2008 volume 10

# Los Alamos Summer School Research Reports

## Sponsors:

### Los Alamos National Laboratory

The Los Alamos National Laboratory Institute for Advanced Studies, part of the National Security Education Center

The Quantum Institute

### The University of New Mexico

Department of Physics and Astronomy and Extended University Graduate and Upper Division Programs, and the National Science Foundation: Research Experience for Undergraduates (REU)

### Contact

James Colgan, Theoretical Division, Mail Stop B283, Los Alamos National Laboratory,
Los Alamos, NM 87545, jcolgan@lanl.gov

### Publication:

Sharon Mikkelson, Graphic Artist, IRM-CAS/ADTSC
Sue King, Writer/Editor, IRM-CAS/ADTSC
Pictures and cover photo taken by Summer School Students

**September 2008/LA-UR-08-05482**


Los Alamos
Summer School

2008 volume 10

# Los Alamos Summer School Research Reports

## *Table of Contents*


Los Alamos
Summer School

2008 volume 10

# Los Alamos Summer School Research Reports

## *Preface*

This year marked the eighteenth session of the Los Alamos Summer School (LASS) in Physics, a joint educational project between the University of New Mexico (UNM) and the Los Alamos National Laboratory (LANL). The National Science Foundation provides funding to the University side through its Research Experience for Undergraduates (REU) site, and this year the LANL Institute for Advanced Studies supports the Laboratory component. In addition, this year we were again pleased to receive support from the Los Alamos National Laboratory's Quantum Institute. The School recruits nationwide, and the majority of students were junior or senior undergraduates. Twelve students, from a variety of backgrounds and locations, participated in this year's program.

As in previous years, the School employed a dual track of lectures on a wide variety of topics as well as student research projects to give the students their first taste of real research. All the lecturers and mentors who participated in LASS were volunteers. The lecturers were distinguished scientists from LANL and from UNM and discussed topics as diverse as climate modeling, high-energy physics, and the latest developments in astrophysics. The research projects, each individually mentored by a LANL scientist, concentrated on a specific problem and allowed the students to experience how research is undertaken at a National Laboratory. Research projects were in many areas of science including atomic physics, high-energy physics, computational neuroscience, and supernovae modeling. Many of the students presented their research at the annual Los Alamos Student Symposium, which was held at UNM-LA on August 5-6, 2008. Two students won outstanding presentation awards. This year we were also fortunate to tour the Very Large Array (VLA) facility in Socorro, NM, as well as the research laboratories of the University of New Mexico in Albuquerque.

This report collates the research findings of the LASS students in their ten-week research experience. Although the short ten-week span of the school usually only allows the students to contribute to a project in progress, or to make a start on a new research project, this year several of the projects are expected to result in journal publications, leading to advances in the field. More importantly, this summer's experiences have allowed these students to participate in a working research environment, which will be of immense help in deciding their future career paths as they complete their undergraduate education within the coming year. We believe that these project reports demonstrate the remarkable achievements of all the students.

LANL
James Colgan
Norm Magee

UNM
Sally Seidel


Los Alamos
Summer School

**2008** *volume 10*

# Los Alamos Summer School Class

| | |
|---|---|
| Robert Hembree | Hampden-Sydney College |
| David Stalla | Lewis University |
| David vanMaanen | Pennsylvania State University |
| Sara Goltry | The College of Idaho |
| Kyle Martin | University of New Mexico |
| Jakob Kotas | Cornell University |
| Tim Berkelbach | New York University |
| Darrell Lim | California State University, Fullerton |
| Julia Scheevel | Rice University |
| Kate Hessler | Lehigh University |
| Daniel Weflen | Drake University |
| David Brenman | Lewis & Clark College |


Los Alamos
Summer School

# Los Alamos Summer School Mentors and Students

| *Mentor* | *Student* |
|---|---|
| Alexander Balatsky (T-11) | Robert Hembree |
| Christopher Fryer (CCS-2) | David Stalla |
| Garrett Kenyon (P-25) | David vanMaanen |
| Hanna Makaruk (P-21) | Sara Goltry |
| Siming Liu (T-6) | Kyle Martin |
| James Colgan (T-4) | Jakob Kotas |
| Joe Abdallah (T-4) | Tim Berkelbach |
| Hans Ziock (EES-6) | Darrell Lim |
| Christof Teuscher (CCS-3) | Julia Scheevel |
| Gerd Kunde (P-22) | Kate Hessler |
| Daniel Horner (T-4) | Daniel Weflen |
| Steve Valone (MST-8) | David Brenman |


Los Alamos
Summer School

# 2008 volume 10
# Los Alamos Summer School Lectures

| | |
|---|---|
| **Rob Duncan (IAS/UNM)** | *Self-organized criticality and critical phenomena* |
| **Brian O'Shea (T-6)** | *A brief introduction to cosmology* |
| **Kent Budge (CCS-2)** | *Neutrinos in supernovae* |
| **Cory Hauck (CCS-2)** | *Differential equations* |
| **Christof Teuscher (CCS-3)** | *Computers as we don't know them* |
| **Yi Jiang (T-7)** | *Multiscale, cell-based model for biological modeling I & II* |
| **Justin Oelgoetz (X-1)** | *Introduction to plasma kinetics & spectroscopy* |
| **Razvan Teodorescu (T-13)** | *Interface growth in two dimensions: from mathematics to biology and computer science- a physicist's perspective* |
| **Chris Fryer (CCS-2)** | *Understanding nature's largest explosions - the engines behind supernovae and gamma-ray bursts* |
| **Tim Thomas (UNM)** | *Novel computing* |
| **Cynthia Reichhardt (T-12)** | *Local probes in superconducting vortex matter* |
| **Leslie Welser (X-2)** | *Inertial confinement fusion* |
| **Gerd Kunde (P-22)** | *The quark gluon plasma* |
| **Jian-Zhou Zhu (T-07)** | *Fluid turbulence* |
| **Werner Dappen (USC)** | *Helioseismology* |
| **David Roberts (T-13/CNLS)** | *Spontaneous synchronization of coupled oscillators* |
| **Steve Valone (MST-8)** | *Open system density functional theory and the Millikan oil-drop experiment* |
| **Balu Nadiga (CCS-2)** | *An introduction to turbulence in the ocean-atmosphere system* |
| **Dan Horner (T-4)** | *Non-perturbative treatments of photoionization of atoms and molecules* |
| **Susan Atlas (UNM)** | *In Search of H: Two lectures on density functional theory* |
| **Bill Louis III (P-DO)** | *Neutrino physics* |
| **Eli Ben-Naim (T-13)** | *Statistical physics of competitions, statistical physics of granular materials* |
| **Hans Ziock (EES)** | *The scale of energy use and the resulting implications for solutions* |
| **Bette Korber (T-10)** | *Biophysics* |
| **Garrett Kenyon (P-25)** | *Computational neuroscience* |
| **Michael Graesser (T-08)** | *SUSY* |
| **Sally Seidel (UNM)** | *Particle physics* |


Los Alamos
Summer School

# 2008 volume 10

# Los Alamos Summer School Photographs

## *LASS Class of 2008*

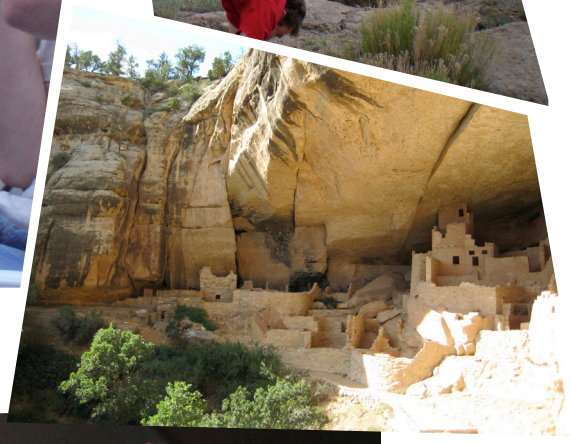
Back row left to right: Sally Seidel (UNM Professor), Kyle Martin, Darrell Lim, David Brenman, Robert Hembree, David vanMaanen, Tim Berkelbach, David Stalla, James Colgan (LANL)

