CALIFORNIA NATIONAL UNIVERSITY

THE EFFICIENCY OF THE RUNGA-KUTTA AND SHOOTING METHODS AS AN AIMING SYSTEM FOR THE M829 ANTI-TANK PROJECTILE

AN ENGINEERING PROJECT REPORT SUBMITTED TO
THE FACULTY OF ENGINEERING
IN CANDIDACY FOR THE DEGREE OF
MASTER OF SCIENCES IN ENGINEERING

BY
CRAIG OWEN SMOOTHEY

NAIROBI, KENYA
23 JUNE 1999

ABSTRACT

The efficiency of a computerized aiming solution system for the M829 projectile, which uses the Runga-Kutta and Shooting Methods, is investigated and evaluated. Computer programs are presented for each algorithm. A realistic ballistic model is used. The number of iterations, and the time required by each algorithm, is measured under selected situations. Each algorithm’s efficiency is judged, and bottlenecks in the system are identified.

TABLE OF CONTENTS

Title Page / 1

Abstract / 2

Table of Contents / 3

The List of Tables / 7

The List of Figures / 8

I. The Problem and Its Setting / 10

- The Statement of the Problem / 10
- The Subproblems / 10
- The Hypotheses / 10
- The Definition of Terms / 11
- The Delimitations / 12
- The Assumptions / 13
- Abbreviations / 13
- The Importance of the Study / 14

II. The Review of the Related Literature / 16

- The M829 APFSDS-T Round / 16
- Target Motion / 17
- Exterior Ballistics / 19
- Initial-Value Methods / 20
- The Shooting Method / 21
- Summary / 22

III. An Overview of the Data and Its Interpretation / 23

- The Data / 23

General Criteria for the Admissibility of the Data / 23

General Treatment of the Data / 24

Systemic Treatment of the Data / 24

The Research Methodology / 24

Treatment of the Data for Each Subproblem / 25

Summary / 32

IV. The Ballistic Model / 33

Introduction / 33

The Definition of the Axes / 33

Gravitational Acceleration / 35

Air Density / 36

Drag Coefficient / 37

Frontal Drag Determination / 39

Z Axis Crosswind Drag Determination / 40

The Axial Acceleration Model / 41

Summary / 42

V. The Runga-Kutta Method and its Tests / 43

The Euler Method Code / 43

The Runga-Kutta Method Code / 44

The Step Size Accuracy Test / 47

The Runga-Kutta Impact Point Determination Code / 47

The Runga-Kutta Impact Point Test / 49

VI. The Shooting Method for Stationary Targets and its Tests / 50

The Method of Convergence / 50

The Test / 51

VII. The Shooting Method for Moving Targets and its Tests / 53

The Method of Convergence / 53

The Test / 53

VIII. The Results / 56

The Runga-Kutta vs. Euler-Method Step Size Accuracy Test / 56

The Runga-Kutta Iteration Test / 57

The Shooting Method for Stationary Targets Test / 61

The Shooting Method for Moving Targets Test / 66

IX. Discussion / 73

The Ballistic Model / 73

The Runga-Kutta Method / 74

The Shooting Method / 75

Efficiency vs. Accuracy / 76

X. Summary, Conclusions, And Recommendations / 78

Summary / 78

Conclusions / 79

Recommendations / 79

References / 80

Appendixes / 83

A. The Axial Acceleration Code / 83

B. The Euler Method Code / 89

C. The Runga-Kutta Code / 91

D. The RK Step Size Accuracy Code / 94

E. The Step Size Accuracy Test Batch File / 96

F. The Runga-Kutta Iteration Test Code / 97

G. The Shooting Method For Stationary Target Code / 99

H. The Shooting Method For Stationary Targets Test Code / 102

I. The Shooting Method For Moving Targets Code / 103

J. The Shooting Method For Moving Targets Test Code / 105

K. Runga-Kutta Method Vs. Euler Method Step Size Accuracy Results / 107

L. The Runga-Kutta Iteration Test Results / 108

M. The Shooting Method For Stationary Targets Test Results / 113

N. The Shooting Method For Moving Targets Test Results / 116

THE LIST OF TABLES

1. The G1 Drag Model / 38
2. The Axial Displacements Resulting from Seven Seconds of Ballistic Flight Time as Determined by the Runga-Kutta and Euler Methods / 56
3. The Minimum, Maximum, Mean and Standard Deviation of the Iteration Count Data from the Runga-Kutta Impact Point Test / 57
4. The Minimum, Maximum, Mean and Standard Deviation of the Computation Time Data from the Runga-Kutta Impact Point Test / 59
5. The Minimum, Maximum, Mean and Standard Deviation of the Iteration Count Data from the Shooting Method for Stationary Targets Test / 62
6. The Minimum, Maximum, Mean and Standard Deviation of the Compute Time Data from the Shooting Method for Stationary Targets Test / 64
7. The Minimum, Maximum, Mean and Standard Deviation of the Iteration Count Data from the Shooting Method for Moving Targets Test / 67
8. The Minimum, Maximum, Mean and Standard Deviation of the Compute Time Data from the Shooting Method for Moving Targets Test / 70

THE LIST OF FIGURES

1. The M829APFSDS-T round prior to being fired / 16
2. The M829 sabot breaking away from the projectile after firing / 17
3. The axial model / 34
4. The method of convergence in the shooting method for stationary targets / 51
5. The method of convergence in the shooting method for moving targets / 54
6. The number of iterations required the Runga-Kutta method to determine the impact point of a projectile launched at a certain elevation / 58
7. The computational time required the Runga-Kutta method to determine the impact point of a projectile launched at a certain elevation / 60
8. The number of iterations required the shooting method to determine an aiming solution against a stationary target / 63
9. The computational time required the shooting method to determine an aiming solution against a stationary target / 65
10. The impact of target speed and target heading on the average number of iterations required the shooting method to determine an aiming solution against a moving target / 68

11. The impact of range on the average number of iterations required the shooting method to determine an aiming solution against a moving target / 69

12. The impact of range on the average computational time required the shooting method to determine an aiming solution against a moving target / 71

13. The impact of target speed and target heading on the average computational time required the shooting method to determine an aiming solution against a moving target / 72

I. THE PROBLEM AND ITS SETTING

The Statement of the Problem

The study investigates and evaluates the efficiency of a computerized aiming system, which uses a combination of the Runga-Kutta and Shooting methods, to determine whether such a system will provide a combat advantage to tank crews. The study considers the M829 APFSDS-T projectile used by the M1A1 Abrams battle tank.

The Subproblems

1. The first subproblem is the investigation and evaluation of the efficiency of a three-dimensional, dynamic step size, fourth-order Runga-Kutta, numerical simulation for the projectile motion of the M829 APFSDS-T projectile.

2. The second subproblem is the investigation and evaluation of the efficiency of the shooting method as an aiming augmentation system for the M829 APFSDS-T projectile.

The Hypotheses

1. A three dimensional, dynamic step size, fourth-order Runga-Kutta, initial-value problem method with a relatively broad step size efficiently models the projectile motion of the M829 APFSDS-T projectile.

2. The Shooting method provides efficient aiming solutions for the M829 APFSDS-T projectile.

The Definitions of Terms

Efficient. A method is efficient if it provides a solution in less time than a human operator could with standard solution tables. This definition is different from the computational complexity definition of the term, where it is used to identify an algorithm requiring a low number of operations for a given task.

Numerical Method. A numerical method is an iterative algorithm for solving a mathematical problem by determining range values at discrete domain intervals throughout the problem domain. Domain interval solutions are determined sequentially or simultaneously. Numerical methods differ from analytical methods, in that analytical methods provide direct solutions at any point in the problem domain, whereas numerical methods are iterative.

Simulation. A simulation is an abstract computer model of a physical system that allows the behavior of the physical system to be reproduced, by computer, to some degree of accuracy.

Augmentation. Augmentation is an iterative process whereby some algorithm successively refines an estimated solution to some problem until it falls within some error bound of the true solution.

Step Size. The step size is the width of the interval between successive solution points in the domain of a problem being solved by a numerical method.

Shooting Method. The shooting method is an algorithm for solving complex mathematical problems by guessing solutions. It is combined with augmentation.

Initial-Value Problem. An initial-value problem is a calculus problem where dependent values (y, y', and perhaps y'') are provided at some initial independent value (x). The problem is to determine dependent values (y, y' and y'') at independent values subsequent to the initial independent value.

The Delimitations

1. The project does not consider internal ballistics. That is to say that the projectile behavior from the moment the shell powder ignites until the time the projectile leaves the weapon's barrel is not be considered.

2. The project does not consider terminal ballistics. That is to say that the project does not consider the behavior of the projectile beyond the moment of impact.

3. The project accepts, as input, the target range, target heading, target speed, air density, wind and local gravity. The project does not consider how this information is gathered.

4. This study does not perform live-fire testing of the computer simulation.

The Assumptions

1. A three dimensional, fourth-order Runga-Kutta initial-value problem method with a dynamic step size accurately models projectile motion.

2. The "shooting" method converges to an accurate aiming solution.

3. The "shooting" method can be optimized to provide rapid convergence.

4. Without loss of generality, the simulation assumes that the tank is level.

5. The project assumes that the target is moving in a straight line.

6. Without loss of generality, the project considers the targeted tank to be at the same altitude as the attacking tank, which is at sea level.

Abbreviations

RK is an abbreviation for Runga-Kutta.

SMST is an abbreviation for Shooting Method for Stationary Targets.

SMMT is an abbreviation for Shooting Method for Moving Targets.

ms is an abbreviation for millisecond.

us is an abbreviation for microsecond.

ns is an abbreviation for nanosecond.

The Importance of the Study

Standard ballistic tables exist for most weapons and rounds. These tables document elevation and azimuth offsets for given target ranges and wind conditions. Battlefield conditions require a tank crew to determine firing solutions quickly. The process may be summarized as follows: (a) determine the target range and wind condition, (b) use the ballistic lookup tables to determine azimuth and elevation aiming offsets, (c) aim the weapon, and (d) fire. A well-trained crew would be able to accomplish the task in five to ten seconds. If the enemy tank commander is aware that he is being targeted, he would certainly use evasion techniques (acceleration, deceleration, veering, zigzagging, and smoke grenades) to make the task of accurately targeting his tank more difficult. Determining an accurate firing solution under such conditions complicates the targeting process so much that a simple table-lookup method becomes impractical for determining firing solutions rapidly. An armor-piercing round must impact the target in order to destroy the tank. A near miss might cause shock waves, but will not destroy the tank. To further complicate the scenario, the enemy commander may be determining a firing solution on his attacker while undertaking evasive action. On the modern battlefield, it is a requirement that a tank commander be able, rapidly, to engage an enemy tank. Any delay in engaging the enemy may result in the hunter becoming the hunted.

II. THE REVIEW OF THE RELATED LITERATURE

The M829 APFSDS-T Round

The M829 120mm APFSDS-T (armor piercing, fin stabilized, discarding sabot, tracer) round (see figure 1) is used by the M1A1 Abrams main battle-tank (Sarson, Sarson, & Zaloga, 1992). The round is fired from the M256, 120mm, smooth bore, cannon barrel. As the round leaves the barrel, the sabot is discarded, revealing a dart-shaped projectile. The dart has a launch speed of 5480ft/s (1670m/s) (Schaefer, 1999a). It weighs 9.41 lbs. (4.277kgs), has diameter is 0.6" (1.524cms), and is 19" (48.26cms) long. The frontal and lateral drag curves for the projectile are not publicly available. A G1 drag profile is therefore used.

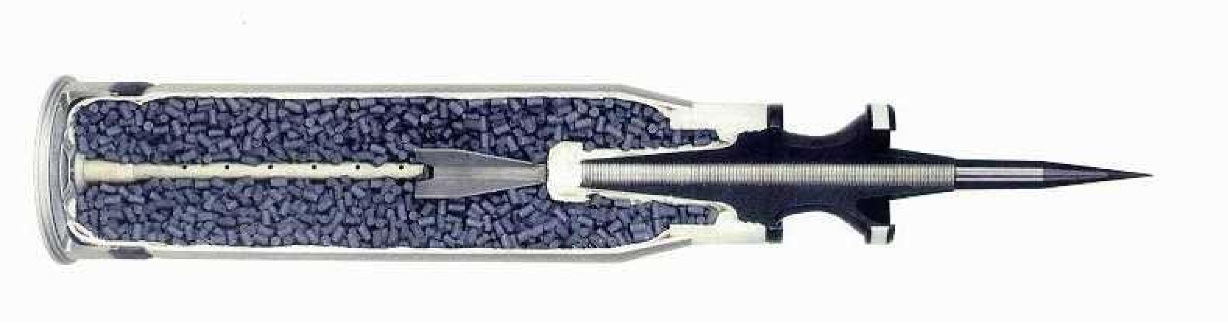

Figure 1. The M829APFSDS-T round prior to being fired[1].

---

[1] From "M829 120mm, APFSDS-T" by John Pike, 1999, Military Analysis Network. Available WWW: http://www.fas.org/man/dod-101/sys/land/m829a1.htm.

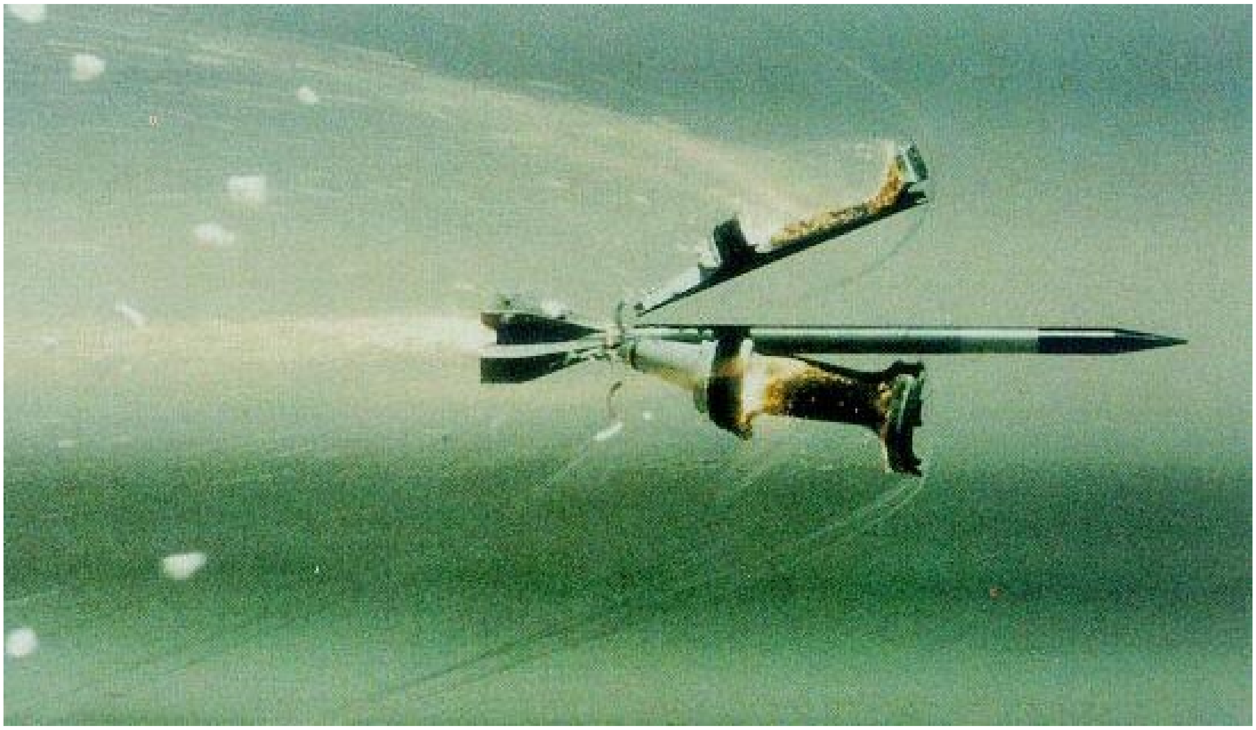

Figure 2. The M829 sabot breaking away from the projectile after firing[2].

Target Motion

Target motion requires that the weapon be aimed at some point that is displaced from the target position (United States Naval Academy). The aiming displacement is called lead. The aiming lead ensures that the weapon impacts the target at its future position.

---

[2] From "120mm Ammunition" by John Pike, 1999, Military Analysis Network. Available WWW: http://www.fas.org/man/dod-101/sys/land/120mm.htm.

In determining the aiming lead, it is assumed that the target will move in a straight line, and will not accelerate (United States Naval Academy). This assumption is valid for the following reasons: (a) the shortest distance between two points is a straight line. This reduces the target's exposure to fire, (b) zigzags, acceleration or curved paths might reduce the targets vulnerability to fire, but also make it more difficult for the target to press home its attack; (c) even if diversionary tactics are used, they average out, over time, to a straight line, or a gentle curve, if the weapon's flight time is minimized.

A general algorithm for compensating for target motion is first to assume that the target is not moving (United States Naval Academy). Determine the flight time of the weapon (TOF). Then, determine a new target position, using TOF seconds of target motion. Determine a firing solution for this new target position. Iterate, until the difference between the new target position after TOF seconds of flight time, and the weapon's previously determined impact point is acceptably minimal.

The M1A1 Abrams battle tank has a maximum cross-country speed of 30mph (Sarson, Sarson, & Zaloga, 1992).

Exterior Ballistics

Any object in motion through a medium experiences a resistive force called drag (Ashour-Abdalla). This force ($F = \frac{1}{2} \rho c_d A v^2$) is proportional to the density of the medium (ρ), the cross-sectional area of the object (A), the drag coefficient (C_d) of the object, and the square of the velocity of object. The resultant acceleration of the object is inversely proportional to the mass of the object by Newton's second ($F = m a$) law.

Air density (ρ) varies with altitude (Williams). Any mathematical method of determining the path of a projectile must consider the changing nature of air density and its impact on drag, as the projectile travels through different altitudes. This project considers a standard atmosphere.

A projectile experiences gravitational attraction (Ashour-Abdalla) force ($F = \frac{G M_e m}{r^2}$), where G is the universal gravitational constant, M_e is the mass of the earth, m is the mass of the object and r is the distance between the center-points of both objects. This causes the projectile to return to the earth, after launch (assuming the projectile is not launched at, or greater than, escape velocity). When the drag force and the gravitational force (Ashour-Abdalla) are balanced, the projectile has reached terminal-velocity (Ashour-Abdalla). Since gravitational acceleration varies with altitude ($g = \frac{G M_e}{r^2}$), any mathematical method of determining the path of a projectile must consider the changing nature of gravity as the projectile travels through different altitudes.

The drag coefficient of any projectile is not constant at all speeds (Schaefer, 1999b). Any mathematical method of determining the path of a projectile must consider the changing nature of the projectile C_d when determining the impact of drag on the projectile. This project considers the G1 drag model for both the frontal and lateral drag calculations.

Initial-Value Methods

An initial-value problem is a problem where $y(x_0)$, $y'(x_0)$, and $y'' = f(x, y, y')$ is known. One is then required to determine subsequent values of $y(x_n)$ and $y'(x_n)$, $n> 0$ (Cassuli & Greenspan, 1988).

In 1768, Euler developed the Euler-Method. It is iterative in nature, using a forward difference approximation. It is illustrated by the formula $y_{n+1} = y_n + hy'(x, y)$, where h is the step size between approximations. The method becomes more accurate as h becomes smaller, and converges to the true value as h approaches 0.

The Runga-Kutta method improves on Euler's method by equating, to a higher order, the series expansion of the numerical method to a Taylor series expansion (Umar). While Euler's method is first-order accurate, the Runga-Kutta method is at least second order accurate. Higher order Runga-Kutta methods exist. The project considers a fourth-order Runga-Kutta method.

Higher order methods are more accurate. For a given step size, the Runga-Kutta method is thus more accurate than Euler's method. One may thus use a larger step size with Runga-Kutta, than that used in Euler, and yet obtain the same accuracy. Less iterations are thus required in Runga-Kutta than in Euler for the same problem size and desired accuracy.

The Shooting Method

The shooting method is a widely used technique for solving calculus problems where the solution is required to pass through some point, subsequent to the initial point (Drasco). The method involves taking an initial guess. The error generated by the guess is then computed. A new guess is made, which should reduce the error. Once more, the error generated by the guess is computed. The process iterates until some error bound is achieved.

The method is optimized by recording guesses and their associated error, and then using Newton's root-finding method to determine subsequent guesses.

Summary

Most of the specifications for the M829 projectile are known. General values are used for parameters that are not known. The primary factors determining the ballistic path are launch velocity, wind, drag and gravity. Dynamic factors such as air density, gravitational acceleration, and the drag coefficient should be computed at each iteration to ensure greater accuracy. The Runga-Kutta initial-value problem method provides high accuracy because it models, more closely, a Taylor series at higher orders. A wider step size may thus be used by the method, resulting in fewer iterations for a given problem size and desired accuracy. The shooting method is a commonly used method for determining solutions to calculus problems where the solution is required to pass through some point. The method uses an iterative sequence of guesses that refine the solution at every iteration. Target motion must be considered when determining a firing solution. An iterative method akin to the shooting method is used to determine lead angles.

III. AN OVERVIEW OF THE DATA AND THE TREATMENT OF THE DATA

The Data

Two data types exist in this research: primary and secondary data.

Primary Data. Data that directly measure the efficiency of the algorithms being tested are considered primary data. Examples of this are: (a) the number of iterations required by the algorithm, (b) the time taken by the algorithm, and (c) the step size used by the algorithm.

Secondary Data. Data that does not directly measure the efficiency of the algorithm being tested are considered secondary data. Examples of this are: (a) impact points, (b) aiming points, and (c) impact point error.

General Criteria for the Admissibility of the Data.

In general, primary data is admissible only if the algorithm being tested has achieved some level of accuracy.

Secondary data are used to determine the admissibility of the related primary data. For example, during the Runga-Kutta step size test, impact points (secondary data) at smaller step sizes are used to determine whether the step size chosen has caused the algorithm to converge on the analytical solution at the desired degree of accuracy. The impact points are thus secondary data generated by the test, which are used to determine whether the step size (primary data) is admissible.

General Treatment of the Data

The data is of the interval type. The mean is thus the appropriate tool for determining the central tendency of the data. However, the curve of the data is not known a priori, so the choice of the arithmetic, geometric, or harmonic mean is made once the data has been collected. The standard deviation is the appropriate method for determining the spread of the data. The range is determined to provide a lower and upper bound of the algorithm under typical usage conditions.

Graphs of the data are produced, which are inspected to determine patterns in the data.

Systemic Treatment of the Data

Since the aim of the project is to determine the efficiency of the system, interpretation of the data is deferred until all tests have been completed. Interpretation of the data then consists of answering the following question: (a) Does the system provide a firing solution in less than 5 seconds? (b) Under which conditions does the system provide firing solutions in less than 5 seconds, (c) Do any noticeable bottlenecks exist in the system, and (d) If bottlenecks exist, where do they exist?

The Research Methodology

A computer program is written, for each test, which simulates the algorithm being tested. The tests build on one another, so that the code for a given test is required as a module in subsequent tests. For example, the Runga-Kutta code is used in the stationary-target aiming-solution tests so that the aiming solution tests incorporate the Runga-Kutta ballistic simulation. This permits the entire system, as well as its components to be tested. The programs are coded in the "C" programming language, and run on an Intel Celeron (Mendocino) 366Mhz processor with 64Mb RAM, running the Linux operating system (Kernel 2.0.36). The GNU "gcc" compiler is used, with all optimizations enabled ("-O3"). The programs are required to record primary and secondary data, and report all data upon completion of the test.

Treatment of the Data for Each Subproblem

Subproblem 1.

The investigation and evaluation of the efficiency of a three dimensional, dynamic step size, fourth-order Runga-Kutta, numerical simulation of the projectile motion of the M829 APFSDF-T projectile.

The Data Needed

Two tests are required for this subproblem. The first test determines the maximum step size, which may be used by the Runga-Kutta method without compromising the accuracy of the Runga-Kutta method. The second test determines the number of iterations and run-time required by a dynamic step size Runga-

Kutta ballistic simulation at selected weapon elevations and wind conditions.

The Location of the Data

The data for both tests are the output of two computer programs that are written to simulate the Runga-Kutta Method. A ballistic model, which simulates a M829 projectile, provides acceleration data.

The Means of Obtaining the Data

The first test uses a weapon elevation of 2°, and a crosswind of 50kts from the left. The test determines the range of the projectile after exactly 7 seconds of flight, using selected step sizes. The step sizes are 1s, 0.1s, 0.01s, 0.001s, 0.0001s, 0.00001s, and 0.000001s. The same test is run using Euler's method, to provide a baseline for evaluating the Runga-Kutta method.

The second test is run using selected weapon elevations, and wind conditions. The algorithm provides dynamic step size control. The weapon elevation varies between 0.05° and 2.00° at 0.05° intervals. The wind varies between 0kts and 50kts from the left at 10kt intervals, and is purely crosswind.

The Criteria for the Admissibility of the Data

For the first test, the maximum step size occurs when the difference between the three-dimensional positions of the projectile, using subsequently smaller step size refinements, is less than one decimal place (0.05) in all axes.

For the second test, the data is admissible only if the projected impact point falls within 0.1m of the projected impact point if an extremely small step size were used. The circular error of the simulation is thus at most 0.1m with respect to the analytical solution. This is not unreasonable since the kill zone of a battle tank is approximately 1m in diameter.

The Treatment of the Data

No analysis of the data from the first test is required since its result (maximum permissible step size) is used as a parameter of the dynamic step-size selection code in the second test. The RK method results are, however, compared to the Euler method results to demonstrate the efficiency of the RK method.

The range, mean and standard deviation of the primary data is determined for selected data sets in the second test. The following data sets are considered: (a) all data, (b) various elevations with 0kt wind, (c) various elevations with a 20kt wind, (d) various elevations with a 50kt wind, (e) various wind speeds at 0.5° elevation, (f) various wind speeds at 1° elevation, and (g) various wind speeds at 1.5° elevation.

The primary data are graphed as follows: (a) elevation vs. number of iterations / elapsed time, and (b) wind vs. number of iterations / elapsed time. The graphs are studied to determine any noticeable patterns.

Subproblem 2. The investigation and evaluation of the efficiency of the shooting method as an aiming augmentation system for the M829 APFSDS-T projectile.

The Data Needed

Two tests are required for this subproblem. The first test determines aiming solutions against a stationary target, while the second test determines aiming solutions against moving targets.

The data required for the first test are: (a) the number of iterations of the shooting method required to determine an aiming solution at the desired accuracy, (b) the time required by the shooting method (including the time required by the ballistic simulation) to determine an aiming solution at the desired accuracy.

The data required for the second test are: (a) the number of iterations of the shooting method required to determine an aiming solution at the desired accuracy, (b) the time required by the shooting method (including the time required by the ballistic simulation and the static target aiming code) to determine an aiming solution at the desired accuracy.

The Location of the Data

The data for the first test are the output of a computer program that is written to simulate the shooting method for stationary targets, with the previously determined Runga-Kutta code providing ballistic simulation information to the shooting method.

The data for the second test are the output of a computer program that is written to simulate the shooting method for moving targets. The previously determined Runga-Kutta code provides ballistic simulation information. The previously determined static target aiming code provides static target aim point information to the shooting method.

The Means of Obtaining the Data

The first test (static target) is run with selected target ranges and wind conditions. The target range varies between 500m and 10000m at 500m intervals. The wind varies between 0kts and 50kts from the left at 10kt intervals.

The second test (moving target) is run with selected target ranges, wind conditions, and target motion vectors. The target range varies between 500m and 10000m at 500m intervals. The wind conditions vary between 0kts and 50kts from the left at 10kt intervals. The target speed varies between 5mph and 30mph at 5mph intervals. The target direction varies between head-on, 45° right of the line of sight (approaching the attacker), and 90° right of the line-of-sight.

The Criteria for the Admissibility of the Data

The primary data generated by the stationary target aiming solution test is admissible only if the impact point of the aiming solution (secondary data) is within 0.25m of the targeted point.

The primary data generated by the moving target aiming solution test is admissible only if the impact point of the aiming solution (secondary data) is within 0.5m of the targeted tank's predicted position.

The Treatment of the Data

Treatment of Data from the First Test. The range, mean and standard deviation of the primary data are determined for selected data sets. The following data sets are considered: (a) all data, (b) varying ranges with 0kt wind, (c) varying ranges with a 20kt wind, (d) varying ranges with a 50kt wind, (e) varying wind speeds at 500m, (f) varying wind speeds at 5000m, and (g) varying wind speeds at 10000m.

The data are graphed as follows: (a) range vs. number of iterations / elapsed time, and (b) wind vs. number of iterations / elapsed time. The graphs are studied to determine any noticeable patterns.

Treatment of Data from the Second Test. The range, mean and standard deviation of the primary data are determined for selected data sets. The following data sets are considered: (a) all data, (b) varying ranges and wind speeds with a target speed of 10mph head-on, (c) varying ranges and wind speeds with a target speed of 10mph 45° right (approaching the attacker), (d) varying ranges and wind speeds with a target speed of 10mph 90° right (e) varying ranges and wind speeds with a target speed of 30mph head-on, and (f) varying ranges and wind speeds with a target speed of 30mph 45° right (approaching the attacker), and (g) varying ranges and wind speeds with a target speed of 30mph 90° right.

The data are graphed as follows: (a) target speed vs. number of iterations / time elapsed, and (b) target direction vs. number of iterations / elapsed time. The graphs are studied to determine any noticeable patterns.

Summary

Two types of data exist in the study: primary and secondary data. Primary data measures the efficiency of an algorithm. Secondary data is any data not related to the efficiency of the algorithm.

Four tests are required for the study. The first test measures the maximum step size for which the accuracy the Runga-Kutta method is not compromised. The second test measures the efficiency of a dynamic step size Runga-Kutta algorithm in determining projectile impact points for selected firing conditions. The third test measures the efficiency of the Shooting Method in determining aiming solutions (elevation and azimuth aiming offsets) against stationary targets. The fourth test measures the efficiency of the Shooting Method in determining aiming solutions against moving targets.

IV. THE BALLISTIC MODEL

Introduction

The study seeks to determine whether tank crews would benefit from the use of a computerized aiming system comprising the Runga-Kutta and Shooting methods. It is therefore required that the secondary data produced by these methods be accurate, even though the accuracy of the secondary data is not a primary concern in this study. The foundation of the accuracy of any ballistic computation is the model of ballistic forces acting on the projectile in flight. This chapter will discuss the ballistic forces considered to act upon the projectile in flight.

The Definition of the Axes

The x-axis is the vertical plane proceeding through the line of sight from the attacker towards the target. It indicates the range towards the target or away from the attacker. Positive displacement in the x-axis is the space extending from the attacker towards the target. Negative displacement in the x-axis is the space extending from the attacker away from the target. Positive velocity and acceleration in the x-axis is in the direction from the attacker towards the target. Negative velocity and acceleration in the x-axis is in the direction from the attacker away from the target.

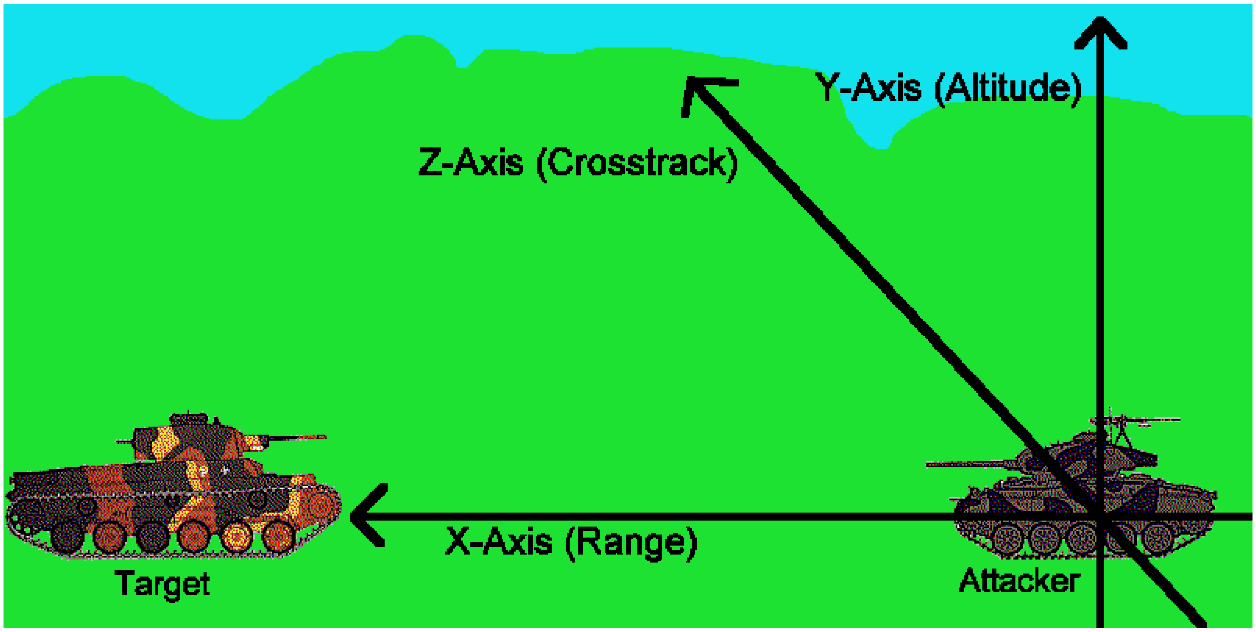

Figure 3. The axial model.

The y-axis is the horizontal plane proceeding through the attacker, extending such that the altitude of the plane is constant. It defines the altitude above sea level. Positive displacement in the y-axis is above sea level. Negative displacement in the y-axis is below sea level. Positive velocity and acceleration in the y-axis is directed from sea level towards above sea level. Negative velocity and acceleration in the y-axis is directed from sea level towards below sea level.

The z-axis is the vertical plane proceeding through the attacker, at 90 degrees from the x-axis and the y-axis. The z-axis defines the cross-track from the line of sight towards the target. Positive displacement in the z-axis is the space proceeding to the right of the line of sight of the target. Negative displacement in the z-axis is the space proceeding to the left of the line of sight of the target. Positive velocity and acceleration in the z-axis is directed towards the right of the line of sight of the attacker. Negative velocity and acceleration in the z-axis is directed towards the left of the line of sight of the target.

Gravitational Acceleration

The gravitational acceleration at a given altitude is determined by the formula $g = \frac{G M_e}{r^2}$. G is the universal gravitational constant, and is 6.67×10^{-11} $kg.m^{-1}.s^{-2}$. M_e is the mass of the earth, and is 5.983×10^{24}kg. The equatorial radius of the earth is 6370km.

Two possible strategies for evaluating g exist: (a) directly evaluating the formula for every evaluation, and (b) setting up an interpolation table, and interpolating g for every evaluation. A test program was written for both methods. Directly evaluating the formula proved to be approximately twice as fast as interpolating (piecewise parabolic) it on the test computer. The code for the g determination is shown in appendix A. On the test machine, a g determination required 180ns.

Air Density

For a standard atmosphere, the air density at a given altitude is provided by the formula

$\rho = \rho_0 (1 - 6.8755856 \times 10^{-6} h)^{4.2558797}$, where rho$_0$ = 1.225kg.m^{-3}, and h is the altitude above sea level, in feet (0 < h < 36,000ft).

Just as for the g determination, there are two possible ways of determining rho: (a) directly evaluating the formula, and (b) interpolating it. Test programs were written to determine which method would prove more efficient. On the test machine, the interpolation (piecewise parabolic) was nearly twice as fast as directly evaluating the formula. Appendix A shows the interpolation code. A setup function is initially (once per program run) called to setup the interpolation tables. A piecewise parabolic interpolation is used. The parabolic interpolation coefficients are determined as follows:

$$c = \frac{y_2 - 2y_1 + y_0}{\delta^2}$$

$$b = \frac{y_1 - y_0 - 2c\delta x_0 - c\delta^2}{\delta}$$

$$a = y_0 - b x_0 - c x_0^2$$

where y_0, y_1 and y_2 are successive range components, x_0 is the domain value of y_0, and δ is the step size between subsequent domain values.

An interpolation function uses the interpolation tables that were initialized by the setup function in order to provide an interpolation using the formula $y = a + bx + cx^2$. The interpolation function requires 610ns on the test machine, and is accurate to 10 decimal places with respect to the analytical solution.

Drag Coefficient

The G1 drag model (see table 1) is used by this project. The only method of determining the drag coefficient is to interpolate it from the piecewise components shown in table 1. A piecewise parabolic interpolation is sufficient for the method. The code for the interpolation is shown in appendix A. The code consists of a setup routine (called once per program run), that sets up the parabolic coefficients table, and an interpolation routine, which uses the parabolic coefficients table to interpolate values. The interpolation method is the same as for the rho interpolation. The interpolation routine requires 650ns per interpolation. Mach 1, for a standard atmosphere, at sea level, is 339.9265028m/s.

