THE PENNSYLVANIA STATE UNIVERSITY Test
SCHREYER HONORS COLLEGE

DEPARTMENT OF DEPARTMENT

TITLE OF THESIS

NAME
SEASON NOW

A thesis
submitted in partial fulfillment
of the requirements
for baccalaureate degrees
in Major1 and Major2
with honors in Area of Honors

Reviewed and approved* by the following:

Some Person
Professor of Something
Thesis Supervisor

Some Other Person
Professor of Something Else
Some Other Position
Honors Adviser

*Signatures are on file in the Schreyer Honors College.

Abstract

Abstract abstract abstract. Abstract? Abstract! Abstract...

Table of Contents

List of Figures

List of Tables

Acknowledgements

To Mr. Scruffles.

Chapter 1

Introduction and Historical Review

1.1 Introduction

Blah blah blah. Here is an example of how to include and cite a figure: see Figure 1.1

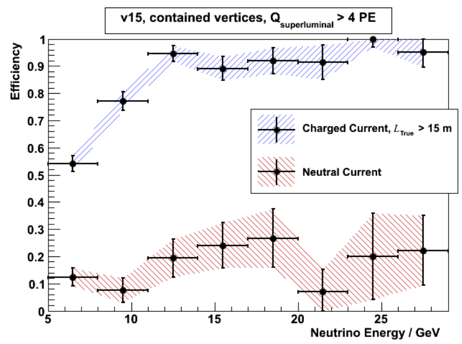

Figure 1.1: Caption to go underneath figure

1.2 Literature Review

Blah blah blah Here's an example of a table: see Table 1.1.

Parameter	**Current Best Value**
Δm^2_{21}	$7.50^{+0.19}_{-0.20} \times 10^{-3}$ eV^2
$\lvert\Delta m^2_{32}\rvert$	$2.32^{+0.12}_{-0.08} \times 10^{-5}$ eV^2
$\sin^2(\theta_{12})$	$0.857^{+0.023}_{-0.025}$
$\sin^2(2\theta_{23})$	> 0.95
$\sin^2(\theta_{13})$	0.098 ± 0.013

Table 1.1: Name of table for caption above table

Here's an example of an equation and some math:

$$U = \begin{pmatrix} c_{12}c_{13} & s_{12}c_{13} & s_{13}e^{i\delta} \\ -s_{12}c_{23} - c_{12}s_{23}s_{13}e^{i\delta} & c_{12}c_{23} - s_{12}s_{23}s_{13}e^{i\delta} & s_{23}c_{13} \\ s_{12}s_{23} - c_{12}c_{23}s_{13}e^{i\delta} & -c_{12}s_{12} - s_{12}c_{23}s_{13}e^{i\delta} & c_{23}c_{13} \end{pmatrix}, \tag{1.1}$$

with $c_{ij} = \cos\theta_{ij}$ and $s_{ij} = \sin\theta_{ij}$.

Here's an example citation: [1].

1.3 Thesis Outline

Impulsive thrust approximations are often used to simply the analysis of the optimal maneuver between circular orbits. In actuality, spacecraft exhibit finite thrust arcs to perform these orbital transfers. This thesis analyzes these minimum burn time, and thus fuel consumption, thrust arcs required for these optimal orbital transfers. Requiring two separate numerical integrations, this is computationally extensive. This thesis then seeks to develop methods to drastically reduce the time required for these computations through means of a C++ implementation utilizing Boost libraries and parallelization through OpenMP. Chapter 2 will detail the finite thrust transfer problem solved within this analysis, while Chapter 3 details the implementation details which were used. Chapter 4 then details the results of these implementations, including speedup comparisons and methods to greater improve the probability of the algorithm approaching an optimal solution.

Chapter 2

Problem Statement

2.1 Problem Definition

This thesis analyzes the optimal finite thrust transfer between two circular orbits. A PSO approach is utlized to attempt to find and optimal solution. The development of this problem is derived from [2].

