 )
PVAR( I1 ) = Y( I1 + DEGREES)
5 CONTINUE

CALL EGNMOT(QVAR, PVAR, DEGREES, VELOC, FORCE)

DO 10, I1 = 1, DEGREES
DYDX( I1 ) = VELOC( I1 )
DYDX( I1 + DEGREES) = FORCE( I1 )
10 CONTINUE
```

Author: Paul Burkhardt
Date: September 28, 2004

Program: DERIV

Calculate derivatives for Hamiltonian Equations of Motion.
Appendix J

General Purpose Makefile

J.1 Makefile

Modify macros and targets as desired.

```make
# Author: Paul Burkhardt
# Date: September 28, 2004
#
# Makefile
#
#
# MACRO DEFINITIONS
#
# Suffixes (file extensions) MAKE should recognize.
.SUFFIXES: .c .cc .C .cpp .f .F .FOR .f77 .f90 .f95 .o
#
# Directories
#
INCDIR = ../../include
MODDIR = ./module
PESDIR = ../../potential

# Directory search paths.
# MAKE will search these paths for files.
# DO NOT modify the default VPATH or vpath names.
VPATH = ./... $(MODDIR) $(PESDIR)

# Filenames and installation paths.

LIB =
LIB_PATH = /usr/lib
EXE =
EXE_PATH = .

# Source code.

CPP_SRC = $(wildcard *.cpp)
F77_SRC = $(wildcard *.f)
F90_SRC = $(wildcard *.f90)
```
# Object files.

CPP_OBJ = $(CPP_SRC:.cpp=.o)
F77_OBJ = $(F77_SRC:.f=.o)
F90_OBJ = $(F90_SRC:.f90=.o)

# Compiler Options

# Compilers.
CPP_COMPILER = g++
F77_COMPILER = g77
F90_COMPILER = g95
ARCHIVE = ar
ARCHIVE_FLAGS = -rvu
GET = get
GET_FLAGS =
WORD_SIZE =
DEBUG = -g
PROFILE = -p6
OVERRIDE = -O2
FAST = -malign-double -ffloat-store -ffast-math
OPTIMIZE = $(OVERRIDE) $(FAST)
ECHO = echo -e
LMAP =

# Includes.
#include =
-I . ./include
-I .../include
-I $(INCDIR)

# Libraries.
LIBRARY =

LINK = $(LIBRARY)

# Compiler flags

CPP_FLAGS = $(include) $(WORD_SIZE)
F77_FLAGS = $(include) $(WORD_SIZE) -ff77
F90_FLAGS = $(include) $(WORD_SIZE)

# TARGET RULES

# Force explicit target.
NULL:
  $(ECHO) ""
  $(ECHO) "***********************************************************************************"
  $(ECHO) "Enter make exe or make lib"
  $(ECHO) "***********************************************************************************"

# Generic target for redirection.
lib: $(LIB)
exe: $(EXE)

# Link and build executable.
$(EXE): $(CPP_OBJ) $(F77_OBJ) $(F90_OBJ)
  $(F77_COMPILER) $? $(LINK) -o $@

# Make library archive.
$(LIB): $(CPP_OBJ) $(F77_OBJ) $(F90_OBJ)
  $(ARCHIVE) $(ARCHIVE_FLAGS) $0 $? ranlib $0
    make tidy

#
# Compile source without linking to build object files.
#
.c.o:  $(CPP\_COMPILER) $(CPP\_FLAGS) −c $<
.f.o:  $(F77\_COMPILER) $(F77\_FLAGS) −c $<
.f90.o: $(F90\_COMPILER) $(F90\_FLAGS) −c $<

# Cleanup
#
tidy:  rm −f $(CPP\_OBJ) $(F77\_OBJ)

clean:  rm −f $(CPP\_OBJ) $(F77\_OBJ) $(LIB) $(EXE)
install:  mv $(LIB) $(LIB\_PATH)
uninstall:  rm −f $(LIB\_PATH)/$(LIB)

# Misc.
#
# Get source files from version control system.
get:  $(CPP\_SRC) $(F77\_SRC) $(F90\_SRC)
     $(GET) $?

# Show files.
show:  $(ECHO) $(CPP\_SRC) $(F77\_SRC) $(F90\_SRC)

# Display recently changed files.
print:  $(ECHO) $(CPP\_SRC) $(F77\_SRC) $(F90\_SRC)
     @$(ECHO) $?
     touch print
Curriculum Vitae

PAUL BURKHARDT

Academic background in theoretical Physical Chemistry. Research topics include development of a fast and accurate algorithm for computing angular momentum coupling coefficients and construction of a symplectic integration algorithm for three-body, classical Hamiltonian dynamics in symmetrized hyperspherical polar coordinates.

EDUCATION

University of Illinois at Urbana-Champaign

• Ph.D. in Physical Chemistry (2004)

Oregon State University at Corvallis

• B.S. in Chemistry (1994)

• Graduating Senior Award

• Phi Lambda Upsilon Chemistry Honor Society

• Phi Theta Kappa International Honor Society

PUBLICATIONS AND PROCEEDINGS

Journal Articles


Doctoral Dissertation [6 P. Burkhardt, Explicit, Multi-Map Symplectic Integrator For

Proceedings