12-2005

Applications of Genetic Algorithms to a Variety of Problems in Physics and Astronomy

Kevin Richard Williams
University of Tennessee - Knoxville

To the Graduate Council:

I am submitting herewith a thesis written by Kevin Richard Williams entitled "Applications of Genetic Algorithms to a Variety of Problems in Physics and Astronomy." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Physics.

M W Guidry, Major Professor

We have read this thesis and recommend its acceptance:

Marianne Breinig, Chia C. Shih

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Kevin Richard Williams entitled "Applications of Genetic Algorithms to a Variety of Problems in Physics and Astronomy." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Physics.

M W Guidry

_____

Major Professor

We have read this thesis
and recommend its acceptance:

Marianne Breinig

_____

Chia C. Shih

_____

Accepted for the Council:

Anne Mayhew

_____

Vice Chancellor and
Dean of Graduate Studies

(Original signatures are on file with official student records)

Applications of Genetic Algorithms to a Variety of Problems in Physics and Astronomy

A Thesis

Presented for the

Masters Degree

The University of Tennessee, Knoxville

Kevin Richard Williams

December 2005

Acknowledgments

I wish to acknowledge and express my appreciation to the following individuals. First, I would like to thank Dr. Guidry, Dr. Breinig, Dr. Shih, and Dr. Messer for taking the time to sit on my committee and for all of their advice and suggestions. I would also like to thank Suzanne Parete-Koon for allowing me to look over a copy of her thesis, which served as a nice example of how I should organize my own thesis. Thanks also to the following individuals who have granted me permission to make use of their figures and diagrams that appear throughout the thesis: Dr. Stern and Meredith Bene, of NASA's Goddard Space Flight Center, Dr. Genzel and Dr. Haiman, of the University of California in Berkeley, and Dr. Trick of the Carnegie Bosch Institute.

Abstract

Genetic algorithms are search techniques that borrow ideas from the biological process of evolution. By means of natural selection, genetic algorithms can be employed as robust numerical optimizers on problems that would normally be extremely problematic due to ill-behaved search spaces. The genetic algorithm has an advantage in that it is a *global* optimization strategy, as opposed to more conventional methods, which will often terminate at local maxima.

The success and resourcefulness of genetic algorithms as problem-solving strategies are quickly gaining recognition among researchers of diverse areas of study. In this thesis I elaborate on applications of a genetic algorithm to several problems in physics and astronomy.

First, the concepts behind functional optimization are discussed, as well as several computational strategies for locating optima. The basic ideas behind genetic algorithms and their operations are then outlined, as well as advantages and disadvantages of the genetic algorithm over the previously discussed optimization techniques. Then the results of several applications of a genetic algorithm are discussed. The majority are relatively simple problems (involving the fitting of only one or two parameters) that nicely illustrate the genetic algorithm's approach to optimization of "fitness," and its ability to reproduce familiar results. The last two problems discussed are non-trivial and demonstrate the genetic algorithm's robustness. The first of these was the calculation of the mass of the radio source Sagittarius A*, believed to be a supermassive black hole at

the center of the Milky Way, which required that the genetic algorithm find several orbital elements associated with an orbiting star. The results obtained with the genetic algorithm were in good agreement with those obtained by Genzel *et al* [19]. Then discussed was the problem of parametrization of thermonuclear reaction rates. This problem is especially interesting because attempts at fitting several rates prior to the implementation of the genetic algorithm proved to be unsuccessful. Some of the rates varied with temperature over many orders of magnitude, and required the genetic algorithm to find as many as twenty-eight parameters. A relatively good fit was obtained for all of the rates.

In the applications of genetic algorithms discussed in this thesis, it has been found that they can outperform conventional optimization strategies for difficult, multidimensional problems, and can perform at least as well as conventional methods when applied to more trivial problems.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Function optimization is a problem encountered quite often in the physical sciences, and a great deal of effort has been spent in inventing and perfecting computational methods that yield optima of difficult functions, the extrema of which cannot be found analytically.

It is the purpose of this thesis to explore one particular method that is quickly gaining recognition as a robust optimization strategy, the genetic algorithm. In this chapter, the reader is introduced to the basic concepts behind optimization, as well as several optimization techniques. In the following chapters, the computational procedures of the genetic algorithm, the results of applications to several problems, and comparisons with other optimization methods will be discussed.

1.1 Optimization Theory

Given a function, $f(x_1, x_2, x_3 \ldots)$, the task of optimization is to find the set of variables, x_i (where $i = 1,2,3\ldots$), for which f takes a maximum value. This parameter set is termed the optimal solution. An optimum value of a function can be one of two types, as is illustrated in Figure 1.1. The figure shows a one-dimensional function bounded by the points *a* and *e*. The two optima shown above are located at the points

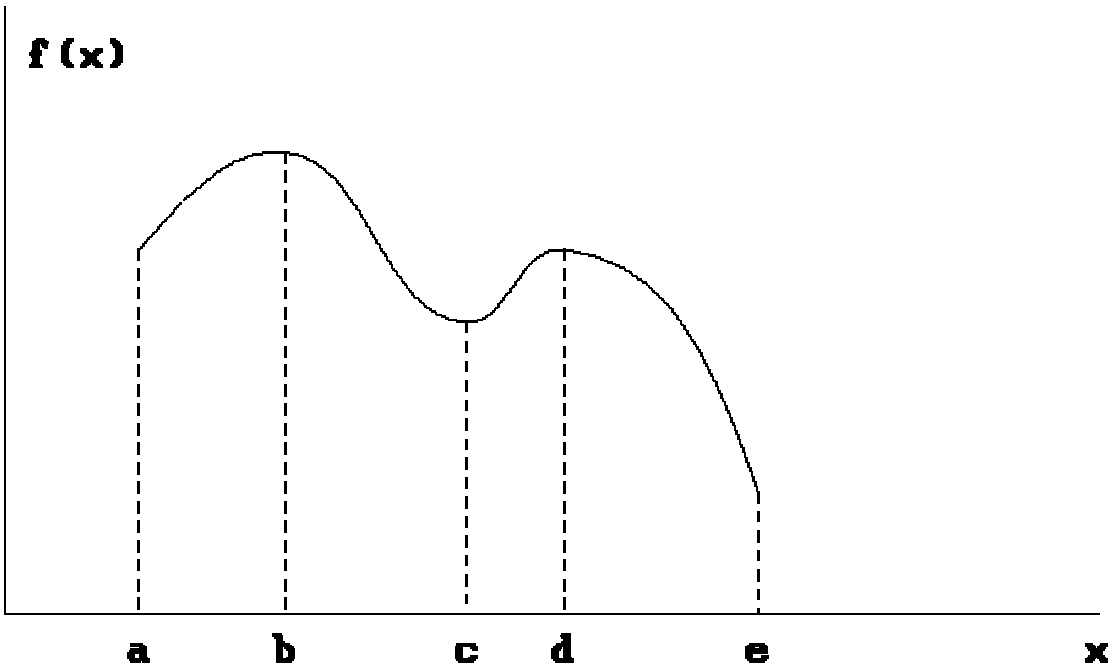

Fig. 1.1 One-dimensional function f(x)

Source: http://mat.gsia.cmu.edu/QUANT/NOTES/chap2/node4.html

$x = b$ and $x = d$. The former is a global optimum, whereas the latter is known as a local optimum. The global optimum is truly the highest peak of f. The local optimum, on the other hand, denotes the highest point in the immediate vicinity. Thus, the point $x = b$ in the illustration need not correspond to a global optimum if the observed region were to be extended beyond the interval $[a,e]$.

The basic calculus approach to the problem of locating an optimum is to differentiate the function and find the value of x that satisfies the condition

$$f'(x) = 0.$$

This x-value corresponds to an extremum of f. Few real world problems, however, can be solved by such simple means. Firstly, the above analytical method is not applicable for many functions, such as transcendental functions, where the task of finding the roots is non-trivial and often itself proves to be a very difficult problem. Also, for a complicated function with several extrema, it would be a difficult task to find all the roots of the above formula.

There is a nice example from optics that illustrates the limitation of the above approach [1]. This problem is encountered when studying the diffraction pattern resulting from a single vertical slit. If x is taken to the distance measured in the direction perpendicular to the slit, then the intensity of the diffraction is related to x through the function

$$I(x) = \left(\frac{\sin x}{x} \right)^2 .$$

An attempt to find the extrema of such a function using the above analytical procedure results in the following transcendental equation.

$$2 \sin x \left(\frac{\cos x}{x} - \frac{\sin x}{x^2} \right) = 0$$

The minima are easily obtained by recognizing that the above equation is satisfied when $\sin x = 0$, and thus $x_n = n\pi$ (where $n = 1,2,3\ldots$). Finding the maxima, on the other hand, would require one to algebraically solve for x, which is not possible for the above formula. In such a case, a more practical approach is to solve the problem numerically.

There are several numerical methods of optimization, the strengths and weaknesses of which are dependent on the kind of problem. Finding the particular method best suited for optimizing a given type of function is thus very important. Given below are descriptions of several common optimization routines, and some strengths and weaknesses associated with each.

1.2 Other Search Techniques

Random Search

In this method, points in the search space are randomly selected and evaluated. After many random evaluations, the point that yields the greatest function value is taken as the maximum. This strategy is rarely used by itself [2].

Gradient Methods

This strategy is restricted to finding local optima. This alone presents a difficulty in applying such a technique to the global optimization problem. The basic procedure behind these so-called "hill-climbing" methods is outlined as follows. Starting at a random location in parameter-space, the direction of steepest ascent in the vicinity of the local landscape is evaluated, usually by making use of gradient information. After moving a specified distance in that direction, the path of steepest ascent is again evaluated. This process is reiterated until the surrounding terrain is downhill in all directions. At finding this optimum, the computation ceases.

The hill-climbing method is serial, meaning that only one point at a time in solution space is evaluated and hence can only search one direction, and thus remains ignorant of the shape and behavior of the overall landscape. The success of the hill-climbing strategy in optimization is therefore highly dependent upon the problem and the search space in question. The method assumes that a derivative exists for the function being optimized. Thus, hill-climbing often fails if a derivative cannot be evaluated. Such would be the case when employing a gradient-based method on a discontinuous function, for example.

Another difficulty encountered in the hill-climbing strategy is the tendency for a search to terminate on local maxima. The hill-climbing strategy would be ideal for finding the optimum in unimodal landscapes, like that of the parabolic function shown in Fig. 1.2. In this particular problem, there exist only one maximum, so there is no question that the global optimum has been found. Consider, on the other hand, the task of finding the global optimum of a function with several *local* optima, like the degree-18 legendre polynomial shown in Fig 1.3. Here it is apparent that the success of the hill-climbing strategy is dependent upon the hill-climber's starting location. If the computation is initiated at a point beyond the local vicinity of the central peak, it is clear that the iterative hill-climbing technique outlined above will converge on a local optimum. For a complex search space, finding a suitable starting location that allows for convergence on the global optimum may become extremely problematic, because one can never be certain if the solution obtained is indeed the desired result, or is instead only a local optimum.

Iterative Hill-Climbing

Iterative Hill-Climbing is a combination of hill-climbing and random search. Like conventional hill-climbing discussed above, this strategy suffers from the drawback of terminating on the first maximum it finds, regardless of whether it is local or global. Upon convergence, however, the procedure is reiterated at a different starting point, thus improving the odds of a global convergence, while the user keeps track of the maxima that have thus far been found. Again, this search is serial and is therefore carried out in local isolation, with the search having no knowledge of the overall terrain of the search space, or of other possible maxima, so a problem is knowing when to terminate the

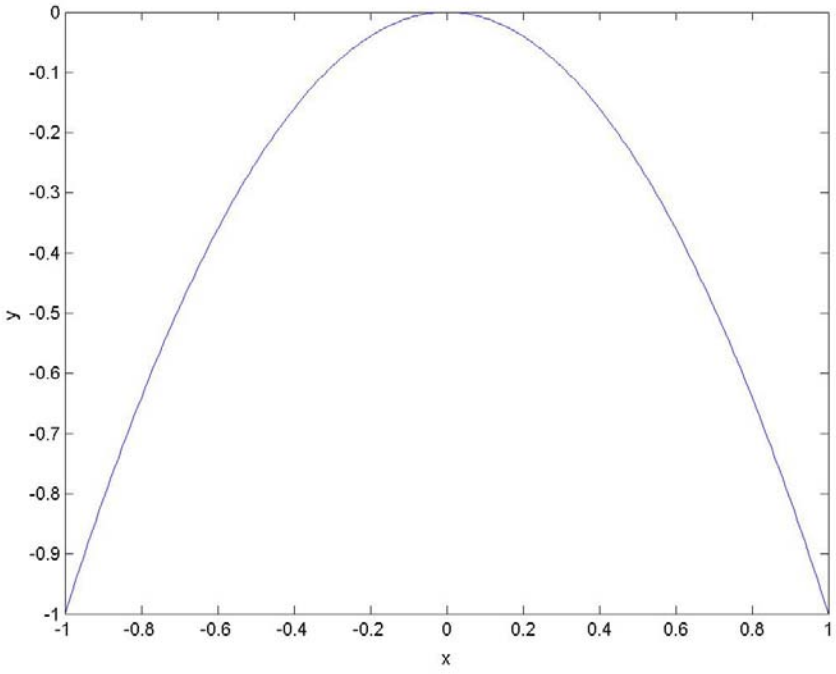

Fig 1.2 Upside-down parabola

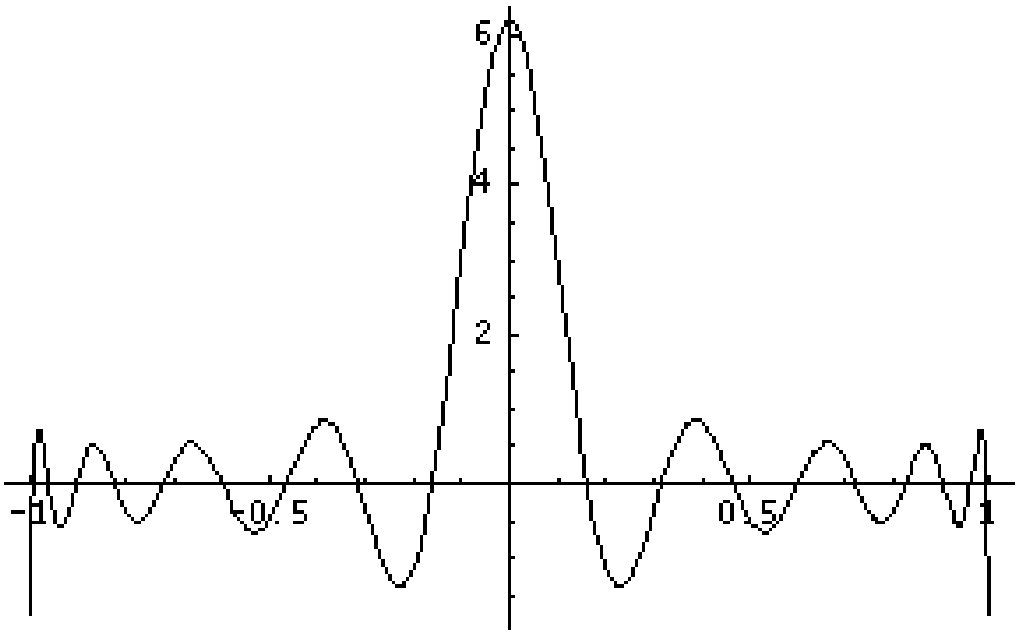

Fig. 1.3 18-degree legendre polynomial

Source: http://math.berkeley.edu/~mhaiman/mathH54/legendre.html

iterative search.

Simulated Annealing

This is a modified version of hill-climbing in which moves are weighted by probabilities, where uphill moves tend to be more probable than downhill ones. Starting from a random location, a move is made, with the probability of downhill moves constantly decreasing. As such, moves in any direction are favored initially, but as time goes on, uphill moves tend to be favored over downhill ones. This approach has the advantage over those discussed above in that, with simulated annealing, there is a possibility, with enough negative moves, of escaping local maxima. Of course, too many negative moves will lead the search away from the global optima. Like the other strategies discussed above, this method only analyzes one solution at a time and thus cannot build an overall picture of the surrounding landscape, and no information from previous moves is used to guide the search. This strategy has proved to be successful in many applications, like the VLSI circuit layout, for example [3].

The Simplex Method

This method is specifically designed to locate extrema in a multi-dimensional space, where each dimension corresponds to a variable defining a solution. Consider an *n*-dimensional space. A simplex is a geometrical figure with $n + 1$ vertices existing in this space. The simplex of a 2D space, for example, would be a triangle, for a 3D space, a tetrahedron, and so on. Given *n* variables, the location of each vertex of the simplex corresponds to a solution ($x_1, x_2, x_3 ... x_n$), as is illustrated in Fig. 1.4. The simplex has the

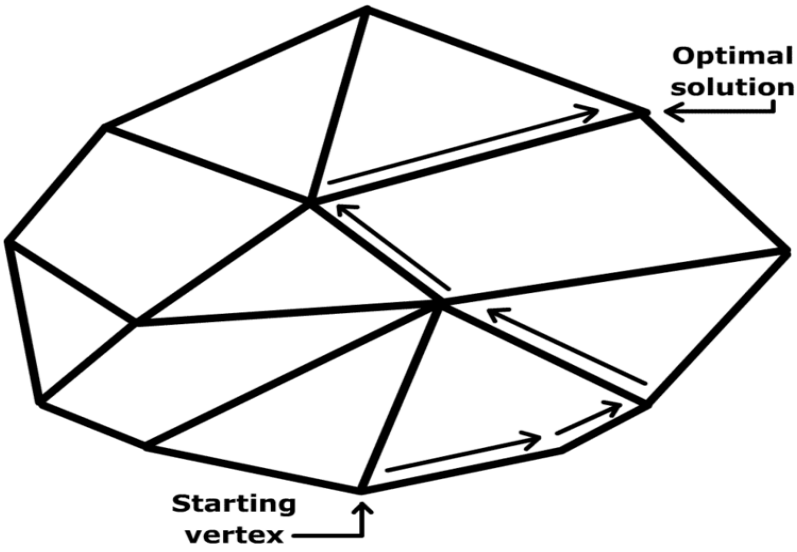

Fig. 1.4 Simplex

Source: http://en.wikipedia.org/wiki/Simplex_algorithm

ability to undergo three types of displacements: contraction, expansion, or reflection. In this way, the vertex with the worst solution is displaced, such that the function value corresponding to that vertex is increased by the move. Having gone through successive such moves in the complex *n*-dimensional topography, the method terminates when no other moves can be found that will increase the function value, and this final "resting place" of the simplex is taken to be the position of a maximum. But, as in the strategies discussed above, the simplex method lacks the ability to distinguish between local and global extrema.

The simplex method does not make uses of derivative information, but only requires function evaluations. Like the above hill-climbing methods, the initial coordinates of the vertices of the simplex often determine whether the global optimum is found [1]. Compared to gradient-based strategies, the simplex method is much slower. It is, however, more successful at navigating a complex search space that might thwart faster, hill-climbing searches, and thus has a better chance of achieving global optimization [1].

Chapter 2

Genetic Algorithms

2.1 An Overview

Genetic algorithms accomplish the optimization process in a manner analogous to biological evolution. This procedure is outlined below in six steps.

1) A random set of model parameter values is generated. This randomness guarantees that no initial bias is present. Each trial solution, corresponding to a specific set of parameter values, can be thought of as an individual in a population.
2) The fitness of each solution, or individual, is computed. This is usually done by a chi square measure of fitness with the data (although other methods of fitness could be employed), where the fitness is defined to be the inverse of chi squared.

$$\text{Fitness} = \frac{1}{x^2}$$

A minimum value of χ^2 will thus yield a maximum value of fitness. Because these parameters were produced in a random fashion, the fitness values of the population should vary over a considerable range. The genetic algorithm, then,

does not seek to maximize an objective function (a physics equation, for example), but a *fitness* function.