This transfer is developed with respect to an initial circular orbit of radius R_1 and a final circular orbit of radius R_2 subject to the conditions $R_1 > R_2$ where the parameter $\beta = R_2/R_1 > 1$. The inertial reference frame selected for this problem definition is centered at the attracting body. The corresponding coordinate frame definition is as follows: The x axis is aligned with the spacecraft at the initial time t_0 and the y axis is located in the orbital plane. The z axis is determined by the right hand rule of these two basis axis. Within this problem v_r denotes the spacecraft radial velocity, v_θ the spacecraft the horizontal component of velocity, r the radisu, and ζ the angular displacement from the x axis. The gravitational parameter of the attracting body is μ_B.

Initial conditions at time t_0 are given by

$$v_r(t_0) = 0 \quad v_\theta(t_0) = \sqrt{\mu_B/R_1} \quad r(t_0) = R_1 \quad \xi(t_0) = 0 \tag{2.1}$$

And final conditions given by

$$v_r(t_f) = 0 \quad v_\theta(t_f) = \sqrt{\mu_B/R_2} \quad r(t_f) = R_2 \tag{2.2}$$

The problem assumes a transfer trajectory of an initial thrust arc, followed by a coasting arc, and ending with a second thrust arc. Optimization of this problem corresponds to determining the thrust pointing angle function corresponding to the minimization of propellant consumption subject to the constraints of equation 2.2.

Additional assumptions are made to simplify the analysis:

1. Throughout the duration of each thrust arc maximum thrust is generated
2. Each thrust arc pointing angle is represented as a third-degree polynomial as a function of time.

T and c are used to represent the spacecraft thrust level and effective exhaust velocity respectively. The previous assumptions indicate that the thrust-to-mass ratio has the following form

$$\frac{T}{m} = \begin{cases} \dfrac{T}{m_0 - \frac{T}{c}t} = \dfrac{cn_0}{c - n_0 t} & 0 \leq t \leq t_1 \\ 0 & t_1 \leq t \leq t_2 \\ \dfrac{T}{m_0 - \frac{T}{c}(t_1 + t - t_2)} = \dfrac{cn_0}{c - n_0(t_1 + t - t_2)} & t_2 \leq t \leq t_f \end{cases} \tag{2.3}$$

Where m_0 is the initial spacecraft mass and n_0 is the initial thrust to mass ratio at t_0. The state space equations of motion for the spacecraft are as follows

$$\dot{v} = -\frac{\mu_B - r v_\theta^2}{r^2} + \frac{T}{m}\sin\delta \tag{2.4}$$

$$\dot{v_\theta} = -\frac{v_r v_\theta}{r} + \frac{T}{m}\cos\delta \tag{2.5}$$

$$\dot{r} = v_r \tag{2.6}$$

$$\dot{\xi} = \frac{v_\theta}{r} \tag{2.7}$$

where $\dfrac{T}{m}$ is given in equation 2.3 and δ is the thrust pointing angle represent by a third-order polynomial of time. The state vector used in this problem is $[x_1 \;\; x_2 \;\; x_3 \;\; x_4]^T = [v_r \;\; v_\theta \;\; r \;\; \xi]^T$.

The thurst pointing angle δ is defined as

$$\delta = \begin{cases} \zeta_0 + \zeta_1 t + \zeta_2 t^2 + \zeta_3 t^3 & 0 \leq t \leq t_1 \\ \nu_0 + \nu_1(t - t_2) + \nu_2(t - t_2)^2 + \nu_3(t - t_3)^3 & t_2 \leq t \leq t_f \end{cases} \tag{2.8}$$

Where the optimum thrust pointing angle coefficients $\{\zeta_0 \;\; \zeta_1 \;\; \zeta_2 \;\; \zeta_3\}$ and $\{\nu_0 \;\; \nu_1 \;\; \nu_2 \;\; \nu_3\}$ are determined by PSO.

