Heterogeneous Catalysis and Reaction Dynamics

by

Mason Milburn

Departmental Honors Thesis
The University of Tennessee at Chattanooga

Project Director: Dr. H. Douglas Kutz
Examination Date: 3/30/00

Doug Kutz
John Lynch
Robert Marlowe
Greg O'Dea
Tom Rybolt

Examining Committee Signatures:

Tom Rybolt

John A. Lynch

Robert L. Marlowe

Chairperson, University Departmental Honors Committee

Table of Contents

I. Abstract

The purpose of this research was to develop computer software that accurately and efficiently displayed the complex dynamics involved with chemical reactions occurring on catalytic surfaces. Through computer-generated three-dimensional animations, this computer program will allow students and teachers to visually examine the details of a particular surface reaction. In order to accomplish this goal, a great deal of preliminary work had first be completed before the actual creation of the computer program could begin.

The study of reaction dynamics has been a focus of research for many years; however, studies into the dynamics of surface reactions was a fairly recent field of focus. The study of reaction dynamics, which was the examination of chemical reactions at the atomic level, involved taking into account the energy of the atoms, the position of the atoms within the system, and the natural movement of the atoms in the system (Polonyi). These essential parts of reaction dynamics are expressed in complex equations and are referred to as the potential energy surface (PES), the coordinate system for the atoms, and the classical equations of motion.

A thorough literature search was performed to find information on the fundamental principles of surface chemistry and to identify the chemical reactions that have been studied and documented. Through this search, a

particular surface reaction system was identified for which the documentation provided all the necessary information to enable the creation of computer-generated animations. This system consisted of the reaction of hydrogen on a pure silicon surface (Kratzer). This surface reaction, in comparison to most systems, was relatively simple but still included the important aspects of surface chemistry, which students should learn from working through the computer program.

The next step was to develop a complete understanding of the complex potential energy surface function for the H_2-Si system. To accomplish this, the mathematical software Maple V was used to dissect the complicated mathematical equations reported in the literature and to generate wire-frame plots of the potential energy surface (Maple V). Once accomplished, the development of the computer program that will perform the animations of the H_2-Si surface reaction will begin.

II. Introduction

In undergraduate chemistry studies and especially in classroom discussion, a student is expected to learn the specifics of chemical reactions but does not always develop a clear picture of what is actually happening at the atomic level. The ability to visualize chemical reactions remains extremely important in order for students to thoroughly conceptualize the course material. For some students, the visualization of chemical reactions is possible without aids; however, this is not the case for many students. The goal of the research was to help undergraduate students to better understand the dynamics of surface reactions through the visualization of actual chemical reactions. This goal will be accomplished by creating a computer program with the capability of generating three-dimensional animations of surface reactions. These animations will accurately depict the significant factors that come into play at the atomic level in surface reactions and will allow the user to observe the role which each factor plays in determining the outcome of the reaction. This computer program will not only expedite the learning process for the student but will also make the process more enjoyable by allowing for a more "hands-on" experience.

In undergraduate physical chemistry courses, reaction dynamics and potential energy surfaces are commonly covered topics. Reaction dynamics attempts to explain chemical reactions at the atomic level in terms of collision geometry, reactant energy, and chemical bonding. The collision geometry of a chemical reaction refers to the angles on a three-dimensional axes with x, y, and z coordinates by which atomic or molecular species collide. The reactant energy is the sum of the different energies associated with reagent atoms or molecules: translational energy, rotational energy, molecular potential energy, and vibrational energy. The translational energy takes into account the speed and mass of an atom or molecule and is equivalent to the kinetic energy, $1/2mv^2$. The rotational energy represents the kinetic energy associated with the rotation of the molecule. The molecular potential energy is a measure of the bonding energy of a molecule and varies with the distance between atoms. The vibrational energy of a molecule measures the energy stored in the oscillations of a molecule. Finally, the chemical bonding takes into account the bonding strength between atoms. One examines these features to determine the initial conditions that favor the formation of products. The program will display these factors occurring within a surface reaction system. The user will be able to see prepared reactions specifically created to display the features of the chosen surface reaction and also to create new animations based on selected initial conditions for the reaction.

In reaction dynamics, examining a chemical reaction in terms of its potential energy surface is extremely informative. The potential energy surface is a plot of the potential energy of a chemical system as a function of the positions of the atoms. Often represented as wire frame plots or contour plots, a potential energy surface can tell one a great deal about the energetics of a particular chemical system (see Figure 1).

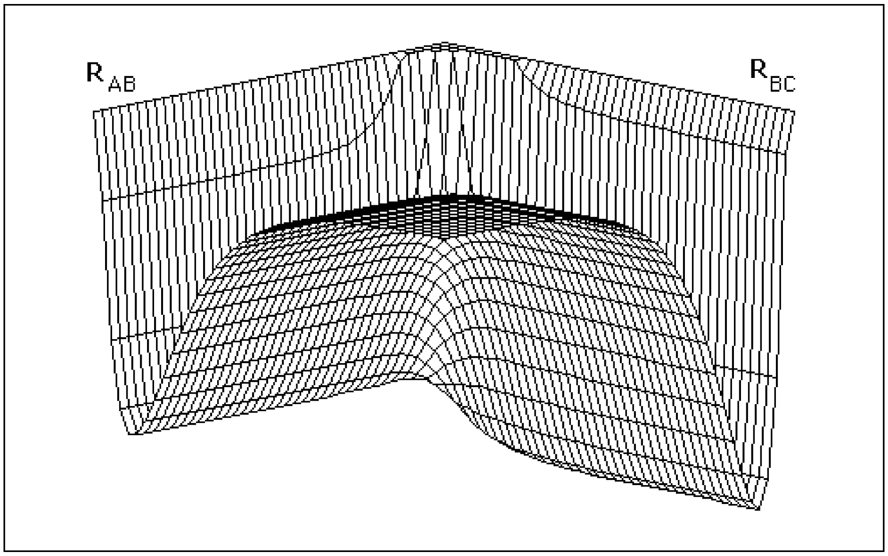


Figure 1

For this potential energy surface, a model A + BC ⇒ AB + C elementary reaction is shown. In this particular plot, the geometry of the collision has been restricted to collinear in that the positions of the three atoms, A-B-C, form a 180° angle and the molecule is not allowed to rotate. Restricting the reaction to collinear geometry is customary so that the potential energy surface may be examined as a function of the two independent variables, R_{AB} and R_{BC}, where R is the distance between atoms. Reactants A + BC are located on the lower left portion of the potential energy surface, and the products AB + C are located on the lower right portion of the surface. The transition state appears as a saddle point in the lower central portion of the surface. One can see the reaction is exothermic (energy releasing) rather than endothermic (energy absorbing) by looking at the potential energy surface and observing that the products are lower in energy than the reactants. The plot also shows that this reaction has a small barrier at the saddle point that must be overcome in order for the formation of products to occur. This potential energy surface represents the simplest of plots. Because more coordinates are needed to describe a surface reaction, the potential energy function for a surface reaction will be more complex than that of a gas-phase reaction. While the gas-phase potential energy surface contains only three variable, a surface reaction potential energy surface can contain more than five variables. Furthermore, a surface reaction can have more than three coordinates and three momenta to integrate while a gas phase can have only three. As one can see, relatively speaking a gas-phase potential energy surface is simple in comparison to the extremely complex potential energy surface of a surface reaction. Within the completed project, the potential energy surfaces for a variety of H_2-Si systems will be included, either in the documentation of the computer program or within the program itself.

The research has for some time dealt with reaction dynamics, potential energy surfaces, and animations of these chemical reaction features. Previous work of a similar nature has been done using computer programs that: 1) perform animations of colinear chemical reactions, 2) depict potential energy surfaces, and 3) generate three-dimensional animations. In previous work, Dr. Kutz and co-workers developed a computer program that simulates animations of colinear chemical reactions and displays potential energy surfaces of these reactions (Copeland). In our work together, a computer program that generates three-dimensional animations of the $H_2 + O \Rightarrow OH + H$ gas-phase reaction have been successfully completed. This program contains several of the programming methods that will be necessary to create accurate animations of a surface reaction. Potential energy plots were also generated for a variety of $H_2 + O \Rightarrow OH + H$ systems. Because of this successful work, we felt confident in our ability to proceed with the challenge of creating a computer program to simulate the much more complex reaction system associated with reactions on surfaces.

III. Overview of Surface Reactions

For hundreds of years scientists have been aware of a surface's ability to catalyze chemical reactions. Recently, studies into the function of surface chemistry as a catalytic process have focused more and more on the interactions at the atomic level. This new field of focus was primarily due to the vast importance surface chemistry has taken on both for the consumer and within industrial production, and is due in part to the advances made in chemical instrumentation. With the ever-increasing importance of surface chemistry, scientists have begun to concentrate on understanding how these catalytic surfaces operate at the atomic level so that one can more accurately predict how a surface will operate for a particular chemical reaction or how to modify the structure of a surface in order to increase its reaction efficiency. These studies delved into several atomic level factors: a description of a surface's dynamic structure at the atomic level, the factors that determine the adsorption of gaseous atoms or molecules on the surface, the mechanism by which surfaces catalyze chemical reactions, the desorption of atoms or molecules into the gaseous state, and the standard mechanisms that most surface reactions follow (Masel pg 443).

Industrially and commercially, surface chemistry is perhaps one of the most important utilizations of chemistry: "90% of all chemicals are produced via a heterogeneously catalyzed process where a reaction occurs on the surface of a catalyst" (Masel pg 1). One can find the importance of surface chemistry industrially in the reduction of nitrogen gas in the presence of hydrogen and an iron catalyst (Masel pg 457) to biologically in catalyzed reactions that "control functioning of the brain and other vital organs" (Somorjai pg 443)**.** Although many reactions, such as $H_2 + 1/2O_2 \Rightarrow H_2O$, are thermodynamically favorable, the rate at which the reaction occurs under normal conditions can take as long as years before significant accumulation of products is observed; whereas, if a platinum catalyst is added to the system, the reaction occurs almost instantaneously. The relatively slow reaction rate is due to the energy barrier that must be overcome before the reaction can occur. This energy barrier is known as the activation energy of the reaction. The ability of surfaces to decrease the energy of activation, thereby increasing the rate of the reaction, is one of the ways that surfaces allow the producer to efficiently make a chemical product in bulk quantities. As surface chemistry continues to gain importance in the everyday world, studies will continue to look further into the specifics of how surfaces operate at the atomic level.