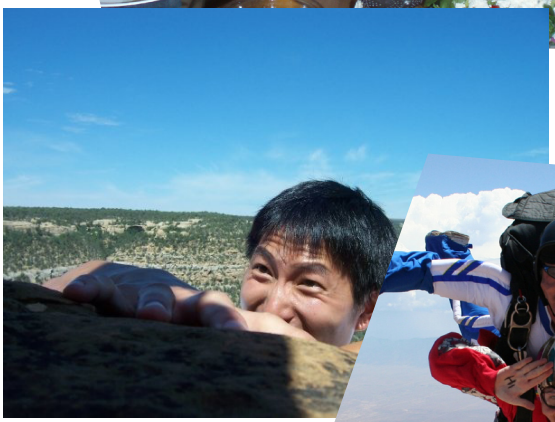
Front row left to right: Julia Scheevel, Kate Hessler, Sara Goltry, Jakob Kotas, Daniel Weflen


Los Alamos
Summer School

# 2008 volume 10

# Los Alamos Summer School Photographs

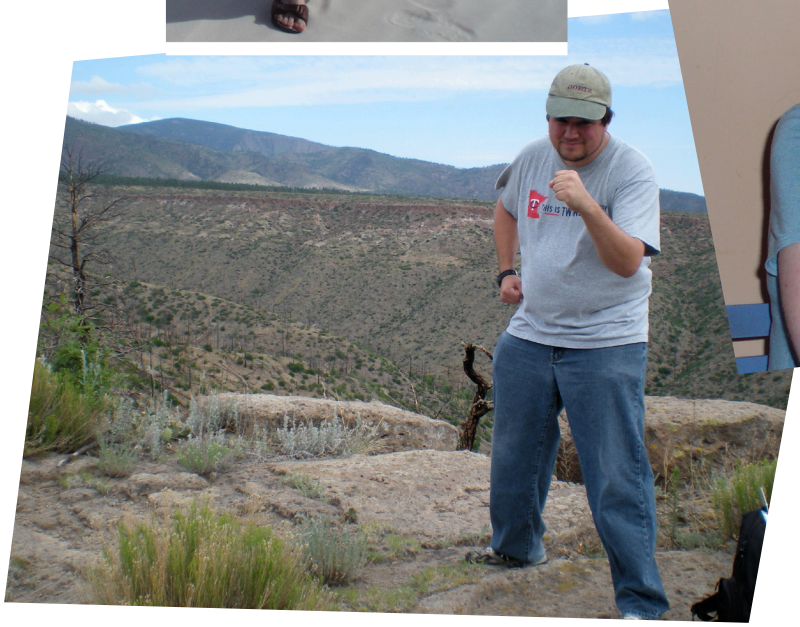





Los Alamos
Summer School

# 2008 volume 10

# Los Alamos Summer School Photographs




Los Alamos
Summer School

**2008** *volume 10*

# Los Alamos Summer School Titles of Papers


Los Alamos
Summer School

# Modeling energy dependence of the *L*-shell x-ray emission produced by femtosecond-pulse laser irradiation of xenon clusters

Timothy C. Berkelbach*
*Departments of Physics and Chemistry, New York University, New York NY 10003*

James Colgan and Joseph Abdallah, Jr.
*Theoretical Division, Los Alamos National Laboratory, Los Alamos NM 87545*

Anatoly Ya. Faenov
*Multicharged Ions Spectra Data Center of VNIIFTRI, Mendeleevo, Moscow Region 141570, Russia*
(Dated: August 27, 2008)

We employ the Los Alamos suite of atomic physics codes to model the $L$-shell x-ray emission spectrum of xenon clusters and compare results with those obtained via femtosecond laser pulse spectroscopy. We find that the commonly employed configuration average approximation breaks down and significant spin-orbit splitting necessitates a detailed level accounting. Additionally, we reproduce an interesting spectral trend for a series of experimental spectra taken with varying pulse energy for fixed pulse duration. To capture the results of an increasing beam energy, we find that spectral modeling requires an increased hot electron fraction, but decreased atomic density and bulk electron temperature. We believe these latter conditions to be a result of partial cluster destruction due to the laser prepulse.

## I. INTRODUCTION

The use of atomic clusters as sources for ultrashort laser-produced plasmas has been of considerable interest lately due to the cluster's more efficient energy absorption than in gaseous or solid counterparts. Potential applications include the industrial production of x-rays, medicine and biology, and even nuclear fusion [1, 2]. As such, the ability to accurately explain and predict the associated plasma kinetics and spectroscopy is highly desirable.

The lifetime of such laser-produced plasmas is remarkably short and as such, experimental determination of many important plasma properties is very difficult. Thus, by comparing an experimental observable such as the emission spectrum, computational modeling both provides a methodology for systematic determination of plasma conditions and helps understand how plasma properties depend on the underlying physics. In this work, we utilize the Los Alamos atomic physics codes - which take the plasma density, electron temperature, and optional hot electron temperature and fraction as input - to model the $L$-shell emission spectrum of Xe. We compare our results with those recently obtained at the Advanced Photon Research Center in Japan via femtosecond laser pulse irradiation of Xe clusters. The influence of hot electrons, which are known to be produced in such high-energy laser pulses, is found to be essential in reproducing the experimental spectra.

In Sec. II we describe the Xe cluster irradiation experiments and present the obtained spectra. Sec. III contains a description of the theoretical methods used, including atomic structure and collisional radiative kinetics code, including an example of the sensitivity of modeled spectra to the level of approximation used. In Sec. IV, we explore the dependence of the modeled spectra on electron temperature, atomic density, and fraction of hot electrons. We compare the modeled spectra to experiment and reproduce an interesting spectral trend obtained for a varying laser pulse energy at fixed duration. In Sec. V, we summarize this work and draw conclusions regarding emission spectra of high-$Z$ atomic clusters and the influence of the laser prepulse.

## II. EXPERIMENTAL METHOD

The Xe cluster experiments were performed with the JAERI (Kyoto, Japan) 100 TW Ti:sapphire laser system based on the technique of chirped pulse amplification, which was designed to generate 20 fs pulses at a 10 Hz repetition rate and capable of producing focusing intensity up to $10^{20}$ W/cm$^2$ [3, 4]. The laser pulses centered at 800 nm were generated at 82.7 MHz by a Ti:sapphire laser oscillator (10 fs). The pulses from the oscillator were stretched to 10 ns, and amplified by a regenerative amplifier and two stages of a multipass amplifier. In this study the amplified pulses were compressed to 30 fs by a vacuum pulse compressor yielding a maximum pulse energy of 360 mJ. In this system, after the regenerative amplifier, the pulses go through two double Pockels cells to reduce the prepulse. The contrast ratio between the main pulse and the prepulse that precedes it by 1 ns is greater than $10^5 : 1$.

In a vacuum target chamber, the measured spot size was 11 $\mu$m at $1/e^2$, which was 1.1 times as large as that of the diffraction limit. Approximately 64% of the laser

*Electronic address: tcb244@nyu.edu

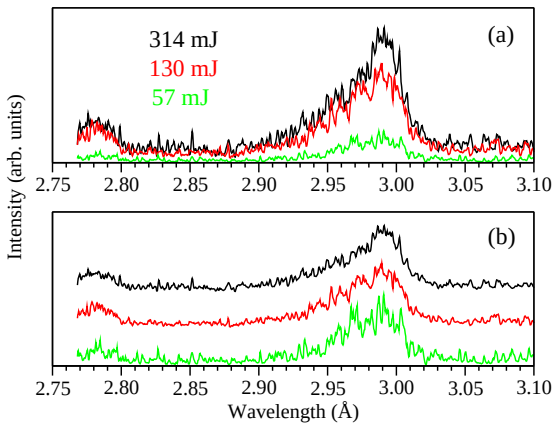



FIG. 1: Experimental emission intensities obtained at a pulse length of 30 fs with a varying energy of 314 mJ (top, black), 130 mJ (middle, red), and 57 mJ (bottom, green). The spectra are shown (a) unscaled, to make clear the intensity dependencies, and (b) scaled to a uniform maximum, to portray the spectral dependencies.

energy was contained in the 11 $\mu$m Gaussian spot. These parameters give a focused laser peak intensity of $1.0 \times 10^{19}$ W/cm$^2$ with a pulse duration of 30 fs and a pulse energy of 200 mJ.