Table 1

The G1 Drag Model[3]

Mach	C_d
0.00	0.2629
0.20	0.2344
0.40	0.2104
0.60	0.2034
0.80	0.2546
1.0	0.4805
1.20	0.6393
1.40	0.6625
1.60	0.6474
1.80	0.6210
2.00	0.5934
2.20	0.5685
2.40	0.5481
2.60	0.5325
2.80	0.5211
3.00	0.5133
3.20	0.5084

---

[3] From "G1" by JBM Small Arms Ballistics, 1996. Available WWW: http://www.lascruces.com/~jbm/download/text/mcg1.txt.

3.40	0.5054
3.60	0.5030
3.80	0.5016
4.00	0.5006
4.20	0.4998
4.40	0.4995
4.60	0.4992
4.80	0.4990
5.00	0.4988

Frontal Drag Determination

The drag force is determined by the formula $F = \frac{1}{2}\rho\, c_d A v^2$. Rho and C_d are dependent on the current altitude and the current speed respectively, and are calculated by the previously discussed code. The M829 projectile is circular in cross section, with a diameter 0.6". The frontal area of the projectile is thus $1.824146925 \times 10^{-4} m^2$. The frontal drag force is determined by evaluating the drag equation. The frontal drag acceleration is then determined by Newton's second law (F=ma), using the projectile's mass of 9.41lbs (4.277kg).

Z Axis Crosswind Drag Determination

Accommodating crosswind (The project only considers z-axis wind) in the model is relatively simple. The effect of crosswind may be viewed as that of a body travelling through some medium - id est drag.

The same method is used for determining the z-axis crosswind acceleration as is used for the frontal drag acceleration. The primary difference is the area (A) used to determine the drag. A secondary difference is that the C_d is substantially different for crosswind determination as compared to frontal acceleration determination since the crosswind speed is substantially different from the frontal speed.

The area of the projectile exposed to crosswind varies with the angle at which the projectile is moving in the x-z plane. Depending on the angle in the x-z plane, some of the frontal / tail area as well as some of the lateral area of the projectile is exposed to crosswind. A simple trigonometric method for determining the area exposed to crosswind is:

Area = LA x XS / XZS + FA x ZS / XZS

where LA is the lateral area, FA is the frontal area, XS is the x-axis speed, ZS is the z-axis speed, and XZS is the xz plane speed. The lateral area was determined from diagrams (see figures 1 and 2) of the projectile to $9.5259457\text{x}10^{-3}\text{m}^2$.

The z-axis crosswind force and acceleration are then determined using the same formulae as for the frontal drag determination.

The Axial Acceleration Model

Determination of the x, y an z-axis accelerations require the determination of: (a) gravitational acceleration, (b) frontal acceleration, and (c) z-axis crosswind acceleration. These accelerations are then broken into axial components, and summed.

The following trigonometric formulae determine the axial acceleration components:

XA = FA x XS / XYZS

YA = FA x YS / XYZS + G

ZA = FA x ZA / XYZS + ZCWA

where XA, YA, and ZA are the x, y, and z-axis acceleration components respectively. FA is the frontal acceleration caused by the frontal drag of the projectile's motion. XYZS is the speed of the projectile. ZCWA is the z-axis crosswind acceleration.

On the test machine, an axial acceleration determination required 2.31us when no crosswind existed, and 3.64us when crosswind existed. The timing differences exist since the crosswind code is only executed when crosswind exists.

Summary

The primary ballistic forces acting on a projectile in flight are: (a) gravitational attraction force, (b) frontal drag force, and (c) wind induced drift force. These forces are determined by standard formulae (taking into consideration the flight conditions at that moment in time), and converted into accelerations by Newton's second law. The forces are then converted in axial components, which are summed per axis to provide a snapshot of axial accelerations experienced by the projectile at a given moment in time.

V. THE RUNGA-KUTTA METHOD AND ITS TESTS

The Euler Method Code

The Euler method is a simple forward difference method for determining subsequent range values in an initial value problem. The method is illustrated by the formula:

$y_{n+1} = y_n + hf(x_n, y_n)$

$x_{n+1} = x_n + h$

where h is the step size, and f is the derivative of y at x_n. The method uses the derivative at (x_n, y_n) to determine, in a linear manner, the subsequent range value y_{n+1} at the next domain location. The method does not consider that the derivative is usually continuously changing. This causes inaccuracies for larger step sizes. However, the method does converge to the true value for small step sizes as h approaches 0.

The Euler Method code (see appendix B for the code) implements the following three-dimensional formulae:

$xs_{n+1} = xs_n + h*xv_n + h^2*xa(xv_n, yv_n, zv_n, xs_n, ys_n, zs_n)/2$

$ys_{n+1} = ys_n + h*yv_n + h^2*ya(xv_n, yv_n, zv_n, xs_n, ys_n, zs_n)/2$

$zs_{n+1} = zs_n + h*zv_n + h^2*za(xv_n, yv_n, zv_n, xs_n, ys_n, zs_n)/2$

$xv_{n+1} = xv_n + h*xa(xv_n, yv_n, zv_n, xs_n, ys_n, zs_n)$

$yv_{n+1} = yv_n + h*ya(xv_n, yv_n, zv_n, xs_n, ys_n, zs_n)$

$zv_{n+1} = zv_n + h*za(xv_n, yv_n, zv_n, xs_n, ys_n, zs_n)$

where xs, ys and zs are the axial displacements; xv, yv, an zv are the axial velocities; and xa, ya and za are the axial accelerations. Euler's method is used to update the axial velocities, while a truncated Taylor series is used to update the axial displacements.

The Runga-Kutta Method Code

The Runga-Kutta method improves on Euler's method by taking into consideration that the derivative changes between domain steps. The Runga-Kutta method thus equates a Taylor series to a higher degree. Practically, the method is a predictor-corrector method. A predictor step equivalent to the Euler method is initially taken. The predictor step is then corrected to provide a more accurate step across the domain interval. In the fourth-order Runga-Kutta method, the corrector step is itself corrected, and that corrector step is itself corrected. This provides a solution that is equivalent to a fourth-order Taylor series. The method is generalized by the following formulae:

$k1 = hf(x_n, y_n)$

$k2 = hf(x_n + h/2, y_n + k1/2)$

$k2 = hf(x_n + h/2, y_n + k2/2)$

$k3 = hf(x_n + h, y_n + k3)$

$y_{n+1} = y_n + (k1 + 2*k2 + 2*k3 + k4)/6$

$x_{n+1} = x_n + h$

The Runga-Kutta code (see appendix C for the code) implements the following three dimensional Runga-Kutta formulae:

$k1.x = h*xa(xv_n, yv_n, zv_n, xs_n, ys_n, zs_n)$

$k1.y = h*ya(xv_n, yv_n, zv_n, xs_n, ys_n, zs_n)$

$k1.z = h*za(xv_n, yv_n, zv_n, xs_n, ys_n, zs_n)$

$k2.x = h*xa(xv_n+k1.x/2, yv_n+k1.y/2, zv_n+k1.z/2, xs_n + h*xv_n/2, ys_n + h*yv_n/2, zs_n + h*zv_n/2)$

$k2.y = h*ya(xv_n+k1.x/2, yv_n+k1.y/2, zv_n+k1.z/2, xs_n + h*xv_n/2, ys_n + h*yv_n/2, zs_n + h*zv_n/2)$

$k2.z = h*za(xv_n+k1.x/2, yv_n+k1.y/2, zv_n+k1.z/2, xs_n + h*xv_n/2, ys_n + h*yv_n/2, zs_n + h*zv_n/2)$

$k3.x = h*xa(xv_n+k2.x/2, yv_n+k2.y/2, zv_n+k2.z/2, xs_n + h*xv_n/2 + h*k1.x/4, ys_n + hyv_n/2 + h*k1.y/4, zs_n + hzv_n/2 + h*k1.z/4)$

$k3.y = h*ya(xv_n+k2.x/2, yv_n+k2.y/2, zv_n+k2.z/2, xs_n + h*xv_n/2 + h*k1.x/4, ys_n + hyv_n/2 + h*k1.y/4, zs_n + hzv_n/2 + h*k1.z/4)$

$k3.z = h*za(xv_n+k2.x/2, yv_n+k2.y/2, zv_n+k2.z/2, xs_n + h*xv_n/2 + h*k1.x/4, ys_n + hyv_n/2 + h*k1.y/4, zs_n + hzv_n/2 + h*k1.z/4)$

$k4.x = h*xa(xv_n+k3.x, yv_n+k3.y, zv_n+k3.z, xs_n + h*xv_n + h*k2.x/2, ys_n + h*yv_n + h*k2.y/2, zs_n + h*zv_n + h*k2.z/2)$

$k4.y = h*ya(xv_n+k3.x, yv_n+k3.y, zv_n+k3.z, xs_n + h*xv_n + h*k2.x/2, ys_n + h*yv_n + h*k2.y/2, zs_n + h*zv_n + h*k2.z/2)$

$k4.z = h*za(xv_n+k3.x, yv_n+k3.y, zv_n+k3.z, xs_n + h*xv_n + h*k2.x/2, ys_n + h*yv_n + h*k2.y/2, zs_n + h*zv_n + h*k2.z/2)$

$xv_{n+1} = xv_n + (k1.x + 2*k2.x + 2*k3.x + k4.x)/6$

$yv_{n+1} = yv_n + (k1.y + 2*k2.y + 2*k3.y + k4.y)/6$

$zv_{n+1} = zv_n + (k1.z + 2*k2.z + 2*k3.z + k4.z)/6$

$xs_{n+1} = xs_n + h*xv_n + h*(k1.x + k2.x + k3.x)/6$

$ys_{n+1} = ys_n + h*yv_n + h*(k1.y + k2.y + k3.y)/6$

$zs_{n+1} = zs_n + h*zv_n + h*(k1.z + k2.z + k3.z)/6$

The Step Size Accuracy Test

The step size accuracy test determines the largest step size for which the Runga-Kutta and Euler methods achieve a one decimal place accuracy (0.05) in all axes. The result of the test is used to set the maximum step size taken during the Runga-Kutta impact point tests.

A program was written (see appendix D for the code) to step across the flight path of the projectile at a step size which is supplied as a command line parameter of the program. The projectile is fired at 2° elevation and 0° azimuth, with a crosswind of 50kts from the left. A batch file (see appendix E for the code) runs the step size test programs with the selected step sizes (1s, 0.1s, 0.01s, 0.001s, 0.0001s, 0.00001s, and 0.000001s). The batch file reports the flight information of the projectile after 7 seconds of flight time.

The maximum step size occurs when the difference between the flight parameters of a given step size, and the flight parameters of a subsequently smaller step size is less than one decimal place (0.05).

The Runga-Kutta Impact Point Determination Code

The Runga-Kutta impact point determination code (see appendix C for the code) accepts the launch elevation, launch azimuth and crosswind as input parameters. The purpose of the code is to determine the range and crosstrack of the impact point of the projectile.

The code consists of two phases: (a) hunting, and (b) converging.

The goal of the "hunting phase" is to determine approximately, in the time domain, where the impact point lies. The code steps across the ballistic flight path, using the maximum step size determined by Runga-Kutta maximum step size test, until the projectile is below 0m altitude. Two bounds exist at this point: (a) the elapsed flight time at which the altitude of the projectile is positive, and (b) the elapsed flight time at which the altitude of the projectile is negative. The impact point (root) then lies somewhere, in the time domain, between these two flight time bounds.

Once the "hunting phase" is completed, the "convergence phase" is initiated. The goal of this phase is to bring the positive altitude and negative altitude flight time bounds closer and closer to the root, until the time between the two bounds reaches some criterion. The time difference between the two bounds is bisected. This bisected time becomes the updated step size. The ballistic flight path is then stepped from the positive altitude bound using the updated step size. If the new point has a positive altitude, it becomes the new positive altitude bound. If the new point has a negative altitude, it becomes the new negative altitude bound. The process iterates until the updated step size is smaller than 10us. At this point, the projectile will not move by more 0.0167m in any axis, which satisfies the accuracy requirements of the ballistic simulation (0.1m circular error probability for the impact point).

The Runga-Kutta Impact Point Test

The purpose of the test (see appendix F for the code) is to determine the number of iterations of the Runga-Kutta method, and the time required, to determine the impact point for a given launch elevation and crosswind speed. The test uses elevations between 0.05° and 2.0° at 0.05° intervals, and wind speeds between 0kts and 50kts from the left at 10kt intervals.

VI. THE SHOOTING METHOD FOR STATIONARY TARGETS AND ITS TESTS

The Method of Convergence

An impact point error function is used to determine the impact point error generated by a given elevation and azimuth offset when targeting a given range and crosstrack position. The function calls the Runga-Kutta impact point determination code to determine the impact point of the projectile. The function returns a negative range error if the projectile falls short of the target, and a positive range error if the projectile falls beyond the target. The function returns a negative crosstrack error if the projective falls to the left of the target, and a positive crosstrack error if the projectile falls to the right of the target.

Two aim points (elevation and azimuth offsets), relatively close to one another, are initially chosen. The choice of the initial aim points are arbitrary, and are chosen to be 0.18° and 0.2°. The impact error generated by both aim points are determined. The derivative of the range error D_e (δxerror/δelevation)and the crosstrack error D_a (δzerror/δazimuth) error is then determined. Newton's root-finding method (see appendix G for the code) is then employed to determine an improved aim point using the following formulae:

$\text{elevation}_{n+1} = \text{elevation}_n - \text{xerror}/D_e$

$azimuth_{n+1} = azimuth_n - zerror/D_a$

The method (see figure 4) iterates until the radial error (sqrt[xerror^2 + zerror^2]) is less than 0.25m.

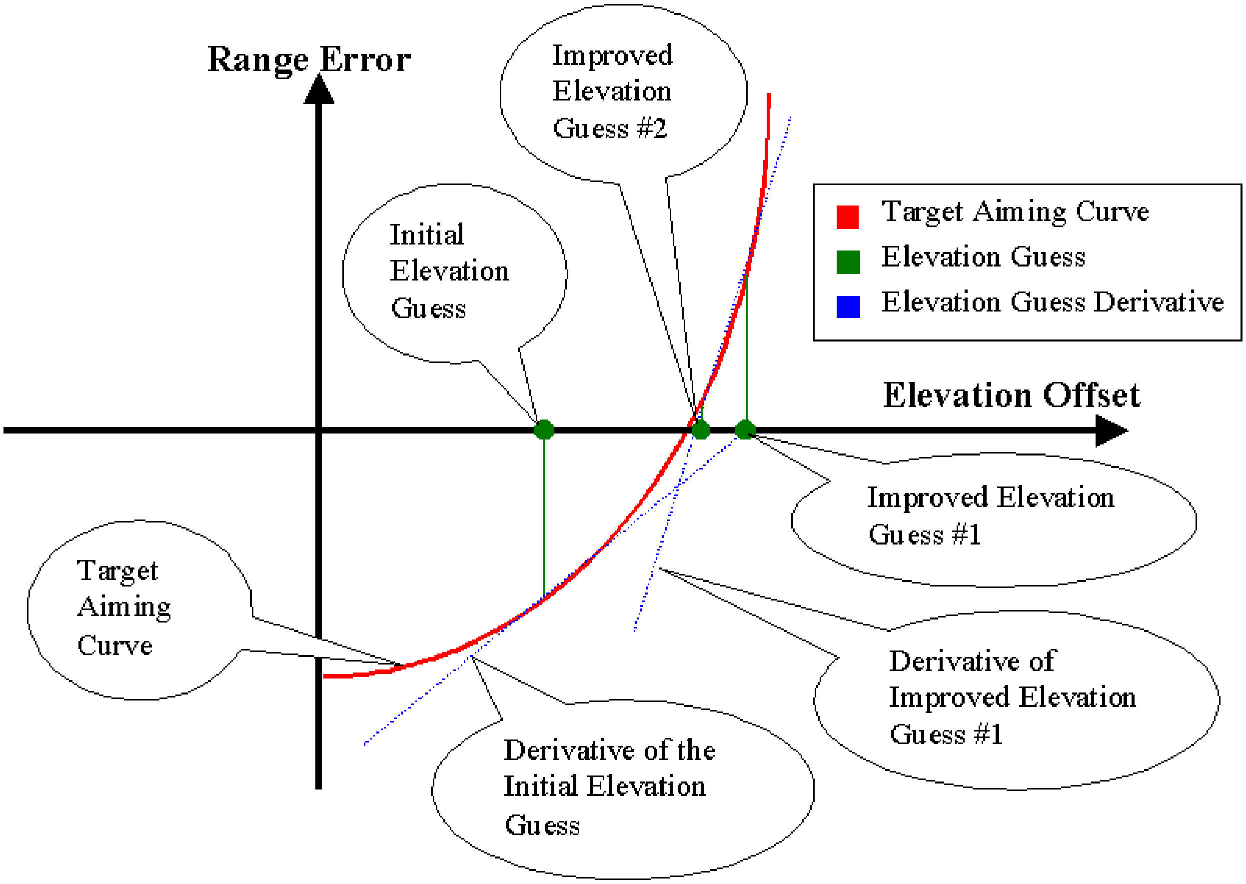

Figure 4. The method of convergence in the shooting method for stationary targets.

The Test

The test (see appendix H for the code) measures the number of iterations of the shooting method, and the time required by the shooting method to converge on the correct aiming point, at the desired degree of accuracy (0.25m). The test uses target ranges between 500m and 10000m at 500m intervals, and wind conditions between 0kts and 50kts from the left at 10kt intervals.

VII. THE SHOOTING METHOD FOR MOVING TARGETS AND ITS TESTS

The Method of Convergence

An aiming solution is initially determined against the target, as if it were stationary. This is achieved by a call to the shooting method for stationary targets code. This determines the flight time (TOF) of the weapon against a stationary target. A new target position is determined by advancing the target position TOF seconds along its velocity vector from the original (stationary) aiming position (see figure 5). An aiming solution is determined against the new target position. Once more, a new target position is determined by advancing the target TOF seconds along the velocity vector of the target from the original (stationary) aiming position. The method (see appendix I for the code) iterates until the radial displacement change between successive iterations is less than 0.5m.

The Test

The test (see appendix J for the code) determines the number of iterations of the shooting method, and the time required by the shooting method, in order to converge on the correct aiming solution with the desired degree of accuracy (0.5m).

The test uses target ranges between 500m and 10000m using 500m intervals, wind speeds between 0kts and 50kts from the left

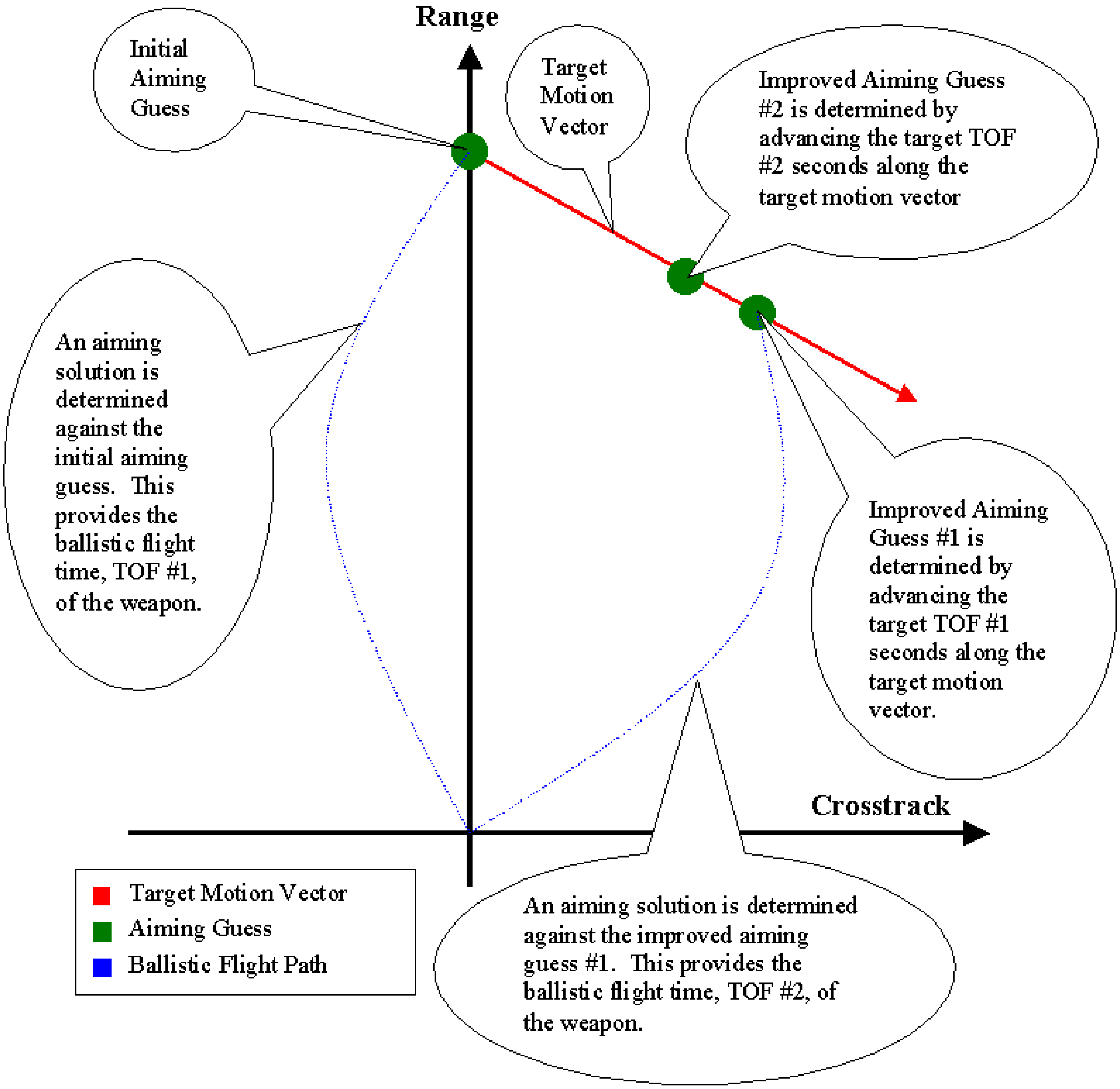

Figure 5. The method of convergence in the shooting method for moving targets.

---

using 10kt intervals, target speeds between 5mph and 30mph at 5mph intervals, and target directions between 90° right and head on at 45° intervals.

VIII. THE RESULTS

The Runga-Kutta vs. Euler-Method Step Size Accuracy Test

The Runga-Kutta method appeared to be internally accurate at all step sizes tested (see Table 2 and Appendix K). The Euler method, on the other hand, only achieved internal accuracy when using a step size of 0.001s. The axial velocity was greatest in the x-axis. Errors resulting from a too large step size were thus more pronounced in the x-axis. While the Runga-Kutta method was internally accurate with a step size of 1s,

Table 2

The Axial Displacements Resulting from Seven Seconds of Ballistic Flight Time as Determined by the Runga-Kutta and Euler Methods

	Runga-Kutta Method			Euler Method		
Step Size	X	Y	Z	X	Y	Z
1	10880.42	149.88	5.25	10864.45	147.22	5.31
0.1	10880.42	149.88	5.25	10878.80	149.61	5.26
0.01	10880.42	149.88	5.25	10880.26	149.85	5.25
0.001	10880.42	149.88	5.25	10880.40	149.88	5.25
0.0001	10880.42	149.88	5.25	10880.42	149.88	5.25
0.00001	10880.42	149.88	5.25	10880.42	149.88	5.25
0.000001	10880.42	149.88	5.25	10880.42	149.88	5.25

the Euler method displayed a 16m (≈52ft) error in the x-axis, with respect to the Runga-Kutta method, when using this step size. The computational time required by the Runga-Kutta method (see Appendix K) was 21.9% greater than that required by the Euler method, at the same step size.

The Runga-Kutta Iteration Test

The step size of the "hunting phase" of the Runga-Kutta iteration test was set to 1s, as determined by the Runga-Kutta step size accuracy test.

The Runga-Kutta method was able to determine the impact

Table 3

The Minimum, Maximum, Mean and Standard Deviation of the Iteration Count Data from the Runga-Kutta Impact Point Test

	Statistical Measures			
Data Set	Minimum	Maximum	M	SD
All Data	8	20	13.525	3.331
0kt Wind	8	20	13.525	3.336
20kt Wind	8	20	13.525	3.336
50kt Wind	8	20	13.525	3.336
0.5 Degrees	14	14	14	0
1 Degree	15	15	15	0
1.5 Degrees	18	18	18	0

point of the projectile using between 8 and 20 iterations (see Table 3). The average number of iterations required by the method was 13.525. The standard deviation was 3.331, implying that most impact points were determined using between 10 and 17

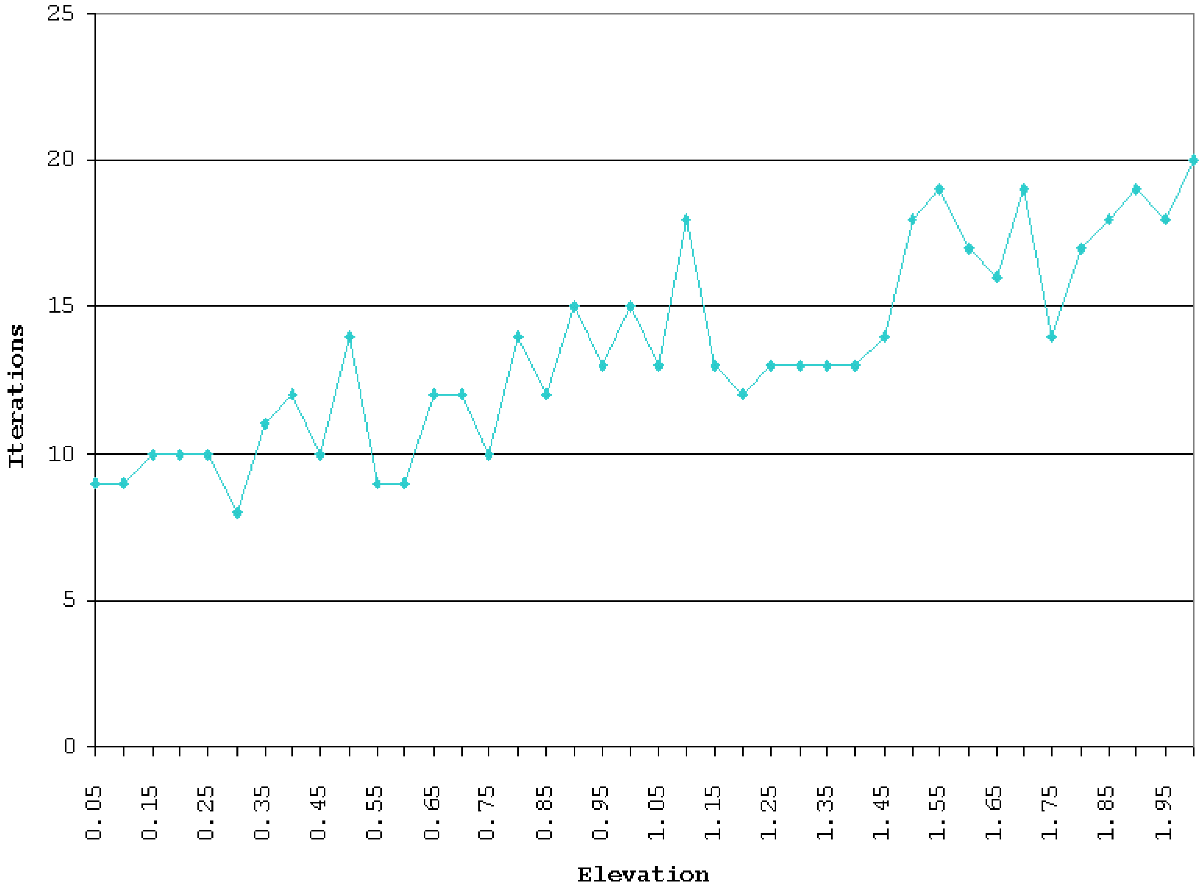

Figure 6. The number of iterations required by the Runga-Kutta method to determine the impact point of a projectile launched at a certain elevation.

iterations. The minimum and maximum number of iterations, at any given elevation, required by the Runga-Kutta method were the same (see Table 3 and Appendix L). It, therefore, appeared that crosswind did not impact (at the tested values) the number of iterations required by the Runga-Kutta method to determine an impact point. The number of iterations required by the method appeared to increase as the elevation increased (see Table 3 and Figure 6). The increase was, however, not linear. It tended to stagger an imaginary increasing line. Globally, the number of iterations appeared to increase as the elevation

Table 4

The Minimum, Maximum, Mean and Standard Deviation of the Computation Time Data from the Runga-Kutta Impact Point Test

	Statistical Measures			
Data Set	Minimum	Maximum	M	SD
All Data	2.790E-04	7.860E-04	4.799E-04	8.409E-05
0kt Wind	2.790E-04	4.110E-04	3.428E-04	3.878E-05
20kt Wind	4.140E-04	6.050E-04	5.059E-04	5.685E-05
50kt Wind	4.110E-04	6.260E-04	5.069E-04	5.962E-05
0.5 degrees	3.440E-04	5.110E-04	4.818E-04	6.754E-05
1 degree	3.550E-04	5.240E-04	4.958E-04	6.899E-05
1.5 degrees	4.060E-04	6.050E-04	5.620E-04	7.666E-05

increased, but at the local level variation existed. The local variation most likely resulted from the alignment of the impact point within the positive altitude and negative altitude bounds at the completion of the "hunting" phase.

The Runga-Kutta method required between 279us and 786us to

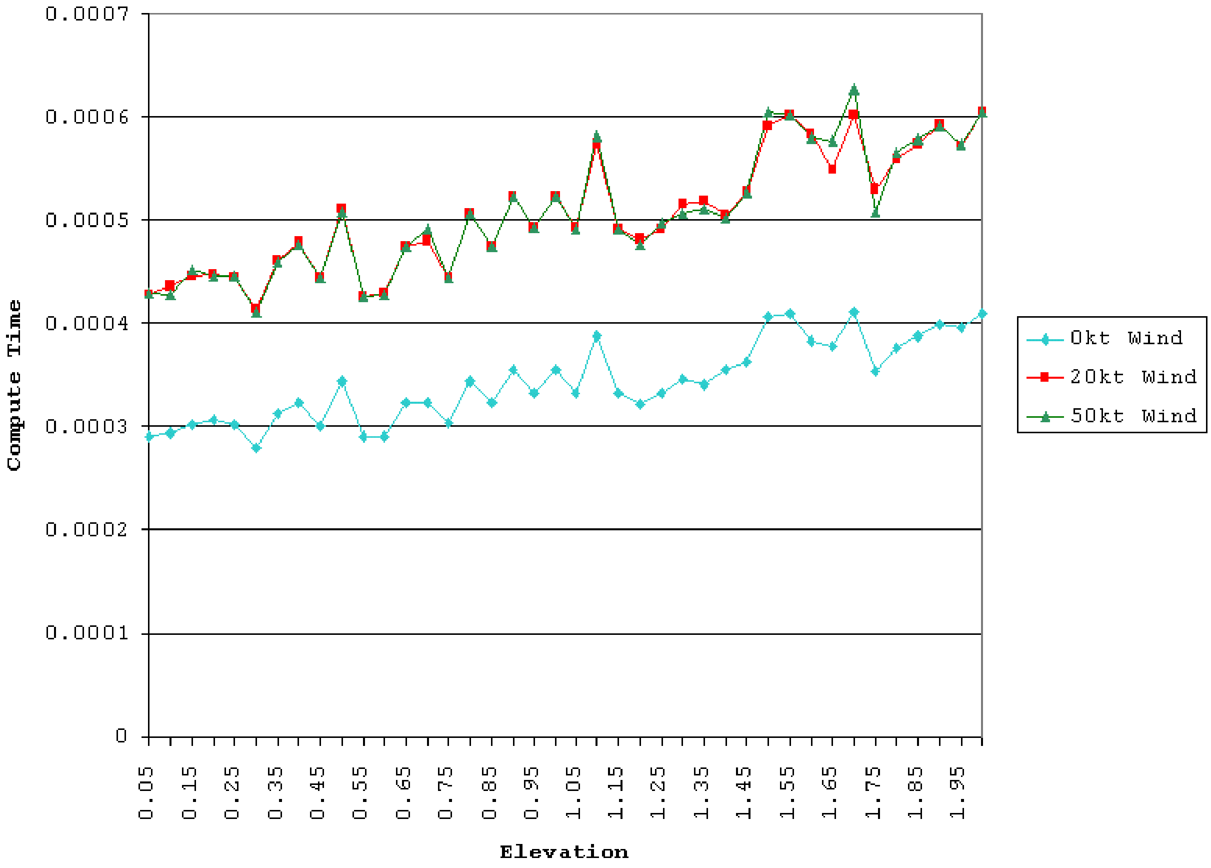

Figure 7. The computational time required the Runga-Kutta method to determine the impact point of a projectile launched at a certain elevation.

determine the impact point of the projectile. The average time required was 479us, with a standard deviation of 84us (see Table 4). There appeared to be a 135us penalty for processing crosswind (see Figure 7). The penalty occurred because the crosswind code only executed when crosswind existed. The computational time required under crosswind conditions (10kts to 50kts) was the same for most crosswind values at a given elevation. Small variations were noted between 1.2° elevation and 1.45° elevation, and between 1.5° elevation and 1.85° elevation (see figure 7). These variations were most likely hardware related since the number of iterations were identical at all crosswind speeds (10kts to 50kts), and the algorithm was the same for all crosswind values (except 0kts).

The code appeared to spend most of its time in the "convergence" phase, and little time in the "hunting" phase. This notion is supported since the step size during the "hunting" phase is 1s, and the flight time required to reach 10000m is approximately 7s. The "convergence" phase successively reduces the step size until it is smaller than 10us.

The Shooting Method for Stationary Targets Test

The shooting method required between 2 and 5 iterations to determine an aiming solution against a stationary target. The average number of iterations required was 3.692, and the standard deviation was 0.696 (see Table 5). The method, therefore, required between 3 and 4 iterations for most test cases. Figure 8 shows that the number of iterations were constant between 500m and 1500m, and once more between 5000m and 10000m. The section of the graph (figure 8) between 1500m and 5000m appears to present a negative sinusoidal curve. The minimum and maximum number of iterations at a given range were the same (see Table 5), with the exception of the ranges between

Table 5

The Minimum, Maximum, Mean and Standard Deviation of the Iteration Count Data from the Shooting Method for Stationary Targets Test

	Statistical Measures			
Data Set	Minimum	Maximum	M	SD
All Data	2	5	3.692	0.696
0kt Wind	2	5	3.700	0.733
20kt Wind	2	5	3.700	0.733
50kt Wind	2	5	3.700	0.657
500m Range	3	3	3	0
5000m Range	4	4	4	0
10000m Range	4	4	4	0

3000m and 5000m. This appears to indicate that crosswind is not a significant factor (see Figure 8) in the number of iterations required by the method. The space between 500m and 1500m required 3 iterations (see Figure 8). The space between 5000m and 10000m required 4 iterations. It, therefore, appears that

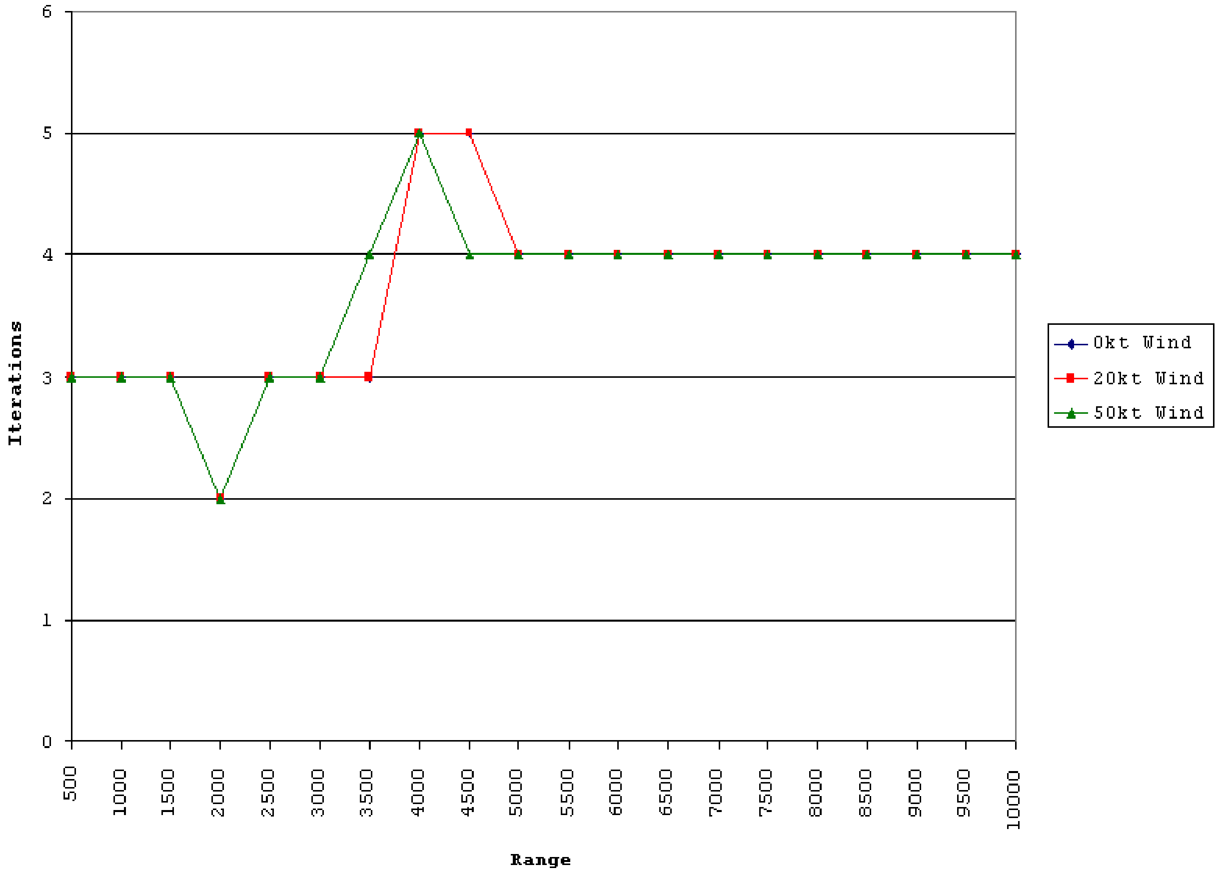

Figure 8. The number of iterations required the shooting method to determine an aiming solution against a stationary target.

range does not significantly impact the number of iterations required by the shooting method.