During the coasting arc the problem consists of a Kepler orbit. As such, the semimajor axis a and eccentricity e of the coasting arc can be computed

$$a = \frac{\mu_B r_1}{2\mu_B - r_1(v_{r1}^2 + v_{\theta 1}^2)} \tag{2.9}$$

$$e = \sqrt{1 - \frac{r_1^2 v_{\theta_1}^2}{\mu_B a}} \tag{2.10}$$

Provided that the orbit is elliptic ($a > 0$) the true anomaly at $t_1(f_1)$ can be computed as

$$\sin f_1 = \frac{v_{r1}}{e}\sqrt{\frac{a(1-e^2)}{\mu_B}} \quad \text{and} \quad \cos f_1 = \frac{v_{\theta_1}}{e}\sqrt{\frac{a(1-e^2)}{\mu_B}} - \frac{1}{e} \tag{2.11}$$

and the eccentric anomaly E_1 as

$$\sin E_1 = \frac{\sin f_1 \sqrt{1-e^2}}{1 + e\cos f_1} \quad \text{and} \quad \cos E_1 = \frac{\cos f_1 + e}{1 + e\cos f_1} \tag{2.12}$$

The PSO parameter ΔE represents the variation in eccentric anomaly througout the coasting arc. Therefore, the eccentric enomaly at t_2 is $E_2 = E_1 + \Delta E$. True anamoly can be thus determined utilizing the results of equations in 2.12

$$\sin f_2 = \frac{\sin E_2 \sqrt{1-e^2}}{1 - e\cos E_2} \quad \text{and} \quad \cos f_2 = \frac{\cos E_2 - e}{1 - e\cos E_2} \tag{2.13}$$

Furthermore, the coasting time interval t_{co} can be calculated through Kepler's law as

$$t_{co} \triangleq t_2 - t_1 = \sqrt{\frac{a^3}{\mu_B}}[E_2 - E_1 - e(\sin E_2 - \sin E_1)] \tag{2.14}$$

The results from 2.9, 2.10, and 2.13 provide the initial conditions required to numerical integrate the spacecraft equations of motion for the second thrust arc beginning at time t_2. These initial conditions for the second thrust arc are computed as

$$v_{r_2} = \sqrt{\frac{\mu_B}{a(1-e^2)}}e\sin f_2 \tag{2.15}$$

$$v_{\theta 2} = \sqrt{\frac{\mu_B}{a(1-e^2)}}(1 + e\cos f_2) \tag{2.16}$$

$$r_2 = \frac{a(1-e^2)}{1 + e\cos f_2} \tag{2.17}$$

$$\xi_2 = \xi_1 + (f_2 - f_1) \tag{2.18}$$

Integrating the second thrust arc with these initial conditions over the second thrust arc time duration with yield the final orbital characteristics.

This system depends on the eight coefficients representing the thrust pointing angles of the first and second thrust arcs, $\{\zeta_0\ \zeta_1\ \zeta_2\ \zeta_3\}$ and $\{\nu_0\ \nu_1\ \nu_2\ \nu_3\}$ respectively, and the time intervals for the first coast arc $\Delta t_1 \triangleq t_1$, the coasting arc, Δt_{co}, and the second coast arc $\Delta t_2 \triangleq t_f - t_2$. These 11 unknowns are sought to be chosen to minimize the objective function

$$J = \Delta t_1 + \Delta t_2 \tag{2.19}$$

Optimizing the unknown coefficients to minimize 2.19 corresponds to the minimization of propellant consumption. Minimizing propellant consumption results in a maximization of the final-to-initial mass ratio given by

$$\frac{m_f}{m_0} = \frac{m_0 - \frac{T}{c}\Delta t_1 - \frac{T}{c}\Delta t_2}{m_0} = 1 - \frac{n_0}{c}(\Delta t_1 + \Delta t_2) \tag{2.20}$$

The coasting arc is required to be elliptic, therefore any particles with $a \leq 0$ are assigned an infinite value.