Xe clusters were produced by expanding high-pressure 2 MPa Xe gas into vacuum through a specially designed pulsed conical nozzle with input and output diameters of 0.5 and 2.0 mm, respectively, and a length of 75 mm. The parameters of this nozzle were calculated using a model developed at the Institute of Mathematical Modeling of the Russian Academy of Sciences [5, 6]. The results of this model have been shown to be consistent with experimentally obtained data [7, 8]. In particular, comparisons of Ar gas density spatial distribution measured by interferometry, and the spatial distribution of cluster sizes measured by light scattering diagnostics have demonstrated the predictive capability of this code.

Simulations of the nozzle employed in this experiment and x-ray emission analysis of similar experiments suggest the production of a dense cluster medium with clusters approximately 1 $\mu$m in size [9–12]. Since the rate of cluster decay is primarily determined by the number of atoms in the cluster, the use of large clusters in conjunction with the prepulse reduction techniques insured the direct interaction between the high-density cluster and the main fs pulse. The laser was then focused about 1.5 mm below the nozzle.

The spatially resolved x-ray spectra were measured using focusing spectrometers with spatial resolution [13]. The spherically bent crystal was placed at a distance of 381.2 $\mu$m from the plasma source and was centered at $\lambda = 4.05$ Å, which corresponds to a Bragg angle of 35.7° for fourth order reflection of the crystal.

Plotted in Fig. 1 is a series of spectra taken for a fixed pulse duration of 30 fs with a varying laser energy. We observe that although the spectrum intensities do indeed decrease with decreasing beam energy, as would be expected, the relative intensity of shorter-wavelength transitions - between 2.95 and 2.97 Å - actually increases. This interesting trend will prove to be a bit counter-intuitive following studies made in Sec. IV, but can in fact be reproduced based on justifiable physical conditions, as will be shown.

## III. THEORETICAL METHOD

All calculations were performed using the Los Alamos suite of atomic physics codes: CATS [14], GIPPER [15], and ATOMIC [16]. Atomic structure and collisional excitation cross sections were computed from the CATS code, based on the Hartree-Fock method of Cowan [17]. GIPPER was used for the calculation of collisional ionization cross sections and all data was fed into ATOMIC, which employs a collisional radiative scheme to determine the ion population distribution and subsequent emission spectrum.

Previous studies utilizing configuration-average-based atomic structure energy levels via the unresolved transition array (UTA) or mixed-UTA (mUTA) methods, have succeeded in describing the x-ray emission spectra of lower-$Z$ atoms [18]. The transition of interest in this work, determined to be $3d \rightarrow 2p$, undergoes severe spin-orbit splitting due to the high-$Z$ effects of Xe. As such, the $L$-shell emission spectrum of interest necessitates a detailed level accounting, including approximately 100,000 levels over 2800 configurations in 13 ion stages, from K-like ($Xe^{35+}$) to Ga-like ($Xe^{23+}$). Each of the 13 ion stages included configurations from the ground state, single promotion from $n = 2$ to $n = (3, 4)$, and double promotion from $n = 3$ to $nl = (4s, 4p)$.

Fig. 2 shows the dependency of the emission spectrum on the level of atomic theory used. The configuration average approximation includes no fine structure via relativistic effects and as such, includes only a single $3d \rightarrow 2p$ transition for each ion stage, subject to permutations of the non-participating electrons. Solution of the full Dirac equation via a Dirac-Fock-Slater procedure results in a so-called relativistic configuration average description, which includes by nature the spin-orbit interaction, resulting in a much more accurate spectrum. To feasibly obtain a fine-structure spectrum, we may go one step further and utilize a semi-relativistic detailed level accounting (DLA).

This detailed level accounting was achieved through the use of a non-relativistic Hamiltonian with the usual semi-relativistic correction terms:

$$\Delta H = -\frac{1}{2mc^2}(E-V)^2 - \frac{\hbar^2}{4m^2c^2}\left(\frac{dV}{dr}\right)\left(\frac{\partial}{\partial r} - \frac{2}{r}\,\mathbf{l}\cdot\mathbf{s}\right),$$

respectively known as the mass-velocity term, the Darwin term, and the spin-orbit term, in cgs units. The spin-

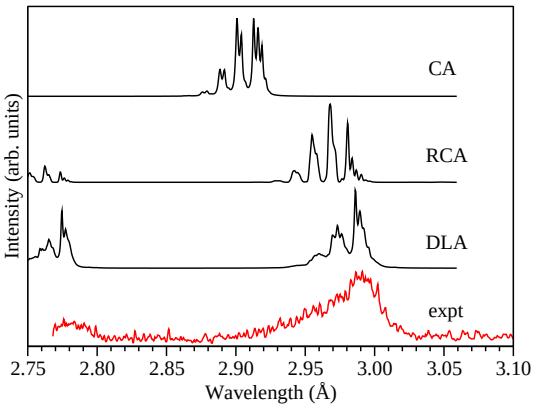



FIG. 2: Theoretical emission spectra for three levels of approximation - configuration average (CA), relativistic configuration average (RCA), and a detailed level accounting (DLA) - and example experimental spectrum, top to bottom respectively. Details regarding each approximation may be found in the text. Each spectrum is calculated at an atomic density of $10^{20}$ cm$^{-3}$ and electron temperature of 300 eV.

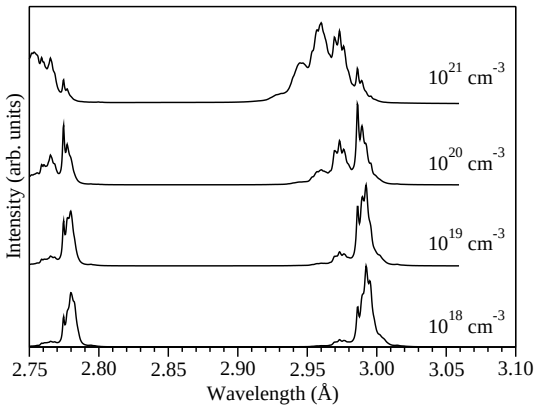



FIG. 3: Theoretical emission spectra for varying atomic densities at fixed electron temperature, $T_e = 300$ eV.

orbit term, which becomes more important for increasing $Z$, is found to play a most important role, resulting in a prominent splitting effect.

It should be noted that a full fine-structure calculation is far too expensive in this case. Therefore, we follow [19] and solve the kinetics rate equations in the configuration average approximation, but use DLA to compute the emission spectrum. Here, we have retained DLA for all transition arrays with less than $10^6$ lines. Furthermore, while our model includes individual detailed levels, there is no configuration-interaction mixing present.

## IV. RESULTS AND DISCUSSION

In all spectra to follow, spectral lines were modeled as Lorentzians with a width approximately equal to the instrumental resolution, $\Delta\lambda/\lambda = 0.0006$. Furthermore, unless otherwise noted, the spectra have been individually rescaled to a uniform maximum for the sake of comparison.

In an effort to characterize the experimental plasma conditions, we first scan the atomic density, $N_a$, at a fixed electron temperature of $T_e = 300$ eV, based on the simulation of a previous experiment with Argon clusters [9, 20]. Fig. 3 shows the obtained spectra taken in the range $N_a = 10^{18} - 10^{21}$ cm$^{-3}$. We see that the spectral features are relatively insensitive to changes in the density until $10^{21}$ cm$^{-3}$ and beyond, which is approaching solid-state densities, and outside the range of reasonable densities for atomic cluster experiments. Again in agreement with Refs. [9, 20], we choose $10^{20}$ cm$^{-3}$ as an initial guess.

With the atomic density fixed, we now seek the spectral temperature dependence. Plotted in Fig. 4 is a series of emission spectra calculated at $N_a = 10^{20}$ cm$^{-3}$, with electron temperature $T_e = 200 - 500$ eV. As the electron temperature is increased, the spectrum begins to favor higher energy transitions. Physically we can understand this trend as the following: each ion stage of Xe has a localized range of $3d \rightarrow 2p$ transitions occurring at successively shorter wavelengths. As we increase the electron temperature, the ion distribution shifts to greater ionization, thus increasing the intensity of shorter wavelength spectral lines.