3) Pairs of solutions are then chosen from the population, where the probability of a particular solution being chosen is proportional to the solution's fitness. This is the computational equivalent of the evolutionary process of natural selection, where the more fit parents tend to reproduce more often. This is the driving force behind evolutionary development. On average, individuals better adapted to their environment tend to produce more offspring compared to "less fit" parents. Here, the measure of fitness that a solution has with a given data set obviously plays the role of biological fitness of an individual with respect to its environment, in that it is a measure of reproductive success.
4) From each pair of selected solutions, new solutions are produced. This reproduction process makes use of two operations: crossover and mutation, which are discussed in greater detail below. These new solutions form the next generation of model parameters.
5) Now, with natural selection acting on this new population, steps 2) through 5) are repeated. This tends to produce successive generations that have a higher fitness to the data. Hence, the solutions tend to "evolve" over time.
6) Termination occurs when the fitness of a solution in the current population exceeds some preset value, or when the computation has been carried out over a specified number of generations.

2.2 The "Breeding" Process

Encoding

As a prerequisite for the subsequent reproduction procedures, each parameter defining a solution is encoded as a string of integers. These are then spliced together, producing a 1-D integer array, or "chromosome." It is this array that undergoes the future operations of crossover and mutation.

As an example, consider the 2-D fitting problem, in which two parameters (x and y) are to be found which maximize the function f(x,y). If the fitted parameters have the following values,

$$(x,y) = (.12348, .71974)$$

then the resulting "chromosome" would be

1234871974.

Note that the decimals of the x and y values have been discarded in the encoding process.

Hence, each set of parameters defining a solution (in this case, the x and y values) can be considered a single chromosome, where each digit is a gene occupying a chromosomal site for which there exists ten possible alleles, or gene values. While it is not biologically accurate to regard a single chromosome as an individual in a population, this characterization will suffice for the purposes of putting the computational steps of the

genetic algorithm in a biological context. At best, these "individuals" can be thought of as possessing a single chromosome, making the two concepts thereby synonymous.

To fully appreciate the analogy with evolutionary biology, two important concepts are of interest here: phenotypes and genotypes. The phenotype can be regarded as the sum of an individual's observable traits; anything that makes up the observable characteristics and behavior of the organism. The genotype consists of the internally coded, genetic information of the individual. It is this information that is passed from one generation to the next via sexual reproduction, and hence is the underlying factor in determining an organism's inheritable physical traits. The encoding process can be regarded as the "uncovering" of an individual's genotype, given its phenotype. Using the above example, the string '1234871974' is the genotype of the phenotype (x,y).

Crossover

Crossover refers to the process in which genetic fragments of the population are used in defining a new generation of trial solutions. This is one of the major features that distinguish the genetic algorithm from other optimization methods. In a biological context, crossover is best regarded as a process that may occur in reproduction. For each site, or gene, in a chromosome, there exists a probability that a crossover operation will occur at that site. This consist of a fragmentation of the chromosome at the location of the site in question, and then the interchanging of fragments with another chromosome. The resulting new chromosomes are regarded as the offspring of the original two in the genetic algorithm.

As an example, consider the following encoded solutions.

1234871974 parent #1

3571092526 parent #2

At a randomly selected site, both parent chromosomes are cut and the resulting fragments are interchanged. If the cutting site were located at site seven, the crossover operation would look like this.

CUT: 123487 1974

357109 2526

SWAP: 123487 2526

357109 1974

SPLICE: 1234872526 offspring #1

3571091974 offspring #2

This process ensures that both offspring have "inherited" information from both parents, analogous to the reproduction of sexual species, in which complementary portions of genetic material are passed on to the new generation. For the sake of simplicity, the example illustrated here is of one-point crossover, where the splicing and recombination occurs at one site in the string. In general, genetic algorithms make use of one-point

crossover. Such is the case for the algorithm used in obtaining the results discussed in the following chapter. There are some genetic algorithms, however, that make use of two-point, or even higher orders of crossover, where the chromosome is cut in more than one location, resulting in a number of fragments. Note that, for one-point crossover, two offspring are produced. These two offspring replace their parents in the generation, so the population size remains a constant throughout the evolutionary run.

To see the significance of the crossover operation in optimization, consider the two parent solutions again, where the digits in bold are assumed, for sake of illustration, to represent those sites that tend to contribute to a greater-than-average fitness. Those not in bold are assumed to contribute little toward maximizing fitness.

123**48**719**7****4** parent #1

3571092**526** parent #2

Making note of the above offspring chromosomes, it is clear that offspring #1

123**48**7**2526**

is more fit than either parent or its sibling,

3**571**09**1**97**4,**

since it has inherited a greater number of those genes that convey a high fitness and thus will have a greater probability of being selected for breeding in the next iteration.

Hence, the crossover operation, combined with natural selection of trial solutions, ensures that each generation "more fit" than the last, meaning that each generation is better able to produce a more accurate model describing the given set of data.

Mutation

The next step in breeding is the application of the mutation operation to the offspring chromosomes. For each integer in the number sequence, there exist a small probability that a random number will replace the digit.

The following is an example of a mutation.

1234872526

12 4872526

1284872526

Note the change in digit value, from 3 to 8, that occurred at the third site from the left. This, of course, is analogous to genetic mutation, where random copying errors in gene values can occur in information being passed from parent chromosomes to offspring chromosomes. Depending upon which gene is affected, mutations can lead to small displacements in parameter space, or to large leaps that result in offspring trial solutions radically different from that of either parent.

The rate of mutation, and hence the probability that a chromosome may undergo a change in one of its digits, is allowed to vary throughout the evolutionary run. A similar process occurs in nature. Some bacterial species will undergo a phase of "hypermutation" when something disruptive occurs in its surrounding environment to threaten the survival of the species, while the mutation rate remains relatively low at other times. In this way, the odds of adapting to new circumstances, and hence survival, are increased in times of severe environmental stress [6]. Just as the rate of mutation in some living things varies in accordance with changes in their surroundings, the mutation rate of a population in a genetic algorithm continually adjusts itself to compensate for fitness (or lack thereof) between solutions and the data. The initial mutation rate is, in general, much less than the crossover probability, so as to ensure that mutation does not undermine any progress made by crossover in breeding more fit solutions.

Decoding

Decoding is simply the inverse of encoding. At this stage the offspring phenotypes are constructed from their corresponding genotypes, to be used in the next generational iteration. The phenotypes of the two offspring chromosomes given above would look like the following.

(x,y) = (.12848, .72526) offspring #1

(x,y) = (.35710, .91974) offspring #2

They have clearly made significant advancements in parameter space as compared to the parents.

(x,y) = (.12348, .71974) parent #1

(x,y) = (.35710, .92526) parent #2

2.3 A Brief History of Genetic Algorithms

Genetic Algorithms began to appear in the late 1950s and early 1960s. Their original use was in modeling natural evolution, but they were soon appreciated as optimization strategies for artificial problems. By 1962, researchers, such as G.J. Friedman and W.W. Bledso, to name a few, had begun to make use of genetic algorithms for optimization purposes. In 1965 a researcher, Ingo Rechenberg, of the Technical University of Berlin, introduced a technique that came to be known as an "evolution strategy." This method made no use of crossover or populations. Instead, a single solution underwent a mutation to produce one offspring. The more fit of the two solutions was kept, to be again subjected to a mutation. The idea of a population of solutions was introduced into later versions [6]. In 1966, L.J. Fogel, A.J. Owens and M.J. Walsh introduced the technique of "evolutionary programming." Like evolution strategies, this technique only made use of mutation. The primary difference between the two methods is the methods of selection. Evolution strategies make use of deterministic selection, based on a function evaluation, whereas the selection techniques employed in

evolutionary programming are probabilistic in nature and based on fitness, much like selection in genetic algorithms. Unlike genetic algorithms, where solutions are encoded during the breeding process, there is no constraint placed on the representation of solutions in evolution strategies or evolutionary programming. Another major distinction between these two methods and genetic algorithms is the manner in which mutation is employed. The mutation operation in evolutionary programming and strategies is weighted with a statistical distribution, with the probability of a change occurring being proportional to the magnitude of the variation. Small variations in the offspring are thus much more probable than substantial ones [7]. In short, the focus of evolution programming and strategies is placed on the behavioral linkage between parent and offspring solutions. There is no emulation of specific genetic operations found in nature, as is present in genetic algorithms [7].

In 1975, the publication of the book *Adaptation in Natural and Artificial Systems* brought about the wide recognition of genetic algorithms as problem solving strategies. This book built on earlier research by John Holland, who was the first to propose the use of crossover and mutation in explicitly mimicking biological evolution as a method of optimization [6]. Also introduced in this same book is the notion of schemata, in which individual solutions were encoded and thought of as being comprised of "building blocks," similar to how proteins are the "building blocks" of DNA chromosomes [6][8]. In the same year, the genetic algorithm's success in navigating complex, discontinuous, and other ill-behaved search spaces was firmly established in a dissertation by Kenneth De Jong [6][9]. By the mid-1980s, there was a widespread interest in genetic algorithms,

and they were being applied to a huge variety of problems, including pipeline flow control, structural optimization, and pattern recognition/classification [9].

Today, genetic algorithms are a thriving field of study, and are used to solve problems in a variety of fields, such as physics, astronomy, aerospace engineering, microchip design, biochemistry and molecular biology [6]. The range of problems that genetic algorithms can handle and "adapt" to is larger than that of any other optimization strategy.

2.4 Advantages and Disadvantages Over Conventional Methods

Genetic algorithms have proven to be successful at virtually any task that can be treated as an optimization problem, and anything described by an equation can be easily treated as such. Of course, for many problems, genetic algorithms may not always be the most efficient way of finding the desired solution(s). This is one form of what has come to be known in the field of evolutionary algorithms as the exploitation vs. exploration dilemma. Exploration and exploitation are both ideal elements of any evolutionary search technique: exploration consists of selecting a large number of diverse solutions, while exploitation builds on those solutions that have been found to have high fitness. In this way, both work together to form a picture of the overall landscape to ensure with great certainty that the optimum found is the global one. So while the application of a genetic algorithm to relatively simple problems will no doubt yield the optimal solution, it may also mean a waste of computational effort when the solution could be obtained by much simpler means, like those of gradient-based methods, for example. It has been

found that analytically solvable problems are usually best done with traditional methods, which are less time consuming and are guaranteed to give an exact solution, unlike genetic algorithms [6]. In other words, applying genetic algorithms to relatively simple tasks could result in more exploitation than what is necessary.

So the question becomes, when is the use of a genetic algorithm called for? Charbonneau lists a few instances where a genetic algorithm might excel where other search techniques fail [6].

(1) Multimodal problems in which one cannot make a reliable guess as to the location of the global optimum.

(2) Problems in which derivatives are extremely difficult or impossible to compute.

(3) Problems which are ill-conditioned, those described by integral equations, for example.

The genetic algorithm, as opposed to conventional methods, is a *global* optimization strategy. This is due to the effectiveness of crossover and mutation. Crossover is the main feature that sets genetic algorithms apart from conventional optimization methods, and without it, the strategy would reduce to that of parallel random searches. Unlike the methods discussed in chapter one, where an individual is confined to searching the space in its local vicinity, the crossover operation in genetic algorithms

allow for the exchange of information between individuals separated by large distances in search spaces. Individuals in one region of the space can therefore benefit from and improve upon what has been learned by individuals elsewhere. In this way, crossover insures a combining of individuals of high fitness, and the possibility that the resulting solutions will have "inherited" the strengths of both parents.

Mutation of offspring solutions is a great advantage in that genetic algorithms, as opposed to conventional numerical optimization methods, are less likely to terminate at local optima, and are better able to navigate complex, even discontinuous fitness landscapes. In regions surrounding local optima, crossover operations do little to further maximize the fitness of parameters because, once a population has converged on a local maximum, segments are being exchanged that are nearly identical. It is here that the significance of the mutation operator becomes obvious. Mutation allows for the production of offspring trail solutions with "genetic segments" that vary from that of the parents to such a degree that these solutions are not confined to local optima. The genetic algorithm can thus make large leaps in the search space even after partial convergence on a local maximum.

This inherent stochasticity of the genetic algorithm can, however, be a two-edged sword. While the probabilistic nature of the genetic algorithm tends to drive the search toward the global optimum, convergence on a global optimum cannot be guaranteed for the same reason. It is important to note that, while genetic algorithms are exceptionally good at finding the global optima in a fitness space, they do not operate with this specific goal. Like evolution, genetic algorithms are inductive. In the natural world, evolution does not have a particular goal of maximizing fitness, but only to evolve away from less-

fit circumstances. And just as the development of a species can terminate at an evolutionary dead end, there is also the possibility of a few solutions coming to dominate the populate, resulting in the genetic algorithm converging on sub-optimal solutions [11]. This is more likely to occur in small populations, where reproductive dominance of a few well-fit individuals prematurely drives down diversity before global optimization is obtained [11]. Genetic algorithms are, however, generally successful at finding "very good" solutions to a problem, if not the optimal solution. Techniques custom designed for solving particular problems, however, are likely to outperform the genetic algorithm in terms of speed and accuracy. Yet, even in cases such as these, improvements have often been made by hybridizing the existing technique with a genetic algorithm [2].

Ironically, one of the biggest advantages of the genetic algorithm would at first glance appear to be a drawback: the genetic algorithm is "blind." Genetic algorithms know nothing about the problems they are being applied to. Instead of relying on specific information about a problem, as do many other search techniques, a fitness function is employed to ascertain whether the random changes resulting from crossover and mutation have made improvements to the overall fitness with data to a theoretical model. In this way, genetic algorithms are not hampered by the user's preconceived notions about a problem, but are concerned only with finding optimal solutions, even if those solutions run contrary to expectations. The disinterested genetic algorithm, therefore, has advantages over conventional methods in that it can explore the fitness landscape without preconceived bias and is thus prone to discovering new and radical solutions that could not have been predicted by *a priori* means. Also, with the solutions being judged solely on a fitness level, local optima are not distinguished from other equally fit points in the

solution space. This ensures the continuation of the iterative breeding process. Points closer to the global optimum will have greater fitness values, and each generation improves upon the fitness of solutions until a convergence criterion is met [10].

It is evident from the above discussion that the fecundity of the crossover and mutation operations employed by genetic algorithms in finding global optima is at its greatest when dealing with populations of diverse solutions. In populations of large diversity, local maxima are not likely to be mistaken for global optima. In less diverse populations, the genetic algorithm can offer little in advantage over conventional methods. This is because the offspring solutions would vary little from their parents, making what would otherwise be advantages of the genetic algorithm redundant.

Another advantage of the genetic algorithm is its ability to search the parameter space in many directions simultaneously. While other heuristic methods perform iterations on a single solution, genetic algorithms make use of an entire population. Unlike most other optimization algorithms, which are serial and confined to exploring the space in one direction at a time, the multiple offspring of the genetic algorithm can explore a large number of regions at once, increasing the odds of a convergence upon the global optimum. Genetic algorithms are thus ideal for problems involving vast search spaces, problems that would be very time-consuming for conventional hill-climbing strategies.

Another advantage of this intrinsic parallelism of the genetic algorithm is its ability to not only evaluate the fitness of each individual, but also to sample all subspaces to which the solution belongs. A subspace can be thought to consist of a group of individuals that share a common gene or set of genes. For example, consider the search

space formed by all possible eight-digit strings, ********, where the * can take on any integer between 0 and 9. Thus, the string 37956812 would be member of this space, but would also be a member of the subspace 3*******, the subspace 37******, the subspace 3******2, and so on. After many iterations, the algorithm can ascertain the average fitness associated with each subspace, thereby making judgments about the many individuals that are members of the space. In this way, it can implicitly evaluate large group of individuals by explicitly evaluating a select few, much like a pollster hopes to learn something about the thoughts and opinions of an ethnic, religious, or social group by sampling a small percentage of the population [6]. After many evaluations, the genetic algorithm can thus "pinpoint" the search space containing the individuals of greatest fitness. This is known as the Schema Theorem in the literature of evolutionary algorithms, and is regarded as the "central advantage" of the genetic algorithm over other optimization methods [6].

There exist, however, "deceptive" search spaces in which genetic algorithms can be thwarted. In such spaces, improvements give misleading information on where the global optimum is to be found [6]. The genetic algorithm operates with the underlying assumption that improved points reveal a neighborhood that is likely to harbor the global optimum. Hence, regions of high-fitness tend to be explored at the expense of regions of low fitness. It is thus easy to imagine how a genetic algorithm might be unsuccessful at locating an optimum that is surrounded on all sides by regions of low fitness. Such a function would be extremely difficult to optimize by any means, and iterative hill-climbing usually wins out over genetic algorithms in such circumstances [2]. But like

natural evolution, genetic algorithms tend to make the best of whatever circumstance it is exposed to and, in such a hypothetical situation, it can at least deliver a fairly impressive solution through convergence on a local maximum. It should also be noted that few problems are as deceptive as the one presented here, and that the location of local improvements usually contribute, to *some* degree, to the discovery of the global optimum's location [6].

The performance of genetic algorithms in finding solutions to problems is often highly sensitive to the values of several parameters. These include the following.

Population size

Number of generations through which the solution is to evolve

Number of significant digits retained in a chromosomal string

Crossover probability

Mutation rate

These variables influence greatly the genetic algorithm's potential for finding optimal solutions and often require a "fine-tuning" by the user for optimal performance. For example, a small population may not allow for a sufficient exploration of the fitness landscape, and hence is unlikely to stumble across the optimal solution. Likewise, if the

rate of genetic change is too high, the algorithm's likelihood of converging may be compromised. Too great of a mutation rate could undermine any progress brought about by crossover, and the search essentially becomes a random one. On the other hand, if the rate of mutation is too low, the risk of terminating on local maxima is increased. The problem, then, is to find a set of parameters that strikes a balance between exploration and exploitation of the solution space. The combination of values required to yield an effectively good solution is not problem-specific, and so finding the choice of the most suitable parameter values for a specific problem becomes an optimization task in and of itself. Evolution has faced similar difficulties in nature. Drastic environmental changes that significantly alter a population size, mutation rates, etc, can result in an extinction of a species [6]. Finding a suitable choice of input parameters can be the biggest obstacle confronted by the user. And due to the stochastic nature of the genetic algorithm, there is no guarantee that a good result can be recreated with the same set of parameters. A parameter set that successfully yields the global optimum on one evolutionary run could produce drastically different results after in a subsequent run for the same problem.

In summary, the four major advantages of genetic algorithms over conventional methods are parallelism, selection, crossover, and mutation. While the individual implementation of any one of these functions would result in only slight improvements to a problem, it is the *combination* of these four operations that give genetic algorithms their power and success at finding global optimums. Beginning with a population of diverse individuals, crossover and mutation allow for an exhaustive search of the space, with selection driving the individuals of each generation toward more promising regions. But by that same token, the unpredictability inherent in the crossover and mutation operations

cannot guarantee global convergence. There is always a risk of sub-optimal convergence, and if one already has specific knowledge of a problem that can help in guiding the search such as the approximate location of the global optimum other techniques are likely to outperform the genetic algorithm. Genetic algorithms tend to be the most effective for complex, multimodal problems involving complex search spaces, outperforming conventional methods in both speed and accuracy.

Chapter 3

Applications

The following is a number of problems from physics and astronomy, and a genetic algorithm's treatment of them. These problems were solved with the genetic algorithm, Pikaia, a general purpose function optimization FORTRAN subroutine. The subroutine can be accessed from the Pikaia homepage [1].

https://www.hao.ucar.edu/Public/models/pikaia/pikaia.html

3.1 Computing the Hubble Constant

From observing the red-shifts of spectra from distant galaxies, the astronomer Edwin Hubble was the first to make note of the expansion of the universe. His observations served to formulate what is today known as Hubble's Law, which states that the velocity, v, at which two galaxies recede from one another is proportional to the distance, D, between them via Hubble's constant, H.

$$v = HD.$$

This relationship, illustrated in Figure 3.1, suggests that the expansion of space is uniform on large scales, where mutual gravitational attraction between galaxies is negligible.