To the naked eye, metallic surfaces could be seen as uniform and non-dynamic structures; however, if viewed at the atomic level, one would see a metallic surface as a dynamic structure made up of metallic ions within a sea of electrons: "The surface… exhibits dynamic restructuring responding to its changing local environment" (Somorjai pg 42). The sharing of electrons and the repulsion of the metals' positive nuclei fix the metal ions in place and give the surface a rigid yet malleable structure. Most metallic catalysts seem to be homogeneous in that the surface is in a single plane and smooth; in fact, "the first physical model of a surface was one of a smooth discontinuity" (Somorjai pg 36). However, in reality most available data points to the conclusion that "solid surfaces are heterogeneous on the atomic scale (Somorjai 41), meaning that the surface is not uniform but instead contains substantial ledges and terraces: "inspection of any surface by an optical microscope… reveals the presence of irregularities and a great deal of roughness" (Somorjai pg 36).

These areas of the surface present many places of opportunity for gaseous atoms or molecules to adsorb.

One of the reasons metallic surfaces act as such efficient catalysts is due to their ability to quickly compensate for the adsorption of a gaseous species. The formation of bonds between gaseous species and surface atoms “induces rearrangement of the substrate atoms at the adsorption site” (Somorjai pg 412). This rearrangement lowers the bonding strength between the atoms of the surface in order for "maximum bonding and stability of the adsorbate-substrate complex" (Somorjai pg 417). When the adsorbed species, through the transference of energy, causes the vibrational excitation of the lattice (Masel pg 377), the coordination ability of the lattice atoms also allows for the bond length contraction and relaxation between the atoms of the lattice so that the bond strength between substrate and adsorbate can be optimized (Somorjai 74). Furthermore, within a sea of electrons the chemical reactants, products, and multiple transition states are stabilized without a distinctive charge separation (Masel pg 451). With the ability to dissipate the bonding energy of the adsorbed atom or molecule within the dynamic lattice, surfaces can form strong bonds with adsorbates while maintaining a cohesive structure.

Unlike the positions of surface atoms, which are essentially fixed, gaseous atoms or molecules are allowed free movement throughout the reaction system. For the surface to be effective at catalyzing a chemical reaction there must be some mechanism that allows the gaseous species to "stick" to the surface. This mechanism by which gaseous atoms or molecules form bonds with the surface atoms is called adsorption. The adsorption of gaseous species on a surface will likely occur whenever the kinetic energy of the gaseous species is sufficiently lower than the “well depth of the attractive surface potential” (Somorjai pg 332). The kinetic energy of the gaseous species can be effected by system factors such as temperature, pressure, and inter-collisions between species. Kinetic energy can also be lessened by the gaseous species' collision with the surface.

When a gaseous species collides with the surface of a catalyst, the atom or molecule will either scatter away from the surface, become trapped by the surface, or stick to the surface (Masel pg 355) (see Figure 2). Elastic scattering occurs whenever an atom or molecule bounces off an undeformable surface with no significant loss of energy (Masel pg 355). In contrast, inelastic scattering occurs whenever the gaseous molecule loses some energy due to the collision with a deformable surface but not enough to stick to the surface (Masel pg 355). In trapping, the gaseous species can lose enough translational energy through the collision to remain on the surface yet not form a bond with the surface atoms (Masel pg 355). When the gaseous species loses sufficient energy to remain on the surface and forms a chemical bond with a surface atom, the molecule is said to "stick" on the surface and, therefore, be adsorbed (Masel pg 356).

Figure 2

The amount of kinetic energy that is lost during the collision of the gaseous species with the surface is dependent on several factors. If the surface is considered undeformable, the gaseous species can bounce off the surface without a significant loss of kinetic energy; however, the gaseous species can lose a great deal of its kinetic energy if the collision is with a deformable surface (Masel pg 355). The incident angle, the angle that the gaseous species strikes the surface relative to the normal plane, is also important in the transfer of energy (Masel pg 359) (see Figure 3).

Figure 3

If the incident angle is small, then the gaseous molecule is said to have a "grazing angle" (Masel pg 366). With this, the gaseous species continues to lose energy along the surface normal since the original collision is not sufficient to bounce the atoms or molecules away from the surface (Masel pg 368). In this situation, the trapping probability will be lower for light molecules with high translational energy and higher for heavy molecules with low translational energy (Masel pg 368). The adsorption potential is further dependent on the impact parameter, *b*, the distance from the center of the surface atom the gaseous species strikes (Masel pg 364). If the gaseous species strikes directly in the center of the surface atom (b=0), the transfer of energy is considerable, while if the impact parameter is greater than zero (b>0), the transfer of energy will be less (Masel pg 364). Once the gaseous species is adsorbed on the surface, the catalytic action of the surface begins.

The specific purpose of a surface is to increase the rate at which a chemical reaction occurs. The surface accomplishes this feat several different ways.

First, by holding an adsorbed molecule or atom in place, the surface makes the collision of a gaseous species with adsorbed species more inelastic than it would be if both were in the gaseous phase. This will make the collision energy and transfer of energy much greater and subsequently make the chemical reaction more likely. A surface catalyst can significantly augment the rate of chemical reactions by decreasing the activation energy, E_a. With the activation energy lowered, the reaction barrier can more easily be crossed. The way that surfaces lower the activation energy of chemical reactions is through the bonding action of surface and adsorbed species (Masel pg 699). When adsorption occurs, the bond formed between the surface atoms and adsorbed species at the transition state will lower the bond strength between the atoms of the adsorbed species (Masel pg 699). With the bond strength lowered or the bond already broken, a molecule colliding with the adsorbed species will be more likely to break the bonds of the molecule and form the products. Because of this, the activation energy of the reaction is lowered.

The last step of surface catalysis occurs when the adsorbed molecule or atom breaks the bond between itself and the surface and re-enters the gaseous phase: desorption. The bond is broken whenever the energy is sufficient to overcome the bonding strength between the surface and the adsorbed species. This can occur either through collisions of gaseous species with the adsorbed species or from energy within the system.

The last step of surface catalysis occurs when the adsorbed molecule or atom breaks the bond between 4).

Figure 4

In the Langmuir-Hinshelwood mechanism, species A and B begin in the gaseous phase and adsorb on the surface. Species B reacts with species A. A chemical bond forms between the two species, and the newly formed molecule desorbs into the gaseous phase. In the Rideal-Eley mechanism, species A first adsorbs on the surface. Species B, still in the gaseous phase, collides with species A and reacts to form a chemical bond. The molecule then desorbs. Finally in the Precursor mechanism, species A adsorbs on the surface. Species B in the gaseous phase collides with the surface and "enters a mobile precursor state". Species B slides across the surface to collide with species A. A chemical bond is formed between species A and B, and the molecule desorbs into the gaseous state (Masel pg 445.

IV. Literature Search

An essential step in the development of a surface reaction program was to build a thorough library of the literature specifically pertaining to studies into the reaction dynamics of surface chemistry. The literature search had a two-fold purpose: 1) Permit an exhaustive overview of the basics of surface chemistry and 2) Give insight into which chemical reactions have been and are currently being researched with respect to their reaction dynamics. Literature sources on general surface chemistry allowed the building of a basic understanding that would later be applied to refining the literature search. A general understanding of surface chemistry was essential in translating the complex information of the detailed studies into the reaction dynamics of specific chemical reactions. Because the study of the reaction dynamics in surface chemistry was a relatively recent focus of study, a broad literature search was necessary to reveal the chemical reactions that have been thoroughly studied. Finally, the initial literature search also allowed for the determination the chemical reaction on which to focus the research.

In preparation for the initial literature search, two main sources were used for general knowledge on surface chemistry: Principles of Adsorption and Reactions on Solid Surfaces by Richard L. Masel, and Introduction to Surface Chemistry and Catalysis by Gabor A. Somorjai. These books gave excellent overviews on the basics of surface chemistry at the atomic level and allowed for the specification the initial search. The refinement of the search terminology that would be used in the literature search was essential in order to obtain a list of references that was both specific and thorough. In the literature search, the search engine STN Chemical Abstracts was employed. With this search, several sources were found that pertained to the current research (Appendix I). This literature search furthered the amount of general information on surface reactions and also revealed several surface reactions that have been studied with respect to their reactions dynamics: $H_2 \Rightarrow H + H$, $H + H \Rightarrow H_2$, $N_2 \Rightarrow N + N$, $CO + O \Rightarrow CO_2$, $H + CO_2 \Rightarrow OH + CO$. As stated previously, in order to simulate chemical reactions one must have a coordinate system for the position of the atoms during the reaction, a potential energy function that specifies the molecular energy between the atoms as a function of atomic distance, and equations of motion to accurately describe the trajectories of the atoms. For each of the reactions listed above, an exhaustive literature search was performed in order to determine if these requirements had been met for any of the surface reactions. Because of the want to work with a fairly complex reaction system that would include at the very least a three-atom reaction, great detail for literature was searched in order to find documentation that would allow for the pursuit of this. One of the reactions that a great deal of reaction dynamics documentation existed on was the $H + CO_2 \Rightarrow OH + CO$ reaction. Most of this research had been documented by one individual, George C. Schatz, so another exhaustive search was done on all the literature he had published on this particular reaction (Appendix I). From this search the necessary requirements that would allow for the use of the reaction in the program were unable to be satisfied. Therefore, the final search yielded positive results only for simple two atom reactions. These results greatly limited the scope of which particular chemical reactions could be simulated and eventually led to the choosing of the H_2-Si reaction as the system of focus. Through the literature search, a great deal has been learned about the knowledge and limitation of dynamics studies of surface chemistry and a surface reaction system has been found; therefore, this goal has been completely satisfied.