The minimum number of iterations occurs at 2000m, and is 2. This range coincides with the elevation pair used as the initial guess of the shooting method (0.18° and 0.2°). The negative sinusoidal curve between 1500m and 5000m in figure 8 is most likely caused by the estimated derivative of the initial elevation guess. As noted earlier, an initial elevation guess of 0.18° and 0.2° is used. The error resulting from these guesses is used to determine the derivative of the error with

Table 6

The Minimum, Maximum, Mean and Standard Deviation of the Compute Time Data from the Shooting Method for Stationary Targets Test

	Statistical Measures			
Data Set	Minimum	Maximum	M	SD
All Data	1.130E-03	3.110E-03	2.276E-03	4.510E-04
0kt Wind	1.130E-03	2.210E-03	1.717E-03	2.980E-04
20kt Wind	1.570E-03	3.050E-03	2.394E-03	4.093E-04
50kt Wind	1.580E-03	2.970E-03	2.387E-03	3.641E-04
500m Range	1.370E-03	1.930E-03	1.822E-03	2.216E-04
5000m Range	1.780E-03	2.570E-03	2.408E-03	3.100E-04
10000m Range	1.860E-03	2.590E-03	2.457E-03	2.924E-04

respect to the elevation. The 0.02° spacing between the initial guesses is probably causing the estimated derivative to be inaccurate. The inaccurately estimated derivative would then cause Newton's root-finding method to overshoot the root on the next iteration, instead of cautiously approaching it.

The shooting method required between 1.13ms and 3.11ms to

Figure 9. The computational time required the shooting method to determine an aiming solution against a stationary target.

determine an aiming solution against a stationary target. The average time required was 2.276ms, and the standard deviation was 451us (see Table 6). A crosswind-processing penalty was noted in figure 9. However, the penalty does not originate in the shooting method, but rather in the Runga-Kutta impact point determination code, which is used by the shooting method to determine impact points for given elevation and azimuth offsets.

The Shooting Method for Moving Targets Test

The shooting method for moving targets required between 2 and 3 iterations to determine an aiming solution against a moving target. The average number of iterations required was 2.031, and the standard deviation was 0.172 (see Table 7). The code, therefore, required 2 iterations for most of the test scenarios. The average number of iterations appeared to increase as the target speed increased (see Figure 10). This occurred because the difference in the ballistic flight time of the weapon between the original position and the impact position changed more rapidly as the target speed increased. It should be noted that the method converged on the correct aim point primary by using ballistic flight time, and secondarily by using target position. The increase in the average number of iterations was amplified when the target was moving towards the attacker. This occurred because the ballistic flight time

Table 7

The Minimum, Maximum, Mean and Standard Deviation of the Iteration Count Data from the Shooting Method for Moving Targets Test

Data Set	Minimum	Maximum	M	SD
	Statistical Measures			
All Data	2	3	2.031	0.172
Target 10mph 90° Right	2	2	2	0
Target 10mph 135° Right	2	2	2	0
Target 10mph 180° Right	2	2	2	0
Target 30mph 90° Right	2	2	2	0
Target 30mph 135° Right	2	3	2.100	0.301
Target 30mph 180° Right	2	3	2.350	0.479
500m Range	2	2	2	0
5000m Range	2	2	2	0
10000m Range	2	3	2.167	0.374
0kt Wind	2	3	2.031	0.172
20kt Wind	2	3	2.031	0.172
50kt Wind	2	3	2.031	0.172

of the weapon changed more rapidly when the target approached the attacker. Crosswind did not appear to be a factor in the number of iterations required to determine an aiming solution.

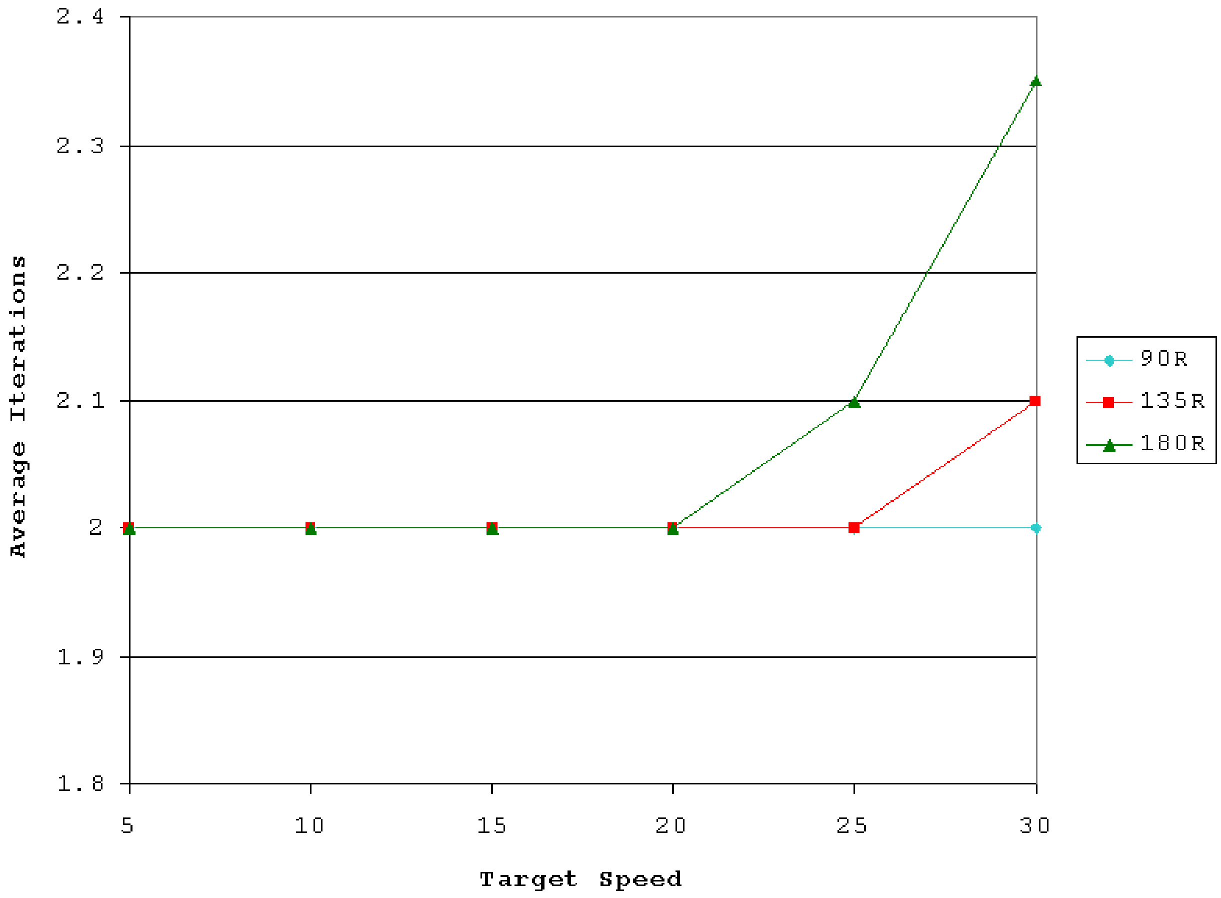

Figure 10. The impact of target speed and target heading on the average number of iterations required the shooting method to determine an aiming solution against a moving target.

The effect of target range (see Figure 11) on the number of iterations required appeared to be parabolic. This occurred because increased range caused increased ballistic flight time. This, in turn, permitted the targets motion to have a greater impact on the ballistic flight time. The method required

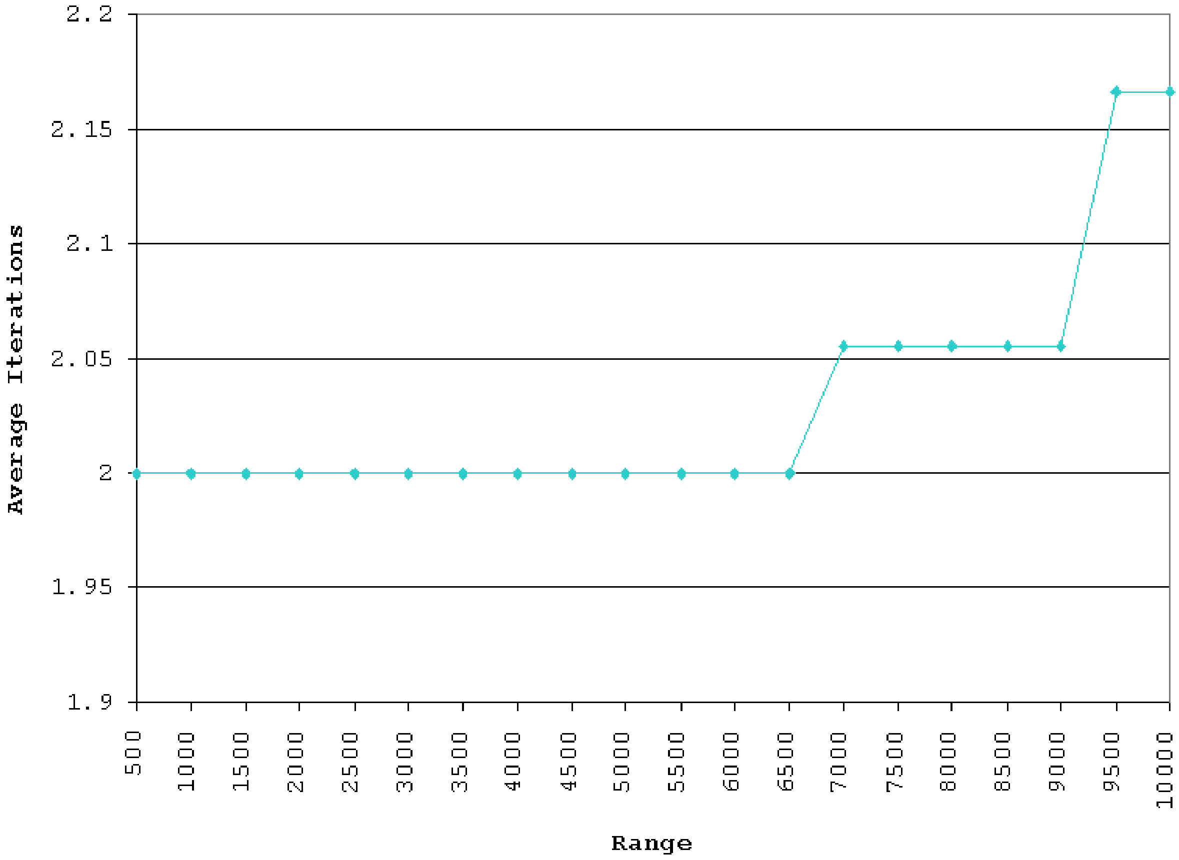

Figure 11. The impact of range on the average number of iterations required the shooting method to determine an aiming solution against a moving target.

between 3ms and 13ms to determine an aiming solution against a moving target (see Table 8). The average time required was 7.473ms, and the standard deviation was 1.576ms. The crosswind penalty caused by the Runga-Kutta impact point determination code was once again evident (see Figure 12). Beyond 4000m, the

Table 8

The Minimum, Maximum, Mean and Standard Deviation of the Compute Time Data from the Shooting Method for Moving Targets Test

	Statistical Measures			
Data Set	Minimum	Maximum	M	SD
All Data	3.000E-03	1.300E-02	7.473E-03	1.576E-03
Target 10mph 90R	3.700E-03	9.000E-03	7.372E-03	1.444E-03
Target 10mph 135R	3.700E-03	8.900E-03	7.338E-03	1.421E-03
Target 10mph 180R	3.600E-03	1.070E-02	7.396E-03	1.412E-03
Target 30mph 90R	3.600E-03	9.700E-03	7.403E-03	1.409E-03
Target 30mph 135R	3.600E-03	1.300E-02	7.530E-03	1.820E-03
Target 30mph 180R	3.500E-03	1.210E-02	8.287E-03	2.343E-03
500m Range	3.700E-03	7.300E-03	6.164E-03	8.765E-04
5000m Range	5.500E-03	9.000E-03	7.912E-03	9.674E-04
10000m Range	5.900E-03	1.300E-02	8.669E-03	1.527E-03
0kt Wind	3.000E-03	8.200E-03	5.479E-03	9.570E-04
20kt Wind	4.400E-03	1.170E-02	7.860E-03	1.414E-03
50kt Wind	4.200E-03	1.300E-02	7.889E-03	1.322E-03

the method appears to require linear increasing computational time (see figure 12). However, the rate of increase is greater under crosswind conditions. This occurs because of the penalty imposed by the crosswind code, which is not run when crosswind

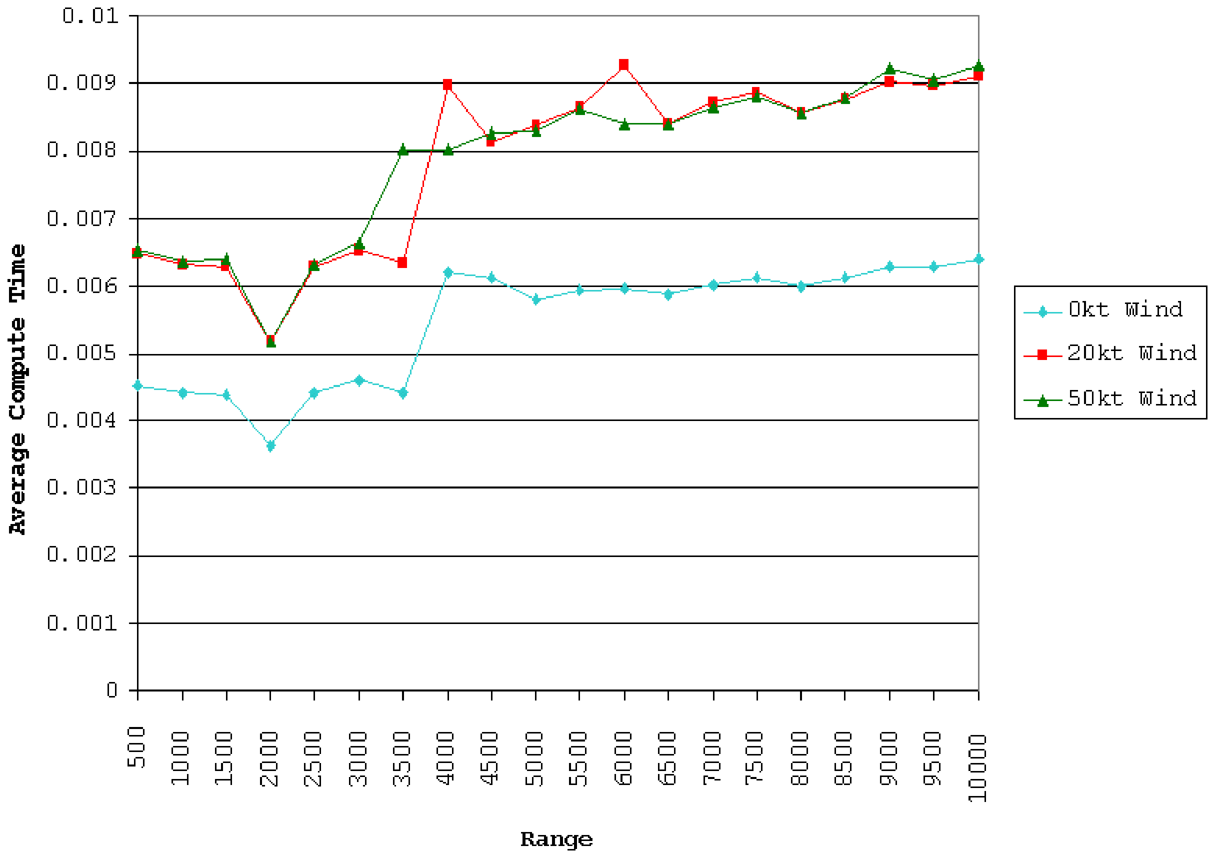

Figure 12. The impact of range on the average computational time required the shooting method to determine an aiming solution against a moving target.

is not present. The depression in figure 12 between 1500m and 4000m is a result of the shooting method for stationary targets code, which is called by the shooting method for moving targets code. The computational time required was noticeably affected

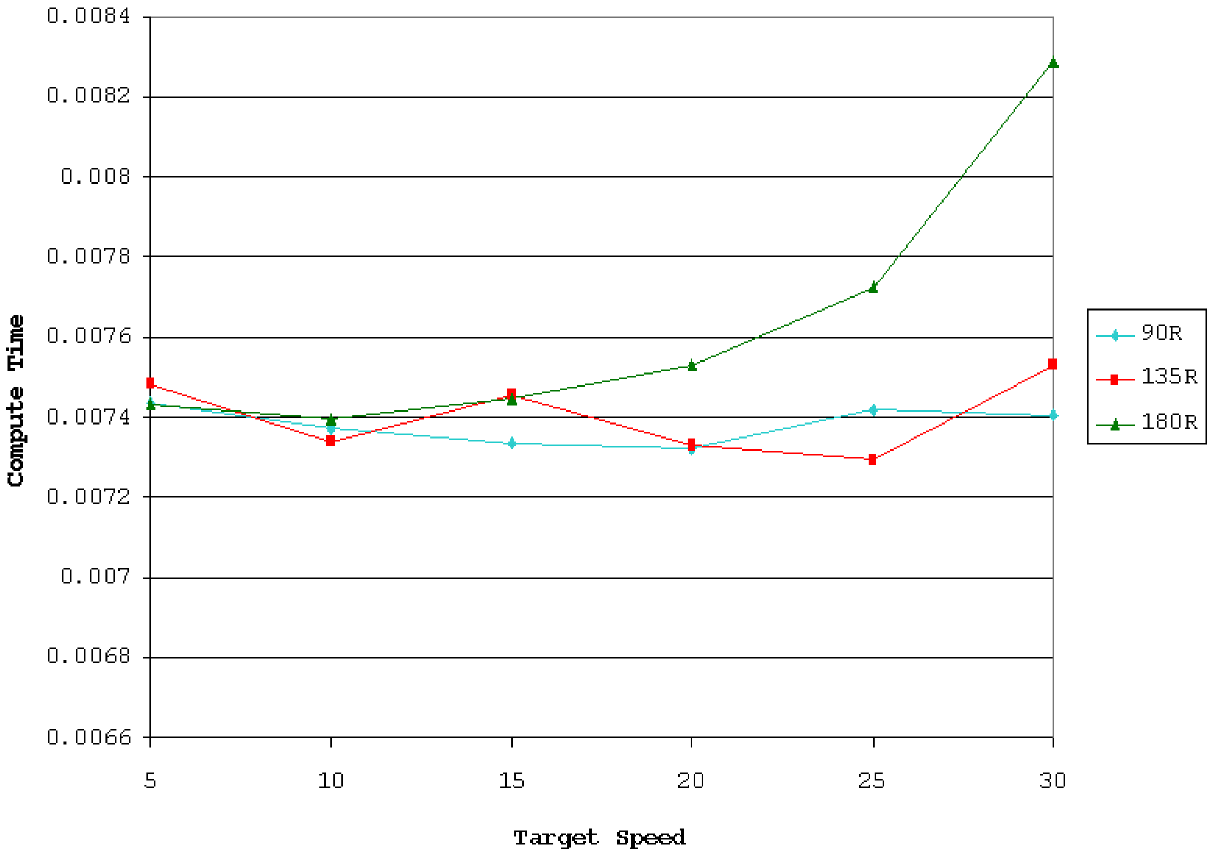

Figure 13. The impact of target speed and target heading on the average computational time required the shooting method to determine an aiming solution against a moving target.

by the target speed, and amplified by the target's direction (see figure 13).

IX. DISCUSSION

The Ballistic Model

The ballistic model presented in this study is fairly accurate for this type of projectile. Since the projectile is not spin stabilized, rotational forces and lifting forces were not considered. These forces would need to be considered for any projectile that is spin stabilized. The computational expense of determining these forces would need to be determined on aiming systems for spin stabilized projectiles.

The drag model employed by the study is accurate at low sub-sonic and at hypersonic speeds (McCoy, 1999). The model is thus relevant to the M829 projectile. Projectiles travelling at low to medium supersonic speeds would require an analysis of the flow field around the projectile at a given speed. The computational expense of determining the flow field would need to be determined for projectiles travelling in this speed range.

The code determined g and rho at every iteration of the Runga-Kutta method. The M829 projectile follows a flat-fire trajectory towards its target. It, therefore, may not be necessary to determine g and rho at every iteration of the Runga-Kutta method. The impact of a constant g and constant rho upon the accuracy of the model deserves investigation.

The Runga-Kutta Method

The Runga-Kutta method performed more efficiently than was anticipated. It was unconditionally accurate at all the tested step sizes. Large portions of the ballistic flight path could thus be stepped over in one time step (1s). In sharp contrast, the Euler method would require the flight path to be stepped at 0.001s intervals in order to achieve internal accuracy. Euler's method would, therefore, require 1000 times as many iterations as the Runga-Kutta method to determine a projectile's position after x seconds of ballistic flight. The Runga-Kutta method only required 21.9% more time per iteration than the Euler method. The Runga-Kutta method was thus 820 times more efficient than the Euler method, in the time domain, for this level of accuracy.

The impact point determination code was efficient, not only as defined by this project, but also in the computational complexity sense of the term. Consider the 0.05° elevation sample, which required 9 iterations. The flight time at this elevation was 0.296s. This implies that the hunting phase consumed 1 iteration, and the convergence phase required 8 iterations. Consider the 1.0° elevation sample, which required 15 iterations. The flight time at this elevation was 5.806s. The hunting phase thus required 6 iterations, and the convergence phase required 9 iterations. Consider the 2.0°

elevation sample, which required 20 iterations. The ballistic flight time was 11.388s. The hunting phase thus required 12 iterations, and the convergence phase required 8 iterations. The amount of time spent in the hunting phase was, thus, linearly proportional to the integer value of the flight time, while the convergence phase was almost constant at around 8 or 9 iterations. The method was, thus, highly predictable and efficient. Improving the algorithm would require an investigation into the improvement of the hunting and convergence phases. Improving the hunting phase would require the investigation of the accuracy of the Runga-Kutta method at step sizes greater than 1s (e.g. 2s, 4s, . . .). This would permit larger time steps to be taken by the hunting phase. It is doubtful that improvements will be found for the convergence phase, since the present method converges from a step size of 1s to a step size of under 10us in 8 to 9 iterations. This represents a 100000 times reduction of step size in only 8 or 9 iterations. The Runga-Kutta impact point determination code is, thus, judged to be computationally efficient.

The Shooting Method

The shooting method for stationary targets determined aiming solutions in 2 to 5 iterations. The method was largely predictable, with most samples requiring either 3 or 4 iterations depending on the range of the target. The negative sinusoidal curve between 1500m and 5000m requires further investigation. It is probable that using an initial guess pair that is more tightly spaced will eliminate the sinusoidal curve, and reduce the range curve to a step curve of either 3 or 4 iterations depending on range, with the exception of the range that coincides with the initial guess pair. It is doubtful that improvements on the present method of convergence will be found, since 2 to 5 iterations is a very low number of iterations for any iterative method; and for most range values, the number of iterations are constant. The method is, thus, judged to be computationally efficient.

The shooting method for moving targets converged on the correct aiming offsets in 2 to 3 iterations. It is extremely unlikely that an improvement on the present algorithm will be discovered. The method is, therefore, judged to be computationally efficient.

Efficiency vs. Accuracy

In any numerical problem, there exists a conflict between efficiency and accuracy. Higher efficiency usually results in reduced accuracy, and vice versa. This project leaned towards accuracy in the ballistic model. In spite of this, the code achieved a best case aiming solution against moving target in 3.11ms, and a worst case aiming solution against a moving target in 13ms. The computational time would be substantially reduced if it were to be determined that a constant g and a constant rho did not significantly impact the operational accuracy of the simulation. The impact of considering extra factors would be akin to the crosswind code penalty noted in this study (see figures 7, 9 and 12), if it were determined that the ballistic model needed to consider more factors in order to be operationally accurate.

The impact point code required at most 20 iteration to determine an impact point. The shooting method for stationary targets code required at most 5 iterations to determine an aiming solution against a stationary target. The shooting method for moving targets code required at most 3 iterations to determine an aiming solution against a moving target. The worst case situation for the simulation is thus 20 x 5 x 3 = 300 iterations of the Runga-Kutta method to determine an aiming solutions against any target. Any systemic efficiency that can be gained in the axial acceleration code would directly impact the computational time required by the code. This also demonstrates that the greatest performance improvement is likely to be gained by improvements in the axial acceleration code, since this code is run most frequently of all code modules.

X. SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

Summary

The project considered a realistic ballistic model as the foundation of all code modules. The Runga-Kutta and Euler methods were compared to determine the step size required to produce an internal accuracy of one decimal place. The Runga-Kutta method was shown to be internally accurate at all the tested step sizes, while the Euler method only achieved internal accuracy at 0.001s. The number of iterations required by the Runga-Kutta method to determine the impact point of a projectile fired at a given elevation was determined. The code required between 8 and 20 iterations to determine the impact point of the projectile. This translated to a computational time requirement of between 279us and 786us. The number of iterations required by the shooting method for stationary targets to determining a firing solution (azimuth and elevation offsets) against a stationary target were determined. The code required between 2 and 5 iterations to determine a firing solution against a stationary target. This translated to between 1.13ms and 3.11ms of computational time. The number of iterations required by the shooting method for moving targets to determine a firing solution against a moving target was determined. The code required between 2 and 3 iterations to determine a firing solution against a moving target. This translated to between 3ms and 13ms of computational time.

Conclusions

All of the hypotheses appear to be supported by the data gathered during this study. The system would definitely provide a combat advantage to tank crews.

Recommendations

The effect of a constant g and a constant rho on the accuracy of the ballistic simulation needs to be investigated. The time required by the simulation will be drastically reduced if these two dynamic factors can be turned into constants.

The study considered a simplified target position and wind system. The software would need to consider targets occurring at altitudes different from the attacker, and wind from any heading; before it could be used in a real world situation.

REFERENCES

Ashour-Abdalla, M. (n.d.). Physics 8A: Physics for Scientists and Engineers. [On-line] Available WWW: http://imed.ucla.edu/physics8a/index.html

Casulli, V., & Greenspan, D. (1988). Numerical Analysis for Applied Mathematics, Science, and Engineering. Addison-Wesley Publishing Company.

Compton's Interactive Encyclopedia [Computer Software]. (1997). Cambridge, USA: Compton's NewMedia, Inc.

Drasco, S. (1998). The Shooting Method. [On-line] Available WWW: http://www.cacr.caltech.edu/~sdrasco/string/node3.html

Fotouhi-Chahouki, R. (1996). The Shooting Method. [On-line] Available WWW: http://www.engr.usask.ca/~ref260/art2/node4.html

Isaacson, J. A., & Vaughan, D. R. (1996). Estimation and Prediction of Ballistic Missile Trajectories. The Rand Corporation.

JBM Small Arms Ballistics. (n.d.). The G1 Drag Model. [On-line] Available WWW: http://www.lascruces.com/~jbm/download/text/mcg1.txt

Kernighan, B. W., & Ritchie, D. M. (1988). The C Programming Language. New Jersey: Prentice-Hall Inc.

Kreyszig, E. (1993). Advanced Engineering Mathematics (Seventh Edition). New York: John Wiley & Sons, Inc.

McCoy, R. L. (1999). Modern Exterior Ballistics: The Launch and Flight Dynamics of Symmetric Projectiles. Schiffer Publishing Ltd.

Federation of American Scientists. (n.d.). M829 120mm, APFSDS-T. [On-line] Available WWW: http://www.fas.org/man/dod-101/sys/ land/m829a1.htm

Federation of American Scientists. (n.d.). 120mm Ammunition. [On-line] Available WWW: http://www.fas.org/man/ dod-101/sys/land/ 120.htm

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (1992). Numerical Recipes in Fortran 77: The Art of Scientific Computing (Second Edition). New York: University of Cambridge.

Restrepo, J. (1999). The Shooting Method. [On-line] Available WWW: http://www.math.arizona.edu/~restrepo/475B/Notes/ source/node21.html

Sarson, S., Sarson, P., & Zaloga. (1993). M1 Abrams: Main Battle Tank. London: New Vanguard.

Schaefer, J. C. (1999a). MORE Power. [On-line] Available WWW: http://home.sprynet.com/~frfrog/power.htm

Schaefer, J. C. (1999b). A Short Course in External Ballistics. [On-line] Available WWW: http://home.sprynet.com/ ~frfrog/extbal.htm

Umar, V. M. (n.d.). Runga-Kutta Methods. [On-line] Available WWW: http://www.npac.syr.edu/projects/csep/ode/node7.html

United States Naval Academy. (n.d.). Fundamentals of Naval Weapons Systems. [On-line] Available WWW: http://www.fas.org/dod-101/navy/docs/fun/

Williams, E. (n.d.). The Aviation Formulary v1.24. [On-line] Available WWW: http://www.best.com/~williams/avform.htm

APPENDIX A: THE AXIAL ACCELERATION CODE

The Implementation File (accel.c)

```
/////////////////////////////////////////////////////////////////////////////
// Filename: accel.c
//
// Purpose: This module provides acceleration data for the projectile
////////////////////////////////////////////////////////////////////////////

///////////////////////////////////////////////////////////////////
// Includes
///////////////////////////////////////////////////////////////////

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include "accel.h"

//////////////////////////////////////////////////////////////////
// Defines
//////////////////////////////////////////////////////////////////

#define     G                 6.67E-11
#define     EARTH_MASS        5.983E+24
#define     EARTH_RADIUS      6370E+3
#define     MACH1             339.9265028
#define     TBL_SCALE         135.9706011
#define     FRONTAL_AREA      1.824146925E-4
#define     LATERAL_AREA      9.5259457E-3
#define     PI                3.1415926535897
#define     MASS              4.2772727272727

/////////////////////////////////////////////////////////////////
// Local structures
/////////////////////////////////////////////////////////////////

struct cd_mach {
      double mach;
      double cd;
};

struct cd_parms {
      double minspeed;
      double maxspeed;
      double a;
      double b;
      double c;
};

struct rho_alt {
      double alt;
      double rho;
};
```

```
struct rho_parms {
      double minalt;
      double maxalt;
      double a;
      double b;
      double c;
};

///////////////////////////////////////////////////////////////////
// Module level global variables
///////////////////////////////////////////////////////////////////

static int tables_setup=0;
static struct cd_parms parabolic_cd[12];
static struct cd_parms parabolic_cd_0;
static struct rho_alt decimated_rho[31];
static struct rho_parms parabolic_rho[15];

//////////////////////////////////////////////////////////////////////
// Standard value functions
//////////////////////////////////////////////////////////////////////

// This function returns the g at a given altitude using standard values
double get_std_g(double alt)
{
      double r;

      r = alt + EARTH_RADIUS;
      return -G*EARTH_MASS/(r*r);
}

// This function return the rho for a given altitude for a standard
atmosphere
double get_std_rho(double alt)
{
      double feet, x;

      feet=alt/0.3048;
      x = 1.0 - feet*6.8755856E-6;
      return 1.225*pow(x, 4.2558797);
}

//////////////////////////////////////////////////////////////////////
// Setup functions
//////////////////////////////////////////////////////////////////////

// This function sets up the drag coefficient interpolation tables
void setup_cd_tables()
{
      int x, y;
      double y2, y1, y0, x0, c, b, a, delta;
      extern struct cd_parms parabolic_cd[12];
      extern struct cd_parms parabolic_cd_0;
      struct cd_mach decimated_cd[] = {
            {0.2, 0.2344},
            {0.4, 0.2104},
            {0.6, 0.2034},
```

```
            {0.8, 0.2546},
            {1.0, 0.4805},
            {1.2, 0.6393},
            {1.4, 0.6625},
            {1.6, 0.6474},
            {1.8, 0.6210},
            {2.0, 0.5934},
            {2.2, 0.5685},
            {2.4, 0.5481},
            {2.6, 0.5325},
            {2.8, 0.5211},
            {3.0, 0.5133},
            {3.2, 0.5084},
            {3.4, 0.5054},
            {3.6, 0.5030},
            {3.8, 0.5016},
            {4.0, 0.5006},
            {4.2, 0.4998},
            {4.4, 0.4995},
            {4.6, 0.4992},
            {4.8, 0.4990},
            {5.0, 0.4988}
      };

      delta=MACH1*0.2;
      for (x=0; x <= 11; x++) {
            y=x*2;
            y2 = decimated_cd[y+2].cd;
            y1 = decimated_cd[y+1].cd;
            y0 = decimated_cd[y].cd;
            x0 = decimated_cd[y].mach*MACH1;
            parabolic_cd[x].minspeed = decimated_cd[y].mach*MACH1;
            parabolic_cd[x].maxspeed = decimated_cd[y+2].mach*MACH1;
            parabolic_cd[x].c = c = (y2 - 2*y1 + y0)/(2*delta*delta);
            parabolic_cd[x].b = b = (y1 - y0 -2*c*delta*x0 -
                  c*delta*delta)/delta;
            parabolic_cd[x].a = a = y0 - b*x0 - c*x0*x0;
      }
      y2 = 0.2104;
      y1 = 0.2344;
      y0 = 0.2629;
      x0 = 0.0;
      parabolic_cd_0.minspeed = 0.0;
      parabolic_cd_0.maxspeed = 0.1*MACH1;
      parabolic_cd_0.c = c = (y2 - 2*y1 + y0)/(2*delta*delta);
      parabolic_cd_0.b = b = (y1 - y0 -2*c*delta*x0 - c*delta*delta)/delta;
      parabolic_cd_0.a = a = y0 - b*x0 - c*x0*x0;
}

// This function sets up the rho interpolation tables
void setup_rho_tables()
{
      int x, y;
      double y2, y1, y0, x0, c, b, a;
      extern struct rho_alt decimated_rho[31];
      extern struct rho_parms parabolic_rho[15];
      double delta;
```

```
	delta = 10.0;

	for (x=0; x <= 30; x+=1) {
		decimated_rho[x].alt = (double)x*delta;
		decimated_rho[x].rho = get_std_rho((double)x*delta);
	}
	for (x=0; x <= 14; x++) {
		y=x*2;
		y2 = decimated_rho[y+2].rho;
		y1 = decimated_rho[y+1].rho;
		y0 = decimated_rho[y].rho;
		x0 = decimated_rho[y].alt;
		parabolic_rho[x].minalt = decimated_rho[y].alt;
		parabolic_rho[x].maxalt = decimated_rho[y+2].alt;
		parabolic_rho[x].c = c = (y2 - 2*y1 + y0)/(2*delta*delta);
		parabolic_rho[x].b = b = (y1 - y0 -2*c*delta*x0 -
			c*delta*delta)/delta;
		parabolic_rho[x].a = a = y0 - b*x0 - c*x0*x0;
	}
}

// This is a one stop setup all interpolation tables function
void setup_interpolation_tables(void)
{
	extern int tables_setup;

	setup_cd_tables();
	setup_rho_tables();
	tables_setup=1;
}

////////////////////////////////////////////////////////////////////
// Interpolation functions
////////////////////////////////////////////////////////////////////

// This function interpolates the drag coefficient for a given speed
double interpolate_cd(double speed)
{
	double dbl_idx, abs_speed;
	int idx;
	extern struct cd_parms parabolic_cd[12];
	extern int tables_setup;

	if (!tables_setup) {
		printf("Interpolation tables not setup.\n");
		exit(0);
	}

	abs_speed=fabs(speed);
	dbl_idx = (abs_speed - parabolic_cd[0].minspeed)/TBL_SCALE;
	dbl_idx = floor(dbl_idx);
	idx=(int)dbl_idx;
	if (idx <  0)
		return parabolic_cd_0.a + parabolic_cd_0.b*abs_speed +
parabolic_cd_0.c*abs_speed*abs_speed;
	if (idx > 11) idx=11;
```

```
      return parabolic_cd[idx].a + parabolic_cd[idx].b*abs_speed +
parabolic_cd[idx].c*abs_speed*abs_speed;
}

// This function interpolates rho for a given altitude
double interpolate_rho(double alt)
{
      double floor_alt;
      int int_alt, idx;
      extern struct rho_alt decimated_rho[31];
      extern struct rho_parms parabolic_rho[15];
      extern int tables_setup;

      if (!tables_setup) {
            printf("Interpolation tables not setup.\n");
            exit(0);
      }

      if (alt < 0) alt=0;
      floor_alt=floor(alt);
      int_alt=(int)floor_alt;
      idx=int_alt/20;
      if (idx < 0) idx=0;
      if (idx > 14) idx=14;

      return parabolic_rho[idx].a + parabolic_rho[idx].b*alt +
            parabolic_rho[idx].c*alt*alt;
}

////////////////////////////////////////////////////////////////
// Axial Acceleration Code
////////////////////////////////////////////////////////////////

// This routine determines the current x, y and z accelerations
struct accel_snapshot get_accel(struct flight_snapshot * m829)
{
      double g;
      double frontal_cd, z_cd;
      double rho;
      double xyz_speed, xz_speed;
      double z_area;
      double frontal_force, z_force;
      double frontal_accel;
      struct accel_snapshot accel;

      // Get 3d air speed
      xyz_speed = sqrt(m829->xv*m829->xv + m829->yv*m829->yv +
            m829->zv*m829->zv);

      // Get g at current altitude
      g = get_std_g(m829->ys);

      // Get rho at the current altitude
      rho = interpolate_rho(m829->ys);

      // Get the projectile's frontal
```

```
	frontal_cd = interpolate_cd(xyz_speed);

	// Determine frontal force and the resulting axial accelerations
	frontal_force = 0.5*rho*FRONTAL_AREA*frontal_cd*xyz_speed*xyz_speed;
	frontal_accel = frontal_force/MASS;
	accel.xa=-1*m829->xv*frontal_accel/xyz_speed;
	accel.ya=-1*m829->yv*frontal_accel/xyz_speed + g;
	accel.za=-1*m829->zv*frontal_accel/xyz_speed;

	// Determine the impact of crosswind
	if (m829->crosswind != 0.0 ) {
		xz_speed = sqrt(m829->xv*m829->xv + m829->zv*m829->zv);
		z_cd = interpolate_cd(m829->crosswind);
		z_area = fabs(m829->xv)*LATERAL_AREA/xz_speed +
			fabs(m829->zv)*FRONTAL_AREA/xz_speed;
		z_force = 0.5*rho*z_area*z_cd*m829->crosswind*m829->crosswind;
		accel.za+=copysign(1,m829->crosswind)*(z_force/MASS);
	}

	return accel;
}
```