Since Δt_{co} can be computed using ΔE, the eccentric anomaly variation replaces the coasting time interval in the problem's 11 unkown parameters. Each particle in the swarm thus consists of the following

$$\chi = [\zeta_0 \quad \zeta_1 \quad \zeta_2 \quad \zeta_3 \quad \nu_0 \quad \nu_1 \quad \nu_2 \quad \nu_3 \quad \Delta t_1 \quad \Delta E \quad \Delta t_2]^T \tag{2.21}$$

To enforce the constraints set forth in equation 2.2 three penalty terms are added to the objective function 2.19.

$$\tilde{J} = \Delta t_1 + \Delta t_2 + \sum_{k=1}^{3} \alpha_k |d_k| \tag{2.22}$$

where

$$d_1 = v_r(t_f) \qquad d_2 = v_\theta(t_f) - \sqrt{\frac{\mu_B}{R_2}} \qquad d_3 = r(t_f) - R_2 \tag{2.23}$$

A maximum acceptable error of 10^-3 is used and the α_k coefficients assigned as follows

$$\alpha_k = \begin{cases} 100 & |d_k| > 10^{-3} \\ 0 & |d_k| < 10^{-3} \end{cases} \tag{2.24}$$

2.2 Canonical Units Definition

This problem employs a canonical set of units to simplify the analysis. One distance unit (DU) is defined as the radius of the initial orbit, and one time unit (TU) definted such that $\mu_B = 1\frac{DU^3}{TU^2}$. The unknown coefficients are thus sought within the following ranges

$$0\,TU \leq t_1 \leq 3\,TU \quad 0 \leq \Delta E \leq 2\pi \quad 0\,TU \leq \Delta t_2 \leq 3\,TU \tag{2.25}$$
$$-1 \leq \xi_k \leq 1 \quad -1 \leq \nu_k \leq 1 \quad (k = 0, 1, 2, 3) \tag{2.26}$$

The effective exhaust velocity c is set to $0.5\frac{DU}{TU^2}$ and the initial thrust-to-mass ratio set to $0.2\frac{DU}{TU^2}$.

2.3 PSO Algorithm

2.3.1 General Overview

Particle Swarm Optimization is a class of algorithm that employs statistical methods to attempt to locate optimal solutions that minimize an objective function, often denoted J, in the global search space. The algorithm was originally designed to model the behavior of flocks of birds or schools of fish but has been adapted to solve a variety of mathematical problems. PSO is an effective method in scenarios where no analytical solution can be found and the optimal solution is unknown.

The potential solutions for PSO algorithms exist within an n dimensional space where n represents the number of unknown parameter values. Within this n dimensional space there is likely to be a variety of local minima in which the swarm, and thus solution, may converge on. As such, one is not guaranteed to find an optimal solution to a problem with a PSO algorithm. Indeed, in many cases it is impossible to know what the true global optimum is. This requires running the algorithm many times to increase the probability of finding a near-optimum solution. For many problems, this may require large computational efforts and extensive time. Reducing the time required for the execution of PSO algorithms allows more runs in a shorter duration, thus probabilistic increasing the chance of a near-optimal solution quicker. This opportunity was one of the core concepts explored within this thesis.

The algorithm consists of a swarm of particles, each initially containing randomly assigned values in the search space for each unknown parameter. These parameter values are referred to as a particles position, with each parameter also having a corresponding velocity. A given number of iterations is set, each in which each particle is evaluated for its fitness as defined by the objective function J. Following each iteration, each of the particles' parameters' position is updated with its velocity terms, and velocity updated based on a variety of factors

1. How far each of the particle's parameters differs from the global best particle ever recorded's parameters

2. How far each of the particle's parameters differs from the parameters of its historical personal best values

3. Its current velocity

This process continues for the set number of iterations. The global best particle, i.e. set of unknown parameters, found within the PSO algorithm is thus the solution. Psuedocode for the algorithm is as follows