V. Overview of the H_2/Si System

Through the literature search, a large library of reaction dynamics information was found on the particular surface reaction system that had been chosen for the program. From these sources, several aspects of the H_2-Si reaction system have been determined to be quite important and should be included in the program. The reaction mechanism by which this chemical reaction proceeds is Rideal-Eley 50% of the time and Precursor the other 50% (Kratzer pg 6752). Further explained, half the time an adsorbed hydrogen atom is struck by a gaseous hydrogen atom to react and form products, while the other half includes an adsorbed hydrogen atom being struck by another hydrogen atom sliding across the surface to form products. Reactions occurring through the Eley-Rideal mechanism “generally show a quite small cross section, of the order of the spatial extension of the reactants ($\sim Å^2$),” while a reaction that uses the Precursor mechanism “may have a large cross section” (Kratzer pg 6753). When formation of products occurs, the energy that is released in the reaction is primarily exhibited as translational energy. These features of the H_2-Si system should be displayed in the surface reaction program.

A great deal about the H_2-Si reaction system can be learned through a study of the potential energy function, coordinate system function, and classical equations of motion for the system. Within the actual computer program, these factors will be used to create accurate three-dimensional animations that will display the features of the H_2-Si chemical system. In order to complete the program, the mathematical functions within the literature must first be replicated and fully understood. The first of these functions that was attempted was the potential energy function. As stated previously, the potential energy surface of a surface reaction can be quite complicated; however, if the system were restricted to a co-linear geometry, the potential energy surface would not be very difficult to produce (see Figure 5).

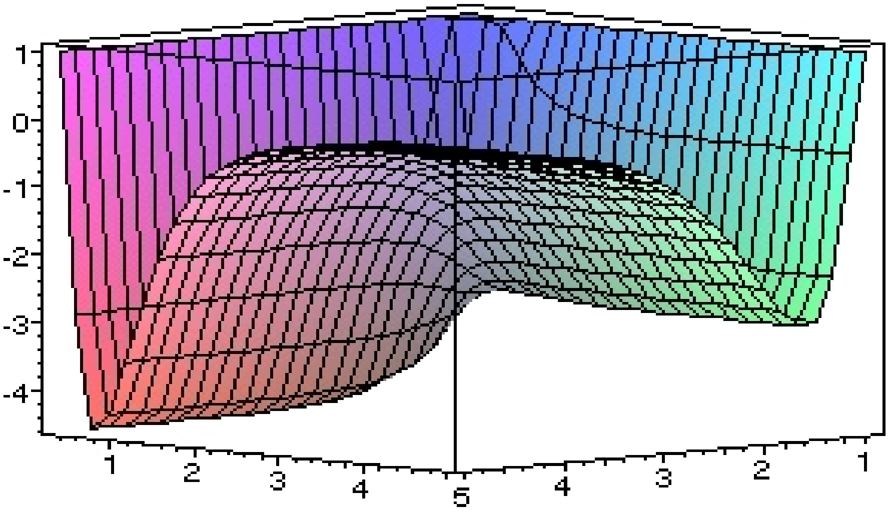


Figure 5

In this system, a hydrogen molecule approached a “naked” silicon surface perpendicular to the surface atoms. The potential energy surface showed the energy of the interaction between the closest hydrogen atom and a silicon atom with respect to atomic distance. The reactants H_2 + Si are found in the lower right portion of the surface, and products H-Si + H are found in the lower left portion of the surface. From the surface one can tell that this reaction is exothermic and has a small barrier (Kratzer pg 6794).

Although informative for this particular system, this potential energy surface did not give a depiction of a realistic system; therefore, it was imperative to produce a potential energy diagram that showed a realistic reaction of this chemical system. Within the literature a potential energy function was found that was based on a system in which a gaseous hydrogen atom interacts with an adsorbed hydrogen atom and two silicon surface atoms (Kratzer pg 6794). This potential energy surface gives an accurate depiction of a frequently studied H_2-Si system (see Figure 6).

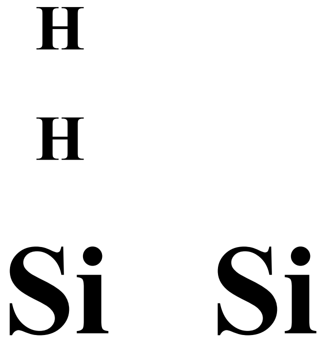


Figure 6

The function took into account the repulsive energy associated with a gaseous hydrogen atom approaching an adsorbed hydrogen atom and the attractive energy of the silicon atoms to the gaseous hydrogen and adsorbed hydrogen with respect to atomic distance (Kratzer pg 6754). In order to reproduce the diagram within the literature, the Maple V software was used to plot the potential energy function (see Figure 7 and Appendix 3).

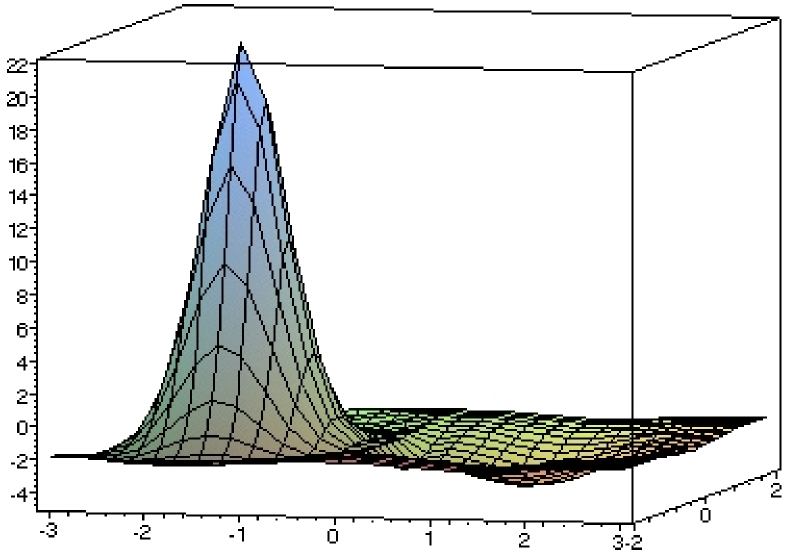


Figure 7

The diagram showed a gaseous-phase hydrogen atom approaching an adsorbed hydrogen atom, left-center of surface, and then approaching a "naked" silicon atom (Kratzer 6754). In the potential energy diagram, one can see the energy of repulsion when the gaseous hydrogen atom is close to the adsorbed hydrogen atom and the energy of attraction as the gaseous hydrogen atom approaches a "naked" silicon atom. This diagram closely matched the diagram found within the literature. In the work with the H_2-Si system, much has been learned about the specifics of this surface reaction system, and one of the more complicated potential energy surfaces has been accurately reproduced.

VI. Programming Language

In the previous work on the gas-phase $O + H_2 \Rightarrow OH + H$ program, the primary programming language was Think Pascal for Macintosh. During this research, we became very familiar and confident with writing computer programs using this particular programming language. Although our competency with Think Pascal was sufficient in order to write the surface chemistry program, the version that had been used for the previous program did not work properly on the Macintosh Quadra or Power Macs. This was not a problem when developing the $O + H_2 \Rightarrow OH + H$ program, which was created on older computers, since the computational requirements were somewhat limited in scope. However, the surface reaction program would require significantly more computational power to run properly. Because of the enormous amount of computation power that would be necessary to properly run the surface reaction program, to continue using the current version of Think Pascal was irrational; therefore, an alternative language needed to be found that could provide the necessary computational power while remaining similar to what was used previously.

In order to find programming software that fit the criteria, several different sources were examined. From this, two possible programming languages to use were uncovered: an updated Think Pascal and CodeWarrior. The updated Think Pascal is a freeware version created by former Think Pascal users and was available only through the Internet. CodeWarrior was commercial programming software by Metrowerks that allowed the user to write programs in a variety of languages and create freestanding applications for both Macintosh and PC computers (CodeWarrior). Both choices were somewhat acceptable since both allowed for the writing of a program in the Pascal language on modern computers. To decide which programming software to use, the different abilities of each were examined very closely to determine which would best accomplish the overall goals of the project.

Several factors were examined to determine the best programming language to use: 1) Were example programs available for reference? 2) Was the Pascal language similar enough to the Think Pascal language that had been previously used in order that older programs would properly run on it? 3) Would the programming language allow for the creation of a program that would run a program on both Macintosh and PC computers? The first of these criteria was extremely important for efficiently creating the program. Without working examples to go by, any small change in the command language could be difficult to detect and might not allow the program to properly run. With working examples one would have a standard to follow when learning how to write the program. CodeWarrior came with a variety of example programs while the updated Think Pascal freeware did not (CodeWarrior). This lack of examples would not be a great deterrent from the updated Think Pascal if the older programs ran without significant changes using the updated version; however, some of the programs did work and some did not. As expected, none of the older programs worked using the Pascal in CodeWarrior without a great deal of effort, but since there were programs to reference, this was not a serious concern. When examining the last factor in determining which programming language to use, only CodeWarrior allowed for the writing of a program that would work on both Macintosh and PC (CodeWarrior). Because the criteria were best satisfied by CodeWarrior, the next step was to learn how to create a program using the Code Warrior software.

Although CodeWarrior allowed for the writing of programs using a Pascal language, parts of the Pascal language structure and program creation process differed a great deal from the Think Pascal version that had been used previously. This was quite evident when examining the Pascal programs and trying to get older programs made with Think Pascal to work. The programs that were created with Think Pascal consistently caused the computer to lock up when the code was imputed verbatim into a CodeWarrior format. The non-functioning of these programs was due to the fact that, although CodeWarrior used a Pascal language, only the mathematics remained completely the same between it and Think Pascal while the visual, user interface and command language could vary significantly. This led to the attempt to use the example programs that came with the CodeWarrior software as basis to get the programs to work. An unfortunate hindrance with this method was that most of the example programs would not run properly; therefore, the first course of action was to make several example programs properly work (CodeWarrior).