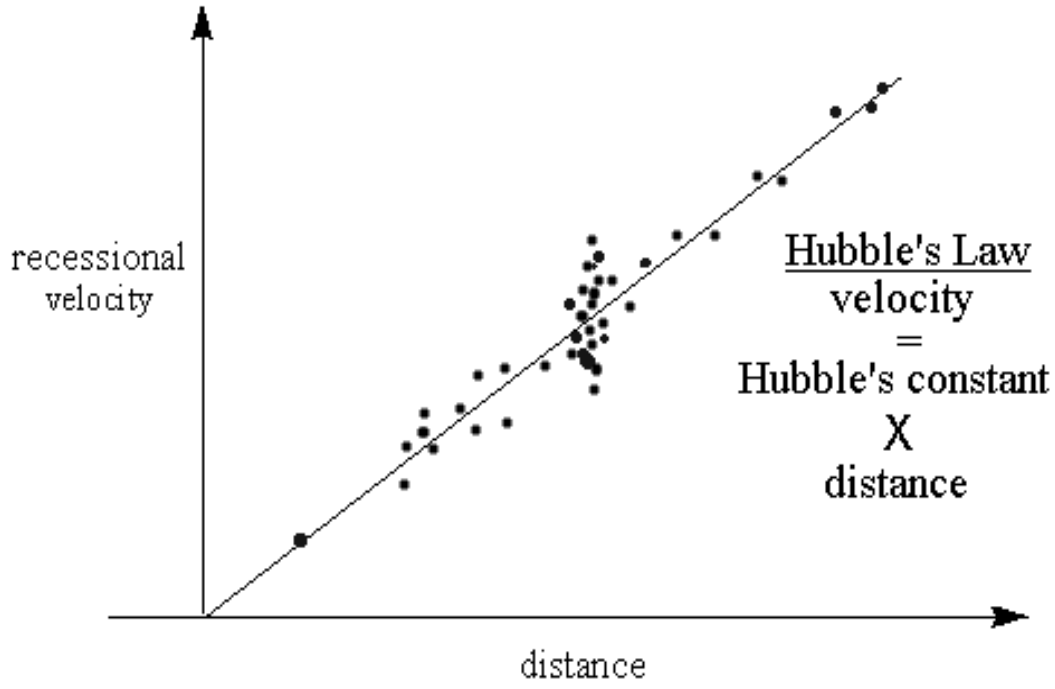

Fig. 3.1 Hubble's Law. The slope of the above line is the Hubble constant, the proportionality constant that relates the distance of receding galaxies to their velocities

Source: http://astrosun2.astro.cornell.edu/academics/courses//astro201/hubbles_law.htm

While there is some degree of uncertainty in the value of *H*, it is thought to be in the range of 45-90 km/sec/Mpc [12], with the best current data suggesting a value near 70 km/sec/Mpc. In this example, the best-fit value for the Hubble constant is found using data taken from a sampling of twelve galaxies. Since *v* and *D* have a linear relationship, the problem simply consists of finding the best-fit slope of a line, given the x-data and y-data (the distances and velocities, respectively). However, the observation data for this problem aren't *v* and *D*, so additional calculations must be carried out in order to obtain values for recessional velocity and distance, which are to be used in the least-squares merit function. Table 3.1 shows twelve galaxies and lists their apparent visual magnitudes. These particular galaxies have a common absolute magnitude of –22. From the data in Table 3.1 the distance to each of these galaxies, in units of parsecs, can be obtained.

(1) $$D = 10^{(m-M+5)/5}$$

The quantity, *M*, in equation (1) denotes absolute magnitude. The corresponding recessional velocities were obtained in the following manner. For each of the above twelve galaxies, the wavelengths of two specific spectral lines, the Ca K and H lines, were measured. Table 3.2 lists these values in angstroms.

Light leaving a receding object is shifted toward the red end of the visible spectrum by the expansion of space. For sufficiently small redshifts, $\Delta\lambda$,

Table 3.1 Galaxies and their Apparent Visual Magnitudes

Object	m
Uma1-2	14.7
Uma1-3	14.6
Uma1-1	14.5
CrBor2	15.5
CrBor1	15.4
Boot2	16.8
Boot3	16.7
Coma2	12.5
Coma3	12.7
Uma2-1	16.6
Uma2-3	16.8
Uma2-2	16.8

Source: K. Duckett, *A Laboratory Textbook for Introductory Astronomy*, Fifth Edition, Contemporary, (1998)

Table 3.2 Galaxies and their corresponding Ca K and H spectral lines

Object	λ_k	λ_h
Uma1-2	4134	4172
Uma1-3	4134	4170
Uma1-1	4136	4172
CrBor2	4216	4254
CrBor1	4218	4256
Boot2	4452	4492
Boot3	4452	4488
Coma2	4024	4058
Coma3	4022	4058
Uma2-1	4472	4514
Uma2-3	4472	4510
Uma2-2	4476	4512

Source: K. Duckett, *A Laboratory Textbook for Introductory Astronomy*, Fifth Edition, Contemporary, (1998)

(2) $$v = \frac{\Delta\lambda}{\lambda_0} c,$$

where c is the speed of light, and λ_o denotes the stationary wavelength, measured in Angstroms. For the Ca K and H lines, these are 3933.67 and 3968.847 Angstroms, respectively. For each galaxy in Table 3.2, equation (2) is used in computing the two recessional velocities associated with each spectral line. The accepted velocities used in obtaining the Hubble constant are taken as the average of these two values.

(3) $$\mathbf{V} = \frac{v_K + v_H}{2}$$

The Pikaia subroutine contains twelve adjustable parameters. These, and the default values are listed in Fig 3.2. For this particular problem, all elements of the *ctrl* array are kept at their default values except the first two. The population number was set to fifty individuals, and the number of generations was given a value of one hundred. As one might expect, the probability of a successful convergence upon an optimal solution generally is highly dependent on the values of these two input parameters. Of course, the inherent randomness involved in the algorithm's exploration and exploitation of a search space does not always guarantee that larger populations and generation counts will yield more fit solutions than smaller ones. For simple problems like this one, however, the returned solution does not seem to vary significantly with changes in input parameter values, especially those of population size and number of generations. Repeated executions of the evolutionary run with increasingly greater values of the two

```
ctrl( 1) - number of individuals in a population (default
           is 100)

ctrl( 2) - number of generations over which solution is
           to evolve (default is 500)

ctrl( 3) - number of significant digits (i.e., number of
           genes) retained in chromosomal encoding (default
           is 6)

ctrl( 4) - crossover probability; must be  <= 1.0 (default
           is 0.85).

ctrl( 5) - mutation mode; 1/2/3/4/5 (default is 2)

ctrl( 6) - initial mutation rate; should be small (default
           is 0.005) (Note: the mutation rate is the proba-
           bility that any one gene locus will mutate in
           any one generation.)

ctrl( 7) - minimum mutation rate; must be >= 0.0 (default
           is 0.0005)

ctrl( 8) - maximum mutation rate; must be <= 1.0 (default
           is 0.25)

ctrl( 9) - relative fitness differential; range from 0
           (none) to 1 (maximum).  (default is 1.)

ctrl(10) - reproduction plan; 1/2/3=Full generational
           replacement/Steady-state-replace-random/Steady-
           state-replace-worst (default is 3)

ctrl(11) - elitism flag; 0/1=off/on (default is 0)
           (Applies only to reproduction plans 1 and 2)

ctrl(12) - printed output 0/1/2=None/Minimal/Verbose
           (default is 0)
```

Fig. 3.2 Control parameters in the genetic algorithm program Pikaia

Source: https://www.hao.ucar.edu/Public/models/pikaia/pikaia.html

aforementioned parameters reveals a convergence in the fitness of the returned solution, where greater values offer no improvement.

With the distance and velocity values obtained from equations (1) and (3), a chi-square measure of fitness is performed using the obtained distance and velocity values.

$$\chi^2 = \left(\frac{HD - V}{\sigma} \right)^2$$

This is carried out in a separate subroutine as shown below.

```
sum=0.
do i=1,numb
sum = sum + ((H*D(i) - V(i))/err(i))**2
End do
```

The integer *numb* denotes the number of data points, in this case, fifteen. The arrays D(i) and V(i) contain the fifteen values of the distances and velocities, respectively, and err(i) denote the error estimates of each data point, assumed here to be constant. The fitness is then defined as the inverse of the sum.

$$\boldsymbol{fit} = \frac{1}{sum}$$

The solution to be returned by the algorithm is defined in the calling program as an array, $x(n)$, bounded in the region [0,1]. The argument n is the number of parameters defining a

a solution. For the present problem of finding the Hubble constant, only one parameter is being fitted, and hence $n = 1$. In general, for an n-dimensional array, $x(n)$, the search space in any one dimension can be extended by multiplication with a real number, M.

$$A = x(n_i)*M$$

For the i'th dimension, then, the search for a global optimum is confined to the interval [0,M], with A denoting the normalized i'th parameter defining a solution. So while it is the elements of the array $x(n)$ that are encoded and manipulated by the operations of the "breeding" process, the fitness measurements require them to be properly rescaled. In order that the returned solution be of the right order of magnitude, the variable H is expressed in the fitness subroutine in terms of the one-dimensional array $x(1)$.

$$H = x(1)*100$$

The best-fit value of H was found to be

$$H = 72.029 \text{ km/s/Mpc}$$

Note that a solution expressed in these units requires the distances computed in eqn (1) be converted from parsecs to mega-parsecs before the fitting procedure is performed.

3.2 Computation of Gravitational Freefall Acceleration Near the Earth's Surface

In the following example, the freefall acceleration of gravity is computed by means of the genetic algorithm, and the results compared with that yielded by a more conventional search technique. Unlike the previous example, which was a linear least-squares fitting problem, the least-squares problem given here is *non*-linear, meaning that the given data is not related in a linear manner. This example illustrates nicely how the fitting procedure carried out is not sensitive to the manner in which the model parameters are related to the given data.

Consider a pendulum of length L, consisting of a string with a small spherical mass attached to the end. If the mass of the string is negligible, and the oscillations are relatively small, the motion of the attached mass can be approximated as simple harmonic motion and is governed by the equation

$$\frac{d^2\theta}{dt^2} = \omega^2\theta \tag{1}$$

where θ is the angle of displacement from equilibrium, ω is defined as

$$\omega = \sqrt{\frac{g}{L}}$$

and g is the gravitational freefall acceleration. The period, the time of one oscillation, is a function of L only and is given by

(2) $$T = 2\pi\sqrt{L/g}\,.$$

Note that this model has a non-linear dependence on g.

Table 3.3 displays measurements made on a simple pendulum. It lists the recorded periods, as well as the corresponding lengths associated with each period value. The data were collected in the following manner. For each length value listed above, the time interval for ten oscillations was recorded. This interval was then divided by ten to yield an average value of the period for that particular length. These averages were recorded as the period values displayed in the table. After each recording of the period for a particular length, the length was altered and the process repeated again. In all, fifteen measurements were made.

Using these data, and equation (2), the value of g is computed, again using a least-squares fit. Again, the estimated error associated with each point is assumed to be a constant.

The returned value of the freefall acceleration for this problem is

g = 9.86480 meters/square second

Because the lengths given in the above data table are expressed in millimeters, a

Table 3.3 Oscillation Periods of a Pendulum and the Corresponding Lengths

Period (s)	L (mm)
1.212	37
1.021	25.6
1.172	33.7
1.25	39
1.194	35
1.112	31
1.006	27.7
0.863	18.4
0.794	15.7
1.169	34
1.887	91
1.678	70
1.547	1.547
1.456	1.456
1.337	1.337

Source: Source: K. Duckett, *A Laboratory Textbook for Introductory Astronomy*, Fifth Edition, Contemporary, 1998

conversion of units is carried out in the code so as to yield the free-fall acceleration in its familiar form expressed in SI units.

This result is now compared with another search method known as a "golden section" search [15]. This procedure is specifically designed to locate the *minima* of functions, rather than optima. The difference between minimization and optimization strategies is a trivial one, however. Recall that the genetic algorithm, in the above problem is, defined fitness as

$$\text{Fitness} = \frac{1}{x^2}.$$

So while the problems that genetic algorithms are applied to are usually referred to as optimization problems, they can also be regarded as ones of minimization, for while the algorithm searches for the global optima of a fitness space, it is doing so by finding a *minimal* value of chi-squared. This is precisely the quantity to be minimized by the golden section search, where

$$\chi^2 = \left(\frac{2\pi\sqrt{\frac{L}{g}} - T}{\sigma} \right)^2. \tag{3}$$

With the above data supplying the values of L and T, the above formula is a function of g

only. The golden section strategy therefore plays the same role in this problem as did the genetic algorithm; namely, to find the value of *g* that minimizes χ^2. Letting the data point (L_i, T_i) define a χ_i^2 via equation (3), the quantity being minimized in this problem is the average χ^2.

$$\chi^2 = \frac{\sum_i^n \chi_i^2}{n}$$

The golden section search accomplishes this task in a manner analogous to that of the root-finding method of bisection, where a root is bracketed on an interval (a,b) and the function is evaluated at some intermediate point *x*, at which point the interval is replaced with either (a,x) or (x,b). The process is reiterated until a termination criterion is satisfied, usually when the length of the bracketed interval reaches a preset value. There is, however, a fundamental difference between the method of bisection and its minimization counterpart, and this is the bracketing process. While bisection, as a root-finding strategy, requires the functional evaluations of two points, bracketing a minimum requires the evaluations of three, where that of the intermediate point is less than those of the other two. For example, consider the bracketed function shown in Fig. 3.3. The bracketing triplet is initially (*a*,*b*,*c*), where

$$f(\mathrm{b}) < f(a) \qquad \text{and} \qquad f(\mathrm{b}) < f(c).$$

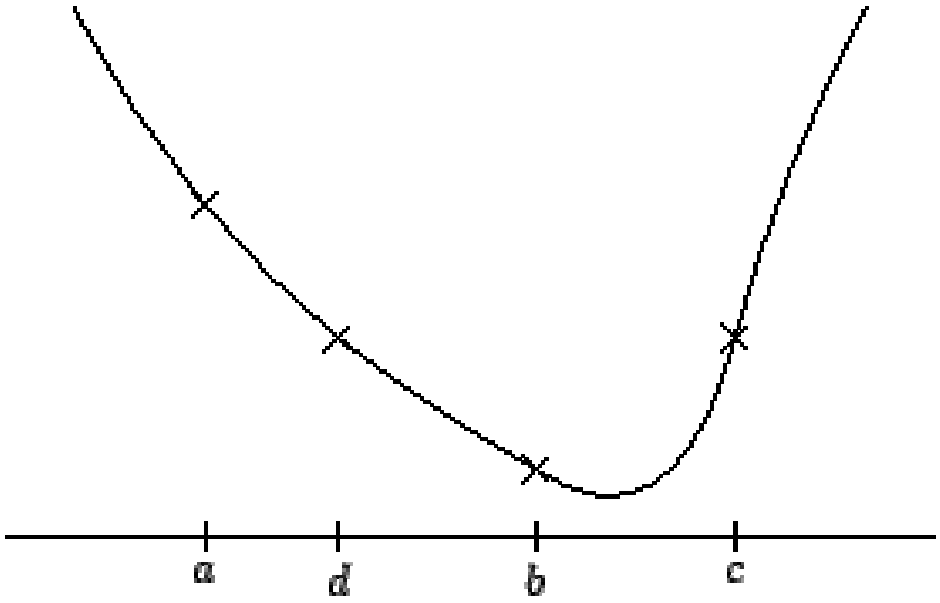

Fig. 3.3 A bracketed minimum

Source: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/BMVA96Tut/node17.html

The function is evaluated at a new point, d, in the interval (a,b). If

$$f(d) > f(b)$$

as in Fig. 3.3, then the bracketing interval (a,b,c) is replaced with (d,b,c). If, however,

$$f(d) < f(b)$$

Then the new bracketing triplet is (a,d,b). This bracketing process is iterated until the distance between the outer points falls below a preset tolerance value.

The method of choosing new values of x is based on the idea of "golden ratios," hence the search name "golden section." As Fig. 3.4 illustrates, two numbers, a and b, are said to be in the golden ratio if the ratio of the larger number to the smaller one is equivalent to that of sum to the larger[23]. This relation takes the following mathematical form.

(4) $$\frac{a}{b} = \frac{a+b}{a}$$

When equation (4) is multiplied by $\frac{a}{b}$, one obtains

(5) $$u^2 = u + 1 ,$$

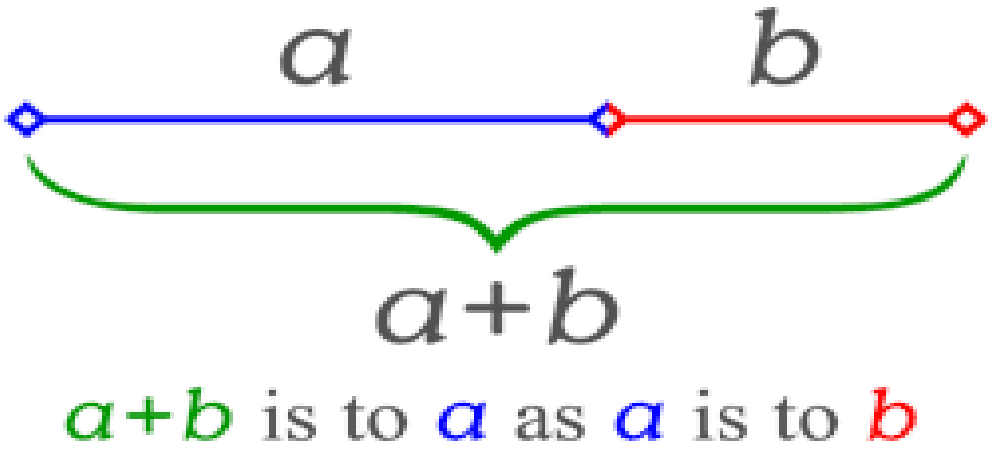

Fig. 3.4 The golden ratio

Source: http://en.wikipedia.org/wiki/Golden_ratio

where $u = \frac{a}{b}$. The roots of equation (5) are known as the golden ratios. So, in algebraic terms, a golden ratio can be defined as a number, the square of which is equal to itself plus one. In the bracketed region, (a,b,c), of the golden search, the middle point, *b*, is chosen so that its distance from one end is .38197 times the total length of the interval, and its fractional distance from the other end is .61803. Note that the ratio of the latter number to the former is approximately 1.61803, the positive root of equation (5). In general, given a bracketing triplet of points, the next number to be evaluated is located at a fraction of .38197 into the larger of the two regions. Hence, each successive bracketing interval decreased by a factor of .61803, ensuring that the distances from the point to both ends of the bracketed region are the self-replicating golden ratios. If the routine is initiated with a bracketing triplet, the segments of which are *not* of the golden ratios, the above procedure for choosing a new point in the larger segment will rapidly reach a convergence, upon which the golden ratios are replicated.

The minimum was found to exist at

g = 9.86484 meters/square second

Note how close this value comes to that given by the genetic algorithm. The two solutions differ by only .00043%, thus demonstrating the genetic algorithm's ability to perform simple optimization/minimization problems of this type. Note also that the golden section search required an initial guess as to the approximate location of the minimum (the search requires that a minimum already be known to exist within the

bracketed interval). The genetic algorithm's independence of prior knowledge of the problem and its insensitivity to its starting location are clearly advantages when dealing with problems in which prior knowledge is lacking.

3.3 The Computation of the Rest Mass of the Electron

Unlike the previous two cases, this next problem is one in which two parameters are being optimized. The genetic algorithm is hence finding the maximum of a function in a 2-D search space, with the returned solution being an array with *two* components. Hence $n = 2$ and each parameter to be fitted serves as a component of the array $x(n)$. The best-fit parameters returned in this problem are the rest mass of the electron and the energy of a gamma ray photon emitted from ^{137}Cs, a radioactive isotope of Cesium.

Compton scattering is an elastic collision between a photon and an electron. A photon strikes an atom and imparts some of its energy to an electron, causing it to recoil. The interaction is illustrated in Figure 3.5. Experimentally, this transfer of energy is evident from the observation that the light after the collision is of a wavelength different than that before the interaction.