Lastly, we examine the influence of hot electrons. The hot electron temperature was chosen to be $T_{\rm hot} = 5$ keV, though previous simulation of Kr clusters have employed a hot electron temperatures of 20 keV [21]. Hot electron temperatures as high as 20 keV were explored in

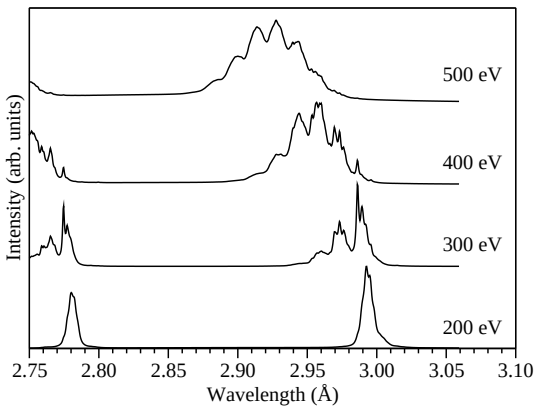



FIG. 4: Theoretical emission spectra for varying electron temperatures at fixed atomic density, $N_a = 10^{20}$ cm$^{-3}$.

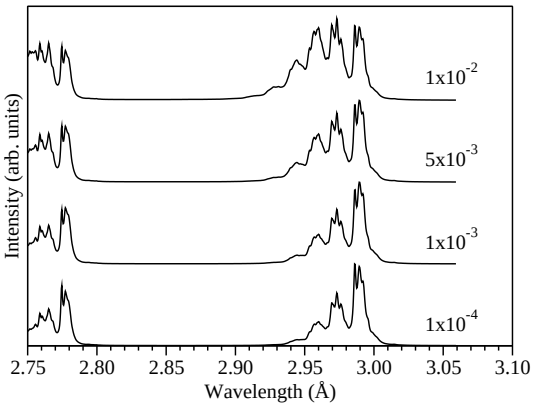



FIG. 5: Theoretical emission spectra for varying hot electron fractions at fixed electron temperature, $T_e = 300$ eV, and atomic density, $N_a = 10^{20}$ cm$^{-3}$.

our study, with no significant difference in the spectral features. This finding is in good agreement with a generalized study [22] showing that neither the functional form nor characteristic energy of the hot electrons has a strong influence on the rates as long as the characteristic energy is greater than the most energetic transition of interest.

Fig. 5 shows a series of spectra taken for a range of hot electron fractions, $f$. Increasing $f$ is observed to be similar to increasing $T_e$, as could be expected. Closer inspection shows that the spectrum is actually widened, with transition peaks from a greater range of ion stages, in much better agreement with experiment. This effect is a result of the two-temperature model, which causes a wider, more balanced ion distribution than found in one-temperature models ($f = 0$).

We now turn to the problem of matching, and thus diagnosing, the experimental spectra. The problem which presents itself, as has been alluded to in previous sections, is that the lower energy - and thus lower intensity - spectra clearly have an increased relative intensity of shorter wavelength (2.95 - 2.97 Å) transitions. Based on the previous dependency investigations, this trend would intuitively be captured by either raising the electron temperature or hot electron fraction. But this act would of course increase the overall intensity as well, in direct opposition to what is seen in experiment, and makes no physical sense.

Thus we seek a physically sensible, variational process which can reproduce high-intensity spectra with decreased relative intensities of shorter wavelength transitions. The features of the modeled spectra are not independent of one another, and thus a linear-type optimization procedure, where each feature is accurately fixed in succession, is not feasible. Rather, we use a method developed only through intuition and experimentation which employs an overshoot-and-correct type procedure. We state here our four step process.

First, we match the mid-intensity spectrum, which is easily accomplished using our results from the previous section. The physical conditions were found to be $N_a = 1.0 \times 10^{20}$ cm$^{-3}$, $T_e = 300$ eV, $f = 0.03$. The problem has now been reduced to merely reproducing the relative differences obtained in either increasing or decreasing the laser energy. Next, we utilize the hot electron fraction to approximate the overall intensities of each spectrum in accordance with Fig. 1. With hindsight on our side, we actually overshoot the difference in intensities, i.e. $f$ too high for the high-energy, 314 mJ spectrum (or too low for the low-energy, 57 mJ spectrum) for exact agreement with the relative intensities of Fig. 1.

While such an effect could alternatively be achieved through the increase of either electron temperature or atomic density, using the hot electron fraction is advantageous in that it promotes the inclusion of more peaks via a broadening of the ion distribution. Furthermore, we make the assumption that the hot electron fraction will be the parameter most influenced by an increased beam energy for fixed pulse duration and thus must most prominently reflect the appropriate physics.

Although this step does indeed reproduce the correct intensity trend, it mis-weights the relative intensities of the shorter wavelength peaks. Thus, our next step is to alter the density in such a way so as to correct the relative peak intensities. This procedure of course unfavorably affects the overall intensity as well, but was compensated for via the overshooting of the hot electron fraction. Lastly, we fine-tune the height of the major peak centered around 2.99 Å, which for small changes, can be uniquely adjusted with the bulk electron temperature.

We demonstrate this procedure, shown in Fig. 6, for the high-energy spectrum. Starting from the "base-

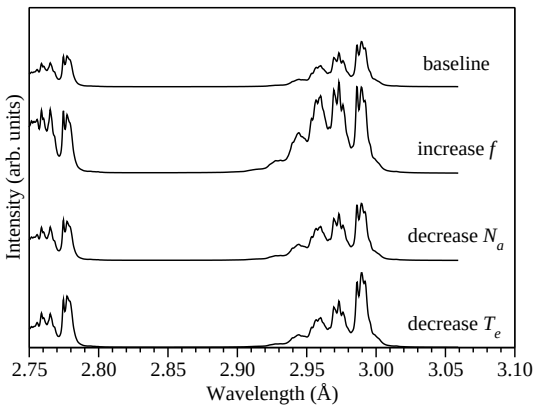



FIG. 6: Example procedure for reproducing the high-energy, 314 mJ spectrum (bottom) from the mid-energy, 130 mJ spectrum, or "baseline" (top). Specific details may be found in the text.

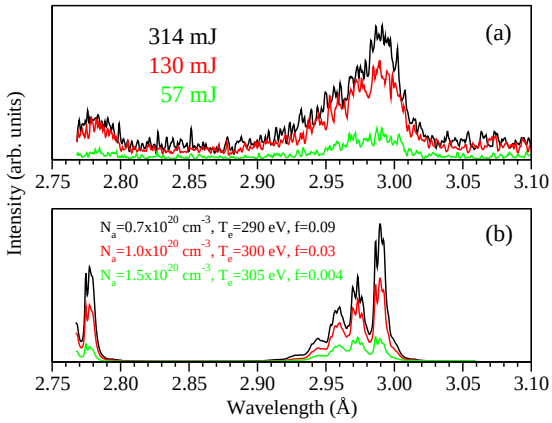



FIG. 7: Comparison of relative intensities for experimental spectra (a) with those of the modeled spectra (b).

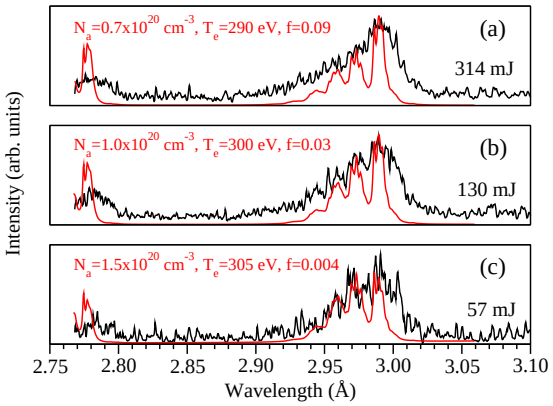



FIG. 8: Comparison of spectral features for each laser energy-dependent spectrum (a)-(c). Experimental results are given in black and modeled results in red.

line" of the mid-energy spectrum, we increase the hot electron fraction from $f = 0.03$ to $f = 0.09$. The intensity has appropriately increased, but unfortunately, so have the intensities of the shorter wavelength peaks. Thus, we now decrease the atomic density from $N_a = 1.0 \times 10^{20}$ cm$^{-3}$ to $N_a = 0.7 \times 10^{20}$ cm$^{-3}$, correcting both the overall intensity as well as the relative intensities of the shorter wavelength peaks. Finally, we fine-tune the major peak by noting that an electron temperature increase will reduce its height whereas a decrease in the electron temperature will increase its height, all the while leaving the other peaks relatively unchanged. Seeking to increase its height, we decrease the electron temperature slightly from $T_e = 300$ eV to $T_e = 290$ eV. An equivalent but opposite procedure for the low-intensity peak results in a set of conditions $N_a = 1.5 \times 10^{20}$ cm$^{-3}$, $T_e = 305$ eV, $f = 0.004$.