The Header File (accel.h)

```
/////////////////////////////////////////////////////////////////
// Filename: rk.h
/////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////
// Global structures
/////////////////////////////////////////////////////////////////

struct accel_snapshot {
	double xa, ya, za;
};

struct flight_snapshot {
	double xs, ys, zs;
	double xv, yv, zv;
	double crosswind;
	double elapsed_time;
	int iterations;
};

/////////////////////////////////////////////////////////////////
// Public functions
/////////////////////////////////////////////////////////////////

double get_std_g(double alt);
double interpolate_cd(double speed);
double get_std_rho(double alt);
double interpolate_rho(double alt);
void setup_interpolation_tables(void);
struct accel_snapshot get_accel(struct flight_snapshot * m829);
```

APPENDIX B: THE EULER METHOD CODE

```
///////////////////////////////////////////////////////////////////////
// Filename: euler_stepsize_accuracy.c
//
// Purpose: Does Euler method as a baseline for the rk tests
///////////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////////
// Includes
//////////////////////////////////////////////////////////////////////

#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <time.h>
#include "accel.h"

//////////////////////////////////////////////////////////////////////
// Module functions
//////////////////////////////////////////////////////////////////////

// This routine is responsible for taking a single step across the flight
path
void do_euler_step(struct flight_snapshot * m829, double step)
{
      struct accel_snapshot accel;

      accel = get_accel(m829);

      m829->xs+=step*m829->xv + step*step*accel.xa/2;
      m829->ys+=step*m829->yv + step*step*accel.ya/2;
      m829->zs+=step*m829->zv + step*step*accel.za/2;

      m829->xv+= step*accel.xa;
      m829->yv+= step*accel.ya;
      m829->zv+= step*accel.za;

      m829->elapsed_time+=step;
}

// This routine steps across the flight path using the selected step size
int main(int argc, char **argv)
{
      struct flight_snapshot input;
      float stepsize;
      time_t start, stop;

      if (argc == 2)
            stepsize=atof(argv[1]);
      else {
            printf("Please supply a step size\n");
            exit(0);
      }
```

```
      setup_interpolation_tables();

      input.xs=0.0;
      input.ys=0.0;
      input.zs=0.0;
      input.xv=1668.982681;
      input.yv=58.282159490;
      input.zv=0.0;
      input.crosswind=25.694;
      input.elapsed_time=0.0;

      start=clock();
      do {
            do_euler_step(&input, stepsize);
            stop=clock();
            printf("%f\t%f\t%f\t%f\t%f\t%f\n",input.elapsed_time, input.xs,
                  input.ys, input.zs, stepsize, ((double)(stop-
                  start))/CLOCKS_PER_SEC);
      } while (input.elapsed_time < 8.0);

      return 0;

}
```

APPENDIX C: THE RUNGA-KUTTA CODE

The Implementation File (rk.c)

```
/////////////////////////////////////////////////////////////////////
// Filename: rk.c
//
// Purpose: Implementation of Runga-Kutta Routines
/////////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////////
// Includes
/////////////////////////////////////////////////////////////////////

#include <math.h>
#include <stdlib.h>
#include "rk.h"

////////////////////////////////////////////////////////////////////
// Defines
////////////////////////////////////////////////////////////////////

#define PI              3.1415926535897
#define LAUNCH_SPEED    1670.304

///////////////////////////////////////////////////////////////////
// Module Functions
///////////////////////////////////////////////////////////////////

// This function takes a single RK step across the flight path
void do_rk_step(struct flight_snapshot * m829, double step)
{
      struct accel_snapshot accel;
      struct flight_snapshot k_flight;
      struct rk {
            double x, y, z;
      } k0, k1, k2, k3;

      // Determine K0
      k_flight = *m829;
      accel = get_accel(&k_flight);
      k0.x=step*accel.xa;
      k0.y=step*accel.ya;
      k0.z=step*accel.za;

      // Determine K1
      k_flight = *m829;
      k_flight.xv+=k0.x/2;
      k_flight.yv+=k0.y/2;
      k_flight.zv+=k0.z/2;
      k_flight.xs+=step*m829->xv/2;
      k_flight.ys+=step*m829->yv/2;
      k_flight.zs+=step*m829->zv/2;
      accel = get_accel(&k_flight);
      k1.x=step*accel.xa;
      k1.y=step*accel.ya;
```

```
        k1.z=step*accel.za;

        // Determine K2
        k_flight = *m829;
        k_flight.xv+=k1.x/2;
        k_flight.yv+=k1.y/2;
        k_flight.zv+=k1.z/2;
        k_flight.xs+=step*m829->xv/2 + step*k0.x/4;
        k_flight.ys+=step*m829->yv/2 + step*k0.y/4;
        k_flight.zs+=step*m829->zv/2 + step*k0.z/4;
        accel = get_accel(&k_flight);
        k2.x=step*accel.xa;
        k2.y=step*accel.ya;
        k2.z=step*accel.za;

        // Determine K3
        k_flight = *m829;
        k_flight.xv+=k2.x;
        k_flight.yv+=k2.y;
        k_flight.zv+=k2.z;
        k_flight.xs+=step*m829->xv + step*k1.x/2;
        k_flight.ys+=step*m829->yv + step*k1.y/2;
        k_flight.zs+=step*m829->zv + step*k1.z/2;
        accel = get_accel(&k_flight);
        k3.x=step*accel.xa;
        k3.y=step*accel.ya;
        k3.z=step*accel.za;

        m829->xs+=step*m829->xv+(k0.x + k1.x + k2.x)*(step/6);
        m829->ys+=step*m829->yv+(k0.y + k1.y + k2.y)*(step/6);
        m829->zs+=step*m829->zv+(k0.z + k1.z + k2.z)*(step/6);

        m829->xv+= (k0.x + 2*k1.x + 2*k2.x + k3.x)/6;
        m829->yv+= (k0.y + 2*k1.y + 2*k2.y + k3.y)/6;
        m829->zv+= (k0.z + 2*k1.z + 2*k2.z + k3.z)/6;

        m829->elapsed_time+=step;
        m829->iterations++;
}

// This routine is responsible for stepping across the entire flight path
// and determining the exact impact point
struct flight_snapshot get_rk_impact_point(struct launch_info * t_0)
{
        struct flight_snapshot current, last_positive;
        double azrad, elevrad, levelspeed;
        double stepsize;

        current.elapsed_time=0.0;
        current.iterations=0;
        current.xs=0.0;
        current.ys=0.0;
        current.zs=0.0;
        current.crosswind=t_0->crosswind;

        elevrad = t_0->elevation * PI / 180.0;
        azrad =t_0->azimuth * PI / 180;
```

```
	current.yv=LAUNCH_SPEED*sin(elevrad);
	levelspeed=LAUNCH_SPEED*cos(elevrad);

	current.zv=levelspeed*sin(azrad);
	current.xv=levelspeed*cos(azrad);

	stepsize=1.0;

	do {
		last_positive = current;
		do_rk_step(&current, stepsize);
		if (current.ys < 0) {
			stepsize=(current.elapsed_time -
				last_positive.elapsed_time)/2;
			current=last_positive;
		}
	} while (stepsize > 1E-5);

	return current;
}
```

The Header File (rk.h)

```
//////////////////////////////////////////////////////////////////////
// Filename: rk.h
//
// Purpose: rk.c header file
//////////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////////
// Dependent includes
//////////////////////////////////////////////////////////////////////

#include "accel.h"

//////////////////////////////////////////////////////////////////////
// Global structures
//////////////////////////////////////////////////////////////////////

struct launch_info {
	double elevation;
	double azimuth;
	double crosswind;
};

/////////////////////////////////////////////////////////////////////
// Public functions
/////////////////////////////////////////////////////////////////////

void do_rk_step(struct flight_snapshot * m829, double step);
struct flight_snapshot get_rk_impact_point(struct launch_info * t_0);
```

APPENDIX D: THE RK STEPSIZE ACCURACY CODE

```
//////////////////////////////////////////////////////////////////////
// Filename: rk_stepsize_accuracy.c
//
// Purpose: Determining maximum stepsize which does not compromise
//          accuracy
//////////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////////
// Include
/////////////////////////////////////////////////////////////////////

#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <time.h>
#include "rk.h"

////////////////////////////////////////////////////////////////////
// Module Functions
////////////////////////////////////////////////////////////////////

int main(int argc, char **argv)
{
      struct flight_snapshot input;
      float stepsize;
      time_t start, stop;

      if (argc == 2)
            stepsize=atof(argv[1]);
      else {
            printf("Please supply a step size\n");
            exit(0);
      }

      setup_interpolation_tables();

      input.xs=0.0;
      input.ys=0.0;
      input.zs=0.0;
      input.xv=1668.982681;
      input.yv=58.282159490;
      input.zv=0.0;
      input.crosswind=25.694;
      input.elapsed_time=0.0;

      start=clock();
      do {
            do_rk_step(&input, stepsize);
            stop=clock();
            printf("%f\t%f\t%f\t%f\t%f\t%f\n",input.elapsed_time, input.xs,
                  input.ys, input.zs, stepsize, ((double)(stop-
                  start))/CLOCKS_PER_SEC);
      } while (input.elapsed_time < 8.0);
```

```
        return 0;
}
```

APPENDIX E: THE STEP SIZE ACCURACY TEST BATCH FILE

```
echo Runga-Kutta
./rk_stepsize_accuracy 1.0 | grep ^7.00000
./rk_stepsize_accuracy 0.1 | grep ^7.00000
./rk_stepsize_accuracy 0.01 | grep ^7.00000
./rk_stepsize_accuracy 0.001 | grep ^7.00000
./rk_stepsize_accuracy 0.0001 | grep ^7.000000
./rk_stepsize_accuracy 0.00001 | grep ^7.000000
./rk_stepsize_accuracy 0.000001 | grep ^7.000000
echo
echo Euler
./euler_stepsize_accuracy 1.0 | grep ^7.00000
./euler_stepsize_accuracy 0.1 | grep ^7.00000
./euler_stepsize_accuracy 0.01 | grep ^7.00000
./euler_stepsize_accuracy 0.001 | grep ^7.00000
./euler_stepsize_accuracy 0.0001 | grep ^7.000000
./euler_stepsize_accuracy 0.00001 | grep ^7.000000
./euler_stepsize_accuracy 0.000001 | grep ^7.000000
```

APPENDIX F: THE RUNGA-KUTTA ITERATION TEST CODE

```
//////////////////////////////////////////////////////////////////////
// Filename: rk_iterations.c
//
// Purpose: Determines time / iterations required to determine
//          the impact point for given launch conditions
//////////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////////
// Includes
/////////////////////////////////////////////////////////////////////

#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <time.h>
#include "rk.h"

/////////////////////////////////////////////////////////////////////
// Defines
/////////////////////////////////////////////////////////////////////

#define ITERATIONS 10000

////////////////////////////////////////////////////////////////////
// Module Functions
////////////////////////////////////////////////////////////////////

int main(int argc, char **argv)
{
      struct launch_info aim;
      struct flight_snapshot impact;
      time_t time1, time2;
      int int_elevation, int_wind;
      double elevation, wind;
      int it;

      setup_interpolation_tables();

      for (int_elevation=1; int_elevation<=40; int_elevation++)
            for(int_wind=0; int_wind<=50; int_wind+=10) {
                  elevation = (double)int_elevation/20;
                  wind = (double)int_wind * 1.85 / 3.6;
                  aim.elevation=elevation;
                  aim.azimuth=0.0;
                  aim.crosswind=wind;
                  time1=clock();
                  for (it=0; it<ITERATIONS; it++)
                        impact=get_rk_impact_point(&aim);
                  time2=clock();
                  printf("%f\t%d\t%f\t%f\t%d\t%f\n", elevation, int_wind,
                        impact.xs,
                        impact.zs,
                        impact.iterations,
```

```
                        ((double)(time2-time1)/CLOCKS_PER_SEC)/ITERATIONS);
        }

        return 0;

}
```

APPENDIX G: THE SHOOTING METHOD FOR STATIONARY TARGET CODE

The Implementation File (smst.c)

```
///////////////////////////////////////////////////////////////
// Filename: smst.c
//
// Purpose: This modules determines aiming solutions
//          against stationary targets
///////////////////////////////////////////////////////////////

///////////////////////////////////////////////////////////////
// Includes
///////////////////////////////////////////////////////////////

#include <math.h>
#include <stdio.h>
#include "smst.h"

//////////////////////////////////////////////////////////////
// Module Functions
//////////////////////////////////////////////////////////////

// Determine impact point error
struct shot_error get_impact_error(struct launch_aim *aim)
{
      struct flight_snapshot ip;
      struct shot_error se;

      ip = get_rk_impact_point(&aim->aim);

      se.boom = ip;
      se.xerror = ip.xs - aim->range;
      se.zerror = ip.zs - aim->crosstrack;

      return se;
}

// Determine correct aim point for the target location
// using the shooting method
struct aiming_solution smst(struct aim_point *ap)
{
      struct launch_aim old, current, next;
      struct shot_error old_error, current_error;
      struct aiming_solution solution;
      double dx, dz, radial_error;
      int iterations;


      old.aim.elevation=0.18;
      old.aim.azimuth=0.18;
      old.aim.crosswind=ap->crosswind;
      old.range=ap->range;
      old.crosstrack=ap->crosstrack;
      old_error=get_impact_error(&old);
```

```
	current.aim.elevation=0.2;
	current.aim.azimuth=0.2;
	current.aim.crosswind=ap->crosswind;
	current.range=ap->range;
	current.crosstrack=ap->crosstrack;
	next=current;
	current_error=get_impact_error(&current);

	iterations=0;
	do {
		iterations++;
		dx=(current_error.xerror - old_error.xerror)/
			(current.aim.elevation - old.aim.elevation);
		dz=(current_error.zerror - old_error.zerror)/
			(current.aim.azimuth - old.aim.azimuth);
		next.aim.elevation = current.aim.elevation -
			current_error.xerror/dx;
		next.aim.azimuth = current.aim.azimuth -
			current_error.zerror/dz;
		old=current;
		old_error=current_error;
		current=next;
		current_error=get_impact_error(&current);
		radial_error = current_error.xerror*current_error.xerror +
			current_error.zerror*current_error.zerror;
		radial_error = sqrt(radial_error);
	} while (radial_error > 0.25);
	solution.elevation=current.aim.elevation;
	solution.azimuth=current.aim.azimuth;
	solution.tof=current_error.boom.elapsed_time;
	solution.iterations=iterations;

	return solution;
}
```

The Header File (smst.h)

```
////////////////////////////////////////////////////////////
// Filename: smst.h
//
// Purpose: Header file for smst code
////////////////////////////////////////////////////////////

///////////////////////////////////////////////////////////
// Dependent includes
///////////////////////////////////////////////////////////

#include "rk.h"

//////////////////////////////////////////////////////////
// Global structures
//////////////////////////////////////////////////////////

struct shot_error {
	struct flight_snapshot boom;
	double xerror;
	double zerror;
```

```
};

struct launch_aim {
	struct launch_info aim;
	double range;
	double crosstrack;
};

struct aim_point {
	double range;
	double crosstrack;
	double crosswind;
};

struct aiming_solution {
	double elevation;
	double azimuth;
	double tof;
	int iterations;
};

///////////////////////////////////////////////////////////////
// Public functions
///////////////////////////////////////////////////////////////

struct shot_error get_impact_error(struct launch_aim *aim);
struct aiming_solution smst(struct aim_point *ap);
```

APPENDIX H: THE SHOOTING METHOD FOR STATIONARY TARGETS TEST CODE

```
///////////////////////////////////////////////////////////////////
// Filename: smst_test.c
//
// Purpose: Test smst functions
///////////////////////////////////////////////////////////////////

///////////////////////////////////////////////////////////////////
// Includes
///////////////////////////////////////////////////////////////////

#include <time.h>
#include <stdio.h>
#include "smst.h"

///////////////////////////////////////////////////////////////////
// Defines
///////////////////////////////////////////////////////////////////

#define ITERATIONS 1000

///////////////////////////////////////////////////////////////////
// Module functions
///////////////////////////////////////////////////////////////////

int main()
{
      struct aim_point ap;
      struct aiming_solution solution;
      int range, wind;
      time_t start, stop;
      int x;

      setup_interpolation_tables();

      for (range=500; range<=10000; range+=500)
            for (wind=0; wind<=50; wind+=10) {
                  ap.range = (double)range;
                  ap.crosstrack = 0.0;
                  ap.crosswind = (double)wind*1.85/3.6;

                  start=clock();
                  for (x=0; x<ITERATIONS; x++)
                        solution=smst(&ap);
                  stop=clock();

                  printf("%d\t%d\t%f\t%f\t%f\t%d\t%f\n", range, wind,
                        solution.elevation, solution.azimuth,
                        solution.tof, solution.iterations,
                        ((double)(stop-start)/CLOCKS_PER_SEC)/ITERATIONS);
            }

      return 0;
}
```

APPENDIX I: THE SHOOTING METHOD FOR MOVING TARGETS CODE

The Implementation File (smmt.c)

```
/////////////////////////////////////////////////////////////////
// Filename: smmt.c
//
// Purpose: Provides aiming solutions against moving targets
/////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////
// Includes
/////////////////////////////////////////////////////////////////

#include <stdio.h>
#include <math.h>
#include "smmt.h"

/////////////////////////////////////////////////////////////////
// Defines
/////////////////////////////////////////////////////////////////

#define PI  3.1415926535897

/////////////////////////////////////////////////////////////////
// Module functions
/////////////////////////////////////////////////////////////////

struct aiming_solution smmt(struct mt_info *mt)
{
      struct aim_point old_aim, current_aim;
      struct aiming_solution old_solution, current_solution;
      double x_delta, z_delta, radial_error;
      double x_speed, z_speed;
      int iterations;

      x_speed = cos(mt->heading * PI / 180) * mt->speed;
      z_speed = sin(mt->heading * PI / 180) * mt->speed;


      current_aim.range=mt->range;
      current_aim.crosstrack=mt->crosstrack;
      current_aim.crosswind=mt->crosswind;
      current_solution = smst(&current_aim);
      iterations=0;

      do {
            iterations++;
            old_aim=current_aim;
            old_solution=current_solution;

            current_aim.range=mt->range + x_speed*old_solution.tof;
            current_aim.crosstrack=mt->crosstrack + z_speed*old_solution.tof;
            current_aim.crosswind=mt->crosswind;
            current_solution=smst(&current_aim);
```

```
            x_delta = old_aim.range - current_aim.range;
            z_delta = old_aim.crosstrack - current_aim.crosstrack;
            radial_error = sqrt(x_delta*x_delta + z_delta*z_delta);

      } while (radial_error > 0.5);

      current_solution.iterations=iterations;

      return current_solution;
}
```

The Header File (smmt.h)

```
///////////////////////////////////////////////////////////////
// Filename: smmt.h
//
// Purpose: smmt.c header file
//////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////
// Includes
//////////////////////////////////////////////////////////////

#include "smst.h"

/////////////////////////////////////////////////////////////
// Global structures
/////////////////////////////////////////////////////////////

struct mt_info {
      double range;
      double crosstrack;
      double speed;
      double heading;
      double crosswind;
};

////////////////////////////////////////////////////////////
// Public functions
////////////////////////////////////////////////////////////

struct aiming_solution smmt(struct mt_info *mt);
```

APPENDIX J: THE SHOOTING METHOD FOR MOVING TARGETS TEST CODE

```
///////////////////////////////////////////////////////////////////////
// Filename: smmt_test.c
//
// Purpose: Tests smmt function
///////////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////////
// Includes
//////////////////////////////////////////////////////////////////////

#include <time.h>
#include <stdio.h>
#include "smmt.h"

/////////////////////////////////////////////////////////////////////
// Defines
/////////////////////////////////////////////////////////////////////

#define ITERATIONS 100

/////////////////////////////////////////////////////////////////////
// Module functions
/////////////////////////////////////////////////////////////////////

int main()
{
      struct mt_info mt;
      struct aiming_solution solution;
      int range, wind, speed, direction, iterations;
      time_t stop, start;

      setup_interpolation_tables();

      for (range=500; range<=10000; range+=500)
        for (wind=0; wind<=50; wind+=10)
          for (speed=5; speed<=30; speed+=5)
            for (direction=90; direction<=180; direction+=45) {
                  mt.range=(double)range;
                  mt.crosstrack=0.0;
                  mt.crosswind=(double)wind;
                  mt.speed=(double)speed*1.6/3.6;
                  mt.heading=(double)direction;

                  start=clock();
                  for (iterations=0; iterations<ITERATIONS; iterations++)
                        solution=smmt(&mt);
                  stop=clock();
                  printf("%d\t%d\t%d\t%d\t%f\t%f\t%d\t%f\n", range, wind,
                        speed, direction, solution.elevation,
                        solution.azimuth, solution.iterations,
                        ((double)(stop-start)/CLOCKS_PER_SEC)/
                              ITERATIONS);
             }
      return 0;
```

```
    }
```

APPENDIX K: RUNGA-KUTTA METHOD VS. EULER METHOD STEPSIZE

ACCURACY RESULTS

Runga-Kutta Method

Flight Time (s)	Range (m)	Altitude (m)	Crosstrack (m)	Stepsize (s)	Compute Time (s)
7.000000	10880.420059	149.878296	5.253693	1.000000	0.000000
7.000000	10880.420659	149.878323	5.253692	0.100000	0.010000
7.000000	10880.420282	149.878327	5.253692	0.010000	0.040000
7.000000	10880.420992	149.878320	5.253693	0.001000	0.440000
7.000000	10880.420253	149.878327	5.253692	0.000100	4.230000
7.000000	10880.420253	149.878327	5.253692	0.000010	44.680000
7.000000	10880.420484	149.878325	5.253692	0.000001	458.610000

Euler Method

Flight Time (s)	Range (m)	Altitude (m)	Crosstrack (m)	Stepsize (s)	Compute Time (s)
7.000000	10864.448508	147.223071	5.310342	1.000000	0.000000
7.000000	10878.798081	149.606268	5.259473	0.100000	0.010000
7.000000	10880.257796	149.851059	5.254271	0.010000	0.030000
7.000000	10880.404741	149.875593	5.253751	0.001000	0.350000
7.000000	10880.418628	149.878054	5.253698	0.000100	3.370000
7.000000	10880.420090	149.878300	5.253692	0.000010	36.260000
7.000000	10880.420467	149.878322	5.253692	0.000001	376.200000

APPENDIX L: THE RUNGA-KUTTA ITERATION TEST RESULTS

Elevation (°)	Wind Speed (kts)	Range (m)	Crosstrack (m)	Iterations (#)	Compute Time (s)
0.050000	0	492.984495	0.000000	9	0.000290
0.050000	10	492.984495	0.000411	9	0.000428
0.050000	20	492.984495	0.001628	9	0.000428
0.050000	30	492.984495	0.003631	9	0.000426
0.050000	40	492.984495	0.006399	9	0.000432
0.050000	50	492.984495	0.009912	9	0.000429
0.100000	0	981.789227	0.000000	9	0.000293
0.100000	10	981.789227	0.001636	9	0.000432
0.100000	20	981.789227	0.006485	9	0.000436
0.100000	30	981.789227	0.014463	9	0.000429
0.100000	40	981.789227	0.025488	9	0.000427
0.100000	50	981.789227	0.039478	9	0.000428
0.150000	0	1466.447747	0.000000	10	0.000301
0.150000	10	1466.447747	0.003664	10	0.000444
0.150000	20	1466.447747	0.014528	10	0.000445
0.150000	30	1466.447747	0.032403	10	0.000444
0.150000	40	1466.447747	0.057102	10	0.000444
0.150000	50	1466.447747	0.088445	10	0.000452
0.200000	0	1947.067707	0.000000	10	0.000307
0.200000	10	1947.067707	0.006487	10	0.000446
0.200000	20	1947.067707	0.025719	10	0.000447
0.200000	30	1947.067707	0.057361	10	0.000446
0.200000	40	1947.067707	0.101085	10	0.000445
0.200000	50	1947.067707	0.156571	10	0.000445
0.250000	0	2423.655897	0.000000	10	0.000302
0.250000	10	2423.655897	0.010092	10	0.000444
0.250000	20	2423.655897	0.040015	10	0.000443
0.250000	30	2423.655897	0.089246	10	0.000444
0.250000	40	2423.655897	0.157275	10	0.000445
0.250000	50	2423.655897	0.243604	10	0.000446
0.300000	0	2896.341919	0.000000	8	0.000279
0.300000	10	2896.341919	0.014472	8	0.000413
0.300000	20	2896.341919	0.057379	8	0.000414
0.300000	30	2896.341919	0.127974	8	0.000414
0.300000	40	2896.341919	0.225523	8	0.000412
0.300000	50	2896.341919	0.349314	8	0.000411
0.350000	0	3365.130656	0.000000	11	0.000313
0.350000	10	3365.130656	0.019615	11	0.000462
0.350000	20	3365.130656	0.077771	11	0.000461
0.350000	30	3365.130656	0.173453	11	0.000461
0.350000	40	3365.130656	0.305671	11	0.000459
0.350000	50	3365.130656	0.473454	11	0.000460
0.400000	0	3830.100129	0.000000	12	0.000324
0.400000	10	3830.100129	0.025512	12	0.000480
0.400000	20	3830.100129	0.101152	12	0.000478
0.400000	30	3830.100129	0.225601	12	0.000476
0.400000	40	3830.100129	0.397568	12	0.000477
0.400000	50	3830.100129	0.615795	12	0.000477
0.450000	0	4291.302624	0.000000	10	0.000300
0.450000	10	4291.302624	0.032153	10	0.000443
0.450000	20	4291.302624	0.127484	10	0.000444

0.450000	30	4291.302624	0.284330	10	0.000444
0.450000	40	4291.302624	0.501065	10	0.000444
0.450000	50	4291.302623	0.776102	10	0.000443
0.500000	0	4748.813545	0.000000	14	0.000344
0.500000	10	4748.813545	0.039530	14	0.000511
0.500000	20	4748.813545	0.156732	14	0.000511
0.500000	30	4748.813545	0.349561	14	0.000509
0.500000	40	4748.813544	0.616020	14	0.000508
0.500000	50	4748.813543	0.954156	14	0.000508
0.550000	0	5202.682999	0.000000	9	0.000290
0.550000	10	5202.682999	0.047632	9	0.000427
0.550000	20	5202.682999	0.188858	9	0.000427
0.550000	30	5202.682998	0.421212	9	0.000428
0.550000	40	5202.682998	0.742287	9	0.000427
0.550000	50	5202.682996	1.149732	9	0.000427
0.600000	0	5652.936669	0.000000	9	0.000290
0.600000	10	5652.936669	0.056451	9	0.000435
0.600000	20	5652.936669	0.223823	9	0.000430
0.600000	30	5652.936669	0.499195	9	0.000429
0.600000	40	5652.936667	0.879714	9	0.000433
0.600000	50	5652.936665	1.362593	9	0.000428
0.650000	0	6099.670433	0.000000	12	0.000324
0.650000	10	6099.670433	0.065977	12	0.000479
0.650000	20	6099.670433	0.261595	12	0.000475
0.650000	30	6099.670432	0.583438	12	0.000477
0.650000	40	6099.670431	1.028173	12	0.000478
0.650000	50	6099.670428	1.592543	12	0.000475
0.700000	0	6542.931410	0.000000	12	0.000323
0.700000	10	6542.931410	0.076203	12	0.000476
0.700000	20	6542.931410	0.302138	12	0.000480
0.700000	30	6542.931409	0.673863	12	0.000491
0.700000	40	6542.931407	1.187525	12	0.000477
0.700000	50	6542.931403	1.839364	12	0.000491
0.750000	0	6982.742886	0.000000	10	0.000303
0.750000	10	6982.742886	0.087118	10	0.000443
0.750000	20	6982.742886	0.345415	10	0.000443
0.750000	30	6982.742885	0.770384	10	0.000443
0.750000	40	6982.742882	1.357622	10	0.000443
0.750000	50	6982.742877	2.102827	10	0.000444
0.800000	0	7419.173982	0.000000	14	0.000344
0.800000	10	7419.173982	0.098714	14	0.000508
0.800000	20	7419.173982	0.391393	14	0.000507
0.800000	30	7419.173981	0.872930	14	0.000508
0.800000	40	7419.173978	1.538334	14	0.000508
0.800000	50	7419.173971	2.382734	14	0.000507
0.850000	0	7852.292610	0.000000	12	0.000323
0.850000	10	7852.292610	0.110983	12	0.000475
0.850000	20	7852.292610	0.440040	12	0.000475
0.850000	30	7852.292608	0.981428	12	0.000475
0.850000	40	7852.292604	1.729537	12	0.000483
0.850000	50	7852.292595	2.678890	12	0.000475
0.900000	0	8282.119633	0.000000	15	0.000355
0.900000	10	8282.119633	0.123917	15	0.000524
0.900000	20	8282.119633	0.491320	15	0.000524
0.900000	30	8282.119631	1.095798	15	0.000524
0.900000	40	8282.119626	1.931087	15	0.000524
0.900000	50	8282.119615	2.991072	15	0.000524

0.950000	0	8708.744074	0.000000	13	0.000333
0.950000	10	8708.744074	0.137507	13	0.000492
0.950000	20	8708.744073	0.545204	13	0.000492
0.950000	30	8708.744071	1.215976	13	0.000491
0.950000	40	8708.744064	2.142874	13	0.000491
0.950000	50	8708.744051	3.319109	13	0.000492
1.000000	0	9132.162633	0.000000	15	0.000355
1.000000	10	9132.162633	0.151744	15	0.000524
1.000000	20	9132.162632	0.601653	15	0.000524
1.000000	30	9132.162629	1.341876	15	0.000524
1.000000	40	9132.162621	2.364743	15	0.000524
1.000000	50	9132.162605	3.662765	15	0.000524
1.050000	0	9552.439841	0.000000	13	0.000333
1.050000	10	9552.439840	0.166621	13	0.000492
1.050000	20	9552.439840	0.660638	13	0.000492
1.050000	30	9552.439836	1.473431	13	0.000491
1.050000	40	9552.439827	2.596578	13	0.000492
1.050000	50	9552.439807	4.021856	13	0.000491
1.100000	0	9969.639151	0.000000	18	0.000388
1.100000	10	9969.639151	0.182130	18	0.000572
1.100000	20	9969.639150	0.722129	18	0.000574
1.100000	30	9969.639146	1.610575	18	0.000573
1.100000	40	9969.639134	2.838263	18	0.000584
1.100000	50	9969.639111	4.396204	18	0.000582
1.150000	0	10383.800523	0.000000	13	0.000333
1.150000	10	10383.800523	0.198262	13	0.000492
1.150000	20	10383.800522	0.786094	13	0.000491
1.150000	30	10383.800517	1.753238	13	0.000491
1.150000	40	10383.800504	3.089673	13	0.000492
1.150000	50	10383.800476	4.785614	13	0.000491
1.200000	0	10794.963311	0.000000	12	0.000322
1.200000	10	10794.963311	0.215011	12	0.000476
1.200000	20	10794.963310	0.852502	12	0.000481
1.200000	30	10794.963304	1.901348	12	0.000479
1.200000	40	10794.963288	3.350683	12	0.000475
1.200000	50	10794.963256	5.189895	12	0.000476
1.250000	0	11203.166212	0.000000	13	0.000333
1.250000	10	11203.166212	0.232368	13	0.000491
1.250000	20	11203.166211	0.921320	13	0.000491
1.250000	30	11203.166204	2.054836	13	0.000492
1.250000	40	11203.166186	3.621170	13	0.000502
1.250000	50	11203.166148	5.608854	13	0.000497
1.300000	0	11608.469309	0.000000	13	0.000345
1.300000	10	11608.469309	0.250326	13	0.000503
1.300000	20	11608.469307	0.992523	13	0.000516
1.300000	30	11608.469299	2.213641	13	0.000545
1.300000	40	11608.469278	3.901026	13	0.000513
1.300000	50	11608.469235	6.042326	13	0.000506
1.350000	0	12010.887941	0.000000	13	0.000341
1.350000	10	12010.887940	0.268877	13	0.000514
1.350000	20	12010.887938	1.066075	13	0.000518
1.350000	30	12010.887929	2.377685	13	0.000501
1.350000	40	12010.887905	4.190117	13	0.000508
1.350000	50	12010.887855	6.490102	13	0.000511
1.400000	0	12410.502363	0.000000	13	0.000354
1.400000	10	12410.502363	0.288015	13	0.000509
1.400000	20	12410.502361	1.141955	13	0.000505