After considerable effort, one of the differences between Think Pascal and CodeWarrior Pascal was discovered. In Think Pascal, the commands library was automatically accessed when one began a new program; however, CodeWarrior required that one manually input some essential command libraries into the program block. Once this was discovered, some of the example programs were made to work properly utilizing this technique. By examining the programming language used in these programs, a great deal was learned about the other programming methods used to write CodeWarrior applications using Pascal language. Because the language structure differed significantly in some respects from the language structure in Think Pascal, this was a very difficult process and took considerable time to work out. By starting out with simple programs that called up PICT files (a standard file structure for Macintosh computers), more complicated programs that showed animations of a molecule rotating were methodically worked up to. In order to feel comfortable in our ability to write a surface reaction program, programs were needed that allowed one to call up PICT files, calculate the positions of rotating circles with time, use offscreen ports that are necessary in showing a multiple frame animation, region off parts of the circles in order that the animations appear three-dimensional, and have a menu bar at the top of the screen that allows the user to quit the application or switch to different parts of the program. Since some of the older programs contained almost all of these, the main goal was to get these older programs to properly run as CodeWarrior applications. The old programs that were used in this were Region, which defined a set region on the screen and then filled it in with color, Rotate, which showed two circles rotating, and Rotate3D, which showed two circles rotating three-dimensionally. These plus parts added along the way were used to develop an understanding of CodeWarrior. Eventually, these old Think Pascal programs were made to work using CodeWarrior (see Appendix II). This success was a major step forward in the development of a surface reaction program. Through our work with CodeWarrior, many of the essential parts that must be utilized in the final program have been made to work and only a small amount of work, if any, still remains to be done with CodeWarrior before the writing of the program (CodeWarrior).

VII. Current Status of Project

In the last year, persistent work towards our goal has brought us closer to developing a surface reaction program. A thorough literature search has significantly furthered our knowledge of the generalizations found within surface chemistry and of the chemical systems that have been studied with respect to their reaction dynamics. Closer examination of the search uncovered the well-documented H_2-Si chemical system on which to base the program. For this system, the potential energy function reported in the literature has been successfully reproduced using Maple V software, giving us confidence in our ability to accurately incorporate this into the program. The next step in this process will most likely include deciphering the coordinate system and equations of motion within the literature. Once the accurate incorporation of these parts into the program is certain, the development of the program source code that will create the surface reaction application will begin using CodeWarrior. This step should be much clearer due to the accomplishment of our goal to get the older programs to run within the CodeWarrior system and our successful work with similar programs. Because these programs contained much of the programming language that will be used to animate the surface reactions, it is likely that the final program could be completed by entering much of the older language structure after only minor changes. Although a great deal of the preliminary work needed to begin the surface reaction program has been accomplished, this still remains a long-range project for which our work during the past year had laid the foundations.

Appendix I: Literature Search

STN Chemical Abstracts Search

```
LOGINID:SSSCOL8F

PASSWORD:HHHHHHHHHHIIIIIIIII#########LPCFM325

PASSWORD:HHHHHHHHHHIIIIIIIII#########0
TERMINAL (ENTER 1, 2, 3, OR ?):3


* * * * * * * * * *   Welcome to STN International   * * * * * * * * * *

Enter NEWS followed by the item number or name to see news on that
specific topic.

  All use of STN is subject to the provisions of the STN Customer
  agreement.  Please note that this agreement limits use to scientific
  research.  Use for software development or design or implementation
  of commercial gateways or other similar uses is prohibited and may
  result in loss of user privileges and other penalties.

* * * * * * * * * * * * * * STN  Columbus  * * * * * * * * * * * * * * *

FILE 'LCA' ENTERED AT 17:17:07 ON 14 JUN 1999
COPYRIGHT (C) 1999 AMERICAN CHEMICAL SOCIETY (ACS)

LCA IS A STATIC LEARNING FILE

THIS FILE CONTAINS CAS REGISTRY NUMBERS FOR EASY AND ACCURATE
SUBSTANCE IDENTIFICATION.

This file contains CAS Registry Numbers for easy and accurate
substance identification.

=> FIC LE CA

COST IN U.S. DOLLARS                    SINCE FILE     TOTAL
                              ENTRY    SESSION
FULL ESTIMATED COST                          0.21       0.21

FILE 'CA' ENTERED AT 17:17:14 ON 14 JUN 1999
USE IS SUBJECT TO THE TERMS OF YOUR STN CUSTOMER AGREEMENT.
PLEASE SEE "HELP USAGETERMS" FOR DETAILS.
COPYRIGHT (C) 1999 AMERICAN CHEMICAL SOCIETY (ACS)


Copyright of the articles to which records in this database refer is
held by the publishers listed in the PUBLISHER (PB) field (available
for records published or updated in Chemical Abstracts after December
26, 1996), unless otherwise indicated in the original publications.

FILE COVERS 1967 - 12 Jun 1999 (19990612/ED)  VOL 130 ISS 25
```

```
  This file contains CAS Registry Numbers for easy and accurate
  substance identification.

  This file supports REG1stRY for direct browsing and searching of
  all substance data from the REGISTRY file.  Enter HELP FIRST for
  more information.

Now you can extend your author, patent assignee, and title searches
back to 1907.  The records from 1907-1966 now have this searchable
data in CAOLD.  You now have electronic access to all of CA: 1907 to
to 1966 in CAOLD and 1967 to the present in CA on STN.

=> S TRAJECTORY(L)  SURFACE?(L)CATALS YS?

        12974 TRAJECTORY
      1305397 SURFACE?
       559071 CATALYS?
L1          14 TRAJECTORY(L)SURFACE?(L)CATALYS?

=> D L1 1-14 IBIB


L1   ANSWER 1 OF 14  CA  COPYRIGHT 1999 ACS
ACCESSION NUMBER:        130:115566  CA
TITLE:                   Long-range diffusion of K promoter on an ammonia
                   synthesis catalyst surface-ionization of excited
                   potassium species in the sample edge fields
AUTHOR(S):               Engvall, Klas; Kotarba, Andrzej; Holmlid, Leif
CORPORATE SOURCE:        Reaction Dynamics Group, Physical Chemistry,
                   Department of Chemistry, Goteborg University,
                   Goteborg, SE-412 96, Swed.
SOURCE:                  J. Catal. (1999), 181(2), 256-264
                   CODEN: JCTLA5; ISSN: 0021-9517
PUBLISHER:               Academic Press

L1   ANSWER 2 OF 14  CA  COPYRIGHT 1999 ACS
ACCESSION NUMBER:        125:257762  CA
TITLE:                   Reaction of gas phase atomic oxygen with chemisorbed
                   hydrogen atoms on a tungsten surface
AUTHOR(S):               Ree, J.; Shin, H. K.
CORPORATE SOURCE:        Department of Chemistry, University of Nevada, Reno,
                   NV, 89557, USA
SOURCE:                  Chem. Phys. Lett. (1996), 258(1,2), 239-247
                   CODEN: CHPLBC; ISSN: 0009-2614

L1   ANSWER 3 OF 14  CA  COPYRIGHT 1999 ACS
ACCESSION NUMBER:        124:236202  CA
TITLE:                   Aerothermal study of Mars pathfinder aeroshell
AUTHOR(S):               Gupta, Roop N.; Lee, K. P.; Scott, Carl D.
CORPORATE SOURCE:        Langley Research Center, NASA, Hampton, VA, 23681, USA
SOURCE:                  J. Spacecr. Rockets (1996), Volume Date 1996, 33(1),
                   61-9
                   CODEN: JSCRAG; ISSN: 0022-4650
```

L1 ANSWER 4 OF 14 CA COPYRIGHT 1999 ACS
ACCESSION NUMBER: 123:18584 CA
TITLE: Dynamic study of N2/Fe(100) interaction
AUTHOR(S): Tien, Peifang; Shen, Xueying
CORPORATE SOURCE: Dep. Electronic Eng., Tsinghua Univ., Beijing, Peop. Rep. China
SOURCE: Zhenkong Kexue Yu Jishu (1994), 14(5), 320-9
CODEN: CKKSDV; ISSN: 0253-9748

L1 ANSWER 5 OF 14 CA COPYRIGHT 1999 ACS
ACCESSION NUMBER: 120:254220 CA
TITLE: Dynamic study of reaction of oxygen molecule with carbon atom adsorbed on Ni(100) surface. II. Trajectory calculations
AUTHOR(S): Yu, Huagen; Cheng, Jiyuan
CORPORATE SOURCE: Chengdu Inst. Org. Chem., Chin. Acad. Sci., Chengdu, 610015, Peop. Rep. China
SOURCE: Cuihua Xuebao (1994), 15(1), 54-7
CODEN: THHPD3; ISSN: 0253-9837

L1 ANSWER 6 OF 14 CA COPYRIGHT 1999 ACS
ACCESSION NUMBER: 119:249396 CA
TITLE: Theoretical aspects of methane dissociation and hydroxylation on metal oxide diatomics in the gas phase
AUTHOR(S): Stiakaki, Maria Aglaia D.; Tsipis, Athanasios C.; Tsipis, Constantinos A.; Xanthopoulos, Constantinos E.
CORPORATE SOURCE: Fac. Chem., Aristotle Univ. Thessaloniki, Thessaloniki, 540 06, Greece
SOURCE: J. Mol. Catal. (1993), 82(2-3), 425-42
CODEN: JMCADS; ISSN: 0304-5102

L1 ANSWER 7 OF 14 CA COPYRIGHT 1999 ACS
ACCESSION NUMBER: 119:147479 CA
TITLE: Navier-Stokes solutions with surface catalysis for Martian atmospheric entry
AUTHOR(S): Chen, Y. K.; Henline, W. D.; Stewart, D. A.; Candler, G. V.
CORPORATE SOURCE: Eloret Inst., Palo Alto, CA, 94303, USA
SOURCE: J. Spacecr. Rockets (1993), 30(1), 32-42
CODEN: JSCRAG; ISSN: 0022-4650

L1 ANSWER 8 OF 14 CA COPYRIGHT 1999 ACS
ACCESSION NUMBER: 117:211841 CA
TITLE: Reaction analysis of potassium promotion of ruthenium-catalyzed carbon monoxide hydrogenation using steady-state isotopic transients
AUTHOR(S): Hoost, T. E.; Goodwin, J. G., Jr.
CORPORATE SOURCE: Chem. Pet. Eng. Dep., Univ. Pittsburgh, Pittsburgh, PA, 15261, USA
SOURCE: J. Catal. (1992), 137(1), 22-35
CODEN: JCTLA5; ISSN: 0021-9517