Conservation of energy states that

$$E_\gamma = E_\gamma' + E_e \tag{1}$$

where E_γ is the energy of the incoming photon, E_γ' is the energy of the scattered

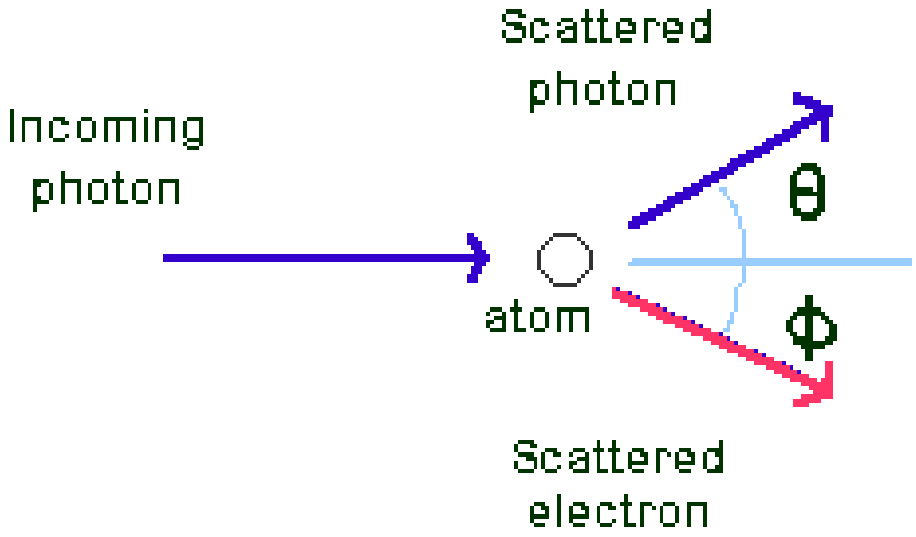

Fig. 3.5 Compton Scattering

Source: NASA's 'Imagine the Universe
http://imagine.gsfc.nasa.gov/docs/science/how_l2/compton_scatter.html

photon and E_e is the energy of the scattered electron. Conservation of linear momentum gives

(2) $$\frac{hf}{c} = \frac{hf'}{c}\cos\theta + mv\cos\phi$$

and

(3) $$\frac{hf'}{c}\sin\theta - mv\sin\phi = 0$$

for the *x* and *y* directions respectively. Here f is the frequency of the incoming photon, f' is the frequency of the scattered photon, h is Plank's constant, and mv is the momentum of the scattered electron. From the above three equations, one can obtain the following formula, where the energy of the scattered photon is a function of initial photon energy and θ,

(4) $$\frac{E'_\gamma}{E_\gamma} = \frac{1}{1 + \varepsilon(1 - \cos\Theta_\gamma)} \quad ,$$

where

$$\varepsilon = E_\gamma / m_e c^2$$

and m_e designates the rest mass of the electron, and is related to the mass of the scattered electron, *m*, by

$$m = \frac{m_e}{\sqrt{1 - \frac{v^2}{c^2}}}$$

where *v* is the recoil velocity.

Table 3.4 shows data taken during a Compton scattering experiment, with ^{137}Cs as the source of the incident photons. The left column contains θ values in radians and the right column lists the corresponding scattered photon energies, expressed in units of mega-electron volts. With this data as input, equation (4) is used in the least-squares fitting problem to find the optimal fitness of the trail solutions for the electron rest mass and E_γ , the energy of a ^{137}Cs gamma ray.

The best-fit values of the photon energy and electron rest mass were found to be

$$E_\gamma = 0.662 MeV$$
$$m_e = 0.512 MeV / c^2 ,$$

Table 3.4 Scattered Photon Angles and Incident Energies

Theta	E
0	0.662
0.1745	0.649
0.349	0.613
0.5236	0.565
0.6981	0.508
0.8726	0.455
1.0471	0.402
1.2216	0.358
1.3962	0.321
1.5707	0.289
1.7452	0.263
1.92	0.242
2.0942	0.225
2.2688	0.212

Source: EG&G ORTEC, *Experiments in Nuclear Science AN34 Laboratory Manual*, Third Edition, 1984

both of which are in close agreement with the accepted values of .662 MeV and .511 MeV/c^2, respectively.

3.4 Computation of the Mass of Saturn

Kepler's third law of planetary motion states that the square of the period, *P*, of a body in a bound orbit around a central mass, *M*, is proportional to the cube of the semi-major axis of the orbit. If the orbit is circular, and the central mass is very large compared to that of the orbiting body, Kepler's third law takes the following form.

$$r^3 = \frac{GM}{4\pi^2} P^2 , \tag{1}$$

Here, *r* is the orbital radius and *G* is the gravitational constant. When the following expression,

$$P = \frac{2\pi r}{v} \tag{2}$$

is inserted in equation (1), the mass of the central body can be expressed as,

$$M = \frac{v^2 r}{G} , \tag{3}$$

where *v* is the velocity of the orbiting body.

In this problem the mass of Saturn is computed using this formula, with Pikaia supplying the best-fit value of *v*, the orbital velocity of the rings, which are treated as being composed of gravitationally bound particles in circular Keplerian orbits about the planet.

Due to the rotation of the rings along our line of sight, the light reflected from the rings on one side of Saturn is red-shifted, while the light coming from the opposite side is blue-shifted. Table 3.5 lists the wavelengths, in angstroms, of light coming from five different points along the plane of the rings, where each row denotes a different radial distance of the ring particles from Saturn. These data were obtained from spectra taken by astronomers at the Lick Observatory [13]. These spectra were observed at an observation angle that guarantees Doppler shifts resulting from direct recession or approach of the ring material. The inclination angle of the plane of the rings with respect to that of the sky is thus taken to be 90 degrees, making the tangential velocities of the ring particles equivalent to the orbital velocities that are to be used in equation (3). The line of observation is along Saturn's equator. Using this data, the Doppler shifts corresponding to each radial point can be calculated, and thereby the velocity determined using the following equation.

$$v = \frac{\Delta\lambda}{4\lambda_o} \tag{4}$$

Table 3.5 Wavelengths of Light Reflected From Saturn's Rings

λ_R	λ_B	λ_0
6219.23	6218.39	6218.67
6230.66	6229.54	6229.82
6252.40	6251.28	6251.84
6264.10	6263.26	6263.54
6269.95	6269.11	6269.39

Source: K. Duckett, *A Laboratory Textbook for Introductory Astronomy*, Fifth Edition, Contemporary, 1998

The first two columns in the above table correspond to opposite sides of the planet. The left column designates the wavelengths of the red-shifted light, while the middle column denotes that of the blue-shifted light. The last column, λ_0, lists the wavelengths of light coming from the region of the rings in which the tangential velocities are perpendicular to our line of sight, and therefore does not experience a Doppler shift. Each row corresponds to a radial distance of ring material from Saturn.

where

$$\Delta\lambda = \lambda_R - \lambda_B \,.$$

The factor of four in the denominator is included as a correction factor. This correction is due to two reasons. First, a factor of two must be included to correct for the fact that the light being observed is not emitted from Saturn's rings, but *reflected.* Also, an additional factor of two is added due to the fact that $\Delta\lambda$ does not denote the usual difference between a shifted wavelength and one of light coming from a stationary source. Instead, it is the *difference* between the red-shifted and blue-shifted light coming from opposite sides of the rings that are used here.

From the above data table and equation (4) used in measuring fitness, Pikaia arrives at a best-fit value for the average velocity of the orbiting rings.

V = 14083.000 meters/second

Since the width of the rings is much smaller than the distances separating Saturn's center of mass from the inner and outer edges of the rings, a mean orbital radius is assumed, and has the following value.

r = 1.2 x 10^8 meters

When this value, along with the returned value of the mean orbital velocity, is placed in equation (3), one obtains the following value for Saturn's mass.

$$\text{Mass of Saturn} = 6.572\, x\ 10^{56}\ \text{kg},$$

This is in good agreement with the accepted value of $5.7\, x 10^{56}$ kg. The percent error is approximately 15%.

3.5 Computation of the Distance to the Small Magellanic Cloud

In this next problem Pikaia is used in the computation of the distance to the Small Magellanic Cloud (hereafter SMC), a satellite galaxy of the Milky Way, using data gathered from observations on Cepheid variables [13].

Cepheid variables are a class of stars that experience a periodic change in apparent magnitude. A correlation exists between the variation period of a Cepheid and it's average absolute magnitude, *M*. Thus, by observing the period, one can deduce the distance modulus (m – M), and hence the distance using the formula,

(1) $$\mathbf{D} = 10\text{^}(\text{m-M+5})/5,$$

where *m* is the apparent magnitude.

Table 3.6 lists some Cepheid variables in the SMC, along with their average apparent magnitudes and the logarithms of their periods of variation. A plot of this data reveals a linear relationship, as is seen in Figure 3.6.

Table 3.6 Observational Data on Cepheid Variable Stars

Variable	log P	m
HV 837	1.63	12.85
HV 1967	1.45	13.52
HV 843	1.13	14.83
HV 2063	1.05	14.47
HV 2019	0.21	16.8
HV 2035	0.3	17.7
HV 844	0.35	16.3
HV 2046	0.41	16
HV 1809	0.45	16.1
HV 1987	0.5	16
HV 1825	0.63	15.6
HV 1903	0.71	15.6
HV 1945	0.81	15.2
HV 2060	1.01	14.3
HV 1873	1.11	14.7
HV 1954	1.22	13.8
HV 847	1.44	13.8
HV 840	1.52	13.4
HV 11182	1.6	13.6
HV 1837	1.63	13.1
HV 1877	1.7	13.1

Source: K. Duckett, *A Laboratory Textbook for Introductory Astronomy*, Fifth Edition, Contemporary, 1998

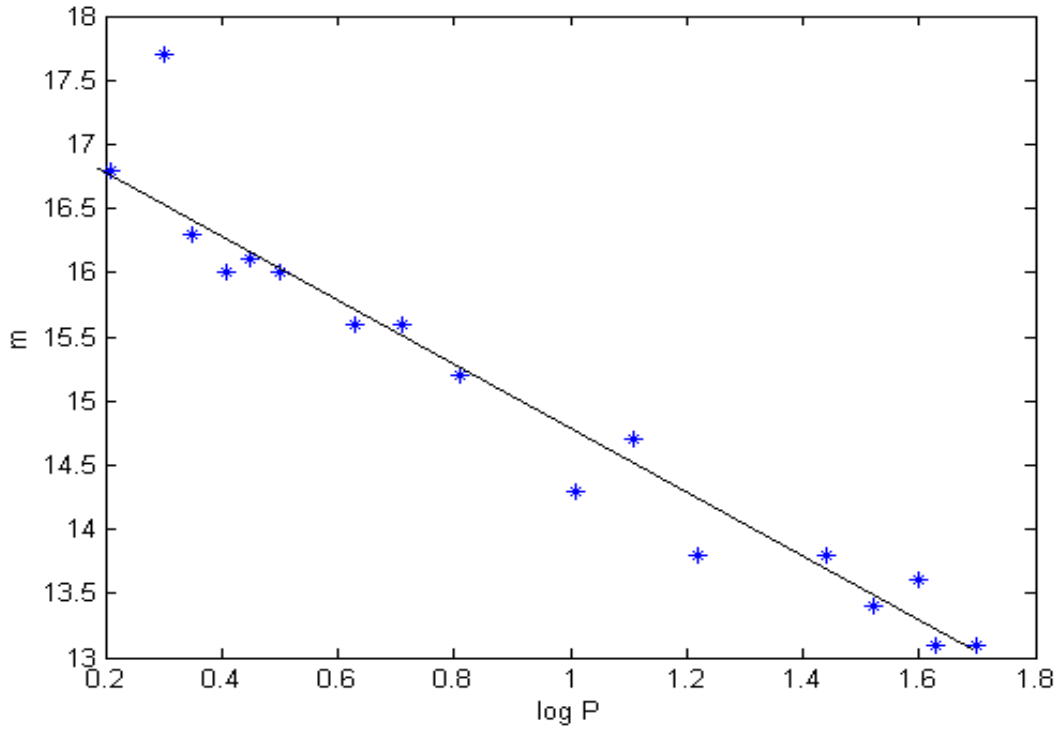

Fig. 3.6 Apparent magnitude vs. log P values for several cepheid variables in the SMC

Since all of the Cepheids are approximately the same distance away, there exists a one-to-one relationship between absolute and apparent magnitude. The distance to the SMC can therefore be calculated if a relationship between the period of varying brightness and absolute magnitude can be determined. Table 3.7, known as a Period- Luminosity Curve, first compiled by the astronomer Harlow Shapely, serves as a calibration by which this relationship is made known. This table, together with Table 3.6 is used in determining the distance modulus, the difference in apparent and absolute magnitude and *log P* is also a linear one, and is shown in Figure 3.7. Pikaia's role in this problem is in finding the best-fit lines through both of the above data sets. Pikaia returns the slope and *y*-intercept of the best-fit lines through both sets of data.

$$\chi^2 = \sum \left(\frac{a \log P_i + b - m_i}{\sigma_i} \right)^2 .$$

The parameters *a* and *b* are the slope and y-intercept, respectively, and the error weights associated with each data point are assumed to be equivalent. This procedure must be carried out twice, once for each data set, with the appropriate values for $\log P_i$ and m_i . In this way, Pikaia returns the average apparent and absolute magnitudes for a Cepheid as a function of its period. This allows the distance modulus to be computed in the following manner. For a given value of *log P*, the difference, *m – M*, can be evaluated. This is done ten times for random *log P* values that are read from an external data file.

Table 3.7 Period-Luminosity Curve

Log P	M
0	0.4
0.2	0.8
0.4	1.2
0.6	1.6
0.8	2.2
1	2.9
1.2	3.6
1.4	4.4
1.6	5.1
1.8	5.8

Source: K. Duckett, *A Laboratory Textbook for Introductory Astronomy*, Fifth Edition, Contemporary, 1998

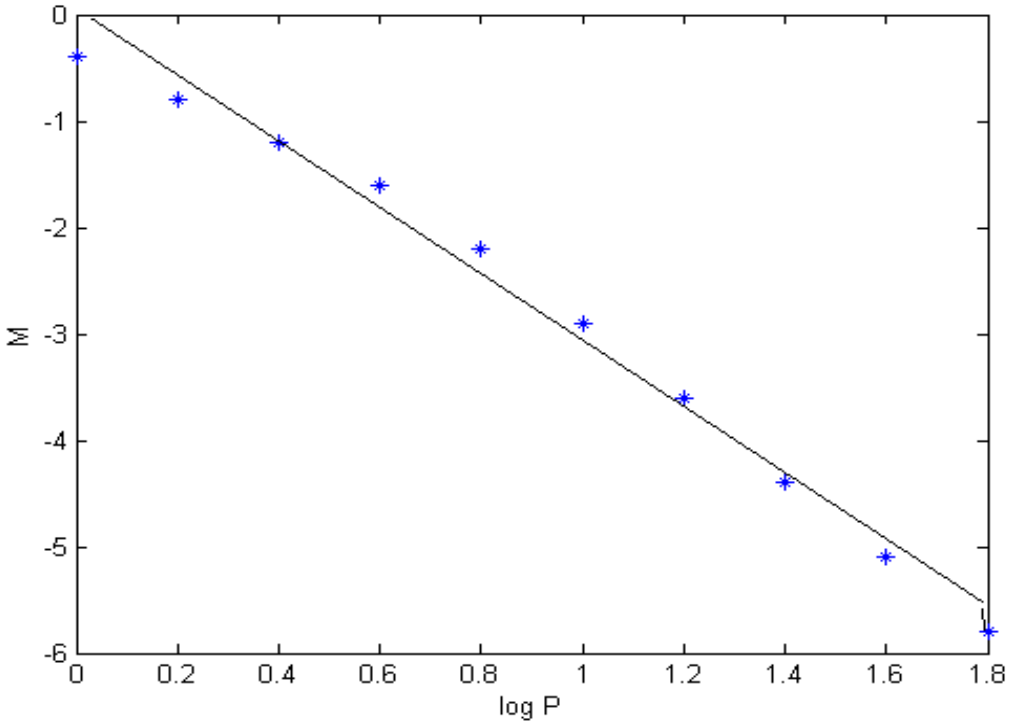

Fig 3.7 A plot of the period-luminosity curve (displayed in Table 3.7)

These ten values of $m - M$ are then averaged. It is from this averaged value of the distance modulus that the distance to the SMC is computed by means of equation (1).

The returned values for the distance modulus and the distance to the SMC are,

distance modulus = 17.85

distance = 42225.368 parsecs

The returned value differs from the currently accepted value of 53,000 Pc by about 20%. Such a significant deviation could possibly be due to convergence on a local optimum. However, for a simple linear fitting problem of this type, it is much more likely due to measurement errors in the data. Firstly, it has been found that the cepheids used by Shapely to determine the distance to the SMC (those listed in Table 3.7) are of a different class than those observed within the SMC. Furthermore, in 1923, studies by Edwin Hubble indicated that an error was introduced into the distance calculation due to interstellar dust [13].

3.6 Computation of the Mass of the Milky Way's Central Black Hole

It is believed that many galaxies harbor supermassive black holes in their centers. Measurements of stellar velocities in the vicinity of the Milky Way's center suggest the presence of a high concentration of mass, presumed to be a black hole [19]. Figure 3.8 shows astrometric data taken over a ten-year period and two thirds of the orbit of the star S2 around the radio source Sagittarius A*, believed to be the black hole. These

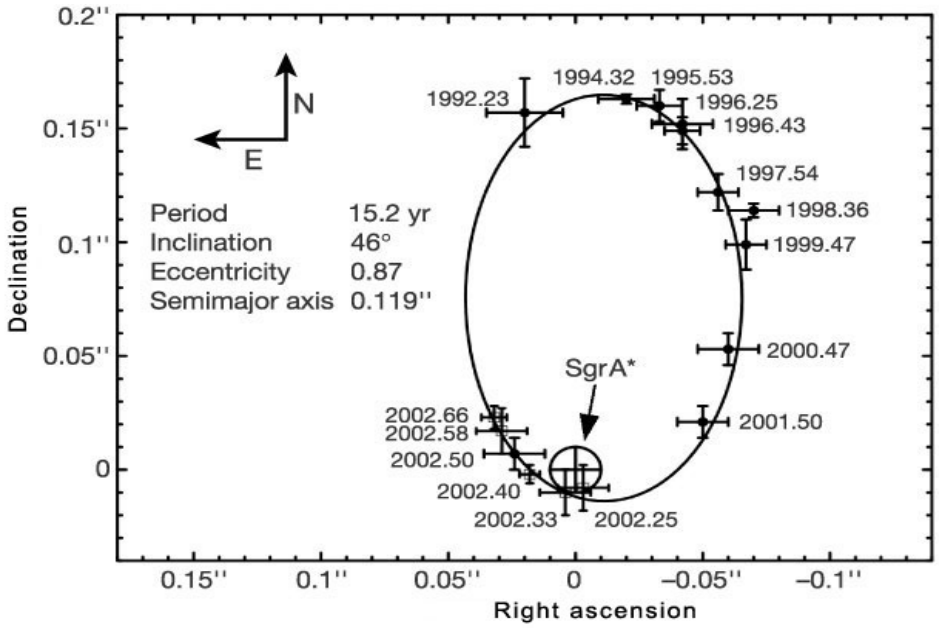

Fig. 3.8 Data showing two thirds of the orbit of S2 around a central mass thought to be a supermassive black hole. The listed orbital parameters were obtained by Schodel *et al* using the publicly available Binary Star Combined Solution Package

Source: R. Genzel *et al*., *Nature*, **419**,694 (2002)

observations, performed by Genzel *et al,* reveal a highly elliptical keplerian orbit. Each data point shown above corresponds to the angular coordinates (right ascension and declination) at which S2 was observed at the displayed time. The vertical and horizontal lines shown with each data point denote the uncertainties in the measurements made of the declination and right ascension, respectively. The solid curve represents the best-fit Keplarian orbit of S2. Genzel *et al* estimated the black hole mass to be of

$(3.7 \pm 1.5)\ x\ 10^6$ solar masses. In this section, the orbital elements and mass of SgrA* are reproduced with a genetic algorithm using the astrometric data of Fig. 3.8.

The procedure used here in obtaining the orbital elements is known as Kowalsky's method [20]. Given the general equation of the apparent orbit (the projection of the true orbit along the plane of the sky),

$$Ax^2 + 2Hxy + By^2 + 2Gx + 2Fy + 1 = 0, \tag{1}$$

this analytical method derives the orbital elements from the coefficients *A, H, B, G,* and *F*. The coordinate system of equation (1) is simply that shown in figure 3.8. The *x* and *y*-axes correspond to right ascension and declination, respectfully, and the origin is taken to be the position of SgrA*.