Algorithm 1 General PSO Algorithm

Set Lower and Upper bounds on position and velocity
for *particle* **in** *Swarm* **do**
  Assign initial position values for each particle
end for
for *iteration* **in** *Num Iterations* **do**
  for *particle* **in** *Swarm* **do**
    Evaluate objective function J
  end for
  Record best set of parameters each particle has achieved $\leftarrow pBest$
  Record best overall position as global best $\leftarrow gBest$
  for *particle* **in** *Swarm* **do**
    Update particle velocity based on $pBest, gBest$, and current velocity
    if $v_i(k) <$ velocity lower bound **then**
      $v_i(k) =$ velocity lower bound
    else if $v_i(k) >$ velocity upper bound **then**
      $v_i(k) =$ velocity upper bound
    end if
    Update particle Position
    if $p_i(k) <$ position lower bound **then**
      $p_i(k) =$ position lower bound
      $v_i(k) = 0$
    else if $p_i(k) >$ position upper bound **then**
      $p_i(k) =$ position upper bound
      $v_i(k) = 0$
    end if
  end for
end for
Use $gBest$ as solution

2.3.2 Finite Thrust Arc Implementation

2.4 Reducing Computational Time

2.4.1 Problem Selection

The finite thrust arc problem explored within this thesis was chosen as a benchmark as a PSO algorithm requiring lots of computational resources. Particles within this problem require lots of execution time for two main reasons

1. Each particle requires two numerical integrations, one for the initial thrust arc and one for the second
2. Many potential solutions within the global search space do not converge during these numerical integrations. This drastically reduces integrations integration step sizes in an attempt to

meet the specified integration tolerances, massively increasing the computational complexity.

Additionally, this problem required a non-trivial 11 unknowns. This broad 11 dimensional search space necessitates a comparatively large swarm population size in the search for optimal values. This combination of individual particles requiring lots of execution with the desired swarm population size makes the problem explored within this thesis was sufficient in evaluating potential speedup methods.

2.4.2 Parallelization

The conditions referenced within section 2.4.1 make the finite thrust transfer problem a good candidate for parallelization. Within each iteration, each particle is evaluated for its value of the objective function J. This contains the computationally heavy numerical integrations. Within this section of the PSO algorithm, each particle is evaluated independently. As such, parallel processing can be used to evaluate multiple particles within the swarm simultaneously. Only after each iteration is complete and position and velocity updating occurs do the particles become dependent on each others' results. Since the updating portion of the algorithm is not computationally expensive, a large potential benefit can be gained from paralellizing the inter-iteration computations. A simplified version of the parallelized algorithm is shown below.

Algorithm 2 Simplified parallel PSO Implementation

```
for iteration in Num Iterations do
    for particle in Swarm do                       ⎫
        Evaluate objective function J              ⎬ in parallel
    end for                                        ⎭
    Update particle velocity based on pBest, gBest, and current velocity
    Update particle position
end for
```

Chapter 3

Methodology

3.1 Particle Swarm Optimization Overview

3.2 Problem Implementation

Stuff detailing the general rk-dopri-5 algorithm goes here

3.2.1 MATLAB

MATLAB implementation details go here e.g. ODE45, etc.

3.2.2 C++ Single Threaded

Details about Boost libraries, timers, etc. go here

3.2.3 C++ Parallelization

Details about OpenMP go Test

Chapter 4

Results

4.1 Numerical Results

4.2 Rehydration Results

4.3 Speedup

References

[1] J. Beringer et al. Review of particle physics. *Phys. Rev. D*, 86:010001, Jul 2012.

[2] Mauro Pontani and Bruce A. Conway. Particle swarm optimization applied to space trajectories. *Journal of Guidance, Control, and Dynamics*, 33(5):1429–1441, 2010.