L1 ANSWER 9 OF 14 CA COPYRIGHT 1999 ACS
ACCESSION NUMBER: 113:151559 CA

TITLE: Oxidation of alcohols by a six-coordinate
ruthenium(IV)-oxo complex
AUTHOR(S): Cundari, Thomas R.; Drago, Russell S.
CORPORATE SOURCE: Dep. Chem., Univ. Florida, Gainesville, FL, 32611, USA
SOURCE: Inorg. Chem. (1990), 29(19), 3904-7
CODEN: INOCAJ; ISSN: 0020-1669

L1 ANSWER 10 OF 14 CA COPYRIGHT 1999 ACS
ACCESSION NUMBER: 108:193432 CA
TITLE: A quasi-classical trajectory study on the adsorption
of oxygen on the surface of silver-gold alloy
AUTHOR(S): Xie, Xiangfang; Luo, Weiwei; Jiang, Fenglin
CORPORATE SOURCE: Dep. Chem., Fudan Univ., Shanghai, Peop. Rep. China
SOURCE: Cuihua Xuebao (1987), 8(4), 423-9
CODEN: THHPD3; ISSN: 0253-9837

L1 ANSWER 11 OF 14 CA COPYRIGHT 1999 ACS
ACCESSION NUMBER: 102:226603 CA
TITLE: Characteristics of the movement of aerosol particles
near catalytically active surfaces
AUTHOR(S): Amelin, A. G.; Kabanov, A. N.; Shchukin, E. R.;
Shulimanova, Z. L.
CORPORATE SOURCE: Mosk. Khim.-Tekhnol. Inst., Moscow, USSR
SOURCE: Kinet. Katal. (1985), 26(2), 498-501
CODEN: KNKTA4; ISSN: 0453-8811

L1 ANSWER 12 OF 14 CA COPYRIGHT 1999 ACS
ACCESSION NUMBER: 99:73020 CA
TITLE: Effect of a heterogeneous catalytic reaction on the
movement of aerosol particles in the catalysis zone
AUTHOR(S): Anelin, A. G.; Kabanov, A. N.
CORPORATE SOURCE: Mosk. Khim.-Tekhnol. Inst., Moscow, USSR
SOURCE: Kolloidn. Zh. (1983), 45(3), 535-7
CODEN: KOZHAG; ISSN: 0023-2912

L1 ANSWER 13 OF 14 CA COPYRIGHT 1999 ACS
ACCESSION NUMBER: 93:97610 CA
TITLE: Transient behavior of fresh and fouled nickel
hydrogenation catalyst
AUTHOR(S): Bilimoria, Maheyar R.; Bruns, Duane D.; Bailey, James
E.
CORPORATE SOURCE: Dep. Chem. Eng., Univ. Houston, Houston, TX, 77004,
USA
SOURCE: AIChE J. (1980), 26(2), 319-21
CODEN: AICEAC; ISSN: 0001-1541

L1 ANSWER 14 OF 14 CA COPYRIGHT 1999 ACS
ACCESSION NUMBER: 90:193057 CA
TITLE: Vibrational excitation from heterogeneous catalysis
AUTHOR(S): Purvis, George D., III; Redmon, Michael J.; Woken,
Jr., George
CORPORATE SOURCE: Chem. Phys. Group, Battelle Columbus Lab., Columbus,
Ohio, USA
SOURCE: J. Phys. Chem. (1979), 83(8), 1027-33
CODEN: JPCHAX; ISSN: 0022-3654

```
=> SURFAC      SURFACE( ?(L)  CATALYS?(L)  POTNE  ENTILA  AL

       1305397 SURFACE?
        559071 CATALYS?
        661144 POTENTIAL
L2        1510 SURFACE?(L)CATALYS?(L)POTENTIAL

=> S   D L2 1-10  30 IBIB

=> ( S (REACTION(W)DYNAMICS ?) ADN   AND (POTENTIAL(W)ENERGY(W)SURFACE?)

       1814455 REACTION
        329577 DYNAMIC?
          2202 REACTION(W)DYNAMIC?
        661144 POTENTIAL
       1203323 ENERGY
       1305397 SURFACE?
         18539 POTENTIAL(W)ENERGY(W)SURFACE?
L3         363 (REACTION(W)DYNAMIC?) AND (POTENTIAL(W)ENERGY(W)SURFACE?)

=> D L3 1-30 IBIB

=> SURFACE       SURFACE(W)CHEMISTRY

       1206232 SURFACE
        155947 CHEMISTRY
L4        2006 SURFACE(W)CHEMISTRY

=> S D L4 1-30 IBIB

=> HETERO      S HETEROGENEOUS CATALYSIS

         67678 HETEROGENEOUS
        106808 CATALYSIS
L5        2686 HETEROGENEOUS CATALYSIS
                 (HETEROGENEOUS(W)CATALYSIS)

=> LOGOFF

ALL L# QUERIES AND ANSWER SETS ARE DELETED AT LOGOFF
LOGOFF? (Y)/N/HOLD:Y

COST IN U.S. DOLLARS                    SINCE FILE      TOTAL
                                   ENTRY    SESSION
FULL ESTIMATED COST                        362.56     362.77

STN INTERNATIONAL LOGOFF AT 17:27:35 ON 14 JUN 1999
```

References Obtained from Literature Search

Books

1. Thom H. Dunning, Jr., Calculation and Characterization of Molecular Potential Energy Surfaces Vol 1., Jai Press Inc., Greenwich, CO, 1990.

2. C. T. Rettner and M. N. R. Ashfold, Dynamics of Gas-Surface Interactions, The Royal Society of Chemistry, Cambridge, 1991.

3. S. Fraga, Computational Chemistry: *Structure, interactions and Reactivity*, Elsevier Science Publishers, Amsterdam**,** 1992, 637-661.

4. Herbert L. Strauss, Gerald T. Babcock, and Stephen R. Leone, Annual Review of Physical Chemistry Vol 46, Annual Reviews Inc., Palo Alto, CA, 1995, 373-394.

5. Herbert L. Strauss, Gerald T. Babcock, and Stephen R. Leone, Annual Review of Physical Chemistry Vol 48, Annual Reviews Inc., Palo Alto, CA, 1997, 243-271, 299-329.

6. A. W. Adamson, Physical Chemistry of Surfaces, 6th ed., John Wiley and Sons Inc., New York, 1996.

7. Density Functional Theory Ab Initio Molecular Dynamics and Combined Density Functional Theory and Molecular Dynamics Simulations. Chap 10. Wei, Dongqing and Salahub, Dennis R. **X** 159-169.

8. Donald L. Thompson, Modern Methods for Multidimensional Dynamics Computations in Chemistry, World Scientific, London, 1998.

9. Donald G. Truhlar and Keiji Morokuma, ACS Symposium Series: Transition State Modeling for Catalysis, American Chemical Society, Washington, D.C., 1998.

10. G. F. Froment and K. C. Waugh, Studies in Surface Science and Catalysis Vol 109: *Dynamics of Surfaces and Reaction Kinetics in Heterogeneous Catalysis*, Elsevier Science, Amsterdam, 1997.

11. J. C. Whitehead, NATO ASI Series: Selectivity in Chemical Reactions Vol 245, Kluwer Academic Publishers, London, 1988.

12. Richard Masel, Principles of Adsorption and Reactions on Solid Surfaces, John Wiley and Sons, New York, 1996.

13. Gabor A. Somorjai, Introduction to Surface Chemistry and Catalysis, John Wiley and Sons, New York, 1994.

14. R. D. Levine, Molecular Reaction Dynamics, Clarendon Press, Oxford, 1974.

15. David M. Hirst, Potential Energy Surfaces: Molecular Structure and Reaction Dynamics, Taylor and Francis, London, 1985.

Articles

1. George D. Purvis III, Michael J. Redmon, and George Woken, Jr., *The Journal of Physical Chemistry*, **83**, 1027 (1979).

2. Tuker Carrington, Jr., Lynn M. Hubbard, Henry F. Schaefer III, and William H. Miller, *Journal of Chemical Physics*, **80**, 4347 (1984).

3. S. M. Foiles, M. I. Baskes, and M. S. Daw, *Physical Review B*, **33**, 7983 (1986).

4. John Brunning, D. Wyn Derbyshire, Ian W. M. Smith, and Martin D. Williams, *Journal of the Chemical Society, Faraday Transitions*, **84**, 105 (1988).

5. Mutsumi Aoyagi and Shigeki Kato, *Journal of Chemical Physics*, **88**, 6409 (1988).

6. Donald W. Brenner, *Physical Review B*, **42**, 9458 (1990).

7. Cuihua Xuebao, *Journal of Catalysis*, **15**, 54 (1994).

8. Paul S. Cremer and Gabor A. Somorjai, *Journal of the Chemical Society*, **91**, 3671 (1995).

9. Ramona S. Taylor and Barbara J. Garrison, *Langmuir*, **11**, 1220 (1995).

10. D. Fulle, H. F. Hamann, H. Hippler, and J. Troe, *Journal of Chemical Physics*, **105**, 983 (1996).

11. V. J. Barclay, W. H. Hung, W. J. Keogh, R. Kuhnemuth, J. C. Polanyi, G. Zhang, Y. Zeiri, D. R. Jennison, and Y. S. Li, *Journal of Chemical Physics*, **105**, 5005 (1996).

12. R. C. Mowrey, G. J. Kroes, G. Wiesenekker, and E. J. Baerends, *Journal of Chemical Physics*, **110**, 2740 (1999).

13. Wei Pan, Tianhai Zhu, and Weitao Yang, *Journal of Chemical Physics*, **107**, 3981 (1997).

14. B. R. Wu, C. Cheng, and S. L. Lee, *Journal of Physical Chemistry A*, **101**, 6545 (1997).

15. Neil C. Filkin, Mintcho S. Tikhov, Alejandra Palermo, and Richard M. Lambert, *Journal of Physical Chemistry A*, **103**, 2680, (1999).

16. Gengyu Cao, Md., Golam Moula, Yuichi Ohno, and Tatsuo Matsushima, *Journal of Physical Chemistry B*, **103**, 3235 (1999).

17. Peter A. Willis, Hans U. Stauffer, Ryan Z. Hinrichs, and H. Floyd Davis, *Journal of Physical Chemistry A*, **103**, 3706 (1999).