Note that this method assumes that the coefficients of equation (1) are known. Knowledge of these constants is crucial in determining the orbital parameters required for the calculation of the black hole's mass. The role of the genetic algorithm in this problem lies in finding the values of *A, B, G, F,* and *H* given the coordinates marking the

measured positions of S2. The most straightforward and obvious way of finding the best-fit coefficients of the ellipse shown above would be to perform a least-squares measure of fitness with equation (1). This direct approach, however, proved to be unsuccessful. For unknown reasons, the genetic algorithm failed to converge, instead carrying out the iterative breeding process beyond the specified bounds placed on each trial solution, eventually producing imaginary values for the coefficients. This method was therefore abandoned in favor of another approach that, although less direct than the above method, proved to be more fruitful. Consider the following alternative expression for the apparent orbit in spherical coordinates,

$$r^2 = \frac{u^2 v^2}{u^2 \cos^2(\theta - \theta_o) + v^2 \sin^2(\theta - \theta_o)}, \tag{2}$$

where u and v are the semi-major and minor axes, respectively, r is the distance from the ellipse center, θ given by

$$\theta = \arctan\left(\frac{y - y_o}{x - x_o}\right)$$

with x_o and y_o denoting the coordinates marking the center, and θ_o is the angle made between the x-axis and the major axis. Using the Pythagorean theorem to compute the distances of each data point from the ellipse center,

(3) $$d_i = \sqrt{(x_o - x_i)^2 + (y_o - y_i)^2},$$

the least-squares fitting problem takes the following form.

(4) $$\chi^2 = \left(\frac{d_i - r(\theta_i)}{\Delta r_i} \right)^2$$

The quantity $r(\theta_i)$ denotes equation (2) evaluated at the angle corresponding to the *i*th data point, and Δr_i is the estimated error in distance between the *i*th point and the center. These error weights were obtained from those associated with the measurements of right ascension and declination through a propagation of errors.

$$\Delta r = \frac{\partial d}{\partial x} \Delta x + \frac{\partial d}{\partial y} \Delta y$$

The quantities $\frac{\partial d}{\partial x}$ and $\frac{\partial d}{\partial y}$ denote the derivatives of equation (3) evaluated at each data point, with Δx and Δy representing the corresponding errors in right ascension and declination. This fitting procedure returns the best-fit values for the following: *u, v,* x_o, y_o, and θ_o. The problem now is to deduce the constants of equation (1) in terms of these parameters.

Re-writing the expression for the apparent ellipse in its more familiar form,

$$\frac{\left(x-x_o\right)^2}{v^2}+\frac{\left(y-y_o\right)^2}{u^2}=1 \quad , \tag{5}$$

the *x* and *y*-intercepts of the ellipse are calculated in terms of the returned parameters. Setting y = 0 in equation (5), the *x*-intercepts, x_1 and x_2 , are found to be

$$x_1 = x_o + \frac{v}{u}\sqrt{u^2 - y_o^2} \qquad x_2 = x_o - \frac{v}{u}\sqrt{u^2 - y_o^2} \quad . \tag{6}$$

Likewise, setting x = 0, it can be shown that

$$y_1 = y_o + \frac{u}{v}\sqrt{v^2 - x_o^2} \qquad y_2 = y_o - \frac{u}{v}\sqrt{v^2 - x_o^2} \quad . \tag{7}$$

Letting y = 0 in equation (1) yields the following.

$$Ax^2 + 2Gx + 1 = 0 \tag{8}$$

Invoking the quadratic formula to find the roots of this expression, one obtains the following expressions for x_1 and x_2 in terms of the unknown coefficients.

(9) $$x_1 = \frac{-G + \sqrt{G^2 - 1}}{A} \qquad x_2 = \frac{-G - \sqrt{G^2 - 1}}{A} .$$

It can be shown from these two equations that

(10) $$G = -\frac{1}{2}\left(x_1 + x_2\right) A .$$

Inserting x_1 into equation (8) gives

(11) $$A = \frac{-2Gx_1 - 1}{x_1^2} .$$

Rearranging equation (10) yields

$$-2Gx_1 = (x_1 + x_2)Ax_1 = (x_1^2 + x_1x_2)A .$$

Inserting this expression into equation (11), the latter formula reduces to

$$A = \frac{1}{x_1 x_2}.$$

Now, putting this expression for A into equation (11), G can be written as

$$G = -\frac{x_1 + x_2}{2x_1 x_2}.$$

Setting x = 0 in equation (1) yields

(12) $$By^2 + 2Fy + 1 = 0$$

and a similar procedure gives B and F in terms of the two y-intercepts of the ellipse. Four of the five desired constants have thus been obtained in terms of the roots.

(13) $$A = \frac{1}{x_1 x_2} \qquad G = -\frac{x_1 + x_2}{2x_1 x_2}$$

$$B = \frac{1}{y_1 y_2} \qquad F = -\frac{y_1 + y_2}{2y_1 y_2}$$

Given the parameters returned by the genetic algorithm, the numerical values of the roots are known from equations (6) and (7), and hence the values of *A, B, G,* and *F* from

equations (13). There still remains the problem of finding H, and again the genetic algorithm is invoked for this purpose. With the measured position coordinates serving as the x and y values in equation (1),

$$Ax^2 + 2Hxy + By^2 + 2Gx + 2Fy + 1 = 0,$$

the only remaining unknown quantity in this formula is H. Isolating H,

$$H = \frac{Ax^2 + By^2 + 2Gx + 2Fy + 1}{2xy} \equiv h(x, y), \tag{14}$$

the chi-square measure of fitness is

$$\chi^2 = \left(2Hx_i y_i - h(x_i, y_i)\right)^2, \tag{15}$$

where $h(x_i, y_i)$ is equation (14) evaluated at the data point (x_i, y_i) with the coefficients given by equations (13).

Having obtained the coefficients of equation (1), Kowalsky's method for determining the orbital element of the system is now undertaken. Illustrated in Fig. 3.9 are several orbital parameters that are of importance in the following derivation of the black hole mass. The origin O corresponds to SgrA*, a focal point of the orbit of S2. If

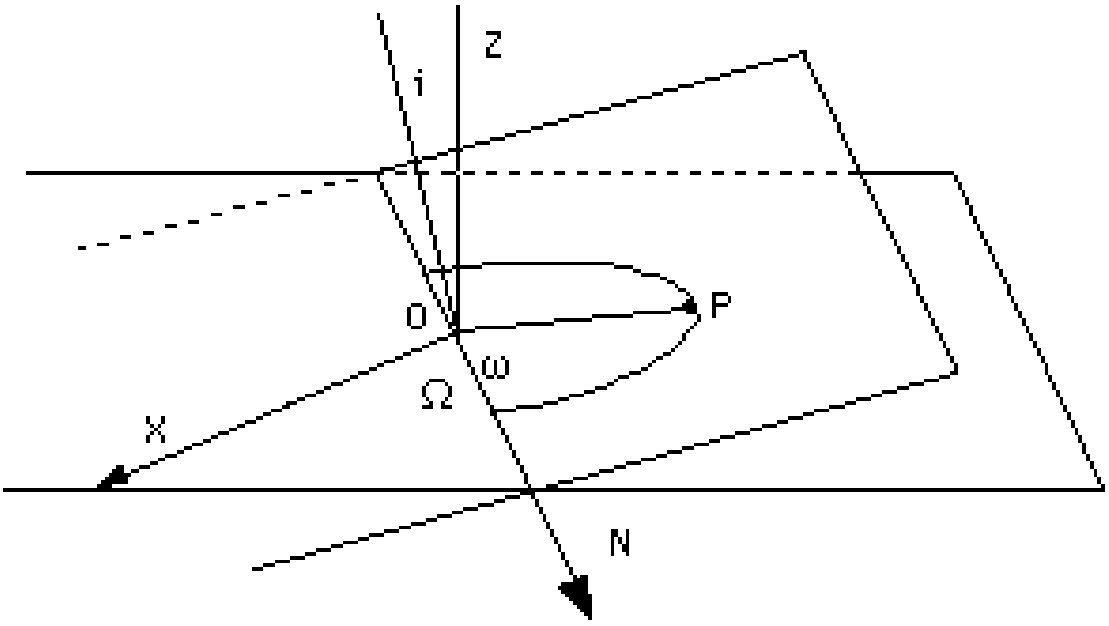

Fig. 3.9 Orbital elements using in deriving the mass of the Milky Way's central black hole.

Source: http://www.phy6.org/stargaze/Smotion.htm

we take the horizontal plane shown above to be the plane of the apparent orbit, and the inclined plane to be that of the true orbit, then the line *N* can be taken as the line of nodes, the line common to both ellipses, the true and the apparent. The angle *i* denotes the inclination and is the angle made between the two planes. The point *P* denotes the periastron, the point of closest approach in the orbit. The angle ω is made between the line of nodes and the major axis of the true orbit, and Ω is the angle made by the intersection of the plane perpendicular to the line of sight and that of the true orbit. Let (*x', y', z'*) denote the coordinates of S2 with respect to rectangular axes made in reference to the *true* orbital plane. The ellipse describing the true orbit can therefore be expressed as

$$\frac{\left(x'+a\varepsilon\right)^2}{a^2}+\frac{y'^2}{b^2}=1 \tag{16}$$

where *a* and *b* are the semi-major and minor axes, respectively, and ε is the eccentricity. Making a change in coordinates, this expression can be re-written in terms of the (*x, y, z)* coordinate system, giving the following equation for the apparent ellipse.

$$\frac{\left(m_2x-l_2y+a\varepsilon n_3\right)^2}{\left(an_3\right)^2}+\frac{(m_1x-l_1y)^2}{(bn_3)^2}=1 \tag{17}$$

where l_1, l_2, m_1, m_2 and n_3 are five of the nine direction cosines, the elements in the transformation matrix that relates the (x,y,z) coordinates to those defined by the plane of the true orbit,

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} l_1 m_1 n_1 \\ l_2 m_2 n_2 \\ l_3 m_3 n_3 \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix},$$

and are defined as follows

(18) $$l_1 = \cos\Omega \cos\omega - \sin\Omega \sin\omega \cos i$$

$$l_2 = -\cos\Omega \cos\omega - \sin\Omega \sin\omega \cos i$$

$$l_3 = \sin\Omega \sin i$$

$$m_1 = \sin\Omega \cos\omega - \cos\Omega \sin\omega \cos i$$

$$m_2 = -\sin\Omega \cos\omega + \cos\Omega \sin\omega \cos i$$

$$m_3 = -\cos\Omega \sin i$$

$$n_1 = \sin\omega \sin i$$

$$n_2 = \cos\ \omega \sin i$$

$$n_3 = \cos i$$

Equations (1) and (17) are equivalent and therefore coefficients of like powers are proportional. It follows that

(19) $$A = \frac{-a}{pn_3^2}\left(\frac{m_2^2}{a^2} + \frac{m_1^2}{b^2}\right), \qquad B = \frac{-a}{pn_3^2}\left(\frac{l_2^2}{a^2} + \frac{l_1^2}{b^2}\right)$$

$$H = \frac{a}{pn_3^2}\left(\frac{l_2 m_2}{a^2} + \frac{l_1 m_1}{b^2}\right)$$

$$G = \frac{-\varepsilon m_2}{pn_3}, \qquad F = \frac{\varepsilon l_2}{pn_3},$$

Where p is the latus rectum of the true ellipse and is defined to be

$$p = \frac{b^2}{a}$$

It follows from equations (18) and (19) that

(20) $$F^2 - G^2 + A - B = \frac{\tan^2 i}{p^2}\cos(2\Omega)$$

(21) $$FG - H = -\frac{\tan^2 i}{2p^2}\sin(2\Omega)$$

(22) $$F^2 + G^2 - (A + B) = \frac{2}{p^2} + \frac{\tan^2 i}{p^2}.$$

These three equations contain three unknown variables: Ω, *p,* and *i*. Dividing equation (21) by (20), one can determine Ω.

(23) $$\Omega = \frac{1}{2}\arctan\left(\frac{-2(FG - H)}{F^2 - G^2 + A - B}\right).$$

Knowing Ω, the value of $\frac{\tan^2 i}{p^2}$ can then be calculated from equation (20) or (21). Equation (22) then allows for the determination of *p*. Having thus found *p*, the inclination, *i*, is easily determined by equations (20), (21), or (22). In addition, equations (18) and (19) can be used to show that

(24) $$\varepsilon \sin \omega = p(G\sin\Omega - F\cos\Omega)\cos i$$

and

(25) $$\varepsilon \cos \omega = - p(G\sin\Omega + F\sin\Omega).$$

Dividing equation (24) by (25) gives

(26) $$\tan \omega = -\frac{G\sin\Omega - F\cos\Omega}{G\cos\Omega + F\sin\Omega}.$$

Taking the inverse tangent of equation (26) thus gives a value for ω. There is now enough information to allow for the calculation of the eccentricity.

$$\varepsilon = \frac{p(G\sin\Omega - F\cos\Omega)\cos i}{\sin\omega}$$

Taking the value of the semi-major axis of the apparent orbit, *u,* returned by the genetic algorithm, the semi-major axis of the *true* orbit is obtained from dividing by the sine of the inclination angle, *i*.

(27) $$a = \frac{u}{\sin i}$$

Note from the above plot of the data that *u*, and therefore *a*, is given in terms of arc seconds and therefore correspond to the angular separation between the apogee and perigee of the orbit. Dividing the quantity given in equation (27) by 3600 gives the angular separation in units of degrees. Our solar system lies approximately 2.63×10^9 astronomical units from the galactic center. If the major axis of S2's orbit is treated as a small segment of the arc of a great circle with a radius of *r*, then a conversion of *a* from degrees to astronomical units is possible by letting $r = 2.63 \times 10^9$ AUs and evoking the formula

(28) $$a = \frac{r\theta}{57.3}$$

where θ is the quantity of equation (27) in degrees and *a* is now the semi-major axis in astronomical units.

To determine the orbital period of S2, consider the formula for the true orbit written in its conic form.

(29) $$\rho = \frac{a(1-\varepsilon^2)}{1+\varepsilon\cos f}$$

Here f denotes the true anomaly, the position angle of S2, and ρ is the radius vector extending from the focal point to the position of S2 along the ellipse. Now consider the ellipse describing the orbit to be a circle of radius a inclined at an angle ϕ = arcsin ε, as shown in Fig 3.10. The angle E is known as the eccentric anomaly and is the angle measured between the major axis of the ellipse and the line joining the center of the auxiliary circle to point Q, which is joined with the point P, marking the true position of S2 on the ellipse, by a vertical line running perpendicular to the major axis. Both the true and eccentric anomalies are measured from the major axis and increase by 360 degrees with each revolution. The rates at which these angles change depend on the position of the star in its orbit and are governed by Kepler's law of areas. From Fig. 3.10, it is seen that $CF = a\varepsilon$ and $CD = a\cos E$. Therefore,

$$DF = CF - CD = a\varepsilon - a\cos E = -\rho\cos f.$$

Rearranging gives

(30) $$\cos f = \frac{a(\cos E - \varepsilon)}{\rho}.$$

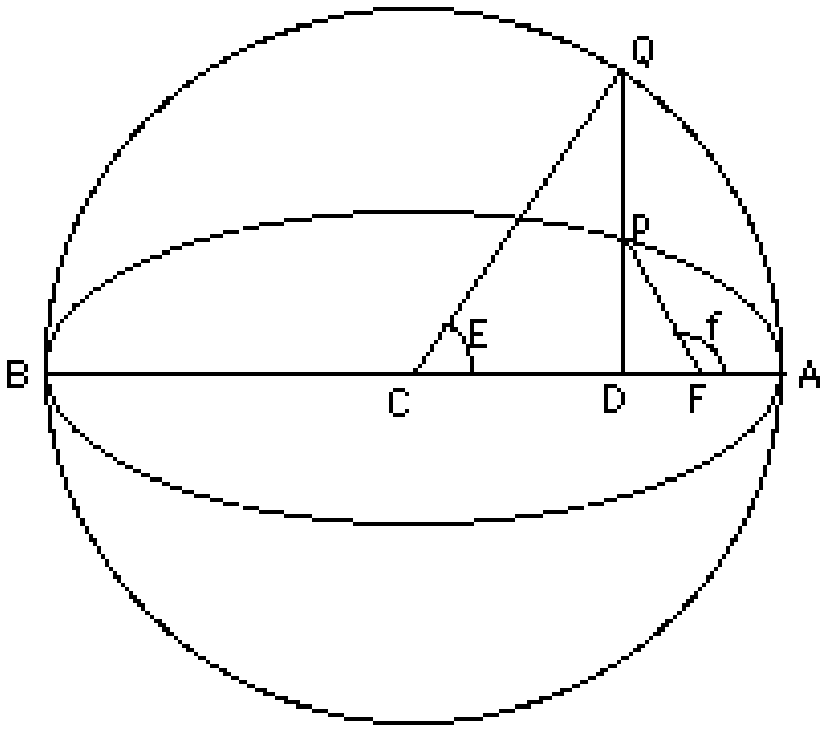

Fig. 3.10 The orbit of S2 inscribed in an auxiliary circle

Source: http://www.phy6.org/stargaze/Smotion.htm

Substituting equation (30) for *cos f* in (29) yields

$$\rho(1+\varepsilon\cos f)=\rho\left(1+\frac{a\varepsilon\cos E}{\rho}-\frac{a\varepsilon^2}{\rho}\right)=\rho+a\varepsilon\cos E-a\varepsilon^2=a(1-\varepsilon^2)$$

Solving for *E* gives

(31)
$$E=\cos^{-1}\left(\frac{a-\rho}{a\varepsilon}\right)$$

Note that the eccentric anomaly is a function of ρ only. Knowing the right ascension and declination of each data point, the Pythagorean theorem allows for the computation of the distances of each position from the black hole.

(32)
$$\rho_i=\frac{\sqrt{x_i^2+y_i^2}}{\sin i}$$

The denominator accounts for the fact that the data points marking the positions of the star define the *apparent* orbit, not the true one. Inserting the calculated values of equation (32) in (31) allows for the determination of the eccentric anomalies corresponding to each recorded position of S2.

The total orbital energy of the star, *Q*, is the sum of its kinetic and its gravitational potential energies,

(33) $$Q = \frac{1}{2} m \left(\frac{d\rho}{dt} \right)^2 + \frac{l^2}{2m\rho^2} - \frac{k}{\rho}$$

where m is the mass of the star and the constant l denotes the magnitude of the angular momentum of the system and is given by

$$l = m\rho^2 \left(\frac{d\theta}{dt} \right)$$

The last term of equation (33) denotes the gravitational binding energy. Rearranging equation (33) gives

(34) $$dt = \frac{d\rho}{\sqrt{\frac{2}{m}\left(Q + \frac{k}{\rho} - \frac{l^2}{2m\rho^2} \right)}}$$

It can be shown that the total energy of the system remains constant [22], and is given by

$$\boldsymbol{Q} = \frac{-k}{2a}.$$

With this expression, equation (34) reduces to

(35) $$dt = \frac{\sqrt{\frac{ma}{k}}\rho d\rho}{\sqrt{\varepsilon^2 a^2 - (\rho - a)^2}} .$$

Equation (31) can be rearranged to yield

(36) $$\rho = a(1 - \varepsilon \cos E) .$$

Differentiating gives

(37) $$d\rho = a\varepsilon \sin E \, dE.$$

Putting equations (36) and (37) into (35) gives

(38) $$\mathrm{dt} = \sqrt{\frac{ma}{k}} a(1 - \varepsilon \cos E) dE .$$

Kepler's third law states that

(39) $$T = 2\pi \sqrt{\frac{ma^3}{k}}$$

where T is the orbital period. Putting this expression into equation (38) yields

(40) $$\frac{2\pi}{T}dt = (1 - \varepsilon \cos E)dE\,.$$

Integrating equation (40), with the condition that E = 0 at perigee, when the star is at its closest approach to the black hole, one obtains

(41) $$\frac{2\pi(t-\tau)}{T} = E - \sin E\,,$$

where τ is the time of periastron passage. The left side of equation (41) is often referred to as the mean anomaly. This angle increases at a steady rate and would mark the position of the star if the eccentricity of the orbit were zero, hence corresponding to a circular orbit in which the orbital velocity is a constant. The mean anomalies of each data point are determined from the eccentric anomalies calculated in equation (31).

(42) $$M_i = E_i - \sin E_i\,.$$

With the data marking the time, in years, of each observation of S2's position, the formula

(43) $$M_i = \frac{2\pi(t_i - \tau)}{T}$$

is used in performing a chi-square measure of fitness.