1.400000	30	12410.502350	2.546920	13	0.000500
1.400000	40	12410.502323	4.488355	13	0.000513
1.400000	50	12410.502265	6.952045	13	0.000502
1.450000	0	12807.326571	0.000000	14	0.000362
1.450000	10	12807.326571	0.307731	14	0.000519
1.450000	20	12807.326568	1.220127	14	0.000527
1.450000	30	12807.326556	2.721270	14	0.000520
1.450000	40	12807.326525	4.795607	14	0.000519
1.450000	50	12807.326460	7.427950	14	0.000527
1.500000	0	13201.395841	0.000000	18	0.000406
1.500000	10	13201.395840	0.328018	18	0.000590
1.500000	20	13201.395837	1.300564	18	0.000591
1.500000	30	13201.395824	2.900669	18	0.000588
1.500000	40	13201.395788	5.111755	18	0.000592
1.500000	50	13201.395714	7.917635	18	0.000605
1.550000	0	13592.766255	0.000000	19	0.000409
1.550000	10	13592.766255	0.348869	19	0.000609
1.550000	20	13592.766251	1.383238	19	0.000601
1.550000	30	13592.766236	3.085060	19	0.000602
1.550000	40	13592.766196	5.436702	19	0.000600
1.550000	50	13592.766113	8.420948	19	0.000601
1.600000	0	13981.471652	0.000000	17	0.000383
1.600000	10	13981.471651	0.370278	17	0.000577
1.600000	20	13981.471647	1.468123	17	0.000583
1.600000	30	13981.471630	3.274379	17	0.000581
1.600000	40	13981.471585	5.770334	17	0.000573
1.600000	50	13981.471492	8.937714	17	0.000579
1.650000	0	14367.566565	0.000000	16	0.000378
1.650000	10	14367.566565	0.392238	16	0.000560
1.650000	20	14367.566560	1.555193	16	0.000549
1.650000	30	14367.566541	3.468574	16	0.000560
1.650000	40	14367.566490	6.112557	16	0.000560
1.650000	50	14367.566386	9.467787	16	0.000577
1.700000	0	14751.062440	0.000000	19	0.000411
1.700000	10	14751.062440	0.414741	19	0.000601
1.700000	20	14751.062435	1.644416	19	0.000602
1.700000	30	14751.062413	3.667569	19	0.000608
1.700000	40	14751.062357	6.463241	19	0.000786
1.700000	50	14751.062240	10.010966	19	0.000626
1.750000	0	15131.991584	0.000000	14	0.000353
1.750000	10	15131.991584	0.437780	14	0.000618
1.750000	20	15131.991578	1.735764	14	0.000529
1.750000	30	15131.991555	3.871304	14	0.000509
1.750000	40	15131.991492	6.822276	14	0.000516
1.750000	50	15131.991362	10.567078	14	0.000508
1.800000	0	15510.427555	0.000000	17	0.000377
1.800000	10	15510.427555	0.461351	17	0.000556
1.800000	20	15510.427549	1.829218	17	0.000560
1.800000	30	15510.427522	4.079737	17	0.000556
1.800000	40	15510.427453	7.189592	17	0.000558
1.800000	50	15510.427309	11.136018	17	0.000565
1.850000	0	15886.380338	0.000000	18	0.000388
1.850000	10	15886.380337	0.485444	18	0.000572
1.850000	20	15886.380330	1.924747	18	0.000573
1.850000	30	15886.380301	4.292797	18	0.000572
1.850000	40	15886.380225	7.565062	18	0.000573
1.850000	50	15886.380066	11.717588	18	0.000578

```
1.900000    0     16259.880517      0.000000    19          0.000399
1.900000    10    16259.880516      0.510054    19          0.000592
1.900000    20    16259.880509      2.022323    19          0.000592
1.900000    30    16259.880476      4.510424    19          0.000589
1.900000    40    16259.880392      7.948579    19          0.000589
1.900000    50    16259.880217      12.311621   19          0.000590
1.950000    0     16630.978740      0.000000    18          0.000395
1.950000    10    16630.978739      0.535175    18          0.000585
1.950000    20    16630.978731      2.121925    18          0.000572
1.950000    30    16630.978696      4.732568    18          0.000573
1.950000    40    16630.978603      8.340058    18          0.000575
1.950000    50    16630.978411      12.917987   18          0.000573
2.000000    0     16999.704428      0.000000    20          0.000410
2.000000    10    16999.704427      0.560800    20          0.000605
2.000000    20    16999.704418      2.223527    20          0.000605
2.000000    30    16999.704380      4.959172    20          0.000608
2.000000    40    16999.704278      8.739395    20          0.000606
2.000000    50    16999.704068      13.536526   20          0.000605
```

APPENDIX M: THE SHOOTING METHOD FOR STATIONARY TARGETS TEST RESULTS

Range (m)	Cross Wind (kts)	Elevation Offset (°)	Azimuth Offset (°)	Flight Time (s)	Iterations (#)	Compute Time (s)
500	0	0.050715	-0.022274	0.300323	3	0.001370
500	10	0.050715	-0.022381	0.300323	3	0.001900
500	20	0.050715	-0.022701	0.300323	3	0.001900
500	30	0.050715	-0.023229	0.300323	3	0.001920
500	40	0.050715	-0.023963	0.300323	3	0.001910
500	50	0.050715	-0.024902	0.300323	3	0.001930
1000	0	0.101872	-0.001098	0.602615	3	0.001390
1000	10	0.101872	-0.001199	0.602615	3	0.001970
1000	20	0.101872	-0.001499	0.602615	3	0.001940
1000	30	0.101872	-0.001993	0.602615	3	0.001920
1000	40	0.101872	-0.002675	0.602615	3	0.001940
1000	50	0.101872	-0.003541	0.602615	3	0.001960
1500	0	0.153477	-0.000034	0.906876	3	0.001350
1500	10	0.153477	-0.000181	0.906876	3	0.001870
1500	20	0.153477	-0.000616	0.906876	3	0.001870
1500	30	0.153477	-0.001331	0.906876	3	0.001880
1500	40	0.153477	-0.002320	0.906876	3	0.001870
1500	50	0.153477	-0.003574	0.906876	3	0.001870
2000	0	0.205535	-0.004927	1.213150	2	0.001130
2000	10	0.205535	-0.005144	1.213150	2	0.001570
2000	20	0.205535	-0.005786	1.213150	2	0.001570
2000	30	0.205535	-0.006844	1.213150	2	0.001560
2000	40	0.205535	-0.008307	1.213150	2	0.001560
2000	50	0.205535	-0.010167	1.213150	2	0.001580
2500	0	0.258046	0.000056	1.521393	3	0.001420
2500	10	0.258046	-0.000190	1.521393	3	0.001980
2500	20	0.258046	-0.000919	1.521393	3	0.001980
2500	30	0.258046	-0.002118	1.521393	3	0.001980
2500	40	0.258046	-0.003775	1.521393	3	0.001980
2500	50	0.258046	-0.005878	1.521393	3	0.001970
3000	0	0.311023	0.000231	1.831680	3	0.001460
3000	10	0.311023	-0.000064	1.831680	3	0.002030
3000	20	0.311024	-0.000938	1.831680	3	0.002030
3000	30	0.311024	-0.002375	1.831680	3	0.002030
3000	40	0.311024	-0.004362	1.831696	3	0.002040
3000	50	0.311025	-0.006882	1.831696	3	0.002020
3500	0	0.364474	0.000512	2.144043	3	0.001340
3500	10	0.364474	0.000170	2.144058	3	0.001900
3500	20	0.364476	-0.000846	2.144058	3	0.001910
3500	30	0.364479	-0.002516	2.144089	3	0.001900
3500	40	0.364485	-0.004824	2.144119	3	0.001910
3500	50	0.364474	-0.008395	2.144058	4	0.002300
4000	0	0.418371	-0.000000	2.458328	5	0.002210
4000	10	0.418373	-0.000399	2.458344	5	0.003070
4000	20	0.418370	-0.001584	2.458328	5	0.003050
4000	30	0.418374	-0.003546	2.458344	5	0.003060
4000	40	0.418389	-0.006147	2.458435	5	0.003110
4000	50	0.418373	-0.009647	2.458328	5	0.002970
4500	0	0.472758	-0.000000	2.774750	5	0.002130

4500	10	0.472760	-0.000451	2.774750	5	0.002940
4500	20	0.472757	-0.001788	2.774734	5	0.002900
4500	30	0.472759	-0.003988	2.774750	5	0.002860
4500	40	0.472778	-0.007030	2.774857	4	0.002520
4500	50	0.472771	-0.010888	2.774811	4	0.002500
5000	0	0.527631	-0.000001	3.093262	4	0.001780
5000	10	0.527631	-0.000504	3.093262	4	0.002480
5000	20	0.527630	-0.001997	3.093262	4	0.002500
5000	30	0.527627	-0.004452	3.093246	4	0.002570
5000	40	0.527627	-0.007844	3.093246	4	0.002560
5000	50	0.527626	-0.012149	3.093246	4	0.002560
5500	0	0.582975	-0.000002	3.413803	4	0.001870
5500	10	0.582974	-0.000558	3.413803	4	0.002590
5500	20	0.582975	-0.002206	3.413803	4	0.002640
5500	30	0.582974	-0.004918	3.413788	4	0.002600
5500	40	0.582974	-0.008666	3.413788	4	0.002610
5500	50	0.582974	-0.013422	3.413788	4	0.002590
6000	0	0.638812	-0.000002	3.736450	4	0.001830
6000	10	0.638811	-0.000612	3.736450	4	0.002550
6000	20	0.638810	-0.002418	3.736435	4	0.002580
6000	30	0.638811	-0.005389	3.736450	4	0.002550
6000	40	0.638812	-0.009495	3.736450	4	0.002540
6000	50	0.638810	-0.014706	3.736435	4	0.002590
6500	0	0.695140	-0.000004	4.061203	4	0.001870
6500	10	0.695142	-0.000666	4.061218	4	0.002560
6500	20	0.695141	-0.002631	4.061203	4	0.002580
6500	30	0.695140	-0.005864	4.061203	4	0.002580
6500	40	0.695141	-0.010332	4.061203	4	0.002560
6500	50	0.695140	-0.016001	4.061203	4	0.002570
7000	0	0.751969	-0.000005	4.388077	4	0.001830
7000	10	0.751969	-0.000722	4.388077	4	0.002560
7000	20	0.751971	-0.002847	4.388092	4	0.002610
7000	30	0.751970	-0.006344	4.388092	4	0.002610
7000	40	0.751969	-0.011176	4.388077	4	0.002610
7000	50	0.751970	-0.017308	4.388092	4	0.002580
7500	0	0.809301	-0.000007	4.717102	4	0.001910
7500	10	0.809301	-0.000779	4.717102	4	0.002660
7500	20	0.809300	-0.003066	4.717102	4	0.002660
7500	30	0.809302	-0.006828	4.717117	4	0.002660
7500	40	0.809301	-0.012028	4.717102	4	0.002640
7500	50	0.809300	-0.018627	4.717102	4	0.002640
8000	0	0.867140	-0.000010	5.048279	4	0.001900
8000	10	0.867140	-0.000836	5.048264	4	0.002610
8000	20	0.867140	-0.003286	5.048264	4	0.002620
8000	30	0.867139	-0.007317	5.048264	4	0.002620
8000	40	0.867138	-0.012888	5.048264	4	0.002620
8000	50	0.867140	-0.019957	5.048264	4	0.002590
8500	0	0.925489	-0.000013	5.381592	4	0.001860
8500	10	0.925489	-0.000895	5.381592	4	0.002580
8500	20	0.925489	-0.003509	5.381577	4	0.002630
8500	30	0.925488	-0.007811	5.381577	4	0.002660
8500	40	0.925487	-0.013755	5.381577	4	0.002660
8500	50	0.925489	-0.021299	5.381592	4	0.002590
9000	0	0.984356	-0.000016	5.717072	4	0.001960
9000	10	0.984355	-0.000954	5.717072	4	0.002730
9000	20	0.984355	-0.003734	5.717072	4	0.002740
9000	30	0.984354	-0.008309	5.717072	4	0.002730

```
9000  40    0.984354    -0.014630   5.717056    4           0.002720
9000  50    0.984356    -0.022653   5.717072    4           0.002740
9500  0     1.043742    -0.000020   6.054733    4           0.001870
9500  10    1.043741    -0.001015   6.054733    4           0.002620
9500  20    1.043741    -0.003962   6.054733    4           0.002590
9500  30    1.043740    -0.008812   6.054733    4           0.002600
9500  40    1.043740    -0.015514   6.054718    4           0.002580
9500  50    1.043742    -0.024019   6.054733    4           0.002600
10000 0     1.103651    -0.000025   6.394577    4           0.001860
10000 10    1.103654    -0.001076   6.394592    4           0.002570
10000 20    1.103653    -0.004192   6.394592    4           0.002570
10000 30    1.103653    -0.009320   6.394592    4           0.002570
10000 40    1.103652    -0.016405   6.394577    4           0.002580
10000 50    1.103652    -0.025398   6.394577    4           0.002590
```

APPENDIX N: THE SHOOTING METHOD FOR MOVING TARGETS TEST RESULTS

Range (m)	Cross Wind (kts)	Target Speed (mph)	Target Heading (°)	Elevation Offset (°)	Azimuth Offset (°)	Iterations (#)	Compute Time (s)
500	0	5	90	0.050722	0.076447	2	0.004900
500	0	5	135	0.050666	0.054057	2	0.005100
500	0	5	180	0.050647	-0.022408	2	0.004300
500	0	10	90	0.050715	0.139418	2	0.004400
500	0	10	135	0.050639	0.109088	2	0.003700
500	0	10	180	0.050579	-0.022544	2	0.004400
500	0	15	90	0.050715	0.206525	2	0.004300
500	0	15	135	0.050572	0.147155	2	0.004400
500	0	15	180	0.050511	-0.022680	2	0.004400
500	0	20	90	0.050716	0.305925	2	0.004900
500	0	20	135	0.050523	0.194622	2	0.004300
500	0	20	180	0.050444	-0.022726	2	0.004400
500	0	25	90	0.050716	0.382422	2	0.005000
500	0	25	135	0.050477	0.242979	2	0.004300
500	0	25	180	0.050378	-0.022864	2	0.004500
500	0	30	90	0.050717	0.458906	2	0.004900
500	0	30	135	0.050429	0.324476	2	0.005100
500	0	30	180	0.050310	-0.023048	2	0.004300
500	10	5	90	0.050722	0.076265	2	0.007100
500	10	5	135	0.050666	0.053875	2	0.007100
500	10	5	180	0.050647	-0.022814	2	0.006200
500	10	10	90	0.050715	0.139205	2	0.006300
500	10	10	135	0.050639	0.109018	2	0.005300
500	10	10	180	0.050579	-0.022951	2	0.006300
500	10	15	90	0.050715	0.206261	2	0.006200
500	10	15	135	0.050572	0.146931	2	0.006300
500	10	15	180	0.050511	-0.023088	2	0.006200
500	10	20	90	0.050716	0.305743	2	0.007100
500	10	20	135	0.050523	0.194363	2	0.006200
500	10	20	180	0.050444	-0.023135	2	0.006200
500	10	25	90	0.050716	0.382241	2	0.007100
500	10	25	135	0.050477	0.242708	2	0.006100
500	10	25	180	0.050378	-0.023274	2	0.006400
500	10	30	90	0.050717	0.458724	2	0.007100
500	10	30	135	0.050429	0.324295	2	0.007300
500	10	30	180	0.050310	-0.023459	2	0.006200
500	20	5	90	0.050725	0.075734	2	0.007000
500	20	5	135	0.050666	0.053342	2	0.007200
500	20	5	180	0.050647	-0.024013	2	0.006200
500	20	10	90	0.050715	0.138583	2	0.006200
500	20	10	135	0.050639	0.108816	2	0.005400
500	20	10	180	0.050579	-0.024153	2	0.006300
500	20	15	90	0.050715	0.205489	2	0.006300
500	20	15	135	0.050572	0.146278	2	0.006300
500	20	15	180	0.050511	-0.024294	2	0.006300
500	20	20	90	0.050716	0.305210	2	0.007000
500	20	20	135	0.050523	0.193605	2	0.006100
500	20	20	180	0.050444	-0.024341	2	0.006300
500	20	25	90	0.050716	0.381708	2	0.007000
500	20	25	135	0.050477	0.241911	2	0.006100
500	20	25	180	0.050378	-0.024484	2	0.006300
500	20	30	90	0.050717	0.458191	2	0.007000

500	20	30	135	0.050429	0.323765	2	0.007300
500	20	30	180	0.050310	-0.024674	2	0.006100
500	30	5	90	0.050729	0.074866	2	0.007100
500	30	5	135	0.050666	0.052476	2	0.007200
500	30	5	180	0.050647	-0.025991	2	0.006200
500	30	10	90	0.050715	0.137581	2	0.006300
500	30	10	135	0.050639	0.108494	2	0.005300
500	30	10	180	0.050579	-0.026137	2	0.006200
500	30	15	90	0.050715	0.204232	2	0.006200
500	30	15	135	0.050572	0.145223	2	0.006300
500	30	15	180	0.050511	-0.026285	2	0.006200
500	30	20	90	0.050716	0.304344	2	0.007000
500	30	20	135	0.050523	0.192373	2	0.006200
500	30	20	180	0.050444	-0.026332	2	0.006200
500	30	25	90	0.050716	0.380841	2	0.007100
500	30	25	135	0.050477	0.240614	2	0.006100
500	30	25	180	0.050378	-0.026481	2	0.006300
500	30	30	90	0.050717	0.457325	2	0.007100
500	30	30	135	0.050429	0.322904	2	0.007200
500	30	30	180	0.050310	-0.026680	2	0.006100
500	40	5	90	0.050735	0.073537	2	0.007100
500	40	5	135	0.050666	0.051294	2	0.007100
500	40	5	180	0.050647	-0.028752	2	0.006200
500	40	10	90	0.050715	0.136230	2	0.006200
500	40	10	135	0.050639	0.108071	2	0.005300
500	40	10	180	0.050579	-0.028907	2	0.006300
500	40	15	90	0.050715	0.202518	2	0.006200
500	40	15	135	0.050572	0.143796	2	0.006200
500	40	15	180	0.050511	-0.029062	2	0.006200
500	40	20	90	0.050716	0.303162	2	0.007000
500	40	20	135	0.050523	0.190693	2	0.006200
500	40	20	180	0.050444	-0.029111	2	0.006200
500	40	25	90	0.050716	0.379659	2	0.007100
500	40	25	135	0.050476	0.267634	2	0.006900
500	40	25	180	0.050378	-0.029268	2	0.006200
500	40	30	90	0.050717	0.456142	2	0.007100
500	40	30	135	0.050429	0.321728	2	0.007200
500	40	30	180	0.050310	-0.029480	2	0.006200
500	50	5	90	0.050733	0.072188	2	0.007000
500	50	5	135	0.050666	0.049811	2	0.007200
500	50	5	180	0.050647	-0.032315	2	0.006200
500	50	10	90	0.050715	0.134565	2	0.006200
500	50	10	135	0.050639	0.107562	2	0.005400
500	50	10	180	0.050579	-0.032481	2	0.006200
500	50	15	90	0.050715	0.200369	2	0.006200
500	50	15	135	0.050572	0.142032	2	0.006300
500	50	15	180	0.050511	-0.032648	2	0.006300
500	50	20	90	0.050716	0.301679	2	0.007000
500	50	20	135	0.050523	0.188592	2	0.006200
500	50	20	180	0.050444	-0.032698	2	0.006300
500	50	25	90	0.050716	0.378176	2	0.007100
500	50	25	135	0.050476	0.266159	2	0.006900
500	50	25	180	0.050378	-0.032867	2	0.006400
500	50	30	90	0.050717	0.454659	2	0.007100
500	50	30	135	0.050429	0.320254	2	0.007200
500	50	30	180	0.050310	-0.004223	2	0.006600
1000	0	5	90	0.101873	0.074939	2	0.004400

1000	0	5	135	0.101776	0.052725	2	0.004300
1000	0	5	180	0.101735	-0.001107	2	0.004300
1000	0	10	90	0.101878	0.140444	2	0.003800
1000	0	10	135	0.101683	0.119410	2	0.004000
1000	0	10	180	0.101598	-0.001116	2	0.004600
1000	0	15	90	0.101873	0.226168	2	0.004400
1000	0	15	135	0.101582	0.160470	2	0.004500
1000	0	15	180	0.101460	-0.001121	2	0.004400
1000	0	20	90	0.101873	0.298154	2	0.004400
1000	0	20	135	0.101485	0.213360	2	0.004500
1000	0	20	180	0.101326	-0.001130	2	0.004400
1000	0	25	90	0.101874	0.383655	2	0.005000
1000	0	25	135	0.101388	0.265354	2	0.004400
1000	0	25	180	0.101190	-0.001135	2	0.004500
1000	0	30	90	0.101876	0.460378	2	0.005000
1000	0	30	135	0.101292	0.313995	2	0.004400
1000	0	30	180	0.101053	-0.001144	2	0.004300
1000	10	5	90	0.101873	0.074555	2	0.006300
1000	10	5	135	0.101776	0.052343	2	0.006200
1000	10	5	180	0.101735	-0.001486	2	0.006200
1000	10	10	90	0.101878	0.140230	2	0.005500
1000	10	10	135	0.101683	0.119243	2	0.005700
1000	10	10	180	0.101598	-0.001495	2	0.006600
1000	10	15	90	0.101873	0.225790	2	0.006300
1000	10	15	135	0.101582	0.160096	2	0.006400
1000	10	15	180	0.101460	-0.001499	2	0.006300
1000	10	20	90	0.101873	0.297833	2	0.006300
1000	10	20	135	0.101485	0.212982	2	0.006400
1000	10	20	180	0.101326	-0.001508	2	0.006300
1000	10	25	90	0.101874	0.383289	2	0.007100
1000	10	25	135	0.101388	0.264991	2	0.006200
1000	10	25	180	0.101190	-0.001512	2	0.006500
1000	10	30	90	0.101876	0.460013	2	0.007100
1000	10	30	135	0.101292	0.313736	2	0.006300
1000	10	30	180	0.101053	-0.001521	2	0.006100
1000	20	5	90	0.101873	0.073430	2	0.006300
1000	20	5	135	0.101776	0.051222	2	0.006100
1000	20	5	180	0.101735	-0.002600	2	0.006200
1000	20	10	90	0.101878	0.139610	2	0.005500
1000	20	10	135	0.101683	0.118760	2	0.005600
1000	20	10	180	0.101598	-0.002607	2	0.006500
1000	20	15	90	0.101873	0.224683	2	0.006300
1000	20	15	135	0.101582	0.158997	2	0.006500
1000	20	15	180	0.101460	-0.002610	2	0.006300
1000	20	20	90	0.101873	0.296881	2	0.006300
1000	20	20	135	0.101485	0.211874	2	0.006400
1000	20	20	180	0.101326	-0.002618	2	0.006300
1000	20	25	90	0.101874	0.382216	2	0.007200
1000	20	25	135	0.101388	0.263922	2	0.006300
1000	20	25	180	0.101190	-0.002621	2	0.006400
1000	20	30	90	0.101876	0.458942	2	0.007200
1000	20	30	135	0.101292	0.312948	2	0.006300
1000	20	30	180	0.101053	-0.002629	2	0.006200
1000	30	5	90	0.101873	0.071599	2	0.006300
1000	30	5	135	0.101776	0.049398	2	0.006100
1000	30	5	180	0.101735	-0.004411	2	0.006200
1000	30	10	90	0.101878	0.138622	2	0.005400

1000	30	10	135		0.101683	0.117991	2		0.005700
1000	30	10	180		0.101598	-0.004417	2		0.006600
1000	30	15	90		0.101873	0.222880	2		0.006300
1000	30	15	135		0.101582	0.157214	2		0.006400
1000	30	15	180		0.101460	-0.004418	2		0.006300
1000	30	20	90		0.101873	0.295298	2		0.006300
1000	30	20	135		0.101485	0.210072	2		0.006300
1000	30	20	180		0.101326	-0.004424	2		0.006400
1000	30	25	90		0.101875	0.380474	2		0.007100
1000	30	25	135		0.101388	0.262177	2		0.006200
1000	30	25	180		0.101190	-0.004425	2		0.006400
1000	30	30	90		0.101876	0.457202	2		0.007200
1000	30	30	135		0.101292	0.311591	2		0.006300
1000	30	30	180		0.101053	-0.004431	2		0.006000
1000	40	5	90		0.101873	0.069096	2		0.006300
1000	40	5	135		0.101776	0.046905	2		0.006200
1000	40	5	180		0.101735	-0.006888	2		0.006200
1000	40	10	90		0.101878	0.137311	2		0.005400
1000	40	10	135		0.101677	0.100079	2		0.006200
1000	40	10	180		0.101598	-0.006891	2		0.006500
1000	40	15	90		0.101873	0.220414	2		0.006300
1000	40	15	135		0.101582	0.154784	2		0.006400
1000	40	15	180		0.101460	-0.006889	2		0.006300
1000	40	20	90		0.101873	0.293083	2		0.006300
1000	40	20	135		0.101485	0.207609	2		0.006400
1000	40	20	180		0.101326	-0.006893	2		0.006400
1000	40	25	90		0.101875	0.378097	2		0.007200
1000	40	25	135		0.101388	0.259779	2		0.006300
1000	40	25	180		0.101190	-0.006891	2		0.006400
1000	40	30	90		0.101876	0.454825	2		0.007200
1000	40	30	135		0.101292	0.309622	2		0.006300
1000	40	30	180		0.101053	-0.006895	2		0.006100
1000	50	5	90		0.101873	0.065952	2		0.006400
1000	50	5	135		0.101776	0.043774	2		0.006200
1000	50	5	180		0.101735	-0.009999	2		0.006200
1000	50	10	90		0.101878	0.135724	2		0.005500
1000	50	10	135		0.101677	0.096945	2		0.006600
1000	50	10	180		0.101598	-0.009999	2		0.006500
1000	50	15	90		0.101873	0.217316	2		0.006300
1000	50	15	135		0.101582	0.151749	2		0.006400
1000	50	15	180		0.101460	-0.009994	2		0.006300
1000	50	20	90		0.101873	0.290236	2		0.006300
1000	50	20	135		0.101485	0.204516	2		0.006400
1000	50	20	180		0.101326	-0.009994	2		0.006300
1000	50	25	90		0.101875	0.375117	2		0.007100
1000	50	25	135		0.101388	0.256753	2		0.006200
1000	50	25	180		0.101190	-0.009989	2		0.006500
1000	50	30	90		0.101876	0.451846	2		0.007100
1000	50	30	135		0.101292	0.307011	2		0.006300
1000	50	30	180		0.101053	-0.009989	2		0.006100
1500	0	5	90		0.153477	0.076913	2		0.004300
1500	0	5	135		0.153329	0.054373	2		0.004600
1500	0	5	180		0.153269	-0.000035	2		0.004500
1500	0	10	90		0.153477	0.155680	2		0.003700
1500	0	10	135		0.153182	0.108768	2		0.004600
1500	0	10	180		0.153061	-0.000036	2		0.004500
1500	0	15	90		0.153478	0.230328	2		0.004300

1500	0	15	135	0.153036	0.161630	2	0.004000
1500	0	15	180	0.152852	-0.000038	2	0.004500
1500	0	20	90	0.153478	0.308352	2	0.004400
1500	0	20	135	0.152888	0.217275	2	0.004500
1500	0	20	180	0.152644	-0.000038	2	0.004500
1500	0	25	90	0.153483	0.385102	2	0.004300
1500	0	25	135	0.152743	0.274100	2	0.004500
1500	0	25	180	0.152439	-0.000038	2	0.004500
1500	0	30	90	0.153501	0.462080	2	0.004200
1500	0	30	135	0.152596	0.326934	2	0.004600
1500	0	30	180	0.152235	-0.000039	2	0.004400
1500	10	5	90	0.153477	0.076362	2	0.006100
1500	10	5	135	0.153329	0.053823	2	0.006500
1500	10	5	180	0.153269	-0.000585	2	0.006500
1500	10	10	90	0.153477	0.155321	2	0.005200
1500	10	10	135	0.153182	0.108218	2	0.006800
1500	10	10	180	0.153061	-0.000585	2	0.006400
1500	10	15	90	0.153478	0.229788	2	0.006100
1500	10	15	135	0.153036	0.161270	2	0.005600
1500	10	15	180	0.152852	-0.000586	2	0.006500
1500	10	20	90	0.153479	0.307817	2	0.006100
1500	10	20	135	0.152890	0.216744	2	0.006500
1500	10	20	180	0.152644	-0.000586	2	0.006600
1500	10	25	90	0.153481	0.384549	2	0.006100
1500	10	25	135	0.152743	0.273764	2	0.006400
1500	10	25	180	0.152439	-0.000586	2	0.006400
1500	10	30	90	0.153502	0.461540	2	0.006100
1500	10	30	135	0.152596	0.326397	2	0.006500
1500	10	30	180	0.152235	-0.000585	2	0.006300
1500	20	5	90	0.153477	0.074746	2	0.006100
1500	20	5	135	0.153329	0.052209	2	0.006500
1500	20	5	180	0.153269	-0.002197	2	0.006500
1500	20	10	90	0.153477	0.154271	2	0.005300
1500	20	10	135	0.153182	0.106605	2	0.006500
1500	20	10	180	0.153061	-0.002196	2	0.006400
1500	20	15	90	0.153478	0.228203	2	0.006200
1500	20	15	135	0.153036	0.160219	2	0.005700
1500	20	15	180	0.152852	-0.002195	2	0.006400
1500	20	20	90	0.153479	0.306249	2	0.006100
1500	20	20	135	0.152890	0.215145	2	0.006400
1500	20	20	180	0.152644	-0.002192	2	0.006600
1500	20	25	90	0.153477	0.382934	2	0.006100
1500	20	25	135	0.152743	0.273097	2	0.006300
1500	20	25	180	0.152439	-0.002190	2	0.006400
1500	20	30	90	0.153492	0.459657	2	0.007100
1500	20	30	135	0.152596	0.324826	2	0.006400
1500	20	30	180	0.152235	-0.002187	2	0.006300
1500	30	5	90	0.153477	0.072120	2	0.006200
1500	30	5	135	0.153329	0.049586	2	0.006500
1500	30	5	180	0.153269	-0.004819	2	0.006500
1500	30	10	90	0.153477	0.152587	2	0.005400
1500	30	10	135	0.153182	0.103983	2	0.006700
1500	30	10	180	0.153061	-0.004814	2	0.006500
1500	30	15	90	0.153478	0.225616	2	0.006200
1500	30	15	135	0.153036	0.158530	2	0.005800
1500	30	15	180	0.152852	-0.004810	2	0.006400
1500	30	20	90	0.153479	0.303716	2	0.006300

1500	30	20	135	0.152890	0.212544	2	0.006400
1500	30	20	180	0.152644	-0.004804	2	0.006600
1500	30	25	90	0.153476	0.380332	2	0.006200
1500	30	25	135	0.152743	0.275969	2	0.006400
1500	30	25	180	0.152439	-0.004797	2	0.006500
1500	30	30	90	0.153501	0.456987	2	0.006900
1500	30	30	135	0.152597	0.322279	2	0.006400
1500	30	30	180	0.152235	-0.004791	2	0.006300
1500	40	5	90	0.153477	0.068535	2	0.006100
1500	40	5	135	0.153329	0.046004	2	0.006500
1500	40	5	180	0.153269	-0.008398	2	0.006700
1500	40	10	90	0.153477	0.150327	2	0.005200
1500	40	10	135	0.153182	0.100404	2	0.006600
1500	40	10	180	0.153061	-0.008388	2	0.006700
1500	40	15	90	0.153478	0.222071	2	0.006100
1500	40	15	135	0.153036	0.156263	2	0.005600
1500	40	15	180	0.152852	-0.008379	2	0.006400
1500	40	20	90	0.153479	0.300299	2	0.006100
1500	40	20	135	0.152890	0.208988	2	0.006500
1500	40	20	180	0.152644	-0.008368	2	0.006700
1500	40	25	90	0.153478	0.376791	2	0.006100
1500	40	25	135	0.152743	0.255921	2	0.006900
1500	40	25	180	0.152439	-0.008357	2	0.006400
1500	40	30	90	0.153480	0.447334	2	0.008000
1500	40	30	135	0.152597	0.318828	2	0.006400
1500	40	30	180	0.152235	-0.008346	2	0.006400
1500	50	5	90	0.153477	0.064040	2	0.006100
1500	50	5	135	0.153329	0.041513	2	0.006500
1500	50	5	180	0.153269	-0.012885	2	0.006600
1500	50	10	90	0.153477	0.147559	2	0.005300
1500	50	10	135	0.153182	0.095916	2	0.006500
1500	50	10	180	0.153061	-0.012869	2	0.006400
1500	50	15	90	0.153478	0.217611	2	0.006100
1500	50	15	135	0.153036	0.153482	2	0.005700
1500	50	15	180	0.152852	-0.012855	2	0.006500
1500	50	20	90	0.153479	0.296127	2	0.006200
1500	50	20	135	0.152890	0.204526	2	0.006400
1500	50	20	180	0.152644	-0.012838	2	0.006500
1500	50	25	90	0.153478	0.372349	2	0.006100
1500	50	25	135	0.152743	0.256716	2	0.006400
1500	50	25	180	0.152439	-0.012821	2	0.006400
1500	50	30	90	0.153480	0.449303	2	0.008500
1500	50	30	135	0.152597	0.314538	2	0.006400
1500	50	30	180	0.152235	-0.012804	2	0.006300
2000	0	5	90	0.205535	0.071945	2	0.003600
2000	0	5	135	0.205335	0.049668	2	0.003700
2000	0	5	180	0.205253	-0.004671	2	0.003600
2000	0	10	90	0.205534	0.154463	2	0.004200
2000	0	10	135	0.205135	0.104032	2	0.003700
2000	0	10	180	0.204973	-0.004416	2	0.003600
2000	0	15	90	0.205536	0.229306	2	0.003600
2000	0	15	135	0.204939	0.163826	2	0.004400
2000	0	15	180	0.204691	-0.004163	2	0.003600
2000	0	20	90	0.205537	0.305287	2	0.003500
2000	0	20	135	0.204730	0.212459	2	0.003000
2000	0	20	180	0.204412	-0.003911	2	0.003700
2000	0	25	90	0.205538	0.382238	2	0.003600

2000	0	25	135	0.204545	0.270116	2	0.003500
2000	0	25	180	0.204133	-0.003660	2	0.003600
2000	0	30	90	0.205539	0.459347	2	0.003600
2000	0	30	135	0.204346	0.324534	2	0.003600
2000	0	30	180	0.203857	-0.003410	2	0.003500
2000	10	5	90	0.205535	0.071127	2	0.005100
2000	10	5	135	0.205335	0.048856	2	0.005200
2000	10	5	180	0.205253	-0.005479	2	0.005100
2000	10	10	90	0.205534	0.153726	2	0.005900
2000	10	10	135	0.205135	0.103219	2	0.005400
2000	10	10	180	0.204973	-0.005219	2	0.005200
2000	10	15	90	0.205536	0.228573	2	0.005200
2000	10	15	135	0.204939	0.163091	2	0.006100
2000	10	15	180	0.204691	-0.004961	2	0.005000
2000	10	20	90	0.205537	0.304504	2	0.005100
2000	10	20	135	0.204730	0.212036	2	0.004300
2000	10	20	180	0.204412	-0.004704	2	0.005100
2000	10	25	90	0.205538	0.381446	2	0.005200
2000	10	25	135	0.204545	0.269349	2	0.005000
2000	10	25	180	0.204133	-0.004448	2	0.005100
2000	10	30	90	0.205539	0.458551	2	0.005100
2000	10	30	135	0.204346	0.323759	2	0.005100
2000	10	30	180	0.203857	-0.004193	2	0.004900
2000	20	5	90	0.205535	0.068723	2	0.005100
2000	20	5	135	0.205335	0.046469	2	0.005200
2000	20	5	180	0.205253	-0.007853	2	0.005200
2000	20	10	90	0.205534	0.151566	2	0.005900
2000	20	10	135	0.205135	0.100832	2	0.005300
2000	20	10	180	0.204973	-0.007577	2	0.005200
2000	20	15	90	0.205536	0.226435	2	0.005100
2000	20	15	135	0.204939	0.160938	2	0.006100
2000	20	15	180	0.204691	-0.007305	2	0.005100
2000	20	20	90	0.205537	0.302206	2	0.005100
2000	20	20	135	0.204730	0.210812	2	0.004400
2000	20	20	180	0.204412	-0.007033	2	0.005100
2000	20	25	90	0.205538	0.379121	2	0.005100
2000	20	25	135	0.204545	0.267102	2	0.005100
2000	20	25	180	0.204133	-0.006762	2	0.005100
2000	20	30	90	0.205539	0.456215	2	0.005200
2000	20	30	135	0.204346	0.321486	2	0.005000
2000	20	30	180	0.203857	-0.006491	2	0.004900
2000	30	5	90	0.205535	0.064805	2	0.005200
2000	30	5	135	0.205335	0.042578	2	0.005200
2000	30	5	180	0.205253	-0.011722	2	0.005100
2000	30	10	90	0.205534	0.148055	2	0.006000
2000	30	10	135	0.205135	0.096940	2	0.005300
2000	30	10	180	0.204973	-0.011421	2	0.005200
2000	30	15	90	0.205536	0.223004	2	0.005100
2000	30	15	135	0.204939	0.157436	2	0.006100
2000	30	15	180	0.204691	-0.011123	2	0.005000
2000	30	20	90	0.205537	0.298470	2	0.005200
2000	30	20	135	0.204730	0.208876	2	0.004200
2000	30	20	180	0.204412	-0.010826	2	0.005700
2000	30	25	90	0.205538	0.375336	2	0.005300
2000	30	25	135	0.204545	0.263452	2	0.005400
2000	30	25	180	0.204133	-0.010530	2	0.005200
2000	30	30	90	0.205538	0.452412	2	0.005100