Figs. 7 and 8 compare the final results of this procedure with experiment. The first figure, Fig. 7 shows the agreement obtained in reproducing the intensity trend of the experimental spectra. The modeled spectra have of course been left completely unscaled, as required by such a comparison. Secondly, Fig. 8 individually overlays each modeled spectrum with its corresponding experimental spectrum to show the excellent agreement obtained in matching the spectral features, most notably the ratio of peaks between 2.95 and 3.00 Å.

The conditions required to model these experimental trends suggests very interesting physical processes taking place with a strong influence on the spectrum. Increasing hot electron fractions for increasing beam energies is easily understandable in terms of the amount of energy imparted on the cluster in a given length of time. The bulk electron temperature and atomic density dependencies are a bit more subtle. We believe these to be effects of cluster destruction due to the laser prepulse. The prepulse for a high-energy beam is more likely to cause a premature expansion, partially destroying the atomic clusters, and thus decreasing the atomic density. Furthermore, we associate this expansion with a cooling of the bulk plasma, resulting in a slight decrease in the electron temperature. Oppositely, the lowest energy beam has a prepulse which minimally affects the integrity of the clusters, which may be modeled by an increased atomic density with slightly hotter electrons.

We present in Fig. 9 a labeling of the Xe ions which contribute most strongly to each peak in the spectrum. Most notably, our Fe-like peak location is in excellent agreement with the findings of Kondo *et al.* in a previous study of the *L*- and *M*-shell emission of Xe [23]. While we could not reproduce a strong theoretical peak just beyond 3.0 Å, as appears to be present in the experimental spectra, there are ion stages with transitions in this region. This experimental peak is believed to be a result of the prepulse, though we could not amplify the intensity of the theoretical peak for any reasonable physical conditions.

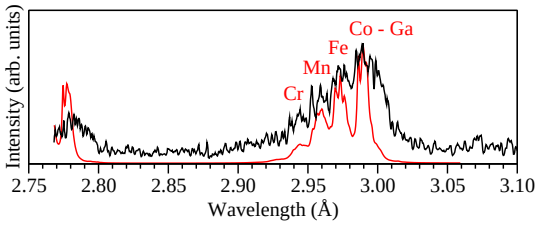



FIG. 9: Identification of major peaks for the modeled spectrum (red) against experiment (black) for the mid-energy, 130 mJ spectrum. Peak wavelength positions are equivalent for all other conditions.

## V. CONCLUSIONS

To summarize, we have performed accurate plasma modeling for the high-$Z$ element xenon, where relativistic effects cannot be ignored. As higher-$Z$ clusters continue to be explored via computational methods, a desire for fine structure detail will necessitate improved methods for retaining relativistic effects in reasonable times. Nonetheless, our atomic physics calculations were performed at a very high level of theory, utilizing relativistic corrections, and plasma emission properties were obtained via a configuration average based solution of the rate equations with a detailed level accounting for all transition arrays with less than $10^6$ lines. This procedure results in an emission spectrum with fine-structure detail which matches experimental results quite well.

Additionally, we presented a procedure for reproducing spectral trends found in experiment, as the laser energy is increased at fixed pulse duration. An increasing beam energy caused an increase in the hot electron fraction but decrease in both the atomic density and bulk electron temperature. This latter characteristic is speculated to be a result of the experimental prepulse, which partially destroys the cluster before the arrival of the main pulse.


### Acknowledgments

I would like to thank my two amazing mentors: Joe Abdallah, whose remarkable physical intuition never ceased to impress me, and James Colgan, for day-to-day assistance and very helpful discussions regarding the work presented here. I would also like to thank our experimentalist collaborator, Anatoly Faenov, for providing both the motivation for this work as well as very helpful suggestions via e-mail correspondence. Furthermore I would like to thank James and Norm Magee for their dedication and hard work in making the summer school the rewarding, enjoyable experience that it was. This work was funded in part by the NSF through the UNM/Los Alamos Summer School in Physics.



[1] T. Ditmire, T. Donnelly, A. M. Rubenchik, R. W. Falcone, and M. Perry, Phys. Rev. A **53**, 3379 (1996).

[2] T. Ditmire, J. Zweiback, V. Yanovsky, T. Cowan, G. Hays, and K. Wharton, Nature (London) **398**, 489 (1999).

[3] K. Yamakawa, M. Ayoama, S. Matsuoka, T. Kase, Y. Akahane, and H. Takuma, Opt. Lett. **23**, 1468 (1998).

[4] Y. Akahane, Ma. Jinlong, Y. Fukuda, M. Aoyoma, H. Kiriyama, J. Sheldakoba, A. Kudryashov, and K. Yamakawa, Rev. Sci. Instrum. **77**, 023102 (2006).

[5] A. S. Boldarev, V. A. Gasilov, and A. Ya. Faenov, JETP Lett. **73**, 514 (2001).

[6] A. S. Boldarev, V. A. Gasilov, and A. Ya. Faenov, Tech. Phys. **49**, 388 (2004).

[7] G. C. Junkel-Vives, J. Abdallah Jr., T. Auguste, P. D'Oliveira, S. Hulin, P. Monot, S. Dobosz, A. Ya. Faenov, A. I. Magunov, T. A. Pikuz, et al., Phys. Rev. E **65**, 036410 (2002).

[8] F. Dorchies, F. Blasco, T. Caillaud, J. Stevefelt, C. Stenz, A. S. Boldarev, and V. A. Gasilov, Phys. Rev. A **68**, 023201 (2003).

[9] J. Abdallah Jr., G. Csanak, Y. Fukuda, Y. Akahane, M. Aoyama, N. Inoue, H. Ueda, K. Yamakawa, A. Ya. Faenov, A. I. Magunov, et al., Phys. Rev. A **68**, 063201 (2003).

[10] Y. Fukuda, K. Yamakawa, Y. Akahane, M. Aoyama, N. Inoue, H. Ueda, J. Abdallah Jr., G. Csanak, A. Ya. Faenov, A. I. Magunov, et al., JETP Lett. **78**, 115 (2003).

[11] Y. Fukuda, Y. Akahane, M. Aoyama, N. Inoue, H. Ueda, Y. Kishimoto, K. Yamakawa, A. Ya. Faenov, A. I. Magunov, T. A. Pikuz, et al., Proc. SPIE **5196**, 234 (2004).

[12] Y. Fukuda, Y. Akahane, M. Aoyama, N. Inoue, H. Ueda, Y. Kishimoto, K. Yamakawa, A. Ya. Faenov, A. I. Magunov, T. A. Pikuz, et al., Laser Part. Beams **22**, 215 (2004).

[13] A. Ya. Faenov, S. A. Pikuz, A. I. Erko, B. A. Bryunetkin, V. M. Dyakin, G. V. Ivanenkov, A. R. Mingaleev, T. A. Pikuz, V. M. Romanova, and T. A. Shelkovenko, Phys. Scr. **50**, 33 (1994).

[14] J. Abdallah, R. E. H. Clark, and R. D. Cowan, Los Alamos National Laboratory LA 11436-M-I (1988).

[15] B. J. Archer, R. E. H. Clark, C. J. Fontes, and H. L. Zhang, Los Alamos Memorandum LA-UR-02-1526 (2002).