(44) $$\chi^2 = \left(\frac{M_i T - 2\pi(t_i - \tau)}{\Delta M_i} \right)^2$$

In this way, the genetic algorithm returns the best-fit values for the period of S2 and the time marking its point of closest approach to the black hole. The quantity ΔM_i denotes the error estimate of the mean anomaly associated with the *i*th data point. Again, these error estimates are obtained through a propagation of errors. Given equation (32) and the error estimates in right ascension and declination, the error in ρ is estimated to be

(45) $$\Delta\rho_i = \frac{\partial\rho}{\partial x}\Delta x_i + \frac{\partial\rho}{\partial y}\Delta y_i$$

The errors associated with the eccentric anomaly values are therefore taken to be

(46) $$\Delta E_i = \frac{\partial E}{\partial \rho}\Delta\rho_i \,,$$

where the derivative $\frac{\partial E}{\partial \rho}$ is that of equation (31) with respect to ρ. Lastly, the derivative of equation (42) is taken and the values of ΔM_i are determined.

(47) $$\Delta M_i = \frac{\partial M}{\partial E} \Delta E_i$$

Knowing the period of S2, there is now enough information to deduce the mass of the Milky Way's central black hole. Kepler's third law states that the square of the period of a body in orbit around a central mass is proportional to the cube of the semi-major axis of the orbit. When the period is expressed in years, and the semi-major axis in astronomical units, as is the case here, the proportionality constant of Kepler's third law is simply the mass of the central body in units of solar masses. Hence, the black hole mass is taken to be

(49) $$M = \frac{a^3}{T^2}$$

Below are listed the values obtained for the orbital parameters of the system, and are in good agreement with those found by other means [19].

inclination = 64.55^o

eccentricity = *.92*

semi-major axis = $.097^o$ = 971.69 AU

Period = 16.47 yrs

The value of the black hole mass was found to be approximately 3.4×10^6 solar masses, again in good agreement with previous estimates.

3.7 Parameterization of Thermonuclear Reaction Rates

The rate of a thermonuclear reaction is strongly dependent on the temperature of the reactants. Parametrization of reaction rates is of extreme importance in astrophysical calculations, as it allows for the compilation of libraries that house such parameters, and can be easily accessed for purposes of astrophysical computations. Here the genetic algorithm is applied to the problem of fitting parameters to rates of reactions that play a role in explosive nucleosynthesis that occurs in nova events or in x-ray bursts. Consider the following functional form that is currently used to parameterize such rates.

reaction rate (T) = exp(a1 + a2/T + a3/T^(1/3) + a4*T^(1/3) + a5*T +
a6*T^(5/3) + a7*ln(T))

Given several values of temperature and rates of reactions involving heavy element production, the goal is to find the parameters a1, a2, a3, etc, that maximizes fitness. If a good fit cannot be found, an additional set of parameters is added. So a functional form with fourteen parameters would be expressed as

$$\text{Rate}(T) =$$

$$\exp(a1+a2/T+a3/T^{(1/3)}+a4*T^{(1/3)}+a5*T+a6*T^{(5/3)}+a7*\ln(T)) +$$

$$\exp(a8+a9/T+a10/T^{(1/3)}+a11*T^{(1/3)}+a12*T+a13*T^{(5/3)}+a14*\ln(T))$$

This problem is an ideal one for genetic algorithms because it is multimodal, requiring the fitting of several parameters, and previous attempts at finding reasonable solutions for certain rates that had proven difficult to fit with other optimization algorithms. It is here that the robustness of the genetic algorithm can be fully appreciated.

A total of nine reaction rates were parameterized in all. In giving a detailed description of the problem of maximizing the fitness of these parameter sets, one particular rate is expounded upon. This reaction is one in which two alpha particles and a free neutron combine to produce a radioactive isotope of beryllium.

$$n + 4He + 4He \rightarrow Be7$$

The data shown in Figure 3.11 is a small sample of data that was taken by the NACRE collaboration, and gives rate values and associated temperatures. There are fifty-nine data points in all. As can be seen, the rates vary by many orders of magnitude over the temperature range of physical interest. Using these data, a chi-square measure of fitness is performed. Finding a good fit for this particular reaction required a rate function with twenty-eight parameters.

T9	Rate
0.1000000000E-02	0.3900000000E-58
0.2000000000E-02	0.2500000000E-46
0.3000000000E-02	0.1350000000E-40
0.4000000000E-02	0.5580000000E-37
0.5000000000E-02	0.2110000000E-34
0.6000000000E-02	0.1960000000E-32
0.7000000000E-02	0.7390000000E-31
0.8000000000E-02	0.1480000000E-29
0.9000000000E-02	0.1880000000E-28
0.1000000000E-01	0.1690000000E-27
0.1100000000E-01	0.1160000000E-26
0.1200000000E-01	0.6370000000E-26
0.1300000000E-01	0.2950000000E-25
0.1400000000E-01	0.1180000000E-24
0.1500000000E-01	0.4170000000E-24
0.1600000000E-01	0.1330000000E-23
0.1800000000E-01	0.1050000000E-22
0.2000000000E-01	0.6290000000E-22
0.2500000000E-01	0.2990000000E-20

Figure 3.11 Rates and corresponding temperatures for the reaction n + 4He + 4He → Be9

Source: http://pntpm.ulb.ac.be/Nacre/nacre_d.htm

Unlike the previous problems discussed here, the returned solutions were much more sensitive to the input parameters governing the evolution of trial solutions, and a great deal of time was spent changing the values of quantities such as population size and crossover probability to ensure a returned solution that would yield a minimal chi-square. Of course, in a problem such as this, it is ultimately impossible to ascertain if the returned solution is the global optimum or merely a local one. There is always the possibility that there exist a combination of input parameters that can yield a better solution. One cannot hope to exhaustively search every possible set of these parameters, and this is in part due to the stochastic nature of the genetic algorithm, which makes itself well known in the problem of parameterization of reaction rates. Due to the inherent randomness involved in the search for an optimal solution, a particular set of input parameter values need not return the same solution for two consecutive executions. After yielding an effectively good solution, the next execution may return an even better one, or a far worse one. In this respect, this problem was much more challenging than those discussed above.

Another distinguishing characteristic that sets this problem apart from the others was the apparent proportionality between fitness of returned solutions and the number of generations through which the solutions are allowed to evolve. Unlike the problems in which returned solutions consists of only one or two fitted parameters and a point is reached where additional generations add nothing to the accuracy of the final result, the fitness of the returned solutions for most of the reactions tended to increase steadily with generation count. Others would eventually reach a convergence. The fitting of the above reaction, for example, was done with 10^7 generations, with a count of 10^8 offering

nothing new. Those evolutionary runs with 10^7 and 10^8 generations would often take several days to converge on an optimal solution, running on a single 2GHz processor.

The most dominant influence on the outcome of the evolutionary runs appeared to be the upper and lower bounds for the allowed range of mutation rate. These can be found in the subroutine *adjmut*. The default values are

rdiflo (minimum rate) = .0005
rdifhi (maximum rate) = .25.

Changing the values had a significant effect on the evolutionary runs and the solutions returned. The success of a given range in mutation rate varied with the reaction being studied. The above variables therefore had to be re-adjusted when a new reaction rate was attempted. For the particular reaction given above, the two values marking the bounds of the mutation rate were set to be .1 and .85.

The problem of fitting parameters to the rates, for the most part, consisted of adjusting the input parameters in a trial and error fashion. A subroutine was introduced into the code that calculated rate values using a returned solution, and then determined the percent difference between this rate value and the one listed in the data table that corresponded to the same temperature. An average of these percent differences was then taken. If the average percent difference for a reaction was too high, some values of the input parameters were altered and the evolutionary run was executed again. Generally, any parameter set yielding rate values that differed from the data by a percent difference less than 30% was kept.

The parameters, a1 though a28, obtained for the above reaction are

-91.850	0.454	-88.852	-59.628	10.824	-45.308	50.398	-36.404	-0.038	48.798
13.758	-70.844	-94.884	18.398	-50.064	-1.088	17.192	19.350	0.020	0.000
-2.114	66.172	-20.002	20.204	-93.172	93.996	-86.004	-38.000		

Using the given temperature values, the reaction rates calculated with these parameters yielded numbers that differed, on average, from the data by approximately 5.7%.

To illustrate the effectiveness of the genetic algorithm's fitting procedure in this problem, Figure 3.12 shows the plots of the data, along with the rates calculated using the returned solution.

The appendix shows the results obtained for eight other reactions

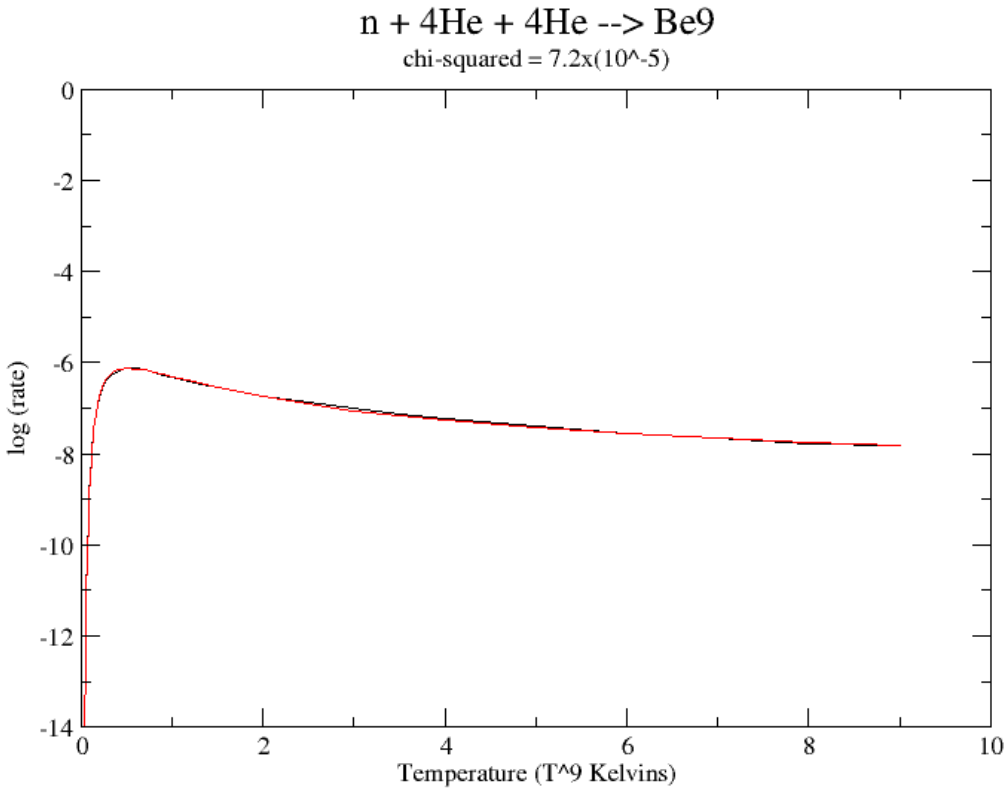

Fig. 3.12 log(rate) vs. temperature. The red curve shows the data, with the dark curve showing the corresponding values computed from the twenty-eight parameters returned by Pikaia. Because the data varies over several orders of magnitude, the graph shows the logarithms of the rates, rather than the rates themselves.

Conclusions

In comparing the performance of genetic algorithms with those of more conventional optimization methods, it has been found that the results are highly dependent upon the particular problem being studied. As was seen in the parameterization of the several reaction rates presented in the appendix, the genetic algorithm was successful at fitting parameters to the rates within a reasonable degree of deviation with the data, excelling where previous strategies have failed. It is complex multimodal problems of this type in which the robustness and many strengths of the genetic algorithm give it distinct advantages over other procedures.

The dependency of the results upon various input parameters (population size, number of generations, mutation rate, etc) also appears to be problem-specific. The solutions obtained when fitting the reaction rates, for example, were highly sensitive to the values of the aforementioned parameters. Other problems involving a 1-D or 2-D fitness space, on the other hand, showed little dependence on input parameters. This is presumably because the problem is simple enough that crossover and mutation need not play a large role in obtaining the desired solution.

It is evident from the problems discussed that genetic algorithms work well for multimodal problems that may thwart other optimization techniques, and perform at least as well as other methods when applied to simpler problems, those involving only 1D and 2D search spaces, for example.

It was the finding of the work presented in this thesis that genetic algorithms generally work well for anything that can be treated as an optimization problem. But it is

only when dealing with problems having many fitting parameters and large, complex search spaces that the use of a genetic algorithm may be necessary and beneficial.

Genetic algorithms offer promising applications in future astrophysics research. One such avenue for genetic algorithms is the modeling of interacting galaxies. Such modeling could provide a deeper understanding of galaxy formation, as well as "peculiar" galaxies that exhibit such structures as bridges and tidal tails. The modeling of interacting galaxies involves the fitting of various orbital parameters, requiring searches in multidimensional solution spaces, and traditional techniques have long suffered the problem of terminating on local optima. Genetic algorithms, with their weak dependence on the location of starting points, provide a great advantage in avoiding such hindrances [24].

There has been recent interest in applying genetic algorithms to the problem of gravitational lens inversion, in which structural detail of the lensing object, such as mass distribution, is recovered from observations of the gravitational lensed images. This requires optimal fitting of several lens parameters as well as surface brightness distribution of the source. Brewer *et al* have demonstrated that genetic algorithms can successfully recover the source configuration of an idealized gravitational lens system, and future work is planned for observed gravitational lens systems [25].

List of References

[1] https://www.hao.ucar.edu

[2] F. Busetti, *Genetic Algorithms Overview*, 2000

[3] http://www.doc.ic.ac.uk

[4] http://mat.gsia.cmu.edu/QUANT/NOTES

[5] http://math.berkeley.edu/~mhaiman/mathH54

[6] A. Marczyk, *Genetic Algorithms and Evolutionary Computation*, http://www.talkorigins.org

[7] http://www.faqs.org

[8] M. Mitchel, *An Introduction to Genetic Algorithms*. MIT Press, 1996

[9] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, 1998

[10] S. Mardle, S. Pascoe, CHEER: Computers in Higher Education Economics Review **13**,2 (1999)

[11] http://subsimple.com

[12] http://astrosun2.astro.cornell.edu/academics

[13] K. Duckett, *A Laboratory Textbook for Introductory Astronomy*, Fifth Edition, Contemporary, 1998

[14] W. Press, *et al*., *Numerical Recipes in Fortran 77*, Second Edition, Cambridge, 1996

[15] http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/ BMVA96Tut/ node17.html

[16] http://en.wikipedia.org

[17] http://imagine.gsfc.nasa.gov

[18] EG&G ORTEC, *Experiments in Nuclear Science AN34 Laboratory Manual*, Third Edition, 1984

[19] R. Genzel *et al*., *Nature*, **419**,694 (2002)

[20] R. Aitken, *Binary Stars*, Dover, 1964

[21] http://www.phy6.org

[22] J. Marion, S.Thornton, *Classical Dynamics of Particles and Systems*, Third Edition, Harcourt Brace Jovanovich & Academic Press, 1988

[23] http://pntpm.ulb.ac.be

[24] C. Theis *et al*., *Galaxies and Genes: Towards an Automatic Modeling of Interacting Galaxies,* arXiv:astro-ph/0107078 v1, 2001

[25] B. Brewer, G. Lewis, *When Darwin Met Einstein: Gravitational Lens Inversion with Genetic Algorithms,* arXiv:astro-ph/0501202 v1, 2005

[26] http://en.wikipedia.org/wiki/Simplex_algorithm

Appendix

The following shows the percent differences between experimentally obtained thermonuclear reaction rates and those computed using the fitted parameters obtained by the genetic algorithm. Also included are the data for each reaction, as well as plots showing both curves (the natural logs of the experimentally obtained rates and those of the calculated rates) for each reaction.

Reaction	Percent difference
$p + {}^{23}Na \rightarrow {}^{24}Mg$	12.4%
$p + {}^{15}N \rightarrow {}^{4}He + {}^{12}C$	7.86%
$p + {}^{27}Al \rightarrow {}^{28}Si$	25.84%
$p + {}^{28}Si \rightarrow {}^{29}P$	23.36%
$p + {}^{27}Al \rightarrow {}^{4}He + {}^{24}Mg$	23.0%
$4He + {}^{18}O \rightarrow {}^{22}Ne$	18.53%
$4He + 4He + 4He \rightarrow {}^{12}C$	14.36%
$n + 4He + 4He \rightarrow {}^{7}Be$	5.7%
$4He + {}^{16}O \rightarrow {}^{20}Ne$	26.92%

Reaction: p + 23Na → 24Mg

T9	Rate
0.1800000000E-01	0.4390000000E-25
0.2000000000E-01	0.6160000000E-24
0.2500000000E-01	0.1210000000E-21
0.3000000000E-01	0.6810000000E-20
0.4000000000E-01	0.4430000000E-16
0.5000000000E-01	0.9030000000E-13
0.6000000000E-01	0.1430000000E-10
0.7000000000E-01	0.5130000000E-09
0.8000000000E-01	0.7340000000E-08
0.9000000000E-01	0.5690000000E-07
0.1000000000E+00	0.2900000000E-06
0.1100000000E+00	0.1100000000E-05
0.1200000000E+00	0.3420000000E-05
0.1300000000E+00	0.9590000000E-05
0.1400000000E+00	0.2610000000E-04
0.1500000000E+00	0.7190000000E-04
0.1600000000E+00	0.1990000000E-03
0.1800000000E+00	0.1350000000E-02
0.2000000000E+00	0.6970000000E-02
0.2500000000E+00	0.1440000000E+00
0.3000000000E+00	0.1070000000E+01
0.3500000000E+00	0.4370000000E+01
0.4000000000E+00	0.1230000000E+02
0.4500000000E+00	0.2690000000E+02
0.5000000000E+00	0.4970000000E+02
0.6000000000E+00	0.1220000000E+03
0.7000000000E+00	0.2290000000E+03
0.8000000000E+00	0.3670000000E+03
0.9000000000E+00	0.5320000000E+03
0.1000000000E+01	0.7250000000E+03
0.1250000000E+01	0.1330000000E+04
0.1500000000E+01	0.2110000000E+04
0.1750000000E+01	0.3090000000E+04
0.2000000000E+01	0.4250000000E+04
0.2500000000E+01	0.7000000000E+04
0.3000000000E+01	0.1010000000E+05
0.3500000000E+01	0.1320000000E+05
0.4000000000E+01	0.1610000000E+05
0.5000000000E+01	0.2100000000E+05
0.6000000000E+01	0.2790000000E+05
0.7000000000E+01	0.3490000000E+05
0.8000000000E+01	0.4220000000E+05
0.9000000000E+01	0.4970000000E+05
0.1000000000E+02	0.5740000000E+05

Reaction: p + 15N → 4He + 12C

T9	Rate
0.5000000000E-02	0.7120000000E-25
0.6000000000E-02	0.1240000000E-22
0.7000000000E-02	0.7590000000E-21

0.8000000000E-02	0.2270000000E-19
0.9000000000E-02	0.4010000000E-18
0.1000000000E-01	0.4760000000E-17
0.1100000000E-01	0.4140000000E-16
0.1200000000E-01	0.2810000000E-15
0.1300000000E-01	0.1560000000E-14
0.1400000000E-01	0.7310000000E-14
0.1500000000E-01	0.2980000000E-13
0.1600000000E-01	0.1080000000E-12
0.1800000000E-01	0.1040000000E-11
0.2000000000E-01	0.7390000000E-11
0.2500000000E-01	0.3740000000E-09
0.3000000000E-01	0.7480000000E-08
0.4000000000E-01	0.5870000000E-06
0.5000000000E-01	0.1310000000E-04
0.6000000000E-01	0.1450000000E-03
0.7000000000E-01	0.1000000000E-02
0.8000000000E-01	0.5040000000E-02
0.9000000000E-01	0.1990000000E-01
0.1000000000E+00	0.6530000000E-01
0.1100000000E+00	0.1870000000E+00
0.1200000000E+00	0.4770000000E+00
0.1300000000E+00	0.1120000000E+01
0.1400000000E+00	0.2440000000E+01
0.1500000000E+00	0.4970000000E+01
0.1600000000E+00	0.9750000000E+01
0.1800000000E+00	0.3280000000E+02
0.2000000000E+00	0.9640000000E+02
0.2500000000E+00	0.8800000000E+03
0.3000000000E+00	0.4660000000E+04
0.3500000000E+00	0.1660000000E+05
0.4000000000E+00	0.4480000000E+05
0.4500000000E+00	0.9860000000E+05
0.5000000000E+00	0.1880000000E+06
0.6000000000E+00	0.4910000000E+06
0.7000000000E+00	0.9850000000E+06
0.8000000000E+00	0.1640000000E+07
0.9000000000E+00	0.2350000000E+07
0.1000000000E+01	0.3370000000E+07
0.1250000000E+01	0.6200000000E+07
0.1500000000E+01	0.9710000000E+07
0.1750000000E+01	0.1340000000E+08
0.2000000000E+01	0.1770000000E+08
0.2500000000E+01	0.2590000000E+08
0.3000000000E+01	0.3810000000E+08
0.3500000000E+01	0.5130000000E+08
0.4000000000E+01	0.6520000000E+08
0.5000000000E+01	0.9430000000E+08
0.6000000000E+01	0.1250000000E+09
0.7000000000E+01	0.1550000000E+09
0.8000000000E+01	0.1860000000E+09
0.9000000000E+01	0.2170000000E+09
0.1000000000E+02	0.2470000000E+09