2000	30	30	135	0.204346	0.317790	2	0.005200
2000	30	30	180	0.203857	-0.010235	2	0.004900
2000	40	5	90	0.205535	0.059431	2	0.005100
2000	40	5	135	0.205335	0.037244	2	0.005200
2000	40	5	180	0.205253	-0.017026	2	0.005100
2000	40	10	90	0.205534	0.143261	2	0.005900
2000	40	10	135	0.205135	0.091604	2	0.005400
2000	40	10	180	0.204973	-0.016688	2	0.005300
2000	40	15	90	0.205536	0.218434	2	0.005100
2000	40	15	135	0.204938	0.152657	2	0.006100
2000	40	15	180	0.204691	-0.016355	2	0.005000
2000	40	20	90	0.205537	0.293371	2	0.005200
2000	40	20	135	0.204730	0.206335	2	0.004200
2000	40	20	180	0.204412	-0.016023	2	0.005400
2000	40	25	90	0.205538	0.370162	2	0.005800
2000	40	25	135	0.204545	0.258482	2	0.005200
2000	40	25	180	0.204133	-0.015692	2	0.005200
2000	40	30	90	0.205538	0.447207	2	0.005100
2000	40	30	135	0.204346	0.312739	2	0.005100
2000	40	30	180	0.203857	-0.015361	2	0.004900
2000	50	5	90	0.205533	0.060020	2	0.006000
2000	50	5	135	0.205335	0.030519	2	0.005100
2000	50	5	180	0.205253	-0.023711	2	0.005100
2000	50	10	90	0.205534	0.137252	2	0.006000
2000	50	10	135	0.205135	0.084876	2	0.005300
2000	50	10	180	0.204973	-0.023326	2	0.005200
2000	50	15	90	0.205522	0.210384	2	0.004200
2000	50	15	135	0.204938	0.146665	2	0.006100
2000	50	15	180	0.204691	-0.022946	2	0.005100
2000	50	20	90	0.205537	0.286985	2	0.005100
2000	50	20	135	0.204730	0.203304	2	0.004200
2000	50	20	180	0.204412	-0.022567	2	0.005200
2000	50	25	90	0.205537	0.363656	2	0.005100
2000	50	25	135	0.204545	0.252276	2	0.005000
2000	50	25	180	0.204133	-0.022190	2	0.005200
2000	50	30	90	0.205538	0.440663	2	0.005100
2000	50	30	135	0.204346	0.306402	2	0.005100
2000	50	30	180	0.203857	-0.021814	2	0.004900
2500	0	5	90	0.258049	0.077658	2	0.004500
2500	0	5	135	0.257796	0.054908	2	0.004400
2500	0	5	180	0.257692	0.000055	2	0.004400
2500	0	10	90	0.258048	0.154967	2	0.005200
2500	0	10	135	0.257548	0.109903	2	0.004500
2500	0	10	180	0.257336	0.000054	2	0.004400
2500	0	15	90	0.258051	0.232131	2	0.004500
2500	0	15	135	0.257294	0.161898	2	0.004400
2500	0	15	180	0.256983	0.000053	2	0.004500
2500	0	20	90	0.258049	0.308776	2	0.004000
2500	0	20	135	0.257043	0.218772	2	0.004600
2500	0	20	180	0.256627	0.000053	2	0.004400
2500	0	25	90	0.258054	0.387302	2	0.004600
2500	0	25	135	0.256794	0.276146	2	0.003800
2500	0	25	180	0.256277	0.000052	2	0.004400
2500	0	30	90	0.258056	0.464800	2	0.004500
2500	0	30	135	0.256546	0.325716	2	0.003900
2500	0	30	180	0.255925	0.000051	2	0.004400
2500	10	5	90	0.258049	0.076736	2	0.006500

2500	10	5	135	0.257796	0.053986	2	0.006200
2500	10	5	180	0.257692	-0.000867	2	0.006300
2500	10	10	90	0.258048	0.154043	2	0.007400
2500	10	10	135	0.257549	0.108983	2	0.006300
2500	10	10	180	0.257336	-0.000867	2	0.006300
2500	10	15	90	0.258051	0.231198	2	0.006500
2500	10	15	135	0.257293	0.160731	2	0.006300
2500	10	15	180	0.256983	-0.000866	2	0.006500
2500	10	20	90	0.258049	0.307885	2	0.005600
2500	10	20	135	0.257043	0.217841	2	0.006500
2500	10	20	180	0.256627	-0.000865	2	0.006400
2500	10	25	90	0.258054	0.386375	2	0.006500
2500	10	25	135	0.256794	0.275338	2	0.005500
2500	10	25	180	0.256277	-0.000865	2	0.006200
2500	10	30	90	0.258056	0.463873	2	0.006500
2500	10	30	135	0.256546	0.324794	2	0.005500
2500	10	30	180	0.255925	-0.000865	2	0.006300
2500	20	5	90	0.258050	0.074033	2	0.006500
2500	20	5	135	0.257796	0.051284	2	0.006200
2500	20	5	180	0.257692	-0.003571	2	0.006300
2500	20	10	90	0.258048	0.151331	2	0.007400
2500	20	10	135	0.257549	0.106286	2	0.006300
2500	20	10	180	0.257336	-0.003566	2	0.006300
2500	20	15	90	0.258050	0.228462	2	0.006500
2500	20	15	135	0.257293	0.157015	2	0.006300
2500	20	15	180	0.256983	-0.003562	2	0.006400
2500	20	20	90	0.258049	0.305290	2	0.005600
2500	20	20	135	0.257043	0.215110	2	0.006600
2500	20	20	180	0.256627	-0.003558	2	0.006400
2500	20	25	90	0.258054	0.383656	2	0.006500
2500	20	25	135	0.256794	0.272991	2	0.005400
2500	20	25	180	0.256277	-0.003554	2	0.006400
2500	20	30	90	0.258055	0.461155	2	0.006500
2500	20	30	135	0.256546	0.322103	2	0.005500
2500	20	30	180	0.255925	-0.003550	2	0.006200
2500	30	5	90	0.258053	0.069639	2	0.006500
2500	30	5	135	0.257796	0.046893	2	0.006200
2500	30	5	180	0.257692	-0.007965	2	0.006300
2500	30	10	90	0.258049	0.150528	2	0.006500
2500	30	10	135	0.257549	0.101903	2	0.006300
2500	30	10	180	0.257336	-0.007954	2	0.006300
2500	30	15	90	0.258050	0.224010	2	0.006500
2500	30	15	135	0.257294	0.156338	2	0.007200
2500	30	15	180	0.256983	-0.007944	2	0.006400
2500	30	20	90	0.258049	0.301124	2	0.005600
2500	30	20	135	0.257043	0.210666	2	0.006500
2500	30	20	180	0.256627	-0.007934	2	0.006400
2500	30	25	90	0.258054	0.379236	2	0.006500
2500	30	25	135	0.256794	0.269245	2	0.005500
2500	30	25	180	0.256278	-0.007924	2	0.006300
2500	30	30	90	0.258055	0.456737	2	0.006400
2500	30	30	135	0.256546	0.317771	2	0.005500
2500	30	30	180	0.255925	-0.007914	2	0.006300
2500	40	5	90	0.258059	0.063643	2	0.006500
2500	40	5	135	0.257797	0.040898	2	0.006300
2500	40	5	180	0.257693	-0.013962	2	0.006300
2500	40	10	90	0.258049	0.143439	2	0.006500

2500	40	10	135	0.257549	0.095920	2	0.006300
2500	40	10	180	0.257336	-0.013944	2	0.006300
2500	40	15	90	0.258050	0.217927	2	0.006500
2500	40	15	135	0.257294	0.150338	2	0.007100
2500	40	15	180	0.256983	-0.013925	2	0.006500
2500	40	20	90	0.258049	0.295547	2	0.005600
2500	40	20	135	0.257043	0.204588	2	0.006500
2500	40	20	180	0.256628	-0.013907	2	0.006400
2500	40	25	90	0.258053	0.373201	2	0.006500
2500	40	25	135	0.256794	0.264276	2	0.005500
2500	40	25	180	0.256278	-0.013889	2	0.006300
2500	40	30	90	0.258054	0.450705	2	0.006400
2500	40	30	135	0.256546	0.311950	2	0.005500
2500	40	30	180	0.255925	-0.013872	2	0.006300
2500	50	5	90	0.258063	0.055870	2	0.007200
2500	50	5	135	0.257798	0.033383	2	0.006200
2500	50	5	180	0.257693	-0.021481	2	0.006300
2500	50	10	90	0.258049	0.135344	2	0.006500
2500	50	10	135	0.257549	0.088419	2	0.006300
2500	50	10	180	0.257337	-0.021452	2	0.006300
2500	50	15	90	0.258050	0.210284	2	0.006500
2500	50	15	135	0.257293	0.148045	2	0.006300
2500	50	15	180	0.256983	-0.021424	2	0.006500
2500	50	20	90	0.258049	0.288745	2	0.005500
2500	50	20	135	0.257045	0.196952	2	0.006500
2500	50	20	180	0.256628	-0.021395	2	0.006400
2500	50	25	90	0.258053	0.365629	2	0.006600
2500	50	25	135	0.256794	0.252024	2	0.006100
2500	50	25	180	0.256278	-0.021367	2	0.006400
2500	50	30	90	0.258054	0.443141	2	0.006500
2500	50	30	135	0.256546	0.304816	2	0.005500
2500	50	30	180	0.255926	-0.021340	2	0.006300
3000	0	5	90	0.311026	0.078703	2	0.004600
3000	0	5	135	0.310727	0.054968	2	0.005000
3000	0	5	180	0.310590	0.000230	2	0.004600
3000	0	10	90	0.311023	0.152540	2	0.004600
3000	0	10	135	0.310414	0.112750	2	0.004800
3000	0	10	180	0.310160	0.000227	2	0.004600
3000	0	15	90	0.311026	0.232277	2	0.004700
3000	0	15	135	0.310109	0.162699	2	0.004600
3000	0	15	180	0.309729	0.000226	2	0.004600
3000	0	20	90	0.311027	0.310253	2	0.004700
3000	0	20	135	0.309804	0.218840	2	0.004500
3000	0	20	180	0.309301	0.000224	2	0.004500
3000	0	25	90	0.311029	0.387991	2	0.004700
3000	0	25	135	0.309502	0.274058	2	0.004700
3000	0	25	180	0.308874	0.000222	2	0.004600
3000	0	30	90	0.311020	0.469565	2	0.004000
3000	0	30	135	0.309199	0.329098	2	0.004500
3000	0	30	180	0.308445	0.000220	2	0.004500
3000	10	5	90	0.311026	0.077604	2	0.006600
3000	10	5	135	0.310742	0.054459	2	0.006700
3000	10	5	180	0.310590	-0.000876	2	0.006500
3000	10	10	90	0.311023	0.151304	2	0.006600
3000	10	10	135	0.310414	0.111639	2	0.006800
3000	10	10	180	0.310160	-0.000876	2	0.006700
3000	10	15	90	0.311026	0.231141	2	0.006600

3000	10	15	135	0.310109	0.161510	2	0.006600
3000	10	15	180	0.309729	-0.000876	2	0.006600
3000	10	20	90	0.311027	0.309125	2	0.006700
3000	10	20	135	0.309804	0.217705	2	0.006400
3000	10	20	180	0.309301	-0.000877	2	0.006500
3000	10	25	90	0.311029	0.386864	2	0.006700
3000	10	25	135	0.309502	0.272934	2	0.006600
3000	10	25	180	0.308874	-0.000877	2	0.006600
3000	10	30	90	0.311020	0.468419	2	0.005700
3000	10	30	135	0.309202	0.327983	2	0.006500
3000	10	30	180	0.308446	-0.000878	2	0.006400
3000	20	5	90	0.311026	0.074384	2	0.006600
3000	20	5	135	0.310737	0.051232	2	0.006600
3000	20	5	180	0.310591	-0.004116	2	0.006600
3000	20	10	90	0.311023	0.147646	2	0.006600
3000	20	10	135	0.310414	0.108388	2	0.006800
3000	20	10	180	0.310161	-0.004113	2	0.006700
3000	20	15	90	0.311026	0.227807	2	0.006600
3000	20	15	135	0.310109	0.158006	2	0.006500
3000	20	15	180	0.309729	-0.004108	2	0.006600
3000	20	20	90	0.311027	0.305817	2	0.006700
3000	20	20	135	0.309804	0.214374	2	0.006400
3000	20	20	180	0.309301	-0.004105	2	0.006500
3000	20	25	90	0.311029	0.383557	2	0.006700
3000	20	25	135	0.309502	0.269635	2	0.006600
3000	20	25	180	0.308874	-0.004101	2	0.006700
3000	20	30	90	0.311020	0.465064	2	0.005700
3000	20	30	135	0.309202	0.324694	2	0.006400
3000	20	30	180	0.308446	-0.004097	2	0.006400
3000	30	5	90	0.311026	0.069151	2	0.006600
3000	30	5	135	0.310733	0.045988	2	0.006600
3000	30	5	180	0.310592	-0.009382	2	0.006500
3000	30	10	90	0.311023	0.141566	2	0.006700
3000	30	10	135	0.310414	0.103116	2	0.006800
3000	30	10	180	0.310162	-0.009372	2	0.006600
3000	30	15	90	0.311026	0.222380	2	0.006700
3000	30	15	135	0.310109	0.152248	2	0.006500
3000	30	15	180	0.309730	-0.009360	2	0.006600
3000	30	20	90	0.311027	0.300434	2	0.006700
3000	30	20	135	0.309804	0.208951	2	0.006500
3000	30	20	180	0.309302	-0.009350	2	0.006400
3000	30	25	90	0.311029	0.378184	2	0.006700
3000	30	25	135	0.309501	0.264268	2	0.006700
3000	30	25	180	0.308875	-0.009339	2	0.006600
3000	30	30	90	0.311020	0.459629	2	0.005700
3000	30	30	135	0.309201	0.319352	2	0.006400
3000	30	30	180	0.308447	-0.009328	2	0.006500
3000	40	5	90	0.311026	0.062012	2	0.006600
3000	40	5	135	0.310729	0.038834	2	0.006600
3000	40	5	180	0.310594	-0.016567	2	0.006500
3000	40	10	90	0.311023	0.138531	2	0.007600
3000	40	10	135	0.310414	0.095939	2	0.006800
3000	40	10	180	0.310164	-0.016547	2	0.006700
3000	40	15	90	0.311025	0.214953	2	0.006600
3000	40	15	135	0.310109	0.144227	2	0.006500
3000	40	15	180	0.309733	-0.016526	2	0.006700
3000	40	20	90	0.311026	0.293081	2	0.006600

3000	40	20	135	0.309804	0.201525	2	0.006500
3000	40	20	180	0.309305	-0.016506	2	0.006500
3000	40	25	90	0.311028	0.370848	2	0.006700
3000	40	25	135	0.309501	0.256931	2	0.006600
3000	40	25	180	0.308878	-0.016485	2	0.006700
3000	40	30	90	0.311020	0.452253	2	0.005800
3000	40	30	135	0.309201	0.312052	2	0.006400
3000	40	30	180	0.308449	-0.016465	2	0.006500
3000	50	5	90	0.311026	0.053067	2	0.006600
3000	50	5	135	0.310727	0.029872	2	0.006600
3000	50	5	180	0.310600	-0.025569	2	0.006600
3000	50	10	90	0.311022	0.129438	2	0.007600
3000	50	10	135	0.310414	0.086968	2	0.006800
3000	50	10	180	0.310170	-0.025537	2	0.006600
3000	50	15	90	0.311025	0.205610	2	0.006700
3000	50	15	135	0.310108	0.138935	2	0.007400
3000	50	15	180	0.309738	-0.025504	2	0.006600
3000	50	20	90	0.311026	0.283850	2	0.006700
3000	50	20	135	0.309803	0.192174	2	0.006400
3000	50	20	180	0.309310	-0.025473	2	0.006500
3000	50	25	90	0.311028	0.361644	2	0.006700
3000	50	25	135	0.309501	0.247715	2	0.006700
3000	50	25	180	0.308883	-0.025441	2	0.006700
3000	50	30	90	0.311019	0.443094	2	0.005700
3000	50	30	135	0.309201	0.302889	2	0.006400
3000	50	30	180	0.308454	-0.025409	2	0.006400
3500	0	5	90	0.364465	0.081015	2	0.004400
3500	0	5	135	0.364104	0.056721	2	0.004300
3500	0	5	180	0.363962	0.000508	2	0.004300
3500	0	10	90	0.364464	0.152446	2	0.004400
3500	0	10	135	0.363741	0.110274	2	0.005000
3500	0	10	180	0.363453	0.000506	2	0.004500
3500	0	15	90	0.364466	0.232285	2	0.004400
3500	0	15	135	0.363384	0.162435	2	0.004500
3500	0	15	180	0.362945	0.000503	2	0.004400
3500	0	20	90	0.364468	0.310650	2	0.004400
3500	0	20	135	0.363025	0.218784	2	0.004400
3500	0	20	180	0.362438	0.000500	2	0.004400
3500	0	25	90	0.364472	0.388776	2	0.004200
3500	0	25	135	0.362669	0.274296	2	0.004400
3500	0	25	180	0.361934	0.000497	2	0.004400
3500	0	30	90	0.364475	0.466801	2	0.004400
3500	0	30	135	0.362311	0.329595	2	0.004500
3500	0	30	180	0.361431	0.000493	2	0.004300
3500	10	5	90	0.364465	0.079757	2	0.006200
3500	10	5	135	0.364104	0.055453	2	0.006200
3500	10	5	180	0.363964	-0.000776	2	0.006200
3500	10	10	90	0.364464	0.151024	2	0.006200
3500	10	10	135	0.363741	0.108976	2	0.007200
3500	10	10	180	0.363454	-0.000777	2	0.006400
3500	10	15	90	0.364465	0.230940	2	0.006200
3500	10	15	135	0.363384	0.161042	2	0.006500
3500	10	15	180	0.362946	-0.000778	2	0.006300
3500	10	20	90	0.364468	0.309317	2	0.006300
3500	10	20	135	0.363025	0.217440	2	0.006300
3500	10	20	180	0.362440	-0.000780	2	0.006300
3500	10	25	90	0.364472	0.387448	2	0.006200

3500	10	25	135	0.362669	0.272967	2	0.006200
3500	10	25	180	0.361935	-0.000781	2	0.006400
3500	10	30	90	0.364475	0.465475	2	0.006200
3500	10	30	135	0.362311	0.328272	2	0.006400
3500	10	30	180	0.361432	-0.000783	2	0.006300
3500	20	5	90	0.364465	0.076070	2	0.006200
3500	20	5	135	0.364104	0.051737	2	0.006200
3500	20	5	180	0.363973	-0.004542	2	0.006100
3500	20	10	90	0.364464	0.146834	2	0.006300
3500	20	10	135	0.363741	0.105160	2	0.007200
3500	20	10	180	0.363463	-0.004537	2	0.006400
3500	20	15	90	0.364465	0.226991	2	0.006300
3500	20	15	135	0.363384	0.156941	2	0.006500
3500	20	15	180	0.362954	-0.004534	2	0.006300
3500	20	20	90	0.364468	0.305409	2	0.006300
3500	20	20	135	0.363025	0.213494	2	0.006400
3500	20	20	180	0.362448	-0.004530	2	0.006300
3500	20	25	90	0.364472	0.383554	2	0.006100
3500	20	25	135	0.362669	0.269066	2	0.006300
3500	20	25	180	0.361943	-0.004527	2	0.006300
3500	20	30	90	0.364475	0.461585	2	0.006200
3500	20	30	135	0.362311	0.324390	2	0.006400
3500	20	30	180	0.361440	-0.004524	2	0.006300
3500	30	5	90	0.364465	0.070081	2	0.006700
3500	30	5	135	0.364104	0.045703	2	0.006600
3500	30	5	180	0.363972	-0.011346	2	0.007500
3500	30	10	90	0.364463	0.144629	2	0.007600
3500	30	10	135	0.363741	0.098951	2	0.007600
3500	30	10	180	0.363462	-0.011330	2	0.007700
3500	30	15	90	0.364465	0.220559	2	0.006800
3500	30	15	135	0.363383	0.150224	2	0.006900
3500	30	15	180	0.362953	-0.011314	2	0.007700
3500	30	20	90	0.364468	0.299048	2	0.006700
3500	30	20	135	0.363025	0.207065	2	0.006800
3500	30	20	180	0.362446	-0.011298	2	0.007700
3500	30	25	90	0.364471	0.377218	2	0.006600
3500	30	25	135	0.362669	0.262716	2	0.006700
3500	30	25	180	0.361941	-0.011282	2	0.007600
3500	30	30	90	0.364474	0.455258	2	0.006700
3500	30	30	135	0.362313	0.318082	2	0.006800
3500	30	30	180	0.361437	-0.011266	2	0.007600
3500	40	5	90	0.364465	0.061915	2	0.007100
3500	40	5	135	0.364104	0.037476	2	0.007000
3500	40	5	180	0.363953	-0.019839	2	0.008900
3500	40	10	90	0.364462	0.136132	2	0.008000
3500	40	10	135	0.363740	0.090472	2	0.008100
3500	40	10	180	0.363442	-0.019811	2	0.009100
3500	40	15	90	0.364465	0.211748	2	0.007100
3500	40	15	135	0.363382	0.145640	2	0.008300
3500	40	15	180	0.362935	-0.019782	2	0.008900
3500	40	20	90	0.364467	0.290349	2	0.007200
3500	40	20	135	0.363025	0.198254	2	0.007200
3500	40	20	180	0.362430	-0.019754	2	0.008800
3500	40	25	90	0.364471	0.368557	2	0.007200
3500	40	25	135	0.362669	0.254028	2	0.007200
3500	40	25	180	0.361923	-0.019726	2	0.008800
3500	40	30	90	0.364476	0.446618	2	0.007100

3500	40	30	135	0.362313	0.309452	2	0.007300
3500	40	30	180	0.361417	-0.019698	2	0.008800
3500	50	5	90	0.364463	0.047482	2	0.008000
3500	50	5	135	0.364104	0.027181	2	0.007100
3500	50	5	180	0.363950	-0.030583	2	0.008800
3500	50	10	90	0.364462	0.125478	2	0.008000
3500	50	10	135	0.363740	0.079840	2	0.008000
3500	50	10	180	0.363442	-0.030602	2	0.009100
3500	50	15	90	0.364464	0.200650	2	0.007100
3500	50	15	135	0.363382	0.135018	2	0.008300
3500	50	15	180	0.362933	-0.030747	2	0.009000
3500	50	20	90	0.364466	0.279418	2	0.007200
3500	50	20	135	0.363024	0.187146	2	0.007200
3500	50	20	180	0.362428	-0.030200	2	0.009900
3500	50	25	90	0.364470	0.357684	2	0.007200
3500	50	25	135	0.362668	0.243102	2	0.007200
3500	50	25	180	0.361927	-0.029457	2	0.008800
3500	50	30	90	0.364474	0.435765	2	0.007100
3500	50	30	135	0.362312	0.298610	2	0.007300
3500	50	30	180	0.361424	-0.029649	2	0.008900
4000	0	5	90	0.418372	0.078248	2	0.006300
4000	0	5	135	0.417958	0.058588	2	0.005500
4000	0	5	180	0.417784	-0.000000	2	0.006800
4000	0	10	90	0.418374	0.156504	2	0.006300
4000	0	10	135	0.417538	0.110663	2	0.006200
4000	0	10	180	0.417192	-0.000000	2	0.006700
4000	0	15	90	0.418379	0.232251	2	0.005600
4000	0	15	135	0.417124	0.165984	2	0.006400
4000	0	15	180	0.416607	-0.000000	2	0.006800
4000	0	20	90	0.418381	0.310973	2	0.005600
4000	0	20	135	0.416710	0.218674	2	0.005700
4000	0	20	180	0.416020	-0.000000	2	0.007900
4000	0	25	90	0.418385	0.389414	2	0.005900
4000	0	25	135	0.416299	0.274454	2	0.005500
4000	0	25	180	0.415436	-0.000000	2	0.006700
4000	0	30	90	0.418388	0.467745	2	0.005600
4000	0	30	135	0.415886	0.329985	2	0.005400
4000	0	30	180	0.414852	-0.000000	2	0.006800
4000	10	5	90	0.418372	0.076749	2	0.008900
4000	10	5	135	0.417958	0.057170	2	0.007800
4000	10	5	180	0.417783	-0.001498	2	0.009700
4000	10	10	90	0.418374	0.155005	2	0.008900
4000	10	10	135	0.417538	0.109167	2	0.008800
4000	10	10	180	0.417195	-0.001496	2	0.009900
4000	10	15	90	0.418379	0.230692	2	0.007900
4000	10	15	135	0.417124	0.164489	2	0.009000
4000	10	15	180	0.416605	-0.001493	2	0.009600
4000	10	20	90	0.418381	0.309431	2	0.008000
4000	10	20	135	0.416710	0.217116	2	0.007800
4000	10	20	180	0.416019	-0.001491	2	0.009600
4000	10	25	90	0.418385	0.387880	2	0.008100
4000	10	25	135	0.416299	0.272915	2	0.007800
4000	10	25	180	0.415435	-0.001489	2	0.009700
4000	10	30	90	0.418390	0.466222	2	0.007900
4000	10	30	135	0.415886	0.328456	2	0.007700
4000	10	30	180	0.414854	-0.001487	2	0.009600
4000	20	5	90	0.418372	0.072353	2	0.009000

4000	20	5	135	0.417956	0.049441	2	0.008900
4000	20	5	180	0.417773	-0.005833	2	0.011700
4000	20	10	90	0.418373	0.150610	2	0.009000
4000	20	10	135	0.417538	0.104782	2	0.008800
4000	20	10	180	0.417193	-0.005880	2	0.010700
4000	20	15	90	0.418379	0.226116	2	0.008000
4000	20	15	135	0.417124	0.160107	2	0.009000
4000	20	15	180	0.416604	-0.005803	2	0.009700
4000	20	20	90	0.418381	0.304906	2	0.008000
4000	20	20	135	0.416710	0.212542	2	0.007800
4000	20	20	180	0.416027	-0.005784	2	0.009900
4000	20	25	90	0.418384	0.383379	2	0.008000
4000	20	25	135	0.416299	0.268398	2	0.007900
4000	20	25	180	0.415436	-0.005754	2	0.009700
4000	20	30	90	0.418390	0.461731	2	0.007900
4000	20	30	135	0.415886	0.323968	2	0.007700
4000	20	30	180	0.414847	-0.005596	2	0.009700
4000	30	5	90	0.418372	0.065209	2	0.009000
4000	30	5	135	0.417956	0.042304	2	0.008700
4000	30	5	180	0.417783	-0.013025	2	0.009700
4000	30	10	90	0.418373	0.143466	2	0.008900
4000	30	10	135	0.417538	0.097653	2	0.008800
4000	30	10	180	0.417193	-0.013007	2	0.009600
4000	30	15	90	0.418378	0.218657	2	0.008000
4000	30	15	135	0.417123	0.152984	2	0.008900
4000	30	15	180	0.416605	-0.012989	2	0.009800
4000	30	20	90	0.418380	0.297543	2	0.007900
4000	30	20	135	0.416710	0.205084	2	0.007800
4000	30	20	180	0.416020	-0.012971	2	0.010100
4000	30	25	90	0.418383	0.376055	2	0.008100
4000	30	25	135	0.416298	0.261043	2	0.007900
4000	30	25	180	0.415434	-0.012953	2	0.009500
4000	30	30	90	0.418389	0.454424	2	0.007900
4000	30	30	135	0.415888	0.316670	2	0.007600
4000	30	30	180	0.414855	-0.012936	2	0.009700
4000	40	5	90	0.418372	0.055455	2	0.008500
4000	40	5	135	0.417956	0.032560	2	0.008300
4000	40	5	180	0.417807	-0.022764	2	0.008700
4000	40	10	90	0.418373	0.133713	2	0.008400
4000	40	10	135	0.417537	0.087924	2	0.008400
4000	40	10	180	0.417219	-0.022732	2	0.008400
4000	40	15	90	0.418378	0.208431	2	0.007500
4000	40	15	135	0.417123	0.143260	2	0.008500
4000	40	15	180	0.416633	-0.022700	2	0.008800
4000	40	20	90	0.418379	0.287467	2	0.007400
4000	40	20	135	0.416708	0.198602	2	0.008300
4000	40	20	180	0.416021	-0.022665	2	0.009300
4000	40	25	90	0.418382	0.366037	2	0.007500
4000	40	25	135	0.416298	0.250971	2	0.007400
4000	40	25	180	0.415438	-0.022633	2	0.009200
4000	40	30	90	0.418387	0.444438	2	0.007400
4000	40	30	135	0.415887	0.306677	2	0.007200
4000	40	30	180	0.414853	-0.022602	2	0.009100
4000	50	5	90	0.418372	0.043227	2	0.008400
4000	50	5	135	0.417955	0.020344	2	0.008300
4000	50	5	180	0.417791	-0.034973	2	0.008100
4000	50	10	90	0.418372	0.121484	2	0.008400

4000	50	10	135	0.417538	0.075733	2	0.008200
4000	50	10	180	0.417202	-0.034924	2	0.008300
4000	50	15	90	0.418376	0.199736	2	0.008500
4000	50	15	135	0.417122	0.131069	2	0.008400
4000	50	15	180	0.416614	-0.034875	2	0.008100
4000	50	20	90	0.418379	0.274792	2	0.007500
4000	50	20	135	0.416709	0.186424	2	0.008100
4000	50	20	180	0.416029	-0.034826	2	0.008300
4000	50	25	90	0.418384	0.353458	2	0.007400
4000	50	25	135	0.416297	0.238290	2	0.007300
4000	50	25	180	0.415445	-0.034777	2	0.008200
4000	50	30	90	0.418388	0.431900	2	0.007400
4000	50	30	135	0.415886	0.294108	2	0.007100
4000	50	30	180	0.414863	-0.034729	2	0.008300
4500	0	5	90	0.472759	0.078495	2	0.006100
4500	0	5	135	0.472282	0.055510	2	0.006100
4500	0	5	180	0.472086	-0.000000	2	0.006800
4500	0	10	90	0.472759	0.157021	2	0.006000
4500	0	10	135	0.471808	0.111028	2	0.006100
4500	0	10	180	0.471415	-0.000000	2	0.006700
4500	0	15	90	0.472763	0.235532	2	0.006000
4500	0	15	135	0.471337	0.166531	2	0.005900
4500	0	15	180	0.470747	-0.000000	2	0.006700
4500	0	20	90	0.472767	0.311240	2	0.005400
4500	0	20	135	0.470865	0.222032	2	0.006100
4500	0	20	180	0.470078	-0.000000	2	0.006700
4500	0	25	90	0.472774	0.389992	2	0.005400
4500	0	25	135	0.470398	0.274571	2	0.005900
4500	0	25	180	0.469417	-0.000000	2	0.006700
4500	0	30	90	0.472777	0.468612	2	0.005500
4500	0	30	135	0.469928	0.330338	2	0.005500
4500	0	30	180	0.468751	-0.000000	2	0.006700
4500	10	5	90	0.472759	0.076801	2	0.008700
4500	10	5	135	0.472282	0.053818	2	0.008800
4500	10	5	180	0.472083	-0.001691	2	0.009800
4500	10	10	90	0.472759	0.155328	2	0.008600
4500	10	10	135	0.471808	0.109338	2	0.008700
4500	10	10	180	0.471414	-0.001689	2	0.009500
4500	10	15	90	0.472763	0.233838	2	0.008800
4500	10	15	135	0.471336	0.164843	2	0.008600
4500	10	15	180	0.470747	-0.001687	2	0.010000
4500	10	20	90	0.472767	0.309485	2	0.007900
4500	10	20	135	0.470865	0.220345	2	0.008800
4500	10	20	180	0.470080	-0.001684	2	0.012000
4500	10	25	90	0.472773	0.388245	2	0.007900
4500	10	25	135	0.470398	0.272818	2	0.007900
4500	10	25	180	0.469414	-0.001682	2	0.010000
4500	10	30	90	0.472777	0.466873	2	0.007800
4500	10	30	135	0.469928	0.328596	2	0.007600
4500	10	30	180	0.468753	-0.001679	2	0.009400
4500	20	5	90	0.472759	0.071835	2	0.008200
4500	20	5	135	0.472282	0.048856	2	0.008500
4500	20	5	180	0.472107	-0.006652	2	0.008300
4500	20	10	90	0.472759	0.150362	2	0.008500
4500	20	10	135	0.471808	0.104383	2	0.008800
4500	20	10	180	0.471437	-0.006643	2	0.008300
4500	20	15	90	0.472763	0.228872	2	0.008100

4500	20	15	135	0.471336	0.159893	2	0.008000
4500	20	15	180	0.470769	-0.006633	2	0.008600
4500	20	20	90	0.472766	0.304336	2	0.007600
4500	20	20	135	0.470864	0.215399	2	0.008100
4500	20	20	180	0.470102	-0.006624	2	0.008100
4500	20	25	90	0.472773	0.383126	2	0.007100
4500	20	25	135	0.470395	0.270904	2	0.008100
4500	20	25	180	0.469438	-0.006615	2	0.008300
4500	20	30	90	0.472776	0.461766	2	0.007500
4500	20	30	135	0.469927	0.323489	2	0.007900
4500	20	30	180	0.468777	-0.006605	2	0.008500
4500	30	5	90	0.472759	0.063763	2	0.008400
4500	30	5	135	0.472282	0.040792	2	0.008300
4500	30	5	180	0.472093	-0.014713	2	0.008100
4500	30	10	90	0.472761	0.142292	2	0.008200
4500	30	10	135	0.471808	0.096329	2	0.008200
4500	30	10	180	0.471422	-0.014692	2	0.008100
4500	30	15	90	0.472762	0.220800	2	0.008200
4500	30	15	135	0.471336	0.151844	2	0.008400
4500	30	15	180	0.470754	-0.014671	2	0.009000
4500	30	20	90	0.472766	0.299311	2	0.008500
4500	30	20	135	0.470864	0.207361	2	0.008300
4500	30	20	180	0.470087	-0.014650	2	0.008000
4500	30	25	90	0.472772	0.374792	2	0.007200
4500	30	25	135	0.470395	0.262872	2	0.008100
4500	30	25	180	0.469423	-0.014630	2	0.008300
4500	30	30	90	0.472778	0.453467	2	0.007400
4500	30	30	135	0.469926	0.318380	2	0.007600
4500	30	30	180	0.468760	-0.014609	2	0.008400
4500	40	5	90	0.472759	0.052743	2	0.008500
4500	40	5	135	0.472282	0.029783	2	0.008400
4500	40	5	180	0.472088	-0.025717	2	0.008400
4500	40	10	90	0.472760	0.131272	2	0.008200
4500	40	10	135	0.471808	0.085334	2	0.008500
4500	40	10	180	0.471419	-0.025680	2	0.008100
4500	40	15	90	0.472761	0.209780	2	0.008100
4500	40	15	135	0.471335	0.140859	2	0.008100
4500	40	15	180	0.470750	-0.025644	2	0.008600
4500	40	20	90	0.472765	0.288290	2	0.008500
4500	40	20	135	0.470863	0.196384	2	0.008600
4500	40	20	180	0.470083	-0.025607	2	0.008600
4500	40	25	90	0.472769	0.366802	2	0.008500
4500	40	25	135	0.470394	0.251906	2	0.008400
4500	40	25	180	0.469419	-0.025571	2	0.008300
4500	40	30	90	0.472774	0.445314	2	0.008400
4500	40	30	135	0.469924	0.307424	2	0.008200
4500	40	30	180	0.468754	-0.025535	2	0.008300
4500	50	5	90	0.472758	0.038927	2	0.008400
4500	50	5	135	0.472281	0.015980	2	0.008500
4500	50	5	180	0.472087	-0.039513	2	0.008400
4500	50	10	90	0.472760	0.117457	2	0.008300
4500	50	10	135	0.471807	0.071551	2	0.008400
4500	50	10	180	0.471417	-0.039457	2	0.008200
4500	50	15	90	0.472760	0.195963	2	0.008200
4500	50	15	135	0.471335	0.127085	2	0.008000
4500	50	15	180	0.470748	-0.039401	2	0.008300
4500	50	20	90	0.472763	0.274473	2	0.008300

4500	50	20	135	0.470862	0.182623	2	0.008300
4500	50	20	180	0.470081	-0.039345	2	0.008000
4500	50	25	90	0.472767	0.352981	2	0.008200
4500	50	25	135	0.470392	0.238160	2	0.008100
4500	50	25	180	0.469417	-0.039290	2	0.008200
4500	50	30	90	0.472771	0.431489	2	0.008400
4500	50	30	135	0.469925	0.293692	2	0.008400
4500	50	30	180	0.468754	-0.039234	2	0.008300
5000	0	5	90	0.527629	0.079011	2	0.005800
5000	0	5	135	0.527088	0.055688	2	0.006000
5000	0	5	180	0.526874	-0.000001	2	0.005900
5000	0	10	90	0.527625	0.157543	2	0.005800
5000	0	10	135	0.526556	0.111397	2	0.005700
5000	0	10	180	0.526120	-0.000001	2	0.005600
5000	0	15	90	0.527627	0.236311	2	0.006000
5000	0	15	135	0.526023	0.167082	2	0.005900
5000	0	15	180	0.525368	-0.000001	2	0.005500
5000	0	20	90	0.527632	0.315082	2	0.005800
5000	0	20	135	0.525494	0.222765	2	0.005700
5000	0	20	180	0.524618	-0.000001	2	0.005600
5000	0	25	90	0.527636	0.393851	2	0.005800
5000	0	25	135	0.524965	0.278444	2	0.005900
5000	0	25	180	0.523872	-0.000001	2	0.006100
5000	0	30	90	0.527642	0.472625	2	0.006100
5000	0	30	135	0.524438	0.334118	2	0.005700
5000	0	30	180	0.523127	-0.000001	2	0.005600
5000	10	5	90	0.527628	0.077246	2	0.008500
5000	10	5	135	0.527088	0.053800	2	0.008100
5000	10	5	180	0.526873	-0.001888	2	0.008100
5000	10	10	90	0.527625	0.155654	2	0.008300
5000	10	10	135	0.526555	0.109511	2	0.008400
5000	10	10	180	0.526119	-0.001886	2	0.008100
5000	10	15	90	0.527627	0.234421	2	0.008500
5000	10	15	135	0.526023	0.165198	2	0.008500
5000	10	15	180	0.525367	-0.001883	2	0.008500
5000	10	20	90	0.527632	0.313192	2	0.008200
5000	10	20	135	0.525493	0.220883	2	0.008100
5000	10	20	180	0.524617	-0.001880	2	0.008100
5000	10	25	90	0.527636	0.391961	2	0.008500
5000	10	25	135	0.524965	0.276564	2	0.008500
5000	10	25	180	0.523869	-0.001878	2	0.008100
5000	10	30	90	0.527641	0.470732	2	0.008200
5000	10	30	135	0.524438	0.332241	2	0.007900
5000	10	30	180	0.523126	-0.001875	2	0.008000
5000	20	5	90	0.527626	0.070149	2	0.008500
5000	20	5	135	0.527088	0.048264	2	0.008500
5000	20	5	180	0.526871	-0.007422	2	0.008300
5000	20	10	90	0.527625	0.150112	2	0.008600
5000	20	10	135	0.526555	0.103982	2	0.008200
5000	20	10	180	0.526117	-0.007412	2	0.008100
5000	20	15	90	0.527629	0.228881	2	0.008300
5000	20	15	135	0.526022	0.159673	2	0.008300
5000	20	15	180	0.525364	-0.007401	2	0.008100
5000	20	20	90	0.527631	0.307649	2	0.008400
5000	20	20	135	0.525493	0.215364	2	0.008200
5000	20	20	180	0.524614	-0.007390	2	0.008300
5000	20	25	90	0.527635	0.386419	2	0.008600