[16] N. H. Magee, J. Abdallah, J. Colgan, P. Hakel, D. P. Kilcrease, S. Mazevet, M. Sherrill, C. J. Fontes, and H. L. Zhang, in *14th APS Topical Conference on Atomic Processes in Plasmas*, edited by J. S. Cohen, S. Mazevet, and D. P. Kilcrease (New York: Melville, 2004), pp. 168–179.

[17] R. D. Cowan, *The Theory of Atomic Structure and Spectra* (University of California Press, 1981).

[18] J. Colgan, J. Abdallah Jr., A. Ya. Faenov, T. A. Pikuz, I. Yu. Skobelev, V. E. Fortov, Y. Fukuda, Y. Akahane, M. Aoyama, N. Inoue, et al., Laser Part. Beams **26**, 83 (2008).

[19] S. Mazevet and J. Abdallah Jr., J. Phys. B **39**, 3419 (2006).

[20] M. Sherrill, J. Abdallah Jr., G. Csanak, Y. Fukuda, Y. Akahane, M. Aoyama, N. Inoue, H. Ueda, K. Yamakawa, A. Ya. Faenov, et al., Phys. Rev. E **73**, 066404 (2006).

[21] S. B. Hansen, K. B. Fournier, A. Ya. Faenov, A. I. Magunov, T. A. Pikuz, I. Yu. Skobelev, Y. Fukuda, Y. Akahane, M. Aoyama, N. Inoue, et al., Phys. Rev. E **71**, 016408 (2005).

[22] S. B. Hansen and A. S. Shlyaptseva, Phys. Rev. E **70**, 036402 (2004).

[23] K. Kondo, A. B. Borisov, C. Jordan, A. McPherson, W. A. Schroeder, K. Boyer, and C. K. Rhodes, J. Phys. B **30**, 2707 (1997).

# Creating a Potential Energy Surface Sensitive to Localized Charge Distribution

David Brenman (Lewis and Clark College and MST-8), Steve Valone (MST-8)

**Abstract**

Atomistic methods are used to study the behavior of materials based on the atomic level model called a potential energy surface. Currently there is a dearth of precise knowledge pertaining to the charge transfer process at the atomistic level, and the models for these processes are limited in what they can do. We use the electronic energy, given by solutions of the Schrödinger equation, to generate parameters for a potential energy surface where we are able to gain knowledge about the atomistic description of molecules [1]. How well we capture information from the solution to the Schrödinger equation depends on the functional form of the potential energy surface that we choose to parameterize. Here we parameterize a new functional form based on localized charges [2]. This form derives from viewing the many electron Hamiltonians as a set of atomic Hamiltonians that are allowed to interact with each other. Further more electrons are allowed to flow between atom sites. By implementing CDFT (constrained density functional theory [3]) to augment traditional density functional theory calculations [4-5], we are able to confine electrons to regions around each atom [3]. Because of this not only are we able to parameterize the atomistic model, but we are able to characterize electron transfer reactions in molecules [3]. The overarching goal is to utilize our new found knowledge of intermolecular charge transfer and create a computationally economical code that will model the interaction and motion of thousands of $H_2O$, $OH^-$, and $H_3O^+$ molecules over small time steps. The $H_2O$ molecule is also a learning platform for constructing advanced potentials for metal-dioxide materials ($MO_2$).

## I. Introduction

When performing research on the properties and characteristics of a material, scientist have become quite adept at designing experiments for this purpose. A problem with such an approach is that these experiments can be quite costly both in money and time. We seek to study the behavior of materials by modeling their behavior on the atomistic scale. We plan on developing a computer code that will model the interactions of numerous atoms or molecules for an extended period of time. With such a program scientist would be able to learn about the properties of a substance quickly, and with a high level of accuracy, provided that the atomic model is accurate.

To simulate material properties we will be utilizing Molecular Dynamics. Under this method the atoms are treated as classical particles, governed by Newtonian dynamics. With the use of Newtonian dynamics come three requirements. The first requirement is the need to specify initial conditions for the entire system. This entails selecting the initial position and velocity for each atom. The second requirement is the need to specify boundary conditions for the system. The last requirement is to specify the force field that will govern each atom.

The force law governing the Interactions between the molecules is

$$F = ma = -\nabla E$$

(1)

By finding a functional form of the potential energy, E, and negating its gradient we are able to determine the forces that will govern our atomic system.

The forces acting on each atom are determined by the current position of all the other atoms. Hence as soon as the atoms relocate any amount the forces acting on our system will change and will need to be recalculated. Because we can only look at the system for distinct times, to create the simulation that the atoms are moving constantly we have to take note of their positions at small time intervals and peace together their trajectories. This process is called finite difference method. The positions of the atoms at each consecutive time step are dependent on the previous time step's location approximations. Thus for long trajectories errors accumulate. To minimize error we choose a very small time step, femtoseconds. For each time step we then take the current force acting on, position of, and velocity of each atom, and integrate our equations of motion to find the molecules new positions, and velocities. This process can be carried out by multiple processes such as Runge-Kutta, Predictor-Corrector, or Verlet. As an example the Verlet equations are

$$\vec{x}(t+\Delta t) = \vec{x}(t) + \vec{v}(t)\Delta t + \frac{1}{2}\vec{a}(\Delta t)^2$$

(2)

$$\vec{v}(t+\Delta t) = \vec{v}(t) + \frac{\vec{a}(t) + \vec{a}(t+\Delta t)}{2}\Delta t$$

(3)

With these new initial condition and the same boundary conditions we then recalculate the effective force function for the system, and continue the process for however many time steps the user wishes. The danger with this method is that no matter how small the time step, as soon as the molecules move the slightest the effective force function will change, thus because our time steps are not infinitely small each iteration of our calculation will build error. Therefore multiple time step calculations will accumulate error.

The best solution then is to recalculate the potential energy function as often as necessary. Currently the problem is that calculating the potential energy function from first principles is an arduous and costly task. Our goal is to create a potential energy function that is highly accurate and can be recalculated quickly. This paper discusses our first steps toward achieving that goal, and our method for creating an accurate easily calculated potential energy surface (PES).

## II. Developing the General Variational Function

### A) Developing Hamiltonian & general form

From general quantum mechanics we know that theoretically the general Schrödinger equation

$$\mathcal{H}\psi = E\psi$$

(4)

can be used to find the potential energy of any system. However, realistically, the Schrödinger equation can be solved analytically for only one molecular system, the hydrogen atom. For use with a water molecule, composed of 10 electrons and 3 nuclei, we must simplify the Schrödinger equation. Specifically we do this by implementing the Born-Oppenheimer approximation to the molecular Hamiltonian. The molecular Hamiltonian

is

$$\mathcal{H} = -\frac{\hbar^2}{2}\sum_\alpha \frac{1}{m_\alpha}\nabla_\alpha^2 - \frac{\hbar^2}{2m}\sum_i \nabla_i^2 + \sum_\alpha \sum_{\beta>\alpha} \frac{Z_\alpha Z_\beta e^2}{r_{\alpha i}} - \sum_\alpha \sum_i \frac{Z_\alpha e^2}{r_{\alpha i}} + \sum_j \sum_{i>j} \frac{e^2}{r_{ij}}$$

(5)

where α and β represent nuclei and i and j represent electrons.

The Born-Oppenheimer approximation makes the assumption that because the electrons are so much lighter than the nuclei and hence move so much faster than the nuclei that when compared to the rapid motion of the electrons, the nuclei are virtually stationary [1]. The result of this theory is that the full Hamiltonian can be broken down into the electronic and nuclear parts. Separating out the nuclear terms the full Hamiltonian (5) can be simplified to the electronic Hamiltonian (6).

$$\mathcal{H}_{el} = -\frac{\hbar^2}{2m}\sum_i \nabla_i^2 - \sum_\alpha \sum_i \frac{Z_\alpha e^2}{r_{\alpha i}} + \sum_j \sum_{i>j} \frac{e^2}{r_{ij}}$$

$$-\frac{\hbar^2}{2}\sum_\alpha \frac{1}{m_\alpha}\nabla_\alpha^2, \quad \sum_\alpha \sum_{\beta>\alpha} \frac{Z_\alpha Z_\beta e^2}{r_{\alpha i}}$$

(6)

Looking at the nuclear part (6), the first term represents the kinetic energy of the nuclei and, due the Born Oppenheimer approximation, can be ignored. The second term is the repulsion of the nuclei and is already a potential energy, thus it can be added to the potential energy solution given by the Schrödinger equation at a later time.