Reaction: p + 27Al → 28Si

T9	Rate
0.2500000000E-01	0.1670000000E-22
0.3000000000E-01	0.3390000000E-20
0.4000000000E-01	0.2440000000E-17
0.5000000000E-01	0.1250000000E-15
0.6000000000E-01	0.8370000000E-14
0.7000000000E-01	0.1310000000E-11
0.8000000000E-01	0.6730000000E-10
0.9000000000E-01	0.1450000000E-08
0.1000000000E+00	0.1690000000E-07
0.1100000000E+00	0.1260000000E-06
0.1200000000E+00	0.6710000000E-06
0.1300000000E+00	0.2750000000E-05
0.1400000000E+00	0.9210000000E-05
0.1500000000E+00	0.2620000000E-04
0.1600000000E+00	0.6520000000E-04
0.1800000000E+00	0.2990000000E-03
0.2000000000E+00	0.1010000000E-02
0.2500000000E+00	0.9550000000E-02
0.3000000000E+00	0.4640000000E-01
0.3500000000E+00	0.1580000000E+00
0.4000000000E+00	0.4380000000E+00
0.4500000000E+00	0.1050000000E+01
0.5000000000E+00	0.2270000000E+01
0.6000000000E+00	0.8200000000E+01
0.7000000000E+00	0.2280000000E+02
0.8000000000E+00	0.5210000000E+02
0.9000000000E+00	0.1030000000E+03
0.1000000000E+01	0.1800000000E+03
0.1250000000E+01	0.5200000000E+03
0.1500000000E+01	0.1100000000E+04
0.1750000000E+01	0.1910000000E+04
0.2000000000E+01	0.2930000000E+04
0.2500000000E+01	0.5450000000E+04
0.3000000000E+01	0.8350000000E+04
0.3500000000E+01	0.1140000000E+05
0.4000000000E+01	0.1440000000E+05
0.5000000000E+01	0.1990000000E+05
0.6000000000E+01	0.2450000000E+05
0.7000000000E+01	0.4030000000E+05
0.8000000000E+01	0.5880000000E+05
0.9000000000E+01	0.7960000000E+05
0.1000000000E+02	0.1020000000E+06

Reaction: p + 28Si → 29P

T9	Rate
0.3000000000E-01	0.5970000000E-24
0.4000000000E-01	0.6610000000E-21
0.5000000000E-01	0.9630000000E-19
0.6000000000E-01	0.4290000000E-17
0.7000000000E-01	0.8870000000E-16

0.8000000000E-01	0.1080000000E-14
0.9000000000E-01	0.9010000000E-14
0.1000000000E+00	0.6440000000E-13
0.1100000000E+00	0.6330000000E-12
0.1200000000E+00	0.8550000000E-11
0.1300000000E+00	0.9880000000E-10
0.1400000000E+00	0.8450000000E-09
0.1500000000E+00	0.5460000000E-08
0.1600000000E+00	0.2790000000E-07
0.1800000000E+00	0.4180000000E-06
0.2000000000E+00	0.3590000000E-05
0.2500000000E+00	0.1640000000E-03
0.3000000000E+00	0.1980000000E-02
0.3500000000E+00	0.1140000000E-01
0.4000000000E+00	0.4110000000E-01
0.4500000000E+00	0.1090000000E+00
0.5000000000E+00	0.2350000000E+00
0.6000000000E+00	0.7140000000E+00
0.7000000000E+00	0.1520000000E+01
0.8000000000E+00	0.2620000000E+01
0.9000000000E+00	0.3920000000E+01
0.1000000000E+01	0.5330000000E+01
0.1250000000E+01	0.8970000000E+01
0.1500000000E+01	0.1290000000E+02
0.1750000000E+01	0.1870000000E+02
0.2000000000E+01	0.2930000000E+02
0.2500000000E+01	0.7860000000E+02
0.3000000000E+01	0.1850000000E+03
0.3500000000E+01	0.2750000000E+03
0.4000000000E+01	0.3820000000E+03
0.5000000000E+01	0.6510000000E+03
0.6000000000E+01	0.9920000000E+03
0.7000000000E+01	0.1410000000E+04
0.8000000000E+01	0.1900000000E+04
0.9000000000E+01	0.2480000000E+04
0.1000000000E+02	0.3160000000E+04

Reaction: p + 27Al → 4He + 24Mg

T9	Rate
0.2500000000E-01	0.7800000000E-22
0.3000000000E-01	0.1550000000E-19
0.4000000000E-01	0.1050000000E-16
0.5000000000E-01	0.4920000000E-15
0.6000000000E-01	0.6070000000E-14
0.7000000000E-01	0.3630000000E-13
0.8000000000E-01	0.1780000000E-12
0.9000000000E-01	0.1320000000E-11
0.1000000000E+00	0.1140000000E-10
0.1100000000E+00	0.7640000000E-10
0.1200000000E+00	0.3800000000E-09
0.1300000000E+00	0.1480000000E-08
0.1400000000E+00	0.4740000000E-08
0.1500000000E+00	0.1300000000E-07

0.1600000000E+00	0.3140000000E-07
0.1800000000E+00	0.1410000000E-06
0.2000000000E+00	0.5450000000E-06
0.2500000000E+00	0.2460000000E-04
0.3000000000E+00	0.7030000000E-03
0.3500000000E+00	0.8240000000E-02
0.4000000000E+00	0.5220000000E-01
0.4500000000E+00	0.2190000000E+00
0.5000000000E+00	0.6910000000E+00
0.6000000000E+00	0.3910000000E+01
0.7000000000E+00	0.1390000000E+02
0.8000000000E+00	0.3730000000E+02
0.9000000000E+00	0.8520000000E+02
0.1000000000E+01	0.1750000000E+03
0.1250000000E+01	0.8170000000E+03
0.1500000000E+01	0.2870000000E+04
0.1750000000E+01	0.7930000000E+04
0.2000000000E+01	0.1810000000E+05
0.2500000000E+01	0.6250000000E+05
0.3000000000E+01	0.1500000000E+06
0.3500000000E+01	0.2890000000E+06
0.4000000000E+01	0.4760000000E+06
0.5000000000E+01	0.9690000000E+06
0.6000000000E+01	0.1550000000E+07
0.7000000000E+01	0.2900000000E+07
0.8000000000E+01	0.4820000000E+07
0.9000000000E+01	0.7330000000E+07
0.1000000000E+02	0.1040000000E+08

Reaction: 4He + 18O → 22Ne

T9	Rate
0.7000000000E-01	0.1530000000E-23
0.8000000000E-01	0.5080000000E-22
0.9000000000E-01	0.9200000000E-21
0.1000000000E+00	0.2730000000E-19
0.1100000000E+00	0.1090000000E-17
0.1200000000E+00	0.2720000000E-16
0.1300000000E+00	0.4200000000E-15
0.1400000000E+00	0.4360000000E-14
0.1500000000E+00	0.3290000000E-13
0.1600000000E+00	0.1920000000E-12
0.1800000000E+00	0.3620000000E-11
0.2000000000E+00	0.3790000000E-10
0.2500000000E+00	0.2950000000E-08
0.3000000000E+00	0.7330000000E-07
0.3500000000E+00	0.9380000000E-06
0.4000000000E+00	0.7160000000E-05
0.4500000000E+00	0.3630000000E-04
0.5000000000E+00	0.1350000000E-03
0.6000000000E+00	0.9740000000E-03
0.7000000000E+00	0.3940000000E-02
0.8000000000E+00	0.1110000000E-01
0.9000000000E+00	0.2460000000E-01

0.1000000000E+01	0.4630000000E-01
0.1250000000E+01	0.1510000000E+00
0.1500000000E+01	0.4340000000E+00
0.1750000000E+01	0.1420000000E+01
0.2000000000E+01	0.4710000000E+01
0.2500000000E+01	0.3270000000E+02
0.3000000000E+01	0.1270000000E+03
0.3500000000E+01	0.3340000000E+03
0.4000000000E+01	0.6880000000E+03
0.5000000000E+01	0.1850000000E+04
0.6000000000E+01	0.3470000000E+04
0.7000000000E+01	0.6050000000E+04
0.8000000000E+01	0.9220000000E+04
0.9000000000E+01	0.1270000000E+05
0.1000000000E+02	0.1640000000E+05

Reaction: 4He + 4He + 4He → 12C

T9	Rate
0.1000000000E-01	0.2930000000E-70
0.1100000000E-01	0.5940000000E-68
0.1200000000E-01	0.6590000000E-66
0.1300000000E-01	0.4460000000E-64
0.1400000000E-01	0.2010000000E-62
0.1500000000E-01	0.6400000000E-61
0.1600000000E-01	0.1530000000E-59
0.1800000000E-01	0.4220000000E-57
0.2000000000E-01	0.5450000000E-55
0.2500000000E-01	0.1110000000E-50
0.3000000000E-01	0.1460000000E-46
0.4000000000E-01	0.5310000000E-40
0.5000000000E-01	0.1040000000E-35
0.6000000000E-01	0.1200000000E-32
0.7000000000E-01	0.3000000000E-30
0.8000000000E-01	0.9680000000E-28
0.9000000000E-01	0.2520000000E-25
0.1000000000E+00	0.2380000000E-23
0.1100000000E+00	0.9640000000E-22
0.1200000000E+00	0.2070000000E-20
0.1300000000E+00	0.2720000000E-19
0.1400000000E+00	0.2430000000E-18
0.1500000000E+00	0.1600000000E-17
0.1600000000E+00	0.8220000000E-17
0.1800000000E+00	0.1220000000E-15
0.2000000000E+00	0.1020000000E-14
0.2500000000E+00	0.4220000000E-13
0.3000000000E+00	0.4570000000E-12
0.3500000000E+00	0.2330000000E-11
0.4000000000E+00	0.7490000000E-11
0.4500000000E+00	0.1780000000E-10
0.5000000000E+00	0.3450000000E-10
0.6000000000E+00	0.8620000000E-10
0.7000000000E+00	0.1550000000E-09
0.8000000000E+00	0.2270000000E-09

0.9000000000E+00	0.2930000000E-09
0.1000000000E+01	0.3480000000E-09
0.1250000000E+01	0.4300000000E-09
0.1500000000E+01	0.4490000000E-09
0.1750000000E+01	0.4370000000E-09
0.2000000000E+01	0.4160000000E-09
0.2500000000E+01	0.3920000000E-09
0.3000000000E+01	0.4160000000E-09
0.3500000000E+01	0.4770000000E-09
0.4000000000E+01	0.5550000000E-09
0.5000000000E+01	0.7040000000E-09
0.6000000000E+01	0.8030000000E-09
0.7000000000E+01	0.8480000000E-09
0.8000000000E+01	0.8520000000E-09
0.9000000000E+01	0.8280000000E-09
0.1000000000E+02	0.7900000000E-09

Reaction: n + 4He + 4He → Be7

T9	Rate
0.1000000000E-02	0.3900000000E-58
0.2000000000E-02	0.2500000000E-46
0.3000000000E-02	0.1350000000E-40
0.4000000000E-02	0.5580000000E-37
0.5000000000E-02	0.2110000000E-34
0.6000000000E-02	0.1960000000E-32
0.7000000000E-02	0.7390000000E-31
0.8000000000E-02	0.1480000000E-29
0.9000000000E-02	0.1880000000E-28
0.1000000000E-01	0.1690000000E-27
0.1100000000E-01	0.1160000000E-26
0.1200000000E-01	0.6370000000E-26
0.1300000000E-01	0.2950000000E-25
0.1400000000E-01	0.1180000000E-24
0.1500000000E-01	0.4170000000E-24
0.1600000000E-01	0.1330000000E-23
0.1800000000E-01	0.1050000000E-22
0.2000000000E-01	0.6290000000E-22
0.2500000000E-01	0.2990000000E-20
0.3000000000E-01	0.5050000000E-18
0.4000000000E-01	0.1900000000E-14
0.5000000000E-01	0.2580000000E-12
0.6000000000E-01	0.6430000000E-11
0.7000000000E-01	0.6140000000E-10
0.8000000000E-01	0.3240000000E-09
0.9000000000E-01	0.1150000000E-08
0.1000000000E+00	0.3120000000E-08
0.1100000000E+00	0.6920000000E-08
0.1200000000E+00	0.1330000000E-07
0.1300000000E+00	0.2270000000E-07
0.1400000000E+00	0.3570000000E-07
0.1500000000E+00	0.5230000000E-07
0.1600000000E+00	0.7250000000E-07
0.1800000000E+00	0.1220000000E-06

0.2000000000E+00	0.1820000000E-06
0.2500000000E+00	0.3480000000E-06
0.3000000000E+00	0.4990000000E-06
0.3500000000E+00	0.6140000000E-06
0.4000000000E+00	0.6900000000E-06
0.4500000000E+00	0.7330000000E-06
0.5000000000E+00	0.7500000000E-06
0.6000000000E+00	0.7340000000E-06
0.7000000000E+00	0.6840000000E-06
0.8000000000E+00	0.6220000000E-06
0.9000000000E+00	0.5600000000E-06
0.1000000000E+01	0.5000000000E-06
0.1250000000E+01	0.3780000000E-06
0.1500000000E+01	0.2890000000E-06
0.1750000000E+01	0.2260000000E-06
0.2000000000E+01	0.1800000000E-06
0.2500000000E+01	0.1210000000E-06
0.3000000000E+01	0.8870000000E-07
0.3500000000E+01	0.6830000000E-07
0.4000000000E+01	0.5480000000E-07
0.5000000000E+01	0.3830000000E-07
0.6000000000E+01	0.2880000000E-07
0.7000000000E+01	0.2260000000E-07
0.8000000000E+01	0.1810000000E-07
0.9000000000E+01	0.1480000000E-07

Reaction: 4He + 16O → 20Ne

T9	Rate
0.1000000000E+00	0.7960000000E-26
0.1100000000E+00	0.1090000000E-24
0.1200000000E+00	0.1100000000E-23
0.1300000000E+00	0.8680000000E-23
0.1400000000E+00	0.5600000000E-22
0.1500000000E+00	0.3040000000E-21
0.1600000000E+00	0.1430000000E-20
0.1800000000E+00	0.2220000000E-19
0.2000000000E+00	0.2550000000E-18
0.2500000000E+00	0.4960000000E-15
0.3000000000E+00	0.3560000000E-12
0.3500000000E+00	0.3970000000E-10
0.4000000000E+00	0.1360000000E-08
0.4500000000E+00	0.2140000000E-07
0.5000000000E+00	0.1960000000E-06
0.6000000000E+00	0.5530000000E-05
0.7000000000E+00	0.6120000000E-04
0.8000000000E+00	0.3740000000E-03
0.9000000000E+00	0.1520000000E-02
0.1000000000E+01	0.4670000000E-02
0.1250000000E+01	0.3430000000E-01
0.1500000000E+01	0.1250000000E+00
0.1750000000E+01	0.3070000000E+00
0.2000000000E+01	0.5910000000E+00
0.2500000000E+01	0.1430000000E+01

```
0.3000000000E+01    0.2540000000E+01
0.3500000000E+01    0.3860000000E+01
0.4000000000E+01    0.5400000000E+01
0.5000000000E+01    0.9620000000E+01
0.6000000000E+01    0.1700000000E+02
0.7000000000E+01    0.2990000000E+02
0.8000000000E+01    0.5120000000E+02
0.9000000000E+01    0.8240000000E+02
0.1000000000E+02    0.1240000000E+03
```