18. Min Qui, Ming Jiang, Yu-Jun Zhao, and Pei-Lin Cao, *Journal of Chemical Physics*, **110**, 10738 (1999).

19. Takaaki Isoda, Seiichiro Maemoto, Katsuki Kusakabe, and Shigeharu Morooka, *Energy and Fuels*, **13**, 617 (1999).

20. P. Kratzer, *Journal of Chemical Physics*, **106**, 6752 (1997).

21. A. C. Luntz and P. Kratzer, *Journal of Chemical Physics*, **104**, 3075 (1996).

22. P. Kratzer, B. Hammer, and J. K. Norshov, *Physical Review B*, **51**, 13432 (1995).

23. P. Bratu, W. Brenig, A. Groβ, M. Hartmann and U. Hofer, P. Kratzer, and R. Russ, *Physical Review B*, **54**, 5978 (1996).

24. W. Brenig, A. Gross, and R. Russ, *Zeitschrift Fur Physik B*, **96**, 231 (1994).

25. X. Y. Zhu, *Proceedings of SPIE*, **3272**, 267 (1998).

26. Johnathon H. Copeland, H. Douglas Kutz, and George T. Mathai, Reaction Dynamics: Collinear Chemical Reactions, Trinity Software Inc., Campton, NH, 1994.

27. Johnathon H. Copeland, H. Douglas Kutz, and George T. Mathai*, The Journal of Chemical Education: Software*, 1992, IVc, 19-21,39-41.

28. J. C. Polanyi, *Accounts of Chemical Research*, **5**, 161 (1972).

29. David Feller, Earl S. Huyser, Weston Thatcher Borden, and Ernest R. Davidson, *Journal of the American Chemical Society*, **105**, 1459 (1983).

30. J. J. Sloan, *Journal of Physical Chemistry*, **92**, 18 (1988).

Sources by George C. Schatz

1. George C. Schatz, Michael S. Fitzcharles, and Lawrence B. Harding, *Faraday Discussions of the Chemical Society*, **84**, 358 (1987).

2. Lester L. Gibson and George C. Schatz,, *1992, IVc, 19-21,39-41.*

3. Thom H. Dunning, Jr., Lawrence B. Harding, Albert F. Wagner, George C. Schatz, and Joel M. Bowman, *Science,* **240**, 453 (1988).

4. J. C. Whitehead ed., Selectivity in Chemical Reactions, Kluwer Academic Publishers, London, 1988, George C. Schatz and Michael S. Fitzcharles, 353-364.

5. George C. Schatz, *Reviews of Modern Physics*, **61**, 669 (1989).

6. T. Dunning ed., Calculations and Characteristics of Molecular Potential Energy Surfaces, JAI Press, Greenwich, CN, 1990, George C. Schatz, 85-127.

7. Kathleen Kudla and George C. Schatz, *Journal of Physical Chemistry*, **95**, 8267 (1991).

8. Kathleen Kudla, George C. Schatz, and Albert F. Wagner, *Journal of Chemical Physics*, **95**, 1635 (1991).

9. George C. Schatz and Jeff Dyck, *Chemical Physics Letters*, **188**, 11 (1992).

10. Kathleen Kudla, Antonios G. Koures, Lawrence B. Harding, and George C. Schatz, *Journal of Chemical Physics*, **96**, 7465 (1992).

11. David C. Clary and George C. Schatz, *Journal of Chemical Physics*, **99**, 4578 (1993).

12. Evelyn M. Goldfield, Stephen K. Gray, and George C. Schatz, *Journal of Chemical Physics*, **102**, 8807 (1995).

13. K. Liu and A. Wagner eds., Advanced Series in Physical Chemistry Vol. 6: The Chemical Dynamics and Kinetics of Small Radicals, World Scientific, New Jersey, 1995, Kathleen Kudla and George C. Schatz, 438-465.

14. Herbert L. Strauss, Gerald T. Babcock, and Stephen R. Leone, Annual Review of Physical Chemistry Vol 46, Annual Reviews Inc., Palo Alto, CA, 1995, Joel M. Bowman and George C. Schatz, 169-195.

15. Kimberly S. Bradley and George C. Schatz, *Journal of Chemical Physics*, **106**, 8464 (1997).

16. Josie V. Setzler, Hua Guo, and George C. Schatz, *Journal of Physical Chemistry B.*, **101**, 5352 (1997).

17. Josie V. Setzler, Jason Bechtel, Hua Guo, and George C. Schatz, *Journal of Physical Chemistry*, **107**, 9176, (1997).