The electronic part of the Hamiltonian (6) is now in a simple enough form that we can solve the electronic Schrödinger (7) equation analytically.

$$\mathcal{H}_{el}\psi_{el} = E_{el}\psi_{el}$$

(7)

This equation holds within it most of the information we need in creating our PES. $E_{el}$ represents the effective PES for atoms moving in an averaged distribution of electrons.

Instead of solving for $E_{el}$ as an eigen value, we hypothesize that rewriting the electronic energy in a variational form will provide better guidance for a more accurate PES.

To do this we multiply both sides of the electronic Schrödinger equation by $\psi_{el}^*$.

$$\psi_{el}^* \mathcal{H}_{el}\psi_{el} = \psi_{el}^* E_{el}\psi_{el}$$

(8)

We then integrate over all space to integrate over all possible values

$$\int \psi_{el}^* \mathcal{H}_{el}\psi_{el} d\tau = \int \psi_{el}^* E_{el}\psi_{el}\, d\tau$$

(9)

Where $d\tau$ represents the volume element for all the electrons. Then we are able to pull out the electronic potential energy. Finally we divide by the integral of $\psi_{el}^*\psi_{el}$ to isolate $E_{el}$. For short hand we write <> to symbolize the integration over all space. We are then left with the variational form of the potential energy.

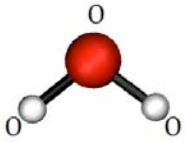
$$E_{el} = \frac{<\psi_{el}|\,\mathcal{H}_{el}\,|\psi_{el}>}{<\psi_{el}\,|\,\psi_{el}>}$$

(10)

**B) Developing appropriate wave function**

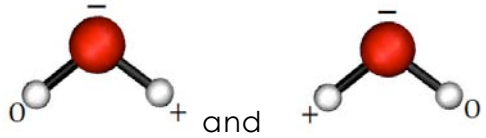
One of the major aspects of our project is to make for certain that the potential energy function is able to accommodate any distribution of charge inside our water molecule and calculate an appropriate potential energy. To do so we choose to break up the electronic wave function into 3 key charge distributions. The new electronic wave function is then composed of a linear combination of these three charge distributions. The first, an arrangement where each atom is neutral, is denoted by $C_{000}\psi_{000}$. The C is called the expansion coefficient and is responsible for determining what

amount of this wave function will be used in the final electronic wave function. Visually this is represented by



(11)

The other two basis charge distributions are $C_{0-+}\psi_{0-+}$ and $C_{+-0}\psi_{+-0}$. These are visually represented by



(12)

The total electronic wave function, composed of a liner combination of these three charge distributions, is written as

$$\psi_{el} = C_{000}\psi_{000} + C_{0-+}\psi_{0-+} + C_{+-0}\psi_{+-0}$$

(13)

For shorthand we will from now on write this formula as

$$\psi_{el} = C_0\psi_0 + C_1\psi_1 + C_2\psi_2$$

(14)

Now that we have the electronic wave function, and we have the electronic Hamiltonian we can insert these into the variational energy equation (10), expand the electronic wave function and receive the final general variational function [2].

$$E_{el} = \frac{C_0^2H_{00} + 2C_0C_1H_{01} + C_1^2H_{11} + 2C_0C_2H_{02} + C_2^2H_{22} + 2C_1C_2H_{12}}{C_0^2 + 2C_0C_1S_{01} + C_1^2 + 2C_0C_2S_{02} + C_2^2 + 2C_1C_2S_{12}}$$

(15)

This equation calculates the potential energy of a water molecule and is a function of 5 variables, 3 spatial and 2 variational,

$$E(Bond_L, Bond_R, \theta, q_L, q_R)$$

(16)

where $q_L$ and $q_R$ are the charges on both hydrogen atoms (see section C), $Bond_L$ and $Bond_R$ are the bond distance between the left and right hydrogen atoms and the oxygen atom, and $\theta$ being the interior angle between the hydrogen atoms. In total five variables. Consequently we have created a five dimensional PES. The inclusion of charge distribution has given us two detentions of increased accuracy over simply using a variable geometry, and gives us reason to believe that this PES will lead to more accurate molecular models.

**C) Relating expansion coefficients to charges**

The next challenge is to solve for the expectation coefficients in the general veriational function (15). The first step is to note that each expectation value will change depending on the original charge distribution that we designate. We begin finding these expectation values by noting that the charge on an atom is determined by the positive nuclear charge of the atom subtracted by the number of electrons currently orbiting around that atom.

$$q_A = Z_A - \vec{N}_A$$

(17)

Where $Z_A$ represent s the nuclear charge for atom A and $\vec{N}_A$ represents the number of electrons around atom A.

In finding the expansion coefficients specifically for our $H_2O$ molecule we will use equation (17) twice for both the left and right hydrogen atoms.

Next we know that $\vec{N}_A$ can be represented by

$$\vec{N}_A = \frac{C_0^2N_{00}^A + C_1^2N_{11}^A + C_2^2N_{22}^A}{C_0^2 + C_1^2 + C_2^2}$$

(18)

To simplify this equation we then divide through by $C_0^2$ on top and bottom. Equation (18) now has the form

$$\vec{N}_A = \frac{N_{00}^A + \frac{C_1^2}{C_0^2}N_{11}^A + \frac{C_2^2}{C_0^2}N_{22}^A}{1 + \frac{C_1^2}{C_0^2} + \frac{C_2^2}{C_0^2}}$$

(19)

We now insert $\vec{N}_A$ back in to equation (15). Adapting this new form for our left and right hydrogen atoms we have the equations

$$q_L = 1 - \frac{N_{00}^L + \frac{C_1^2}{C_0^2}N_{11}^L + \frac{C_2^2}{C_0^2}N_{22}^L}{1 + \frac{C_1^2}{C_0^2} + \frac{C_2^2}{C_0^2}}$$

(18)

$$q_R = 1 - \frac{N_{00}^R + \frac{C_1^2}{C_0^2}N_{11}^R + \frac{C_2^2}{C_0^2}N_{22}^R}{1 + \frac{C_1^2}{C_0^2} + \frac{C_2^2}{C_0^2}}$$

(19)

By knowing the charge distribution on the water molecule for each basis state we fill in the numerical values for all terms with the general form $N_{aa}^R$. The numbers we insert for $N_{aa}^R$ are the number of electrons on the specific hydrogen atom for the charge distribution specified. For example in equation (19) these values would be, from left to right, 1, 1, 0. See equations (12,13,14) for a visual representation of the electron positions. From here we have two equations with two unknowns, and the values for $\frac{C_1^2}{C_0^2}$ and $\frac{C_2^2}{C_0^2}$ can be solved for by basic algebra.
Finally we insert these expansion coefficients into the general variational function (15).

**D) Estimating Hamiltonian matrix elements**

We used a program called NWChem to perform the computationally expensive calculations necessary to find the numerical values for the matrix elements, denoted by $\mathcal{H}_{00}$, $\mathcal{H}_{10}$ or any such form in equation (15). These matrix elements are the ground state potential energy values for the charge distributions depicted in their subscripts. For example $\mathcal{H}_{10}$ stands for the transition potential energy for a change in distribution from $\psi_{0-+}$ to $\psi_{000}$.

To estimate these matrix elements NWChem implements an approximation used to simplify the Schrödinger equation other then the Born-Oppenheimer approximation. This approximation is called Constrained Density Functional Theory (CDFT).