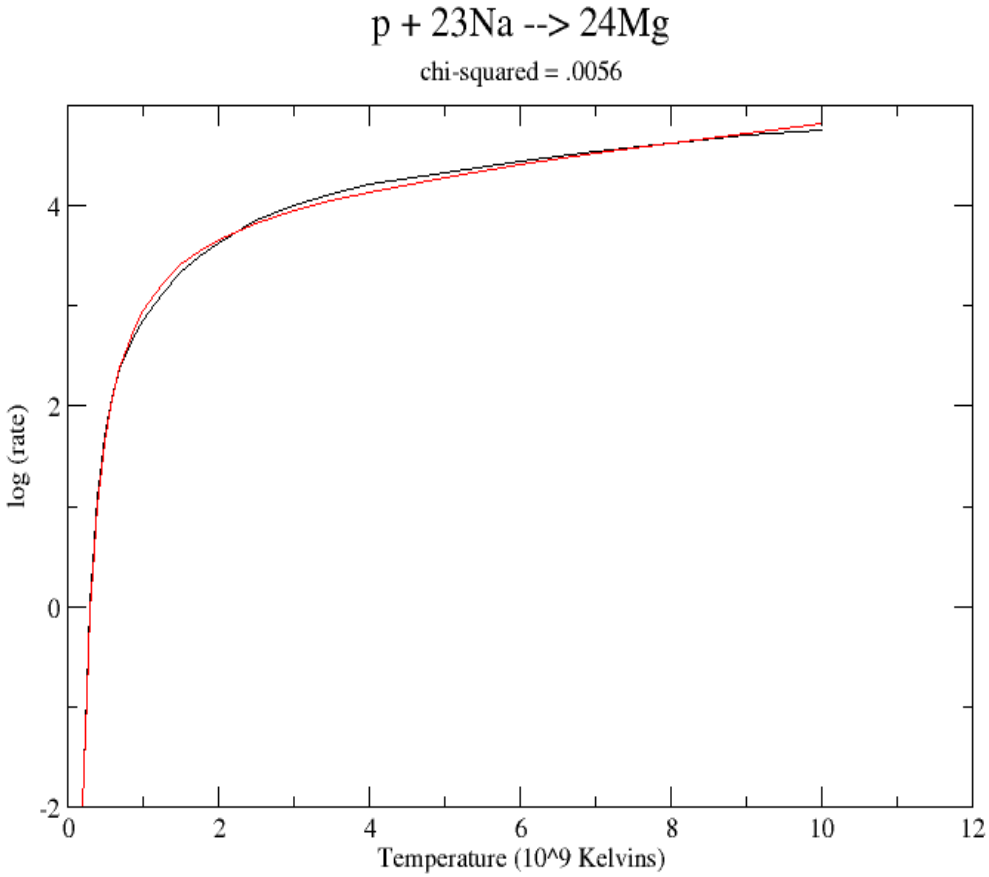

Fig. A1 log (rate) vs. temperature for p + 23Na → 24Mg

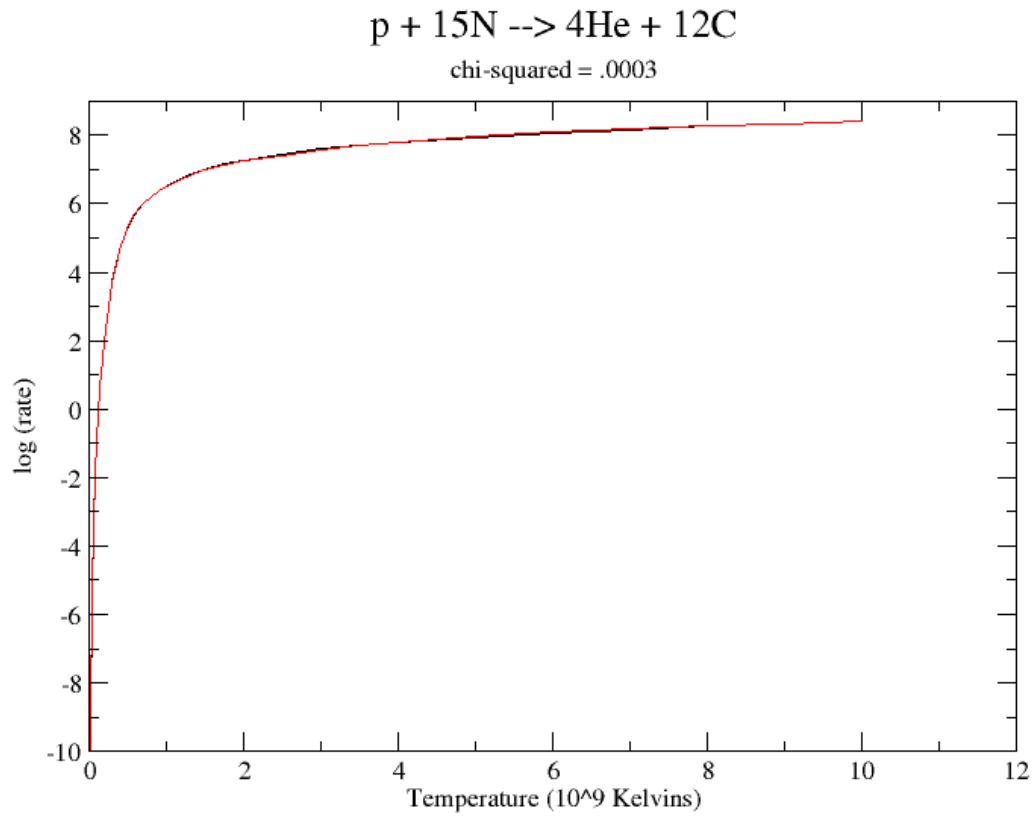

Fig. A2 log (rate) vs. temperature for p + 15N → 4He + 12C

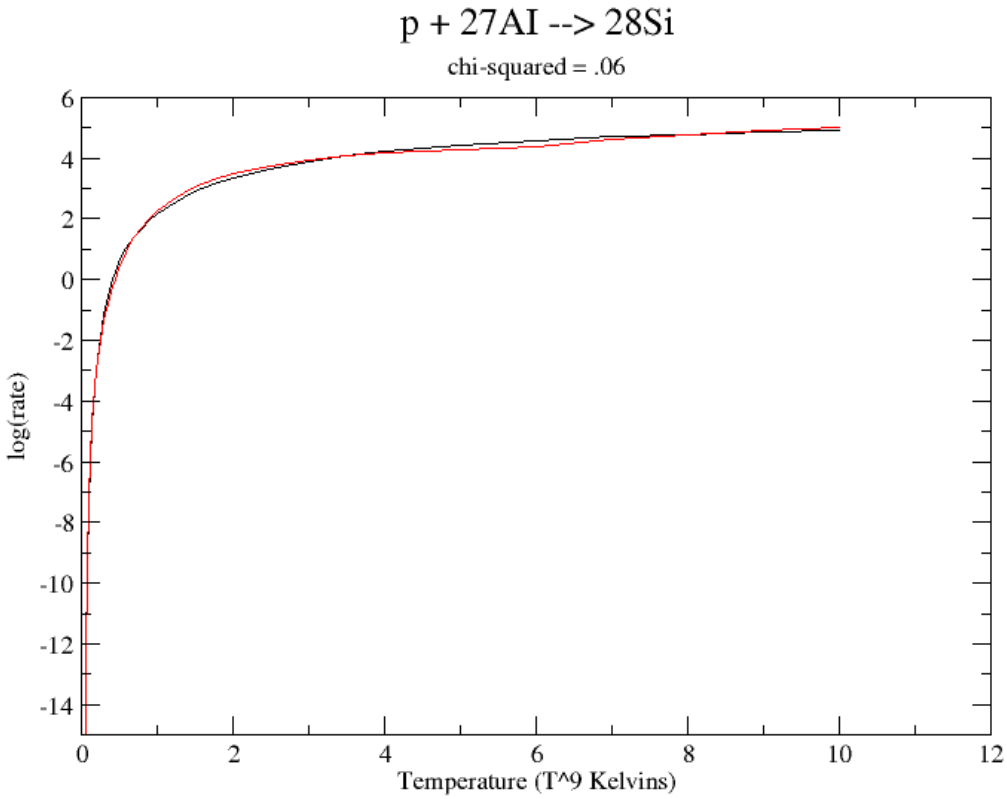

Fig. A3 log (rate) vs. temperature for p + 27Al → 28Si

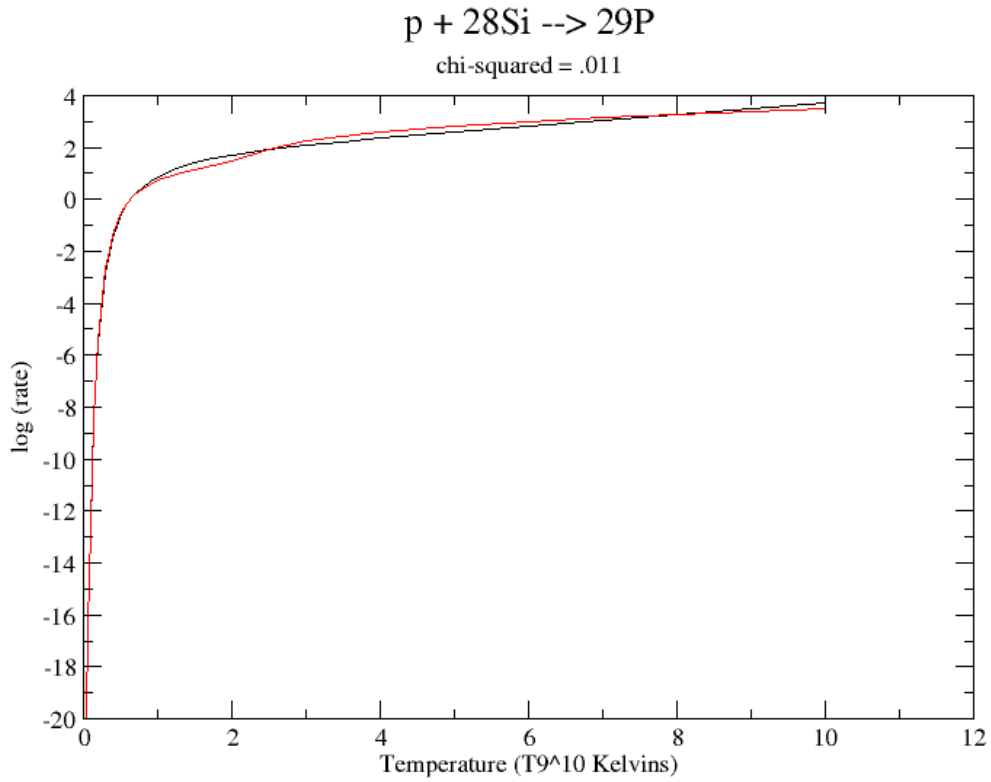

Fig A4 log (rate) vs. temperature for p + 28Si → 29P

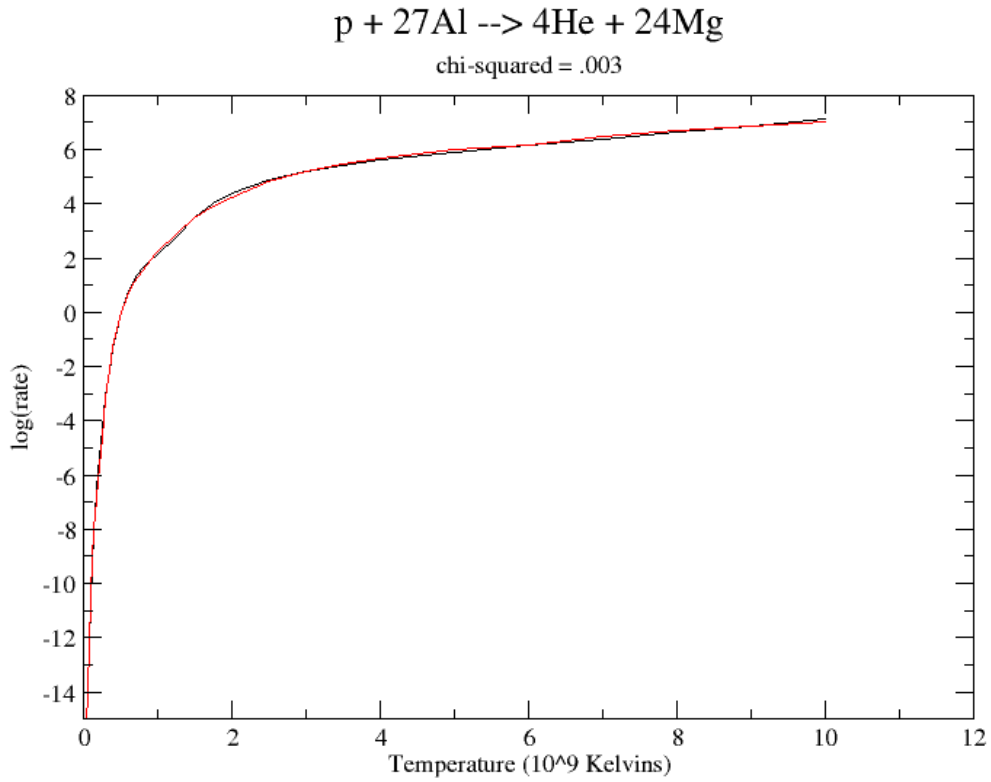

Fig A5 log (rate) vs. temperature for p + 27Al → 4He + 24Mg

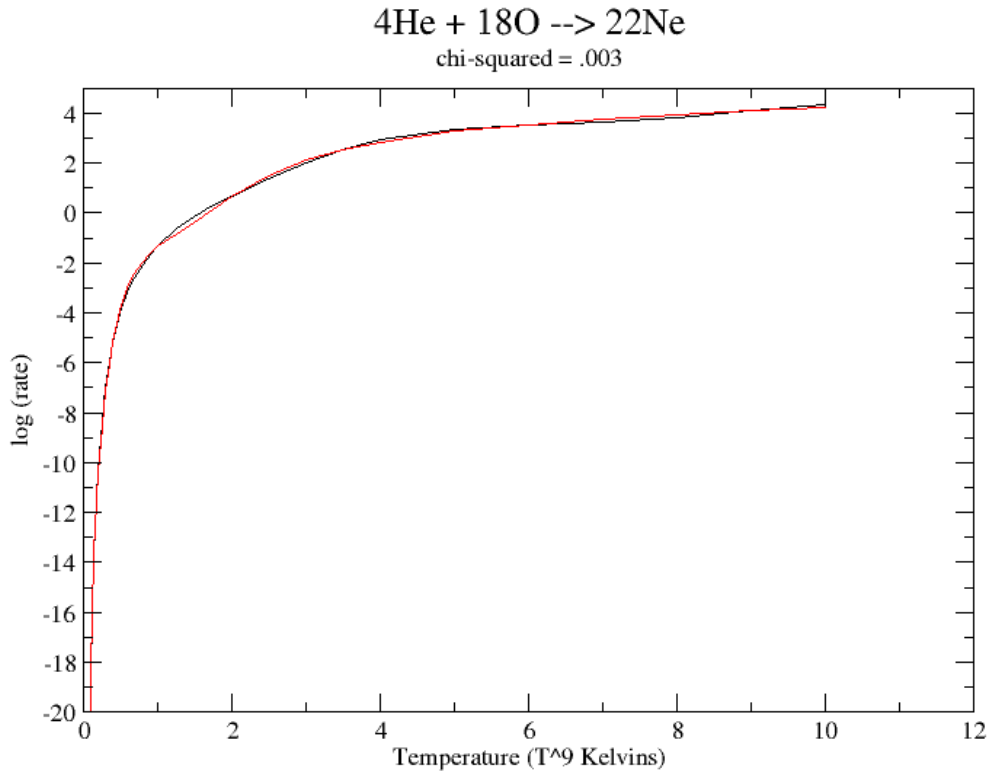

Fig. A6 log (rate) vs. temperature for 4He + 18O → 22Ne

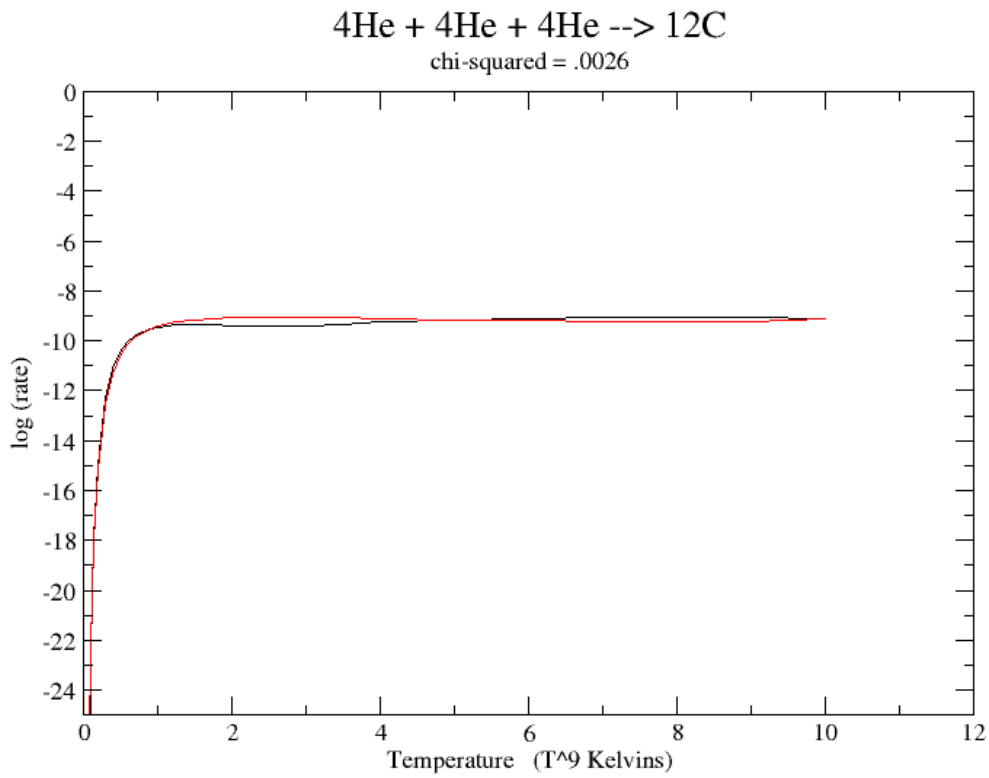

Fig. A7 log (rate) vs. temperature for 4He + 4He + 4He → 12C

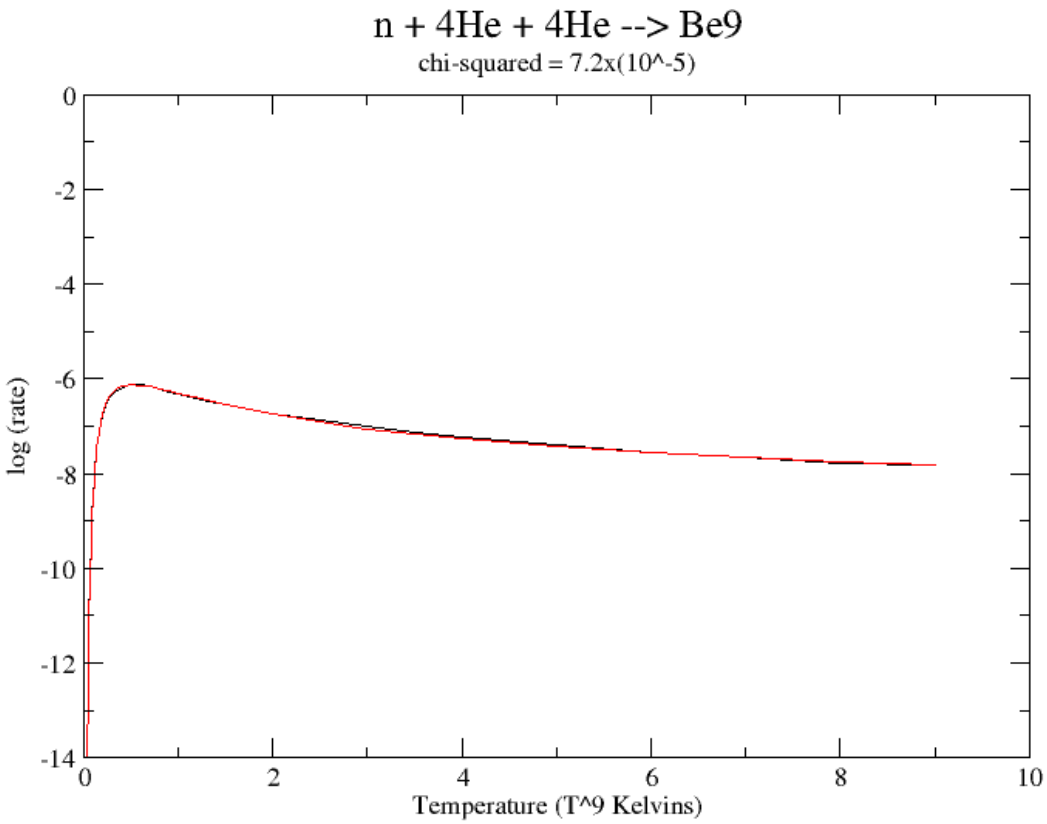

Fig. A8 log (rate) vs. temperature for n + 4He + 4He → Be7

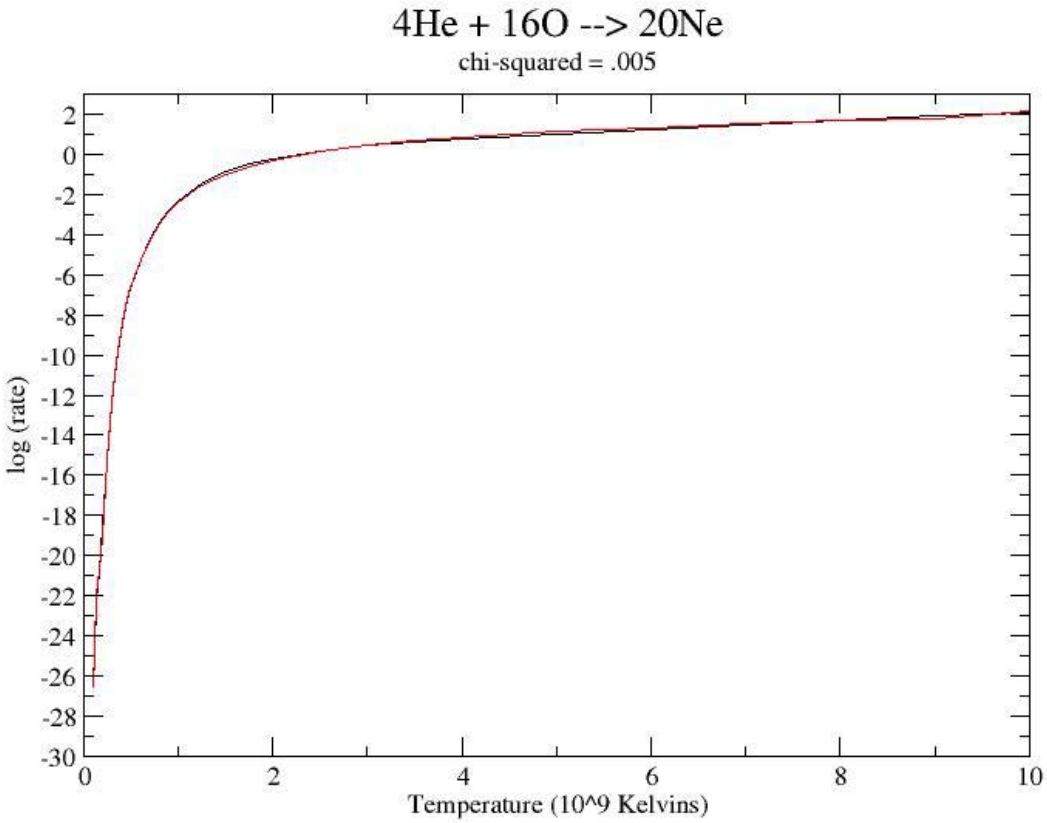

Fig A9 log (rate) vs. temperature for 4He + 16O → 20Ne

Vita

Kevin Richard Williams was born in Greenville, SC on October 3, 1976. He was raised in Williamston, SC and went to grade school at West Pelzer Primary School, and junior high school at Palmetto Middle School. He graduated from Palmetto High School in 1995. He then attended Furman University in Greenville, SC, where he earned a B.S. in physics in 2000. From there, he enrolled in the University of Tennessee, Knoxville in the M.S. program.