18. George C. Schatz, *Journal of Chemical Physics*, **107**, 2340 (1997).

Appendix II: CodeWarrior

This is the code for a program created using the CodeWarrior programming language. The program has two main parts: 1) The first allows the user to see a PICT file on the screen and 2) The second shows two circles rotating in three-dimensions. The program allows the user to switch between the two different parts using the menu bar.

```
unit MasonI;

//
……………………………………………………………………………………………………
……………………………………………………………………………………………………
……… includes
interface

uses

            { Universal Interfaces. }
             Fonts, Menus, Processes, Resources,  QuickDraw, OSUtils, SegLoad, Windows,
TextEdit, Dialogs,
       Processes, Events, QuickDrawText, Sound;

//
……………………………………………………………………………………………………
……………………………………………………………………………………………………
…… constants

const

rWindowResourceID = 128;
rPictureResourceID = 128;

//
……………………………………………………………………………………………………
…………………………………………………………………………… routine declarations
{procedure        doInitManagers;
procedure         doNewWindow;
procedure         doDrawPictAndString; }
procedure       MasonI;
```

```
implementation


// ◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊
initManagers

{procedure doInitManagers;

            begin
            MaxApplZone;
            MoreMasters;

            InitGraf(@qd.thePort);
            InitFonts;
            InitWindows;
            InitMenus;
            TEInit;
            InitDialogs(nil);

            InitCursor;
            end;}

// ◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊
doNewWindow

procedure doNewWindow;

            var
            windowPtrB : WindowPtr;
            windowRecPtr : Ptr;

            begin
            windowRecPtr := NewPtr(sizeof(WindowRecord));
            if (windowRecPtr = nil) then
                        begin
                        SysBeep(10);
                        ExitToShell;
                        end;

            windowPtrB := GetNewCWindow(rWindowResourceID, windowRecPtr,
WindowPtr(-1));
            if (windowPtrB = nil) then
                        begin
                        SysBeep(10);
                        ExitToShell;
                        end;

            SetPort(windowPtrB);
            TextFont(systemFont);
            end;
```

```
                { of procedure doNewWindow }

// ◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊
doDrawPictAndString

procedure doDrawPictAndString;

        var
        pictureHdl : PicHandle;
        pictureRect : Rect;

        begin
        pictureHdl := GetPicture(rPictureResourceID);
        if (pictureHdl = nil) then
                begin
                SysBeep(10);
                ExitToShell;
                end;

        SetRect(pictureRect,0,0,640,480);

        HNoPurge(Handle(pictureHdl));
        DrawPicture(pictureHdl,pictureRect);
        HPurge(Handle(pictureHdl));


        ReleaseResource(Handle(pictureHdl));

        end;
                { of procedure doDrawPictAndString }

//
◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊
◊◊◊◊◊ main
procedure MasonI;

begin

{doInitManagers;}
doNewWindow;
doDrawPictAndString;
while not Button do
        begin
        end;
        end;
end.

unit Event;

interface
```

```
 uses
 Processes,  Fonts,Sound;
{ , Resources, Menus,QuickDraw, TextUtils,Types, OSUtils, SegLoad, Windows, TextEdit,
Dialogs,Processes, Events, QuickDrawText, Appearance, Devices,  GestaltEqu, LowMem,
ToolUtils, fp}
 const
  pi = 3.1415927;
rWindowResourceID = 128;

 var
  DrawRect: Rect;
  Beta: integer;
  Rgn1Rect, OffRect, OnRect: Rect;
  Rgn1: RgnHandle;
  X1, X2, Y1, Y2: Real;
  h1, h2, v: Integer;
 { RGBWhite, RGBBlue, RGBRed, RGBLtBlue, RGBBlack, RGBGreen: RGBColor;}
  OvalRect: Rect;
gWhiteColour : RGBColor;
gBlackColour : RGBColor;
gRedColour : RGBColor;
gYellowColour : RGBColor;
gGreenColour : RGBColor;
gBlueColour : RGBColor;
RgnRect: Rect;

{procedure          doInitManagers;
procedure           doNewWindow;}
procedure         Event;

implementation
// ◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊
initManagers

{procedure doInitManagers;

              begin
              MaxApplZone;
              MoreMasters;

              InitGraf(@qd.thePort);
              InitFonts;
              InitWindows;
              InitMenus;
              TEInit;
              InitDialogs(nil);

              InitCursor;
              end;}
```

```
// ◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊
doNewWindow

procedure doNewWindow;

        var
        windowPtrC : WindowPtr;
        windowRecPtr : Ptr;

        begin
        windowRecPtr := NewPtr(sizeof(WindowRecord));
        if (windowRecPtr = nil) then
                begin
                SysBeep(10);
                ExitToShell;
                end;

        windowPtrC := GetNewCWindow(rWindowResourceID, windowRecPtr,
WindowPtr(-1));
        if (windowPtrC = nil) then
                begin
                SysBeep(10);
                ExitToShell;
                end;

        SetPort(windowPtrC);
        TextFont(systemFont);
        end;

 procedure SetColors;
 begin
        gWhiteColour.red := $FFFF;
        gWhiteColour.green := $FFFF;
        gWhiteColour.blue := $FFFF;

        gBlackColour.red := $0000;
        gBlackColour.green := $0000;
        gBlackColour.blue := $0000;

        gRedColour.red := $AAAA;
        gRedColour.green := $0000;
        gRedColour.blue := $0000;

        gYellowColour.red := $FFFF;
        gYellowColour.green := $CCCC;
        gYellowColour.blue := $0000;

        gGreenColour.red := $0000;
        gGreenColour.green := $9999;
```

```
            gGreenColour.blue := $0000;

            gBlueColour.red := 26214;
            gBlueColour.green := 39321;
            gBlueColour.blue := 26214;

end;


procedure SetRgnRect (var RgnRect: Rect; hor, ver, rad: Integer);
begin
 SetRect(RgnRect, hor - rad, ver - rad, hor + rad, ver + rad);
end;

procedure PaintAtom (hor, ver, rad: Integer);
{ var
OvalRect: Rect;}
begin
SetRect (OvalRect, hor - rad, ver - rad, hor + rad, ver + rad);
 PaintOval(OvalRect);
end;

procedure DrawMolecule;
begin
 if Y2 >= Y1 then              {Atom2 is in back}
  begin
{----------Paint Atom2----------}
   RGBForeColor(gYellowColour);
   PaintAtom(h2, v, 96);
{---------Paint Atom1-----------}
   OpenRgn;
   SetRgnRect(Rgn1Rect, h1, v, 96);
   FrameOval(Rgn1Rect);
   CloseRgn(Rgn1);
   RGBForeColor(gRedColour);
   FillRgn(Rgn1, qd.black);
  end
 else
  begin       {Atom1 is in back}
{---------Paint Atom1-----------}
   RGBForeColor(gRedColour);
   PaintAtom(h1, v, 96);
{--------Paint Atom2------------}
   OpenRgn;
   SetRgnRect(Rgn1Rect, h2, v, 96);
   FrameOval(Rgn1Rect);
   CloseRgn(Rgn1);
   RGBForeColor(gYellowColour);
   FillRgn(Rgn1, qd.black);
  end;
```

```
 end;

 procedure Pause (time: LongInt);
  var
   t: LongInt;

 begin
  t :=TickCount;
  while (TickCount- t < time) do
   begin
   end;
 end;

 procedure Coordinates;
 begin
  X1 := 2 * Sin(Beta * pi / 180);
  Y1 := 2 * Cos(Beta * pi / 180);
  X2 := -2 * Sin(Beta * pi / 180);
  Y2 := -2 * Cos(Beta * pi / 180);
  h1 := 320 + Round(32 * X1);
  h2 := 320 + Round(32 * X2);
  v := 240;
 end;

procedure Event;

begin
 SetColors;
{doInitManagers; }
doNewWindow;
 RGBBackColor(gBlackColour);
 RGBForeColor(gBlueColour);
 Rgn1 := NewRgn;
  Beta := -180;
  repeat
   RGBBackColor(gBlueColour);
   Coordinates;
   DrawMolecule;
   Pause(50);
   RGBForeColor(gBlueColour);
   SetRect(DrawRect, 20,20,620,460);
EraseRect(DrawRect);
 {  RGBForeColor(gBlackColour);
  MoveTo(50,50);
 write( 'X1  =', X1);
  MoveTo(70,70);
  write('Y1 =', Y1);
  MoveTo(90,90);
 write('X2  =', X2);
 MoveTo(110,110);
```

```
 write('Y2  =', Y2);
 MoveTo(200,200);
 write('Beta =', Beta);}
   Beta := Beta + 15;

  until Beta = 180;
  repeat
   until Button;
 DisposeRgn(Rgn1);
 end;
 {DisposeWindow(windowPtr);}
end.

program EventsMain;

uses MasonI , Event,
            { Universal Interfaces. }
            Appearance, Devices, Fonts, GestaltEqu, Menus, PictUtils, Processes, Sound,
TextUtils,
            ToolUtils, Resources;

const
rMenubar = 128;
rWindow = 129;
mApple = 128;
iAbout = 1;
mFile =129;
iQuit = 11;
mDemonstration = 131;
MAXLONG = $7FFFFFFF;
iRegion = 2;
iTitleScreen = 1;

var
gPreAllocatedBlockPtr : Ptr;
gMacOS85Present : Boolean;
gWindowPtr : WindowPtr;
gDone : boolean;
gCursorRegion: RgnHandle;
gInBackground : boolean;
gSleepTime : SInt32;
osError : OSErr;
response : SInt32;
mainMenubarHdl : Handle;
mainMenuHdl : MenuHandle;
mainErr : OSErr;

procedure DoInitManagers; forward;
procedure EventLoop; forward;
procedure DoEvents({const} var theEvent : EventRecord); forward;
```

```
procedure DoMenuChoice(menuChoice : SInt32); forward;

procedure DoInitManagers;
        var
        osError : OSErr;
        begin
        MaxApplZone;
        MoreMasters;
        InitGraf(@qd.thePort);
        InitFonts;
        InitWindows;
        InitMenus;
        TEInit;
        InitDialogs(nil);
        InitCursor;
        FlushEvents(everyEvent, 0);
        {osError := RegisterAppearanceClient;}
        end;
                { of procedure DoInitManagers }

procedure EventLoop;
        var
        theEvent : EventRecord;
        gotEvent : Boolean;
        begin
        gDone := false;
        gSleepTime := MAXLONG;
        gCursorRegion := nil;
        while not gDone do
                begin
                gotEvent := WaitNextEvent(everyEvent, theEvent, gSleepTime,
gCursorRegion);
                if gotEvent then
                        begin
                        DoEvents(theEvent);
                        end;
                end;
        end;
                { of procedure EventLoop }

//
◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊
◊ DoEvents

procedure DoEvents({const} var theEvent : EventRecord);
        var
        theWindowPtr : WindowPtr;
        partCode : SInt16;
        charCode : SInt8;
        ignored : OSStatus;
```

```
    begin
    case (theEvent.what) of
        mouseDown: begin
            partCode := FindWindow(theEvent.where, theWindowPtr);
            case partCode of
                inMenuBar: begin
                    DoMenuChoice(MenuSelect(theEvent.where));
                    end;
                inContent: begin
                    if (theWindowPtr <> FrontWindow) then
                        begin
                        SelectWindow(theWindowPtr);
                        end;
                    end;
                inDrag: begin
                    DragWindow(theWindowPtr, theEvent.where,
qd.screenBits.bounds);
                    end;
                end;
            end;
        keyDown, autoKey: begin
            charCode := BAnd(theEvent.message, charCodeMask);
            if (BAnd(theEvent.modifiers, cmdKey) <> 0) then
                begin
                DoMenuChoice(MenuEvent(theEvent));
                end;
            end;
        updateEvt: begin
            BeginUpdate(WindowPtr(theEvent.message));
            EndUpdate(WindowPtr(theEvent.message));
            end;
        osEvt: begin
            case BAnd(BSR(theEvent.message, 24), $000000FF) of

                suspendResumeMessage: begin
                    if (BAnd(theEvent.message, resumeFlag) = 1)
then
                        begin
                        gInBackground := false;
{$ifc TARGET_CPU_PPC}
                        if (gMacOS85Present = true) then
                            begin
                            ignored :=
SetThemeCursor(kThemeArrowCursor);
                            end
                        else begin
                            SetCursor(qd.arrow);
                            end;
{$elsec}
                        SetCursor(qd.arrow);
```

```
{$endc}
                        end
                    else begin
                        gInBackground := true;
                        end;
                    end;
                mouseMovedMessage: begin
                    {DoChangeCursor(FrontWindow,
gCursorRegion);}
                    end;
                otherwise begin
                    end;
                end;
                    { of case statement }
            end;
        otherwise begin
            end;
        end;
            { of case statement }
    end;
        { of procedure DoEvents }

// ◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊◊
DoMenuChoice

procedure DoMenuChoice(menuChoice : SInt32);
    var
    menuID, menuItem : SInt16;
    itemName : Str255;
    daDriverRefNum : SInt16;
    ignored : OSStatus;
    begin
    menuID := HiWord(menuChoice);
    menuItem := LoWord(menuChoice);
    if (menuID = 0) then
        begin
        Exit(DoMenuChoice);
        end;
    case menuID of
        mFile: begin
            if (menuItem = iQuit) then
                begin
                gDone := true;
                end;
            end;
        mDemonstration: begin
            case menuItem of
                iRegion: begin
                    Event;
                    end;
```

```
              iTitleScreen: begin
                  MasonI;
                  end;
                otherwise begin
                    end;
                end;
                    { of case statement }
            end;
        otherwise begin
            end;
        end;
            { of case statement }
    HiliteMenu(0);
    end;
        { of procedure DoMenuChoice }


begin

gMacOS85Present := false;

gPreAllocatedBlockPtr := NewPtr(sizeof(DialogRecord));
if (gPreAllocatedBlockPtr = nil) then
        begin
        ExitToShell;
        end;

DoInitManagers;

osError:=noErr;
if ((osError = noErr) and (response >= $00000850)) then
        begin
        gMacOS85Present := true;
        end;

    mainMenubarHdl := GetNewMBar(rMenubar);
    if (mainMenubarHdl = nil) then
        begin
        ExitToShell;
        end;
    SetMenuBar(mainMenubarHdl);

    DrawMenuBar;

    mainMenuHdl := GetMenuHandle(mApple);
    if (mainMenuHdl = nil) then
        begin
        ExitToShell;
        end
    else begin
```

```
                    AppendResMenu(mainMenuHdl, 'DRVR');
                    end;

gWindowPtr:= GetNewCWindow(rWindow, nil, WindowPtr(-1));
SetPort(gWindowPtr);
TextSize(10);
EventLoop;
end.
```