5000	20	25	135	0.524965	0.271050	2	0.008700
5000	20	25	180	0.523868	-0.007380	2	0.008300
5000	20	30	90	0.527642	0.465195	2	0.008500
5000	20	30	135	0.524437	0.326731	2	0.008400
5000	20	30	180	0.523122	-0.007369	2	0.008200
5000	30	5	90	0.527625	0.062138	2	0.008600
5000	30	5	135	0.527088	0.039264	2	0.008300
5000	30	5	180	0.526870	-0.016417	2	0.008200
5000	30	10	90	0.527624	0.141106	2	0.008500
5000	30	10	135	0.526555	0.094995	2	0.008300
5000	30	10	180	0.526113	-0.016394	2	0.008200
5000	30	15	90	0.527628	0.219873	2	0.008600
5000	30	15	135	0.526022	0.150693	2	0.008400
5000	30	15	180	0.525362	-0.016370	2	0.008100
5000	30	20	90	0.527630	0.298641	2	0.008200
5000	30	20	135	0.525492	0.206393	2	0.008200
5000	30	20	180	0.524613	-0.016347	2	0.008700
5000	30	25	90	0.527636	0.377413	2	0.008700
5000	30	25	135	0.524964	0.262087	2	0.008400
5000	30	25	180	0.523866	-0.016324	2	0.008100
5000	30	30	90	0.527640	0.456182	2	0.008400
5000	30	30	135	0.524438	0.317777	2	0.008300
5000	30	30	180	0.523121	-0.016301	2	0.008200
5000	40	5	90	0.527624	0.049924	2	0.008400
5000	40	5	135	0.527088	0.026978	2	0.008300
5000	40	5	180	0.526868	-0.028697	2	0.008300
5000	40	10	90	0.527624	0.128810	2	0.008600
5000	40	10	135	0.526555	0.082726	2	0.008400
5000	40	10	180	0.526114	-0.028656	2	0.008300
5000	40	15	90	0.527627	0.207577	2	0.008400
5000	40	15	135	0.526024	0.138435	2	0.008600
5000	40	15	180	0.525361	-0.028615	2	0.008100
5000	40	20	90	0.527631	0.286346	2	0.008100
5000	40	20	135	0.525494	0.194146	2	0.008100
5000	40	20	180	0.524611	-0.028574	2	0.008300
5000	40	25	90	0.527633	0.365112	2	0.008500
5000	40	25	135	0.524965	0.249854	2	0.008400
5000	40	25	180	0.523864	-0.028533	2	0.008400
5000	40	30	90	0.527640	0.443884	2	0.009000
5000	40	30	135	0.524437	0.305556	2	0.008300
5000	40	30	180	0.523119	-0.028493	2	0.008500
5000	50	5	90	0.527624	0.034534	2	0.008600
5000	50	5	135	0.527088	0.011575	2	0.008200
5000	50	5	180	0.526867	-0.044092	2	0.008100
5000	50	10	90	0.527623	0.113394	2	0.008500
5000	50	10	135	0.526554	0.067346	2	0.008500
5000	50	10	180	0.526113	-0.044029	2	0.008000
5000	50	15	90	0.527625	0.192159	2	0.008300
5000	50	15	135	0.526023	0.123066	2	0.008200
5000	50	15	180	0.525362	-0.043966	2	0.007900
5000	50	20	90	0.527629	0.270927	2	0.008300
5000	50	20	135	0.525493	0.178791	2	0.008800
5000	50	20	180	0.524612	-0.043904	2	0.008600
5000	50	25	90	0.527633	0.349696	2	0.008300
5000	50	25	135	0.524963	0.234514	2	0.008200
5000	50	25	180	0.523863	-0.043841	2	0.008100
5000	50	30	90	0.527638	0.428466	2	0.008400

5000	50	30	135	0.524437	0.290232	2	0.008300
5000	50	30	180	0.523120	-0.043779	2	0.008200
5500	0	5	90	0.582972	0.079073	2	0.006000
5500	0	5	135	0.582378	0.055862	2	0.005800
5500	0	5	180	0.582132	-0.000002	2	0.005800
5500	0	10	90	0.582974	0.158067	2	0.005900
5500	0	10	135	0.581783	0.111768	2	0.006000
5500	0	10	180	0.581294	-0.000002	2	0.006100
5500	0	15	90	0.582978	0.237097	2	0.006200
5500	0	15	135	0.581191	0.167635	2	0.006000
5500	0	15	180	0.580454	-0.000001	2	0.005800
5500	0	20	90	0.582982	0.316128	2	0.005900
5500	0	20	135	0.580601	0.223502	2	0.005900
5500	0	20	180	0.579622	-0.000001	2	0.005800
5500	0	25	90	0.582985	0.395158	2	0.006000
5500	0	25	135	0.580013	0.279364	2	0.006000
5500	0	25	180	0.578788	-0.000001	2	0.006000
5500	0	30	90	0.582994	0.474192	2	0.006000
5500	0	30	135	0.579427	0.335219	2	0.005800
5500	0	30	180	0.577960	-0.000001	2	0.005800
5500	10	5	90	0.582972	0.076989	2	0.008600
5500	10	5	135	0.582378	0.053776	2	0.008400
5500	10	5	180	0.582132	-0.002086	2	0.008400
5500	10	10	90	0.582974	0.155980	2	0.008500
5500	10	10	135	0.581783	0.109685	2	0.008400
5500	10	10	180	0.581293	-0.002083	2	0.008600
5500	10	15	90	0.582978	0.235009	2	0.008700
5500	10	15	135	0.581191	0.165554	2	0.008500
5500	10	15	180	0.580454	-0.002080	2	0.008300
5500	10	20	90	0.582982	0.314039	2	0.008500
5500	10	20	135	0.580601	0.221423	2	0.008500
5500	10	20	180	0.579621	-0.002077	2	0.008600
5500	10	25	90	0.582988	0.393073	2	0.008500
5500	10	25	135	0.580013	0.277286	2	0.008200
5500	10	25	180	0.578788	-0.002074	2	0.008500
5500	10	30	90	0.582993	0.472104	2	0.008700
5500	10	30	135	0.579427	0.333144	2	0.008700
5500	10	30	180	0.577959	-0.002071	2	0.008200
5500	20	5	90	0.582972	0.070878	2	0.008500
5500	20	5	135	0.582378	0.047659	2	0.008600
5500	20	5	180	0.582131	-0.008200	2	0.008900
5500	20	10	90	0.582974	0.149858	2	0.008900
5500	20	10	135	0.581783	0.103577	2	0.008500
5500	20	10	180	0.581292	-0.008188	2	0.008600
5500	20	15	90	0.582977	0.228888	2	0.008800
5500	20	15	135	0.581191	0.159452	2	0.008800
5500	20	15	180	0.580455	-0.008176	2	0.008300
5500	20	20	90	0.582981	0.307917	2	0.008500
5500	20	20	135	0.580600	0.215326	2	0.008400
5500	20	20	180	0.579620	-0.008165	2	0.008700
5500	20	25	90	0.582986	0.386948	2	0.009300
5500	20	25	135	0.580012	0.271196	2	0.008700
5500	20	25	180	0.578787	-0.008153	2	0.008600
5500	20	30	90	0.582991	0.465978	2	0.008600
5500	20	30	135	0.579426	0.327060	2	0.008700
5500	20	30	180	0.577958	-0.008141	2	0.008200
5500	30	5	90	0.582974	0.060951	2	0.008600

5500	30	5	135	0.582378	0.037715	2	0.008500
5500	30	5	180	0.582131	-0.018137	2	0.008300
5500	30	10	90	0.582973	0.139909	2	0.008400
5500	30	10	135	0.581782	0.093650	2	0.008500
5500	30	10	180	0.581292	-0.018111	2	0.008500
5500	30	15	90	0.582976	0.218936	2	0.008900
5500	30	15	135	0.581190	0.149532	2	0.008700
5500	30	15	180	0.580455	-0.018085	2	0.008400
5500	30	20	90	0.582982	0.297967	2	0.008600
5500	30	20	135	0.580599	0.205416	2	0.008800
5500	30	20	180	0.579620	-0.018059	2	0.008600
5500	30	25	90	0.582986	0.376998	2	0.008700
5500	30	25	135	0.580013	0.261296	2	0.008300
5500	30	25	180	0.578787	-0.018033	2	0.008600
5500	30	30	90	0.582991	0.456028	2	0.008700
5500	30	30	135	0.579426	0.317171	2	0.008400
5500	30	30	180	0.577958	-0.018007	2	0.008300
5500	40	5	90	0.582974	0.047269	2	0.010600
5500	40	5	135	0.582378	0.024141	2	0.008400
5500	40	5	180	0.582130	-0.031703	2	0.008400
5500	40	10	90	0.582975	0.126328	2	0.008900
5500	40	10	135	0.581782	0.080098	2	0.008400
5500	40	10	180	0.581291	-0.031657	2	0.008500
5500	40	15	90	0.582975	0.205353	2	0.008500
5500	40	15	135	0.581189	0.135991	2	0.008400
5500	40	15	180	0.580454	-0.031611	2	0.008300
5500	40	20	90	0.582979	0.284382	2	0.008500
5500	40	20	135	0.580601	0.191888	2	0.008500
5500	40	20	180	0.579619	-0.031566	2	0.008700
5500	40	25	90	0.582984	0.363410	2	0.008700
5500	40	25	135	0.580012	0.247781	2	0.008600
5500	40	25	180	0.578788	-0.031521	2	0.008600
5500	40	30	90	0.582990	0.442441	2	0.008700
5500	40	30	135	0.579424	0.303666	2	0.008400
5500	40	30	180	0.577957	-0.031476	2	0.008600
5500	50	5	90	0.582986	0.030602	2	0.008600
5500	50	5	135	0.582378	0.007122	2	0.008500
5500	50	5	180	0.582130	-0.048711	2	0.008800
5500	50	10	90	0.582974	0.109299	2	0.008600
5500	50	10	135	0.581782	0.063108	2	0.008300
5500	50	10	180	0.581292	-0.048641	2	0.008300
5500	50	15	90	0.582976	0.188324	2	0.008800
5500	50	15	135	0.581191	0.119015	2	0.010600
5500	50	15	180	0.580454	-0.048571	2	0.008600
5500	50	20	90	0.582980	0.267351	2	0.008400
5500	50	20	135	0.580599	0.174927	2	0.008400
5500	50	20	180	0.579619	-0.048501	2	0.008400
5500	50	25	90	0.582982	0.346380	2	0.008500
5500	50	25	135	0.580010	0.230835	2	0.008700
5500	50	25	180	0.578786	-0.048432	2	0.008600
5500	50	30	90	0.582990	0.425411	2	0.008700
5500	50	30	135	0.579424	0.286740	2	0.008200
5500	50	30	180	0.577957	-0.048362	2	0.008200
6000	0	5	90	0.638811	0.079330	2	0.005900
6000	0	5	135	0.638154	0.056018	2	0.006200
6000	0	5	180	0.637880	-0.000002	2	0.006000
6000	0	10	90	0.638812	0.158595	2	0.005900

6000	0	10	135	0.637497	0.112143	2	0.005900
6000	0	10	180	0.636954	-0.000002	2	0.005900
6000	0	15	90	0.638816	0.237886	2	0.006000
6000	0	15	135	0.636843	0.168193	2	0.005800
6000	0	15	180	0.636029	-0.000002	2	0.006000
6000	0	20	90	0.638819	0.317178	2	0.006000
6000	0	20	135	0.636192	0.224243	2	0.005900
6000	0	20	180	0.635109	-0.000002	2	0.005900
6000	0	25	90	0.638825	0.396470	2	0.005900
6000	0	25	135	0.635543	0.280288	2	0.006100
6000	0	25	180	0.634189	-0.000002	2	0.006100
6000	0	30	90	0.638832	0.475764	2	0.006000
6000	0	30	135	0.634895	0.336327	2	0.006000
6000	0	30	180	0.633274	-0.000002	2	0.005900
6000	10	5	90	0.638810	0.077045	2	0.008600
6000	10	5	135	0.638154	0.053732	2	0.008700
6000	10	5	180	0.637881	-0.002287	2	0.008900
6000	10	10	90	0.638812	0.156308	2	0.008800
6000	10	10	135	0.637497	0.109861	2	0.008600
6000	10	10	180	0.636953	-0.002283	2	0.008400
6000	10	15	90	0.638816	0.235599	2	0.008500
6000	10	15	135	0.636843	0.165913	2	0.008500
6000	10	15	180	0.636030	-0.002280	2	0.011300
6000	10	20	90	0.638819	0.314891	2	0.009700
6000	10	20	135	0.636191	0.221965	2	0.011500
6000	10	20	180	0.635107	-0.002277	2	0.011600
6000	10	25	90	0.638825	0.394183	2	0.010700
6000	10	25	135	0.635543	0.278012	2	0.011500
6000	10	25	180	0.634190	-0.002273	2	0.010800
6000	10	30	90	0.638834	0.473478	2	0.008800
6000	10	30	135	0.634894	0.334055	2	0.008600
6000	10	30	180	0.633273	-0.002270	2	0.008600
6000	20	5	90	0.638810	0.070346	2	0.008900
6000	20	5	135	0.638154	0.047028	2	0.008800
6000	20	5	180	0.637880	-0.008985	2	0.008800
6000	20	10	90	0.638811	0.149602	2	0.008700
6000	20	10	135	0.637497	0.103169	2	0.008600
6000	20	10	180	0.636954	-0.008971	2	0.008500
6000	20	15	90	0.638815	0.228892	2	0.008500
6000	20	15	135	0.636842	0.159228	2	0.008900
6000	20	15	180	0.636029	-0.008959	2	0.010100
6000	20	20	90	0.638820	0.308184	2	0.010900
6000	20	20	135	0.636190	0.215286	2	0.010600
6000	20	20	180	0.635108	-0.008945	2	0.011000
6000	20	25	90	0.638825	0.387477	2	0.009300
6000	20	25	135	0.635542	0.271339	2	0.009700
6000	20	25	180	0.634189	-0.008933	2	0.009100
6000	20	30	90	0.638831	0.466769	2	0.008400
6000	20	30	135	0.634893	0.327387	2	0.008700
6000	20	30	180	0.633274	-0.008920	2	0.009200
6000	30	5	90	0.638810	0.059459	2	0.008400
6000	30	5	135	0.638153	0.036132	2	0.008500
6000	30	5	180	0.637880	-0.019872	2	0.008400
6000	30	10	90	0.638810	0.138702	2	0.008300
6000	30	10	135	0.637496	0.092293	2	0.008300
6000	30	10	180	0.636954	-0.019843	2	0.008400
6000	30	15	90	0.638814	0.217991	2	0.008300

6000	30	15	135	0.636841	0.148361	2	0.008300
6000	30	15	180	0.636028	-0.019814	2	0.008400
6000	30	20	90	0.638818	0.297281	2	0.008500
6000	30	20	135	0.636192	0.204430	2	0.008400
6000	30	20	180	0.635108	-0.019786	2	0.008400
6000	30	25	90	0.638825	0.376575	2	0.008500
6000	30	25	135	0.635543	0.260495	2	0.008400
6000	30	25	180	0.634188	-0.019757	2	0.008400
6000	30	30	90	0.638830	0.455867	2	0.008500
6000	30	30	135	0.634894	0.316553	2	0.008400
6000	30	30	180	0.633273	-0.019728	2	0.008300
6000	40	5	90	0.638811	0.044605	2	0.008300
6000	40	5	135	0.638153	0.021257	2	0.008500
6000	40	5	180	0.637880	-0.034735	2	0.008400
6000	40	10	90	0.638812	0.123823	2	0.008300
6000	40	10	135	0.637496	0.077447	2	0.008300
6000	40	10	180	0.636954	-0.034684	2	0.008400
6000	40	15	90	0.638815	0.203111	2	0.008300
6000	40	15	135	0.636843	0.133527	2	0.008300
6000	40	15	180	0.636028	-0.034634	2	0.008400
6000	40	20	90	0.638818	0.282401	2	0.008200
6000	40	20	135	0.636190	0.189611	2	0.008300
6000	40	20	180	0.635108	-0.034584	2	0.008400
6000	40	25	90	0.638824	0.361694	2	0.008400
6000	40	25	135	0.635541	0.245688	2	0.008500
6000	40	25	180	0.634188	-0.034534	2	0.008500
6000	40	30	90	0.638830	0.440986	2	0.008500
6000	40	30	135	0.634893	0.301762	2	0.008600
6000	40	30	180	0.633273	-0.034484	2	0.008300
6000	50	5	90	0.638815	0.025997	2	0.008300
6000	50	5	135	0.638153	0.002606	2	0.008600
6000	50	5	180	0.637881	-0.053370	2	0.008400
6000	50	10	90	0.638811	0.105169	2	0.008300
6000	50	10	135	0.637496	0.058836	2	0.008400
6000	50	10	180	0.636952	-0.053292	2	0.008400
6000	50	15	90	0.638813	0.184452	2	0.008400
6000	50	15	135	0.636842	0.114930	2	0.008300
6000	50	15	180	0.636029	-0.053215	2	0.008400
6000	50	20	90	0.638818	0.263743	2	0.008300
6000	50	20	135	0.636189	0.171030	2	0.008400
6000	50	20	180	0.635106	-0.053138	2	0.008300
6000	50	25	90	0.638822	0.343034	2	0.008400
6000	50	25	135	0.635541	0.227127	2	0.008500
6000	50	25	180	0.634189	-0.053061	2	0.008500
6000	50	30	90	0.638827	0.422324	2	0.008500
6000	50	30	135	0.634893	0.283219	2	0.008600
6000	50	30	180	0.633272	-0.052985	2	0.008400
6500	0	5	90	0.695141	0.079596	2	0.006200
6500	0	5	135	0.694421	0.056086	2	0.006000
6500	0	5	180	0.694120	-0.000004	2	0.005700
6500	0	10	90	0.695142	0.159126	2	0.006100
6500	0	10	135	0.693702	0.112519	2	0.005900
6500	0	10	180	0.693105	-0.000004	2	0.005800
6500	0	15	90	0.695148	0.238681	2	0.006000
6500	0	15	135	0.692984	0.168754	2	0.005900
6500	0	15	180	0.692091	-0.000004	2	0.005800
6500	0	20	90	0.695153	0.318235	2	0.006000

6500	0	20	135		0.692271	0.224989	2		0.005800
6500	0	20	180		0.691081	-0.000004	2		0.005800
6500	0	25	90		0.695159	0.397793	2		0.005900
6500	0	25	135		0.691559	0.281218	2		0.005800
6500	0	25	180		0.690074	-0.000003	2		0.005700
6500	0	30	90		0.695165	0.477348	2		0.006000
6500	0	30	135		0.690848	0.337441	2		0.005800
6500	0	30	180		0.689069	-0.000003	2		0.005800
6500	10	5	90		0.695141	0.077110	2		0.008500
6500	10	5	135		0.694421	0.053601	2		0.008400
6500	10	5	180		0.694122	-0.002489	2		0.008100
6500	10	10	90		0.695142	0.156637	2		0.008600
6500	10	10	135		0.693702	0.110037	2		0.008300
6500	10	10	180		0.693104	-0.002485	2		0.008400
6500	10	15	90		0.695147	0.236192	2		0.008600
6500	10	15	135		0.692984	0.166274	2		0.008400
6500	10	15	180		0.692092	-0.002482	2		0.008300
6500	10	20	90		0.695152	0.315747	2		0.008500
6500	10	20	135		0.692271	0.222511	2		0.008300
6500	10	20	180		0.691082	-0.002478	2		0.008200
6500	10	25	90		0.695159	0.395304	2		0.008500
6500	10	25	135		0.691558	0.278743	2		0.008200
6500	10	25	180		0.690073	-0.002474	2		0.008200
6500	10	30	90		0.695164	0.474858	2		0.008400
6500	10	30	135		0.690848	0.334968	2		0.008300
6500	10	30	180		0.689071	-0.002471	2		0.008200
6500	20	5	90		0.695141	0.069821	2		0.008500
6500	20	5	135		0.694421	0.046311	2		0.008400
6500	20	5	180		0.694122	-0.009776	2		0.008200
6500	20	10	90		0.695144	0.149343	2		0.008500
6500	20	10	135		0.693702	0.102757	2		0.008300
6500	20	10	180		0.693104	-0.009762	2		0.008300
6500	20	15	90		0.695146	0.228895	2		0.008800
6500	20	15	135		0.692984	0.159001	2		0.008600
6500	20	15	180		0.692092	-0.009748	2		0.008500
6500	20	20	90		0.695151	0.308451	2		0.008500
6500	20	20	135		0.692270	0.215245	2		0.008400
6500	20	20	180		0.691082	-0.009733	2		0.008200
6500	20	25	90		0.695156	0.388006	2		0.008400
6500	20	25	135		0.691557	0.271482	2		0.008300
6500	20	25	180		0.690073	-0.009719	2		0.008100
6500	20	30	90		0.695164	0.467562	2		0.008500
6500	20	30	135		0.690849	0.327717	2		0.008300
6500	20	30	180		0.689071	-0.009705	2		0.008300
6500	30	5	90		0.695141	0.057975	2		0.008500
6500	30	5	135		0.694420	0.034460	2		0.008600
6500	30	5	180		0.694122	-0.021622	2		0.008300
6500	30	10	90		0.695143	0.137485	2		0.008500
6500	30	10	135		0.693701	0.090926	2		0.008400
6500	30	10	180		0.693104	-0.021590	2		0.008400
6500	30	15	90		0.695145	0.217036	2		0.008600
6500	30	15	135		0.692983	0.147180	2		0.008700
6500	30	15	180		0.692092	-0.021559	2		0.008300
6500	30	20	90		0.695151	0.296591	2		0.008600
6500	30	20	135		0.692268	0.203434	2		0.008400
6500	30	20	180		0.691082	-0.021527	2		0.008200
6500	30	25	90		0.695156	0.376145	2		0.008500

6500	30	25	135		0.691558	0.259686	2		0.008300
6500	30	25	180		0.690073	-0.021496	2		0.008200
6500	30	30	90		0.695165	0.455703	2		0.008500
6500	30	30	135		0.690846	0.315928	2		0.008400
6500	30	30	180		0.689071	-0.021464	2		0.008200
6500	40	5	90		0.695141	0.041809	2		0.008500
6500	40	5	135		0.694420	0.018277	2		0.008400
6500	40	5	180		0.694120	-0.037793	2		0.008300
6500	40	10	90		0.695142	0.121298	2		0.008500
6500	40	10	135		0.693701	0.074776	2		0.008400
6500	40	10	180		0.693105	-0.037737	2		0.008400
6500	40	15	90		0.695145	0.200848	2		0.008600
6500	40	15	135		0.692984	0.131042	2		0.008500
6500	40	15	180		0.692091	-0.037682	2		0.008200
6500	40	20	90		0.695150	0.280402	2		0.008600
6500	40	20	135		0.692269	0.187313	2		0.008400
6500	40	20	180		0.691080	-0.037628	2		0.008200
6500	40	25	90		0.695154	0.359953	2		0.008500
6500	40	25	135		0.691555	0.243578	2		0.008300
6500	40	25	180		0.690074	-0.037573	2		0.008200
6500	40	30	90		0.695161	0.439509	2		0.008700
6500	40	30	135		0.690845	0.299838	2		0.008300
6500	40	30	180		0.689069	-0.037518	2		0.008300
6500	50	5	90		0.695142	0.021551	2		0.008600
6500	50	5	135		0.694420	-0.002016	2		0.008400
6500	50	5	180		0.694122	-0.058068	2		0.008600
6500	50	10	90		0.695141	0.101004	2		0.008800
6500	50	10	135		0.693700	0.054530	2		0.008400
6500	50	10	180		0.693106	-0.057983	2		0.008400
6500	50	15	90		0.695143	0.180550	2		0.008500
6500	50	15	135		0.692983	0.110811	2		0.008400
6500	50	15	180		0.692092	-0.057899	2		0.008300
6500	50	20	90		0.695149	0.260103	2		0.008400
6500	50	20	135		0.692267	0.167099	2		0.008400
6500	50	20	180		0.691082	-0.057814	2		0.008200
6500	50	25	90		0.695154	0.339656	2		0.008500
6500	50	25	135		0.691555	0.223385	2		0.008200
6500	50	25	180		0.690073	-0.057730	2		0.008200
6500	50	30	90		0.695160	0.419209	2		0.008400
6500	50	30	135		0.690844	0.279665	2		0.008300
6500	50	30	180		0.689070	-0.057646	2		0.008200
7000	0	5	90		0.751971	0.079866	2		0.005900
7000	0	5	135		0.751191	0.057570	2		0.005900
7000	0	5	180		0.750859	-0.000005	2		0.005800
7000	0	10	90		0.751972	0.159661	2		0.006000
7000	0	10	135		0.750399	0.112898	2		0.005800
7000	0	10	180		0.749751	-0.000005	2		0.005800
7000	0	15	90		0.751976	0.239478	2		0.005900
7000	0	15	135		0.749620	0.169319	2		0.006100
7000	0	15	180		0.748646	-0.000005	2		0.006000
7000	0	20	90		0.751981	0.319299	2		0.005900
7000	0	20	135		0.748841	0.225738	2		0.006000
7000	0	20	180		0.747546	-0.000005	2		0.006000
7000	0	25	90		0.751989	0.399120	2		0.005900
7000	0	25	135		0.748064	0.282153	2		0.005900
7000	0	25	180		0.746448	-0.000005	2		0.005900
7000	0	30	90		0.751998	0.478942	2		0.005900

7000	0	30	135	0.747291	0.338561	2	0.006000
7000	0	30	180	0.745351	-0.000005	3	0.007800
7000	10	5	90	0.751971	0.077178	2	0.008500
7000	10	5	135	0.751190	0.055373	2	0.008400
7000	10	5	180	0.750858	-0.002693	2	0.008400
7000	10	10	90	0.751972	0.156970	2	0.008500
7000	10	10	135	0.750401	0.110214	2	0.008500
7000	10	10	180	0.749750	-0.002689	2	0.008300
7000	10	15	90	0.751976	0.236788	2	0.008500
7000	10	15	135	0.749619	0.166637	2	0.008800
7000	10	15	180	0.748645	-0.002685	2	0.008600
7000	10	20	90	0.751981	0.316608	2	0.008400
7000	10	20	135	0.748840	0.223058	2	0.008700
7000	10	20	180	0.747545	-0.002681	2	0.008500
7000	10	25	90	0.751988	0.396429	2	0.008500
7000	10	25	135	0.748064	0.279475	2	0.008500
7000	10	25	180	0.746447	-0.002677	2	0.008500
7000	10	30	90	0.751997	0.476251	2	0.008500
7000	10	30	135	0.747290	0.335886	2	0.008600
7000	10	30	180	0.745352	-0.002673	3	0.011200
7000	20	5	90	0.751971	0.069295	2	0.008500
7000	20	5	135	0.751188	0.045863	2	0.009400
7000	20	5	180	0.750858	-0.010575	2	0.008400
7000	20	10	90	0.751971	0.149081	2	0.008500
7000	20	10	135	0.750401	0.102342	2	0.008500
7000	20	10	180	0.749751	-0.010559	2	0.008300
7000	20	15	90	0.751975	0.228897	2	0.008500
7000	20	15	135	0.749619	0.158772	2	0.008900
7000	20	15	180	0.748646	-0.010544	2	0.008600
7000	20	20	90	0.751981	0.308717	2	0.008500
7000	20	20	135	0.748839	0.215201	2	0.008600
7000	20	20	180	0.747545	-0.010528	2	0.008500
7000	20	25	90	0.751989	0.388538	2	0.008500
7000	20	25	135	0.748065	0.271627	2	0.008500
7000	20	25	180	0.746448	-0.010513	2	0.008500
7000	20	30	90	0.751996	0.468359	2	0.008500
7000	20	30	135	0.747291	0.328043	2	0.008500
7000	20	30	180	0.745353	-0.010497	3	0.011200
7000	30	5	90	0.751971	0.056483	2	0.008500
7000	30	5	135	0.751187	0.031693	2	0.008500
7000	30	5	180	0.750859	-0.023387	2	0.008400
7000	30	10	90	0.751973	0.136258	2	0.008600
7000	30	10	135	0.750400	0.089548	2	0.008500
7000	30	10	180	0.749751	-0.023352	2	0.008300
7000	30	15	90	0.751975	0.216073	2	0.008500
7000	30	15	135	0.749617	0.145988	2	0.008700
7000	30	15	180	0.748646	-0.023318	2	0.008600
7000	30	20	90	0.751981	0.295892	2	0.008600
7000	30	20	135	0.748840	0.202430	2	0.008600
7000	30	20	180	0.747543	-0.023284	2	0.008500
7000	30	25	90	0.751987	0.375711	2	0.008300
7000	30	25	135	0.748062	0.258867	2	0.008500
7000	30	25	180	0.746446	-0.023250	2	0.008600
7000	30	30	90	0.751993	0.455529	2	0.008500
7000	30	30	135	0.747290	0.315298	2	0.008600
7000	30	30	180	0.745351	-0.023216	3	0.011200
7000	40	5	90	0.751970	0.038999	2	0.008400

7000	40	5	135	0.751186	0.014785	2	0.008400
7000	40	5	180	0.750858	-0.040878	2	0.008400
7000	40	10	90	0.751971	0.118752	2	0.008500
7000	40	10	135	0.750399	0.072084	2	0.008500
7000	40	10	180	0.749750	-0.040817	2	0.008300
7000	40	15	90	0.751973	0.198564	2	0.008500
7000	40	15	135	0.749619	0.128538	2	0.008800
7000	40	15	180	0.748645	-0.040757	2	0.008600
7000	40	20	90	0.751980	0.278384	2	0.008600
7000	40	20	135	0.748838	0.184994	2	0.008700
7000	40	20	180	0.747544	-0.040697	2	0.008700
7000	40	25	90	0.751984	0.358199	2	0.008600
7000	40	25	135	0.748062	0.241450	2	0.008500
7000	40	25	180	0.746447	-0.040638	2	0.008500
7000	40	30	90	0.751991	0.438019	2	0.008500
7000	40	30	135	0.747289	0.297897	2	0.008600
7000	40	30	180	0.745352	-0.040578	3	0.011100
7000	50	5	90	0.751971	0.017089	2	0.008500
7000	50	5	135	0.751185	-0.006989	2	0.008400
7000	50	5	180	0.750859	-0.062808	2	0.008300
7000	50	10	90	0.751970	0.096807	2	0.008500
7000	50	10	135	0.750399	0.050192	2	0.008400
7000	50	10	180	0.749751	-0.062715	2	0.008300
7000	50	15	90	0.751973	0.176616	2	0.008500
7000	50	15	135	0.749617	0.106660	2	0.008700
7000	50	15	180	0.748646	-0.062623	2	0.008600
7000	50	20	90	0.751978	0.256432	2	0.008600
7000	50	20	135	0.748838	0.163138	2	0.008500
7000	50	20	180	0.747543	-0.062531	2	0.008500
7000	50	25	90	0.751983	0.336249	2	0.008500
7000	50	25	135	0.748061	0.219613	2	0.008400
7000	50	25	180	0.746446	-0.062439	2	0.008600
7000	50	30	90	0.751991	0.416068	2	0.008500
7000	50	30	135	0.747286	0.276084	2	0.008500
7000	50	30	180	0.745351	-0.062348	3	0.011100
7500	0	5	90	0.809301	0.080139	2	0.006000
7500	0	5	135	0.808468	0.056873	2	0.006000
7500	0	5	180	0.808095	-0.000007	2	0.006100
7500	0	10	90	0.809305	0.160200	2	0.006200
7500	0	10	135	0.807598	0.113280	2	0.006000
7500	0	10	180	0.806895	-0.000007	2	0.005800
7500	0	15	90	0.809308	0.240283	2	0.006100
7500	0	15	135	0.806752	0.169888	2	0.006200
7500	0	15	180	0.805696	-0.000007	2	0.005900
7500	0	20	90	0.809314	0.320368	2	0.006100
7500	0	20	135	0.805907	0.226493	2	0.005900
7500	0	20	180	0.804501	-0.000007	2	0.006000
7500	0	25	90	0.809322	0.400455	2	0.006100
7500	0	25	135	0.805066	0.283093	2	0.006100
7500	0	25	180	0.803312	-0.000007	2	0.006000
7500	0	30	90	0.809329	0.480541	2	0.006100
7500	0	30	135	0.804228	0.339688	2	0.005800
7500	0	30	180	0.802125	-0.000007	3	0.008000
7500	10	5	90	0.809301	0.077247	2	0.008600
7500	10	5	135	0.808471	0.053728	2	0.009500
7500	10	5	180	0.808097	-0.002900	2	0.008700
7500	10	10	90	0.809304	0.157304	2	0.008600

7500	10	10	135	0.807598	0.110392	2	0.008400
7500	10	10	180	0.806894	-0.002895	2	0.008300
7500	10	15	90	0.809308	0.237388	2	0.008800
7500	10	15	135	0.806751	0.167002	2	0.008700
7500	10	15	180	0.805696	-0.002891	2	0.008400
7500	10	20	90	0.809313	0.317473	2	0.008700
7500	10	20	135	0.805906	0.223610	2	0.008500
7500	10	20	180	0.804503	-0.002887	2	0.008700
7500	10	25	90	0.809321	0.397560	2	0.008900
7500	10	25	135	0.805065	0.280213	2	0.008800
7500	10	25	180	0.803311	-0.002882	2	0.008700
7500	10	30	90	0.809330	0.477646	2	0.008700
7500	10	30	135	0.804227	0.336809	2	0.008300
7500	10	30	180	0.802124	-0.002878	3	0.011400
7500	20	5	90	0.809300	0.068766	2	0.008500
7500	20	5	135	0.808447	0.045260	2	0.011200
7500	20	5	180	0.808095	-0.011381	2	0.008700
7500	20	10	90	0.809304	0.148816	2	0.008700
7500	20	10	135	0.807598	0.101924	2	0.008300
7500	20	10	180	0.806893	-0.011364	2	0.008500
7500	20	15	90	0.809306	0.228898	2	0.008600
7500	20	15	135	0.806751	0.158540	2	0.008600
7500	20	15	180	0.805697	-0.011347	2	0.008400
7500	20	20	90	0.809314	0.308984	2	0.008700
7500	20	20	135	0.805907	0.215157	2	0.008500
7500	20	20	180	0.804501	-0.011330	2	0.008500
7500	20	25	90	0.809320	0.389068	2	0.008800
7500	20	25	135	0.805066	0.271769	2	0.008700
7500	20	25	180	0.803312	-0.011313	2	0.008600
7500	20	30	90	0.809329	0.469155	2	0.008700
7500	20	30	135	0.804225	0.328373	2	0.008300
7500	20	30	180	0.802125	-0.011297	3	0.011300
7500	30	5	90	0.809300	0.054982	2	0.008500
7500	30	5	135	0.808470	0.031455	2	0.009600
7500	30	5	180	0.808096	-0.025167	2	0.008700
7500	30	10	90	0.809302	0.135020	2	0.008700
7500	30	10	135	0.807597	0.088159	2	0.008400
7500	30	10	180	0.806894	-0.025130	2	0.008400
7500	30	15	90	0.809307	0.215099	2	0.008600
7500	30	15	135	0.806749	0.144786	2	0.008700
7500	30	15	180	0.805695	-0.025093	2	0.008500
7500	30	20	90	0.809313	0.295185	2	0.008700
7500	30	20	135	0.805906	0.201415	2	0.008400
7500	30	20	180	0.804502	-0.025056	2	0.008600
7500	30	25	90	0.809318	0.375268	2	0.008700
7500	30	25	135	0.805063	0.258040	2	0.008700
7500	30	25	180	0.803311	-0.025019	2	0.008600
7500	30	30	90	0.809328	0.455355	2	0.008700
7500	30	30	135	0.804224	0.314660	2	0.008300
7500	30	30	180	0.802126	-0.024982	3	0.011400
7500	40	5	90	0.809300	0.036172	2	0.008600
7500	40	5	135	0.808459	0.013968	2	0.008500
7500	40	5	180	0.808095	-0.043989	2	0.008800
7500	40	10	90	0.809303	0.116187	2	0.008700
7500	40	10	135	0.807596	0.069371	2	0.008300
7500	40	10	180	0.806895	-0.043924	2	0.008400
7500	40	15	90	0.809306	0.196264	2	0.008700