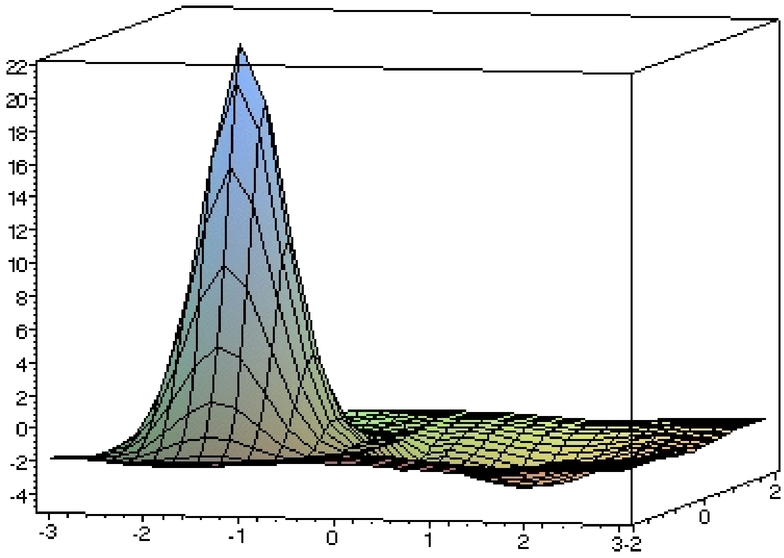
Appendix III: Maple V

```
BigOne:= proc(xi,yi) if xi<0 then ((D1*exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-
zj)^2)^.5-RE1))*(exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))-
2)+DD1*exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))*(exp(-
B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))+2))/2) + ((D2*exp(-B2*(((xj-
xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-RE2))*(exp(-B2*(((xj-xa)^2 + (yj-ya)^2 +
(zj-za)^2)^.5-RE2))-2)+DD2*exp(-B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-
RE2))*(exp(-B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-RE2))+2))/2) +
((D3*exp(-B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-RE3))*(exp(-B3*(((xi-
xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-RE3))-2)+DD3*exp(-B3*(((xi-xa)^2 + (yi-
ya)^2 + (zi-za)^2)^.5-RE3))*(exp(-B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-
RE3))+2))/2) - ((((D1*exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-
RE1))*(exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))-2)-DD1*exp(-
B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))*(exp(-B1*(((xi-xj)^2 + (yi-
yj)^2 + (zi-zj)^2)^.5-RE1))+2))/2) * ((D1*exp(-B1*(((xi-xj)^2 + (yi-yj)^2 +
(zi-zj)^2)^.5-RE1))*(exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))-2)-
DD1*exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))*(exp(-B1*(((xi-
xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))+2))/2)) + (((D2*exp(-B2*(((xj-xa)^2 +
(yj-ya)^2 + (zj-za)^2)^.5-RE2))*(exp(-B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-
za)^2)^.5-RE2))-2)-DD2*exp(-B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-
RE2))*(exp(-B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-RE2))+2))/2) *
((D2*exp(-B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-RE2))*(exp(-B2*(((xj-
xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-RE2))-2)-DD2*exp(-B2*(((xj-xa)^2 + (yj-
ya)^2 + (zj-za)^2)^.5-RE2))*(exp(-B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-
RE2))+2))/2)) + (((D3*exp(-B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-
RE3))*(exp(-B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-RE3))-2)-DD3*exp(-
B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-RE3))*(exp(-B3*(((xi-xa)^2 + (yi-
ya)^2 + (zi-za)^2)^.5-RE3))+2))/2) * ((D3*exp(-B3*(((xi-xa)^2 + (yi-ya)^2 +
(zi-za)^2)^.5-RE3))*(exp(-B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-RE3))-
2)-DD3*exp(-B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-RE3))*(exp(-
B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-RE3))+2))/2)) - (((D1*exp(-
B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))*(exp(-B1*(((xi-xj)^2 + (yi-
yj)^2 + (zi-zj)^2)^.5-RE1))-2)-DD1*exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-
zj)^2)^.5-RE1))*(exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))+2))/2)
* ((D2*exp(-B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-RE2))*(exp(-
B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-RE2))-2)-DD2*exp(-B2*(((xj-
xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-RE2))*(exp(-B2*(((xj-xa)^2 + (yj-ya)^2 +
(zj-za)^2)^.5-RE2))+2))/2)) - (((D1*exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-
zj)^2)^.5-RE1))*(exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))-2)-
DD1*exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))*(exp(-B1*(((xi-
```

```
xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))+2))/2) * ((D3*exp(-B3*(((xi-xa)^2 +
(yi-ya)^2 + (zi-za)^2)^.5-RE3))*(exp(-B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-
za)^2)^.5-RE3))-2)-DD3*exp(-B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-
RE3))*(exp(-B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-RE3))+2))/2)) -
(((D2*exp(-B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-RE2))*(exp(-B2*(((xj-
xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-RE2))-2)-DD2*exp(-B2*(((xj-xa)^2 + (yj-
ya)^2 + (zj-za)^2)^.5-RE2))*(exp(-B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-
RE2))+2))/2) * ((D3*exp(-B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-
RE3))*(exp(-B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-RE3))-2)-DD3*exp(-
B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-RE3))*(exp(-B3*(((xi-xa)^2 + (yi-
ya)^2 + (zi-za)^2)^.5-RE3))+2))/2)))^.5 else ((D1*exp(-B1*(((xi-xj)^2 + (yi-
yj)^2 + (zi-zj)^2)^.5-RE1))*(exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-
RE1))-2)+DD1*exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))*(exp(-
B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))+2))/2) + ((D2*exp(-B2*(((xj-
xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-RE2))*(exp(-B2*(((xj-xa)^2 + (yj-ya)^2 +
(zj-za)^2)^.5-RE2))-2)+DD2*exp(-B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-
RE2))*(exp(-B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-RE2))+2))/2) +
((D3*exp(-B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-RE3))*(exp(-B3*(((xi-
xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-RE3))-2)+DD3*exp(-B3*(((xi-xa)^2 + (yi-
ya)^2 + (zi-za)^2)^.5-RE3))*(exp(-B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-
RE3))+2))/2) - ((((D1*exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-
RE1))*(exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))-2)-DD1*exp(-
B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))*(exp(-B1*(((xi-xj)^2 + (yi-
yj)^2 + (zi-zj)^2)^.5-RE1))+2))/2) * ((D1*exp(-B1*(((xi-xj)^2 + (yi-yj)^2 +
(zi-zj)^2)^.5-RE1))*(exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))-2)-
DD1*exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))*(exp(-B1*(((xi-
xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))+2))/2)) + (((D2*exp(-B2*(((xj-xa)^2 +
(yj-ya)^2 + (zj-za)^2)^.5-RE2))*(exp(-B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-
za)^2)^.5-RE2))-2)-DD2*exp(-B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-
RE2))*(exp(-B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-RE2))+2))/2) *
((D2*exp(-B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-RE2))*(exp(-B2*(((xj-
xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-RE2))-2)-DD2*exp(-B2*(((xj-xa)^2 + (yj-
ya)^2 + (zj-za)^2)^.5-RE2))*(exp(-B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-
RE2))+2))/2)) + (((D3*exp(-B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-
RE3))*(exp(-B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-RE3))-2)-DD3*exp(-
B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-RE3))*(exp(-B3*(((xi-xa)^2 + (yi-
ya)^2 + (zi-za)^2)^.5-RE3))+2))/2) * ((D3*exp(-B3*(((xi-xa)^2 + (yi-ya)^2 +
(zi-za)^2)^.5-RE3))*(exp(-B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-RE3))-
2)-DD3*exp(-B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-RE3))*(exp(-
B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-RE3))+2))/2)) - (((D1*exp(-
B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))*(exp(-B1*(((xi-xj)^2 + (yi-
```

```
yj)^2 + (zi-zj)^2)^.5-RE1))-2)-DD1*exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-
zj)^2)^.5-RE1))*(exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))+2))/2)
* ((D2*exp(-B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-RE2))*(exp(-
B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-RE2))-2)-DD2*exp(-B2*(((xj-
xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-RE2))*(exp(-B2*(((xj-xa)^2 + (yj-ya)^2 +
(zj-za)^2)^.5-RE2))+2))/2)) - (((D1*exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-
zj)^2)^.5-RE1))*(exp(-B1*(((xi-xj)^2 + (yi-yj)^2 + (zi-zj)^2)^.5-RE1))-2)-
DD1*exp(-B1*(((xi-xj)^2+(yi-yj)^2+(zi-zj)^2)^.5-RE1))*(exp(-B1*(((xi-
xj)^2+(yi-yj)^2+(zi-zj)^2)^.5-RE1))+2))/2)*((D2*exp(-B2*(((xj-xa)^2+(yj-
ya)^2+(zj-za)^2)^.5-RE2))*(exp(-B2*(((xj-xa)^2 + (yj-ya)^2 + (zj-za)^2)^.5-
RE2))+2))/2) * ((D3*exp(-B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-
RE3))*(exp(-B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-RE3))-2)-DD3*exp(-
B3*(((xi-xa)^2 + (yi-ya)^2 + (zi-za)^2)^.5-RE3))*(exp(-B3*(((xi-xa)^2 + (yi-
ya)^2 + (zi-za)^2)^.5-RE3))+2))/2)))^.5 + (sqrt(7.36 + 2.41*cos(.82*xi) -
4.45*cos(2*.82*xi) + 0.20*cos(3*.82*xi))-(1-cos(2*0.82*yi)))*(exp((-
2*alpha)*(zi-(1.11 + 0.67*cos(.82*xi) - 0.17*cos(2*.82*xi) -
0.17*cos(3*.82*xi)-0.46*(1-cos(2*0.82*yi)))))-2*exp(-alpha*(zi-(1.11 +
0.67*cos(.82*xi) - 0.17*cos(2*.82*xi) -0.17*cos(3*.82*xi)-0.46*(1-
cos(2*0.82*yi)))))) fi end;
```

```
> plot3d(BigOne,-3..3,-2..2);
```


22
20
18
16
14
12
10
8
6
4
2
0
-2
-4
-3
-2
-1
0
1
2
3
-2
0
2

References

1. P. Kratzer, *Journal of Chemical Physics*, **106**, 6752 (1997).

2. Richard Masel, Principles of Adsorption and Reactions on Solid Surfaces, John Wiley and Sons, New York, 1996.

3. Gabor A. Somorjai, Introduction to Surface Chemistry and Catalysis, John Wiley and Sons, New York, 1994.

4. Johnathon H. Copeland, H. Douglas Kutz, and George T. Mathai, Reaction Dynamics: Collinear Chemical Reactions, Trinity Software Inc., Campton, NH, 1994.

5. Johnathon H. Copeland, H. Douglas Kutz, and George T. Mathai*, The Journal of Chemical Education: Software*, 1992, IVc, 19-21,39-41.

6. J. C. Polanyi, *Accounts of Chemical Research*, **5**, 161 (1972).

7. CodeWarrior Professional: Release 4, Metrowerks Inc., Austin, TX, 1998.

8. Maple V: Release 5.1, Waterloo Maple Inc. 1998.

Acknowledgments

I would like to acknowledge my research advisor for the past two years Dr. Kutz for his patience and support in my efforts to learn about and research reaction dynamics and computer programming of this important area of chemistry. I would also like to acknowledge the financial support of the Grote Endowment which allows students like myself to pursue undergraduate research.