7500	40	15	135	0.806750	0.126012	2	0.008700
7500	40	15	180	0.805696	-0.043859	2	0.008500
7500	40	20	90	0.809311	0.276346	2	0.008600
7500	40	20	135	0.805906	0.182659	2	0.008400
7500	40	20	180	0.804501	-0.043794	2	0.008600
7500	40	25	90	0.809317	0.356430	2	0.008700
7500	40	25	135	0.805062	0.239301	2	0.008700
7500	40	25	180	0.803312	-0.043729	2	0.008500
7500	40	30	90	0.809324	0.436514	2	0.008800
7500	40	30	135	0.804224	0.295940	2	0.008300
7500	40	30	180	0.802124	-0.043664	3	0.011400
7500	50	5	90	0.809300	0.012598	2	0.008600
7500	50	5	135	0.808451	-0.010966	2	0.009600
7500	50	5	180	0.808097	-0.067589	2	0.008700
7500	50	10	90	0.809302	0.092577	2	0.008600
7500	50	10	135	0.807598	0.045820	2	0.008400
7500	50	10	180	0.806894	-0.067488	2	0.008400
7500	50	15	90	0.809303	0.172648	2	0.008700
7500	50	15	135	0.806749	0.102476	2	0.008700
7500	50	15	180	0.805695	-0.067389	2	0.008500
7500	50	20	90	0.809308	0.252729	2	0.008700
7500	50	20	135	0.805905	0.159144	2	0.008500
7500	50	20	180	0.804503	-0.067288	2	0.008700
7500	50	25	90	0.809315	0.332810	2	0.008800
7500	50	25	135	0.805060	0.215809	2	0.008700
7500	50	25	180	0.803311	-0.067189	2	0.008600
7500	50	30	90	0.809322	0.412893	2	0.008700
7500	50	30	135	0.804221	0.272469	2	0.008400
7500	50	30	180	0.802124	-0.067090	3	0.011400
8000	0	5	90	0.867140	0.080416	2	0.006100
8000	0	5	135	0.866219	0.057000	2	0.005700
8000	0	5	180	0.865838	-0.000010	2	0.006000
8000	0	10	90	0.867142	0.160741	2	0.005900
8000	0	10	135	0.865300	0.113666	2	0.005900
8000	0	10	180	0.864541	-0.000010	2	0.005800
8000	0	15	90	0.867148	0.241091	2	0.006000
8000	0	15	135	0.864388	0.170460	2	0.005800
8000	0	15	180	0.863248	-0.000010	2	0.005800
8000	0	20	90	0.867152	0.321444	2	0.005900
8000	0	20	135	0.863476	0.227252	2	0.005900
8000	0	20	180	0.861958	-0.000009	2	0.005800
8000	0	25	90	0.867161	0.401798	2	0.005900
8000	0	25	135	0.862566	0.284039	2	0.005800
8000	0	25	180	0.860673	-0.000009	2	0.005800
8000	0	30	90	0.867170	0.482152	2	0.006000
8000	0	30	135	0.861662	0.340820	2	0.005800
8000	0	30	180	0.859392	-0.000009	3	0.007800
8000	10	5	90	0.867139	0.077317	2	0.008500
8000	10	5	135	0.866220	0.053917	2	0.008300
8000	10	5	180	0.865837	-0.003108	2	0.008500
8000	10	10	90	0.867141	0.157640	2	0.008500
8000	10	10	135	0.865300	0.110572	2	0.008400
8000	10	10	180	0.864540	-0.003104	2	0.008200
8000	10	15	90	0.867147	0.237990	2	0.008700
8000	10	15	135	0.864387	0.167369	2	0.008200
8000	10	15	180	0.863248	-0.003099	2	0.008400
8000	10	20	90	0.867151	0.318342	2	0.008300

8000	10	20	135	0.863475	0.224165	2	0.008600
8000	10	20	180	0.861957	-0.003094	2	0.008200
8000	10	25	90	0.867159	0.398695	2	0.008600
8000	10	25	135	0.862566	0.280954	2	0.008300
8000	10	25	180	0.860672	-0.003090	2	0.008200
8000	10	30	90	0.867171	0.479051	2	0.008600
8000	10	30	135	0.861661	0.337738	2	0.008200
8000	10	30	180	0.859392	-0.003085	3	0.011200
8000	20	5	90	0.867139	0.068234	2	0.008500
8000	20	5	135	0.866222	0.044883	2	0.008200
8000	20	5	180	0.865838	-0.012194	2	0.008600
8000	20	10	90	0.867140	0.148549	2	0.008500
8000	20	10	135	0.865302	0.101502	2	0.008400
8000	20	10	180	0.864541	-0.012176	2	0.008300
8000	20	15	90	0.867145	0.228897	2	0.008600
8000	20	15	135	0.864386	0.158306	2	0.008300
8000	20	15	180	0.863246	-0.012158	2	0.008300
8000	20	20	90	0.867151	0.309249	2	0.008500
8000	20	20	135	0.863474	0.215110	2	0.008500
8000	20	20	180	0.861959	-0.012139	2	0.008200
8000	20	25	90	0.867158	0.389601	2	0.008500
8000	20	25	135	0.862566	0.271910	2	0.008300
8000	20	25	180	0.860673	-0.012121	2	0.008300
8000	20	30	90	0.867169	0.469955	2	0.008600
8000	20	30	135	0.861661	0.328702	2	0.008200
8000	20	30	180	0.859393	-0.012103	3	0.011200
8000	30	5	90	0.867139	0.053471	2	0.008500
8000	30	5	135	0.866232	0.029853	2	0.008700
8000	30	5	180	0.865837	-0.026964	2	0.008500
8000	30	10	90	0.867141	0.133772	2	0.008500
8000	30	10	135	0.865301	0.086760	2	0.008300
8000	30	10	180	0.864540	-0.026923	2	0.008300
8000	30	15	90	0.867145	0.214118	2	0.008600
8000	30	15	135	0.864385	0.143575	2	0.008200
8000	30	15	180	0.863247	-0.026883	2	0.008300
8000	30	20	90	0.867150	0.294468	2	0.008500
8000	30	20	135	0.863474	0.200393	2	0.008600
8000	30	20	180	0.861957	-0.026843	2	0.008200
8000	30	25	90	0.867158	0.374820	2	0.008500
8000	30	25	135	0.862565	0.257206	2	0.008300
8000	30	25	180	0.860674	-0.026803	2	0.008300
8000	30	30	90	0.867167	0.455172	2	0.008500
8000	30	30	135	0.861660	0.314014	2	0.008200
8000	30	30	180	0.859391	-0.026763	3	0.011100
8000	40	5	90	0.867139	0.033325	2	0.008500
8000	40	5	135	0.866216	0.009907	2	0.010200
8000	40	5	180	0.865839	-0.047128	2	0.008500
8000	40	10	90	0.867140	0.113600	2	0.008600
8000	40	10	135	0.865300	0.066638	2	0.008300
8000	40	10	180	0.864541	-0.047057	2	0.008300
8000	40	15	90	0.867145	0.193943	2	0.008500
8000	40	15	135	0.864386	0.123467	2	0.008300
8000	40	15	180	0.863246	-0.046987	2	0.008400
8000	40	20	90	0.867150	0.274292	2	0.008500
8000	40	20	135	0.863473	0.180304	2	0.008500
8000	40	20	180	0.861959	-0.046917	2	0.008200
8000	40	25	90	0.867155	0.354642	2	0.008400

8000	40	25	135	0.862564	0.237137	2	0.008400
8000	40	25	180	0.860673	-0.046847	2	0.008300
8000	40	30	90	0.867164	0.434994	2	0.008400
8000	40	30	135	0.861657	0.293963	2	0.008200
8000	40	30	180	0.859393	-0.046778	3	0.011200
8000	50	5	90	0.867139	0.008078	2	0.008500
8000	50	5	135	0.866239	-0.015594	2	0.009200
8000	50	5	180	0.865838	-0.072412	2	0.008400
8000	50	10	90	0.867141	0.088314	2	0.008500
8000	50	10	135	0.865299	0.041415	2	0.008300
8000	50	10	180	0.864541	-0.072304	2	0.008200
8000	50	15	90	0.867143	0.168651	2	0.008600
8000	50	15	135	0.864383	0.098259	2	0.008200
8000	50	15	180	0.863248	-0.072196	2	0.008300
8000	50	20	90	0.867146	0.248995	2	0.008400
8000	50	20	135	0.863473	0.155118	2	0.008500
8000	50	20	180	0.861958	-0.072088	2	0.008200
8000	50	25	90	0.867155	0.329346	2	0.008400
8000	50	25	135	0.862562	0.211975	2	0.008400
8000	50	25	180	0.860673	-0.071981	2	0.008200
8000	50	30	90	0.867160	0.409693	2	0.008400
8000	50	30	135	0.861655	0.268827	2	0.008300
8000	50	30	180	0.859392	-0.071874	3	0.011100
8500	0	5	90	0.925489	0.080694	2	0.006000
8500	0	5	135	0.924499	0.057175	2	0.006000
8500	0	5	180	0.924089	-0.000013	2	0.006100
8500	0	10	90	0.925492	0.161286	2	0.006000
8500	0	10	135	0.923512	0.114053	2	0.006100
8500	0	10	180	0.922694	-0.000013	2	0.006000
8500	0	15	90	0.925498	0.241906	2	0.005900
8500	0	15	135	0.922528	0.171035	2	0.006000
8500	0	15	180	0.921303	-0.000013	2	0.006000
8500	0	20	90	0.925502	0.322525	2	0.006000
8500	0	20	135	0.921548	0.228017	2	0.006100
8500	0	20	180	0.919917	-0.000012	2	0.006000
8500	0	25	90	0.925512	0.403149	2	0.005900
8500	0	25	135	0.920573	0.284992	2	0.006000
8500	0	25	180	0.918536	-0.000012	2	0.006000
8500	0	30	90	0.925523	0.483771	2	0.006000
8500	0	30	135	0.919599	0.341959	2	0.005900
8500	0	30	180	0.917157	-0.000012	3	0.008000
8500	10	5	90	0.925488	0.077389	2	0.008700
8500	10	5	135	0.924499	0.053882	2	0.008600
8500	10	5	180	0.924088	-0.003319	2	0.008700
8500	10	10	90	0.925491	0.157978	2	0.008400
8500	10	10	135	0.923512	0.110753	2	0.008700
8500	10	10	180	0.922693	-0.003314	2	0.008600
8500	10	15	90	0.925497	0.238597	2	0.008600
8500	10	15	135	0.922528	0.167737	2	0.008600
8500	10	15	180	0.921305	-0.003309	2	0.008600
8500	10	20	90	0.925504	0.319218	2	0.008700
8500	10	20	135	0.921548	0.224722	2	0.008700
8500	10	20	180	0.919917	-0.003304	2	0.008800
8500	10	25	90	0.925511	0.399839	2	0.008700
8500	10	25	135	0.920572	0.281700	2	0.008700
8500	10	25	180	0.918535	-0.003299	2	0.008600
8500	10	30	90	0.925524	0.480463	2	0.008700

8500	10	30	135	0.919598	0.338672	2	0.008600
8500	10	30	180	0.917157	-0.003294	3	0.011500
8500	20	5	90	0.925488	0.067698	2	0.008700
8500	20	5	135	0.924500	0.044233	2	0.008500
8500	20	5	180	0.924089	-0.013015	2	0.008700
8500	20	10	90	0.925490	0.148279	2	0.008500
8500	20	10	135	0.923511	0.101077	2	0.008800
8500	20	10	180	0.922695	-0.012995	2	0.008500
8500	20	15	90	0.925495	0.228895	2	0.008600
8500	20	15	135	0.922529	0.158070	2	0.008600
8500	20	15	180	0.921303	-0.012975	2	0.008600
8500	20	20	90	0.925503	0.309517	2	0.008600
8500	20	20	135	0.921549	0.215063	2	0.008800
8500	20	20	180	0.919918	-0.012956	2	0.008600
8500	20	25	90	0.925512	0.390138	2	0.008600
8500	20	25	135	0.920570	0.272049	2	0.008600
8500	20	25	180	0.918536	-0.012936	2	0.008600
8500	20	30	90	0.925520	0.470758	2	0.008700
8500	20	30	135	0.919598	0.329031	2	0.008400
8500	20	30	180	0.917158	-0.012917	3	0.011300
8500	30	5	90	0.925488	0.051949	2	0.008700
8500	30	5	135	0.924503	0.028565	2	0.008600
8500	30	5	180	0.924088	-0.028776	2	0.008600
8500	30	10	90	0.925491	0.132514	2	0.008400
8500	30	10	135	0.923510	0.085350	2	0.008700
8500	30	10	180	0.922693	-0.028732	2	0.008600
8500	30	15	90	0.925495	0.213128	2	0.008600
8500	30	15	135	0.922527	0.142354	2	0.008500
8500	30	15	180	0.921305	-0.028689	2	0.008600
8500	30	20	90	0.925501	0.293746	2	0.008700
8500	30	20	135	0.921546	0.199362	2	0.008700
8500	30	20	180	0.919917	-0.028646	2	0.008700
8500	30	25	90	0.925508	0.374365	2	0.008600
8500	30	25	135	0.920568	0.256365	2	0.008800
8500	30	25	180	0.918535	-0.028603	2	0.008600
8500	30	30	90	0.925519	0.454988	2	0.008600
8500	30	30	135	0.919595	0.313361	2	0.008600
8500	30	30	180	0.917156	-0.028560	3	0.011300
8500	40	5	90	0.925488	0.030457	2	0.008700
8500	40	5	135	0.924514	0.006719	2	0.009400
8500	40	5	180	0.924090	-0.050295	2	0.008500
8500	40	10	90	0.925489	0.110994	2	0.008600
8500	40	10	135	0.923512	0.063885	2	0.008600
8500	40	10	180	0.922695	-0.050219	2	0.008600
8500	40	15	90	0.925494	0.191603	2	0.008500
8500	40	15	135	0.922528	0.120903	2	0.008600
8500	40	15	180	0.921304	-0.050143	2	0.008600
8500	40	20	90	0.925498	0.272219	2	0.008600
8500	40	20	135	0.921545	0.177929	2	0.008700
8500	40	20	180	0.919918	-0.050068	2	0.008700
8500	40	25	90	0.925507	0.352838	2	0.008700
8500	40	25	135	0.920569	0.234953	2	0.008600
8500	40	25	180	0.918537	-0.049993	2	0.008600
8500	40	30	90	0.925515	0.433456	2	0.008700
8500	40	30	135	0.919594	0.291970	2	0.008500
8500	40	30	180	0.917158	-0.049918	3	0.011300
8500	50	5	90	0.925491	0.003526	2	0.008500

8500	50	5	135	0.924477	-0.019889	2	0.010500
8500	50	5	180	0.924089	-0.077278	2	0.008500
8500	50	10	90	0.925490	0.084018	2	0.008400
8500	50	10	135	0.923511	0.036978	2	0.008600
8500	50	10	180	0.922695	-0.077162	2	0.008500
8500	50	15	90	0.925492	0.164619	2	0.008500
8500	50	15	135	0.922525	0.094010	2	0.008400
8500	50	15	180	0.921303	-0.077046	2	0.008600
8500	50	20	90	0.925496	0.245231	2	0.008500
8500	50	20	135	0.921544	0.151060	2	0.008700
8500	50	20	180	0.919918	-0.076930	2	0.008400
8500	50	25	90	0.925504	0.325848	2	0.008500
8500	50	25	135	0.920566	0.208110	2	0.008700
8500	50	25	180	0.918536	-0.076814	2	0.008500
8500	50	30	90	0.925512	0.406466	2	0.008500
8500	50	30	135	0.919591	0.265154	2	0.008300
8500	50	30	180	0.917158	-0.076699	3	0.011300
9000	0	5	90	0.984354	0.080975	2	0.006200
9000	0	5	135	0.983294	0.057365	2	0.006000
9000	0	5	180	0.982856	-0.000016	2	0.006200
9000	0	10	90	0.984359	0.161836	2	0.006200
9000	0	10	135	0.982236	0.114445	2	0.006200
9000	0	10	180	0.981358	-0.000016	2	0.006100
9000	0	15	90	0.984363	0.242724	2	0.006200
9000	0	15	135	0.981182	0.171615	2	0.006300
9000	0	15	180	0.979870	-0.000016	2	0.006000
9000	0	20	90	0.984371	0.323616	2	0.006200
9000	0	20	135	0.980131	0.228785	2	0.006200
9000	0	20	180	0.978385	-0.000016	2	0.006200
9000	0	25	90	0.984381	0.404507	2	0.006300
9000	0	25	135	0.979086	0.285949	2	0.006300
9000	0	25	180	0.976904	-0.000016	2	0.006100
9000	0	30	90	0.984391	0.485399	2	0.006300
9000	0	30	135	0.978042	0.343105	2	0.006100
9000	0	30	180	0.975427	-0.000016	3	0.008200
9000	10	5	90	0.984354	0.077462	2	0.008800
9000	10	5	135	0.983294	0.053864	2	0.008600
9000	10	5	180	0.982855	-0.003532	2	0.008800
9000	10	10	90	0.984358	0.158319	2	0.008900
9000	10	10	135	0.982236	0.110936	2	0.008800
9000	10	10	180	0.981360	-0.003527	2	0.008900
9000	10	15	90	0.984363	0.239206	2	0.008800
9000	10	15	135	0.981181	0.168109	2	0.009000
9000	10	15	180	0.979870	-0.003521	2	0.008600
9000	10	20	90	0.984370	0.320096	2	0.008800
9000	10	20	135	0.980133	0.225284	2	0.008800
9000	10	20	180	0.978384	-0.003516	2	0.008900
9000	10	25	90	0.984379	0.400989	2	0.008900
9000	10	25	135	0.979085	0.282451	2	0.009100
9000	10	25	180	0.976903	-0.003511	2	0.008700
9000	10	30	90	0.984391	0.481882	2	0.009100
9000	10	30	135	0.978041	0.339609	2	0.008900
9000	10	30	180	0.975427	-0.003505	3	0.011600
9000	20	5	90	0.984354	0.067159	2	0.008900
9000	20	5	135	0.983295	0.043599	2	0.008500
9000	20	5	180	0.982854	-0.013843	2	0.008800
9000	20	10	90	0.984357	0.148007	2	0.008900

9000	20	10	135	0.982235	0.100649	2	0.008800
9000	20	10	180	0.981359	-0.013822	2	0.008700
9000	20	15	90	0.984363	0.228893	2	0.009000
9000	20	15	135	0.981182	0.157830	2	0.009000
9000	20	15	180	0.979869	-0.013801	2	0.008600
9000	20	20	90	0.984369	0.309782	2	0.009000
9000	20	20	135	0.980131	0.215013	2	0.009000
9000	20	20	180	0.978383	-0.013780	2	0.008900
9000	20	25	90	0.984377	0.390673	2	0.009000
9000	20	25	135	0.979085	0.272191	2	0.009000
9000	20	25	180	0.976902	-0.013759	2	0.008600
9000	20	30	90	0.984390	0.471566	2	0.009100
9000	20	30	135	0.978040	0.329360	2	0.008800
9000	20	30	180	0.975425	-0.013738	3	0.011700
9000	30	5	90	0.984353	0.050417	2	0.008900
9000	30	5	135	0.983296	0.026930	2	0.008500
9000	30	5	180	0.982855	-0.030604	2	0.008800
9000	30	10	90	0.984355	0.131246	2	0.008800
9000	30	10	135	0.982234	0.083930	2	0.008800
9000	30	10	180	0.981360	-0.030558	2	0.008700
9000	30	15	90	0.984360	0.212128	2	0.008900
9000	30	15	135	0.981180	0.141123	2	0.009000
9000	30	15	180	0.979870	-0.030511	2	0.008600
9000	30	20	90	0.984369	0.293017	2	0.009200
9000	30	20	135	0.980130	0.198322	2	0.008900
9000	30	20	180	0.978384	-0.030465	2	0.008900
9000	30	25	90	0.984377	0.373907	2	0.009000
9000	30	25	135	0.979083	0.255515	2	0.008900
9000	30	25	180	0.976903	-0.030419	2	0.008600
9000	30	30	90	0.984387	0.454797	2	0.009100
9000	30	30	135	0.978039	0.312701	2	0.008800
9000	30	30	180	0.975427	-0.030373	3	0.011800
9000	40	5	90	0.984353	0.027571	2	0.008900
9000	40	5	135	0.983305	0.004212	2	0.008500
9000	40	5	180	0.982854	-0.053489	2	0.008800
9000	40	10	90	0.984356	0.108369	2	0.008800
9000	40	10	135	0.982236	0.061111	2	0.008800
9000	40	10	180	0.981359	-0.053408	2	0.008600
9000	40	15	90	0.984360	0.189245	2	0.008900
9000	40	15	135	0.981181	0.118317	2	0.008900
9000	40	15	180	0.979869	-0.053327	2	0.008600
9000	40	20	90	0.984364	0.270129	2	0.009000
9000	40	20	135	0.980129	0.175536	2	0.008800
9000	40	20	180	0.978383	-0.053246	2	0.009000
9000	40	25	90	0.984372	0.351016	2	0.008900
9000	40	25	135	0.979080	0.232752	2	0.009000
9000	40	25	180	0.976902	-0.053166	2	0.008600
9000	40	30	90	0.984384	0.431908	2	0.009000
9000	40	30	135	0.978037	0.289960	2	0.008800
9000	40	30	180	0.975426	-0.053086	3	0.012100
9000	50	5	90	0.984357	-0.001057	2	0.008800
9000	50	5	135	0.983294	-0.024958	2	0.010800
9000	50	5	180	0.982854	-0.082189	2	0.008700
9000	50	10	90	0.984356	0.079690	2	0.008900
9000	50	10	135	0.982235	0.032508	2	0.008900
9000	50	10	180	0.981359	-0.082064	2	0.009300
9000	50	15	90	0.984358	0.160558	2	0.009900

9000	50	15	135	0.981178	0.089729	2	0.009500
9000	50	15	180	0.979869	-0.081939	2	0.008600
9000	50	20	90	0.984364	0.241439	2	0.009000
9000	50	20	135	0.980127	0.146972	2	0.008700
9000	50	20	180	0.978383	-0.081815	2	0.008800
9000	50	25	90	0.984371	0.322324	2	0.008900
9000	50	25	135	0.979079	0.204215	2	0.009000
9000	50	25	180	0.976905	-0.081691	2	0.008600
9000	50	30	90	0.984379	0.403209	2	0.009000
9000	50	30	135	0.978033	0.261451	2	0.008800
9000	50	30	180	0.975428	-0.081568	3	0.011700
9500	0	5	90	1.043740	0.081259	2	0.005900
9500	0	5	135	1.042608	0.057564	2	0.006000
9500	0	5	180	1.042138	-0.000020	2	0.006100
9500	0	10	90	1.043744	0.162389	2	0.005900
9500	0	10	135	1.041477	0.114838	2	0.006000
9500	0	10	180	1.040542	-0.000020	2	0.006000
9500	0	15	90	1.043750	0.243548	2	0.005900
9500	0	15	135	1.040352	0.172199	2	0.005800
9500	0	15	180	1.038950	-0.000020	2	0.006000
9500	0	20	90	1.043758	0.324710	2	0.005900
9500	0	20	135	1.039230	0.229559	2	0.006100
9500	0	20	180	1.037361	-0.000020	2	0.006000
9500	0	25	90	1.043767	0.405873	2	0.006000
9500	0	25	135	1.038112	0.286913	2	0.005900
9500	0	25	180	1.035779	-0.000020	3	0.008000
9500	0	30	90	1.043781	0.487038	2	0.005900
9500	0	30	135	1.036996	0.344272	3	0.008100
9500	0	30	180	1.034203	-0.000020	3	0.007900
9500	10	5	90	1.043743	0.077535	2	0.008400
9500	10	5	135	1.042608	0.053852	2	0.008600
9500	10	5	180	1.042140	-0.003748	2	0.008700
9500	10	10	90	1.043746	0.158662	2	0.008400
9500	10	10	135	1.041477	0.111120	2	0.008600
9500	10	10	180	1.040542	-0.003742	2	0.008500
9500	10	15	90	1.043749	0.239819	2	0.008600
9500	10	15	135	1.040352	0.168484	2	0.008200
9500	10	15	180	1.038949	-0.003736	2	0.008500
9500	10	20	90	1.043757	0.320981	2	0.008400
9500	10	20	135	1.039229	0.225848	2	0.008600
9500	10	20	180	1.037363	-0.003731	2	0.008700
9500	10	25	90	1.043768	0.402145	2	0.008700
9500	10	25	135	1.038111	0.283204	2	0.008400
9500	10	25	180	1.035781	-0.003725	3	0.011400
9500	10	30	90	1.043778	0.483308	2	0.008400
9500	10	30	135	1.036995	0.340567	3	0.011600
9500	10	30	180	1.034203	-0.003719	3	0.011100
9500	20	5	90	1.043742	0.066617	2	0.008400
9500	20	5	135	1.042606	0.042971	2	0.008600
9500	20	5	180	1.042138	-0.014679	2	0.008600
9500	20	10	90	1.043744	0.147732	2	0.008500
9500	20	10	135	1.041476	0.100218	2	0.008600
9500	20	10	180	1.040540	-0.014656	2	0.008500
9500	20	15	90	1.043749	0.228888	2	0.008500
9500	20	15	135	1.040350	0.157589	2	0.008200
9500	20	15	180	1.038948	-0.014634	2	0.008500
9500	20	20	90	1.043756	0.310048	2	0.008400

9500	20	20	135	1.039229	0.214963	2	0.008700
9500	20	20	180	1.037362	-0.014611	2	0.008600
9500	20	25	90	1.043765	0.391210	2	0.008500
9500	20	25	135	1.038110	0.272331	2	0.008400
9500	20	25	180	1.035780	-0.014589	3	0.011500
9500	20	30	90	1.043778	0.472374	2	0.008400
9500	20	30	135	1.036996	0.329704	3	0.011400
9500	20	30	180	1.034202	-0.014567	3	0.011100
9500	30	5	90	1.043742	0.048875	2	0.008400
9500	30	5	135	1.042610	0.025299	2	0.008600
9500	30	5	180	1.042137	-0.032449	2	0.008600
9500	30	10	90	1.043742	0.129968	2	0.008500
9500	30	10	135	1.041478	0.082500	2	0.008600
9500	30	10	180	1.040542	-0.032399	2	0.008500
9500	30	15	90	1.043748	0.211120	2	0.009000
9500	30	15	135	1.040351	0.139882	2	0.009300
9500	30	15	180	1.038950	-0.032350	2	0.009000
9500	30	20	90	1.043755	0.292279	2	0.008700
9500	30	20	135	1.039228	0.197273	2	0.008600
9500	30	20	180	1.037361	-0.032300	2	0.008700
9500	30	25	90	1.043763	0.373439	2	0.008600
9500	30	25	135	1.038108	0.254657	2	0.008400
9500	30	25	180	1.035779	-0.032251	3	0.011400
9500	30	30	90	1.043774	0.454600	2	0.008600
9500	30	30	135	1.036992	0.312048	3	0.011500
9500	30	30	180	1.034203	-0.032202	3	0.011100
9500	40	5	90	1.043741	0.024664	2	0.008500
9500	40	5	135	1.042615	0.001210	2	0.008500
9500	40	5	180	1.042139	-0.056713	2	0.008700
9500	40	10	90	1.043742	0.105723	2	0.008500
9500	40	10	135	1.041477	0.058317	2	0.008500
9500	40	10	180	1.040541	-0.056627	2	0.008500
9500	40	15	90	1.043745	0.186867	2	0.008500
9500	40	15	135	1.040350	0.115714	2	0.008300
9500	40	15	180	1.038949	-0.056540	2	0.008500
9500	40	20	90	1.043752	0.268023	2	0.008400
9500	40	20	135	1.039226	0.173124	2	0.008700
9500	40	20	180	1.037363	-0.056454	2	0.008600
9500	40	25	90	1.043760	0.349181	2	0.008600
9500	40	25	135	1.038107	0.230533	2	0.008400
9500	40	25	180	1.035781	-0.056367	3	0.011500
9500	40	30	90	1.043771	0.430341	2	0.008600
9500	40	30	135	1.036991	0.287947	3	0.011300
9500	40	30	180	1.034203	-0.056282	3	0.011100
9500	50	5	90	1.043742	-0.005672	2	0.008400
9500	50	5	135	1.042624	-0.029695	2	0.009500
9500	50	5	180	1.042139	-0.087144	2	0.008700
9500	50	10	90	1.043742	0.075330	2	0.008500
9500	50	10	135	1.041476	0.028007	2	0.008600
9500	50	10	180	1.040541	-0.087011	2	0.008500
9500	50	15	90	1.043745	0.156464	2	0.008400
9500	50	15	135	1.040347	0.085416	2	0.008400
9500	50	15	180	1.038949	-0.086877	2	0.008400
9500	50	20	90	1.043751	0.237614	2	0.008400
9500	50	20	135	1.039224	0.142853	2	0.008600
9500	50	20	180	1.037362	-0.086745	2	0.008700
9500	50	25	90	1.043757	0.318769	2	0.008500

9500	50	25	135	1.038105	0.200288	2	0.008500
9500	50	25	180	1.035781	-0.086612	3	0.011400
9500	50	30	90	1.043766	0.399925	2	0.008600
9500	50	30	135	1.036989	0.257734	3	0.011400
9500	50	30	180	1.034202	-0.086481	3	0.011200
10000	0	5	90	1.103652	0.081545	2	0.006100
10000	0	5	135	1.102445	0.057767	2	0.006000
10000	0	5	180	1.101945	-0.000025	2	0.006000
10000	0	10	90	1.103655	0.162946	2	0.006000
10000	0	10	135	1.101240	0.115235	2	0.005900
10000	0	10	180	1.100244	-0.000025	2	0.005900
10000	0	15	90	1.103661	0.244376	2	0.006000
10000	0	15	135	1.100043	0.172788	2	0.006200
10000	0	15	180	1.098547	-0.000025	2	0.006200
10000	0	20	90	1.103669	0.325811	2	0.006100
10000	0	20	135	1.098845	0.230338	2	0.006100
10000	0	20	180	1.096857	-0.000025	2	0.006200
10000	0	25	90	1.103680	0.407247	2	0.006000
10000	0	25	135	1.097655	0.287882	2	0.006000
10000	0	25	180	1.095171	-0.000025	3	0.008200
10000	0	30	90	1.103694	0.488683	2	0.005900
10000	0	30	135	1.096467	0.345431	3	0.008200
10000	0	30	180	1.093492	-0.000024	3	0.008000
10000	10	5	90	1.103652	0.077610	2	0.008700
10000	10	5	135	1.102445	0.053843	2	0.008600
10000	10	5	180	1.101945	-0.003966	2	0.008600
10000	10	10	90	1.103657	0.159006	2	0.008500
10000	10	10	135	1.101240	0.111306	2	0.008500
10000	10	10	180	1.100243	-0.003960	2	0.008400
10000	10	15	90	1.103663	0.240437	2	0.008600
10000	10	15	135	1.100043	0.168860	2	0.008800
10000	10	15	180	1.098546	-0.003954	2	0.008900
10000	10	20	90	1.103671	0.321871	2	0.008600
10000	10	20	135	1.098847	0.226415	2	0.008600
10000	10	20	180	1.096857	-0.003947	2	0.008800
10000	10	25	90	1.103681	0.403306	2	0.008600
10000	10	25	135	1.097653	0.283962	2	0.008400
10000	10	25	180	1.095170	-0.003941	3	0.011700
10000	10	30	90	1.103693	0.484742	2	0.008500
10000	10	30	135	1.096466	0.341516	3	0.011600
10000	10	30	180	1.093491	-0.003935	3	0.011400
10000	20	5	90	1.103652	0.066071	2	0.008400
10000	20	5	135	1.102445	0.042343	2	0.008600
10000	20	5	180	1.101946	-0.015522	2	0.008700
10000	20	10	90	1.103656	0.147455	2	0.008400
10000	20	10	135	1.101242	0.099784	2	0.008700
10000	20	10	180	1.100242	-0.015498	2	0.008400
10000	20	15	90	1.103660	0.228882	2	0.008400
10000	20	15	135	1.100041	0.157346	2	0.008900
10000	20	15	180	1.098548	-0.015474	2	0.008700
10000	20	20	90	1.103669	0.310315	2	0.008600
10000	20	20	135	1.098844	0.214912	2	0.008700
10000	20	20	180	1.096856	-0.015451	2	0.008900
10000	20	25	90	1.103679	0.391751	2	0.008500
10000	20	25	135	1.097652	0.272471	2	0.008500
10000	20	25	180	1.095172	-0.015427	3	0.011700
10000	20	30	90	1.103692	0.473186	2	0.008700

10000	20	30	135	1.096466	0.330036	3	0.011600
10000	20	30	180	1.093490	-0.015403	3	0.011500
10000	30	5	90	1.103654	0.047322	2	0.008600
10000	30	5	135	1.102446	0.023664	2	0.008500
10000	30	5	180	1.101945	-0.034311	2	0.008600
10000	30	10	90	1.103653	0.128682	2	0.008400
10000	30	10	135	1.101241	0.081059	2	0.008500
10000	30	10	180	1.100244	-0.034258	2	0.008500
10000	30	15	90	1.103661	0.210105	2	0.008500
10000	30	15	135	1.100041	0.138633	2	0.008800
10000	30	15	180	1.098547	-0.034205	2	0.008700
10000	30	20	90	1.103667	0.291535	2	0.008600
10000	30	20	135	1.098843	0.196215	2	0.008700
10000	30	20	180	1.096857	-0.034152	2	0.008800
10000	30	25	90	1.103677	0.372969	2	0.008600
10000	30	25	135	1.097652	0.253792	2	0.008400
10000	30	25	180	1.095171	-0.034100	3	0.011700
10000	30	30	90	1.103689	0.454401	2	0.008600
10000	30	30	135	1.096464	0.311375	3	0.011700
10000	30	30	180	1.093492	-0.034047	3	0.011400
10000	40	5	90	1.103654	0.021738	2	0.008600
10000	40	5	135	1.102450	-0.001799	2	0.008600
10000	40	5	180	1.101944	-0.059967	2	0.008600
10000	40	10	90	1.103653	0.103058	2	0.008400
10000	40	10	135	1.101239	0.055503	2	0.008500
10000	40	10	180	1.100243	-0.059874	2	0.008400
10000	40	15	90	1.103658	0.184472	2	0.008500
10000	40	15	135	1.100038	0.113091	2	0.008800
10000	40	15	180	1.098546	-0.059782	2	0.008700
10000	40	20	90	1.103663	0.265897	2	0.008500
10000	40	20	135	1.098843	0.170694	2	0.008700
10000	40	20	180	1.096857	-0.059690	2	0.008800
10000	40	25	90	1.103674	0.347330	2	0.008500
10000	40	25	135	1.097650	0.228296	2	0.008400
10000	40	25	180	1.095173	-0.059598	3	0.011700
10000	40	30	90	1.103684	0.428759	2	0.008600
10000	40	30	135	1.096459	0.285903	3	0.011600
10000	40	30	180	1.093491	-0.059507	3	0.011400
10000	50	5	90	1.103654	-0.010317	2	0.008600
10000	50	5	135	1.102469	-0.033655	2	0.008600
10000	50	5	180	1.101947	-0.092145	2	0.008600
10000	50	10	90	1.103653	0.070939	2	0.008400
10000	50	10	135	1.101241	0.023474	2	0.008600
10000	50	10	180	1.100243	-0.092003	2	0.008400
10000	50	15	90	1.103656	0.152340	2	0.008500
10000	50	15	135	1.100037	0.081073	2	0.008700
10000	50	15	180	1.098549	-0.091861	2	0.008700
10000	50	20	90	1.103660	0.233759	2	0.008600
10000	50	20	135	1.098840	0.138702	2	0.008700
10000	50	20	180	1.096857	-0.091720	2	0.008800
10000	50	25	90	1.103669	0.315187	2	0.008400
10000	50	25	135	1.097647	0.196332	2	0.008500
10000	50	25	180	1.095172	-0.091578	3	0.012000
10000	50	30	90	1.103679	0.396614	2	0.009700
10000	50	30	135	1.096458	0.253973	3	0.013000
10000	50	30	180	1.093491	-0.091438	3	0.011800