To explain how CDFT works it is necessary to understand how Density Functional Theory (DFT) works. When solving for the potential energy using DFT, the energy is no longer dependent on the electronic wave function but rather on the electron density, depicted by figure (A). Figure A depicts a cross section of the electron density around the Oxygen atom, shown in green.

$$E[\psi_{el}] = E[\rho_{el}]$$

(20)

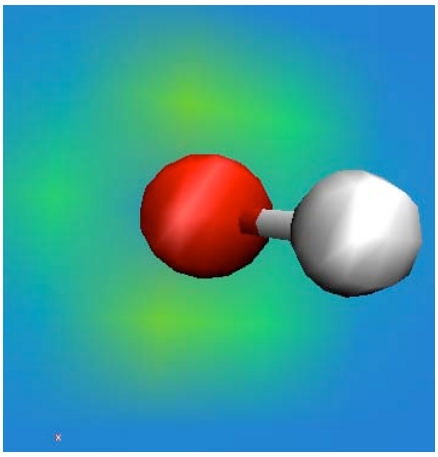


Fig. A: The green cloud is a cross section of the electron density around the center of the oxygen atom.

The cause of this transformation is the Hamiltonian is no longer a functional of the electronic wave function rather; it is now a functional of the electron density and is a much simpler Hamiltonian:

$$\mathcal{H}(\psi_{el}) = \mathcal{H}(\rho_{el})$$

(21)

In practice we only know approximations to $\mathcal{H}(\rho_{el})$. The electron density is a summation of all electron orbitals:

$$\rho_{el}(r) = \sum_{i\sigma} \varphi_{i\sigma}^2(r)$$

(22)

If E[p] is stationary and the orbitals are normalized then the result of these two constraints is the Kohn Sham equations [4,5].

$$\left(-\frac{1}{2}\nabla^2 + V_n(r) + \int \frac{\rho(r')}{|r-r'|}dr' + V_{xc\sigma}(r;\rho)\right)\varphi_{i\sigma} = \epsilon_{i\sigma}\varphi_{i\sigma}$$

(23)

where the φ's are no longer the previous complicated wavefunctions, they are now single particle orbitals. The $-\frac{1}{2}\nabla^2$ stands for the kinetic energy of the electrons. $V_n(r)$ is the external potential and represents the electron nucleus interaction:

$$V_n(r) = -\sum_j \frac{Z_j}{|\vec{r} - \vec{R_j}|}$$

(24)

The third term, $\int \frac{\rho(r')}{|r-r'|}dr'$, is the electron electron interaction. Finally $V_{xc\sigma}(r;\rho)$ is the exchange correlation.

From here the Kohn Sham equations can be solved using self consistent methods. We first make a guess as the orbital for the first electron. With this guess we are then able to determine the corresponding electron density, □. With the electron density we are then able to construct a working Hamiltonian, as seen in the first part of equation (23). We then solve the Kohn Sham equation as an eigenvalue problem and are able to determine the new orbitals for the electrons and its corresponding potential energy. From here we start the process over again, using our newly obtained orbital. We continue this process until the resulting orbitals are very similar to the previous orbital. We then perform the same procedure for each of the electrons, and finally add up all their energies and other contributions to obtain the ground state potential energy for that molecule.

NWChem takes the same basic approach except for the key difference that it adds a constraint to the Kohn sham equations. This constraint forces a certain number of electrons to be on a specific atom at any moment.

To precisely solve for the Hamiltonian matrix elements found in equation (15) one would need to have highly sophisticated wave functions multiply the Hamiltonian that is operating on another set of these highly sophisticated wave functions. For example the matrix element for the basis state 0-+ is denoted $H_{11}$. This notation stands for

$$< \psi_{0-+} |\mathcal{H}| \psi_{0-+} >$$

(25)

By using CDFT we are making the argument that the potential energy given to us by this approximation method is very similar to the real potential energy from the very sophisticated wavefunction.

Just as in DFT we started to solve for the potential energy of a charge distribution by making the approximation that the potential energy can be dependent on the electron density and not the electronic wave function. The constrained Kohn

Sham electric energy equation takes the regular electronic energy and add to it a constraint term, yielding

$$W[\rho, V_c] = E[\rho] + V_c\left(\sum_\sigma \int w_c^\sigma \, \rho^\sigma(r)dr - N_c\right) \quad (26)$$

$$N_c = \sum_\sigma w_c^\sigma(r)\rho^\sigma(r)dr$$

(27) where E[p] represents the unconstrained KS electric energy equation, $V_c$ is a Lagrange multiplier, and $N_c$ (27) is the general constraint to the density.

Just as with the DFT method we require that the orbitals are normalized, and W[p, $V_c$] stationary obtain the constrained version of the KS equation:

$$\left(-\frac{1}{2}\nabla^2 + v_n(r) + \int \frac{\rho(r')}{|r-r'|}dr' + v_{xc\sigma}(r) + V_c w_c^\sigma(r)\right)\varphi_{i\sigma} = \epsilon_{i\sigma}\varphi_{i\sigma}$$

(28)

The term $V_v w_c^\sigma(r)$ in equation (28) is the new constraint. We can solve these equations using self consistent methods, as in the previous example for the unconstrained KS equations. The only difference is that we update $V_c$ along with the orbitals. NWChem then repeats this process until $V_c$ satisfies the constraint equation (27). Once the constraint is met NWChem uses the resultant $\phi_{i\sigma}$, in the next self consistent iteration and yields the ground state energy for the **constrained** system. This process is discussed in vastly greater detail by Wu and Van Voorhis [3].

With the Hamiltonian matrix elements all determined these values are then inserted into the general variational formula (15).

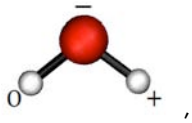
Through this approach we are able to find the specific potential energy for any charge distribution.

## III. Results

Using our method we were able to model the potential energy for a water molecule with any symmetric geometry and charge distribution. The following graphs are calibrated such that zero potential energy is a state where all the electrons have been striped form the nuclei.

Graph (B) is a potential energy surface for a water molecule at optimized geometry, with the right hydrogen constrained to be neutral. The first point on the left represents the state where each atom has a neutral charge. The next point is the charged state where 1/10 of an electron has been shifted from the left hydrogen to the oxygen. It is important to make the connection that this graph is a visualization of the general variational function, equation (15), with all but $q_L$ held constant.

It is also important to make the connection that the point farthest to the right is the precise charge state 0 - +

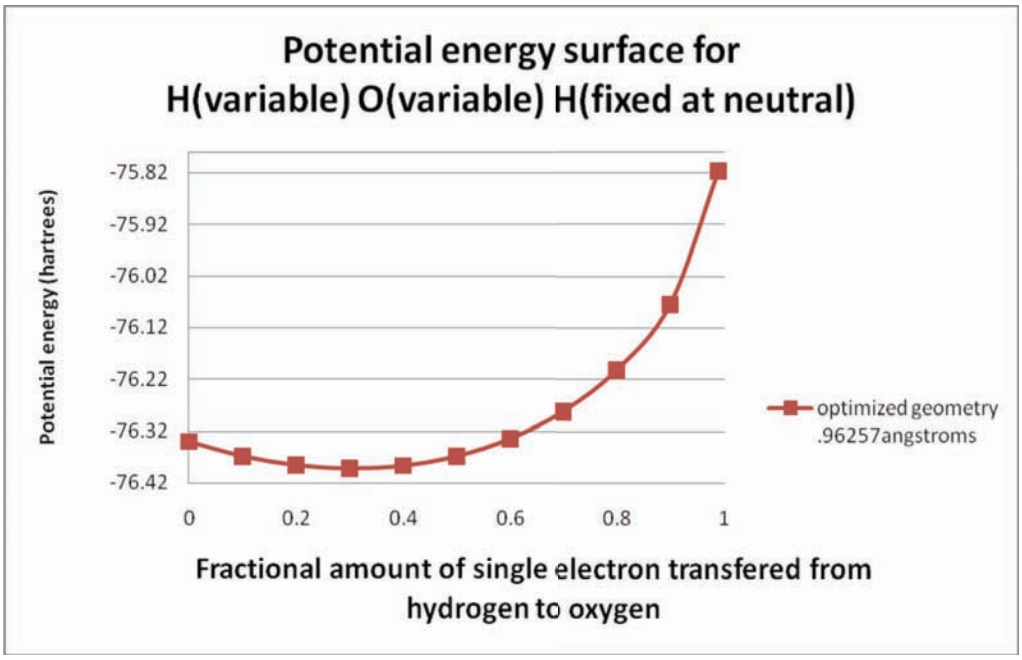



,

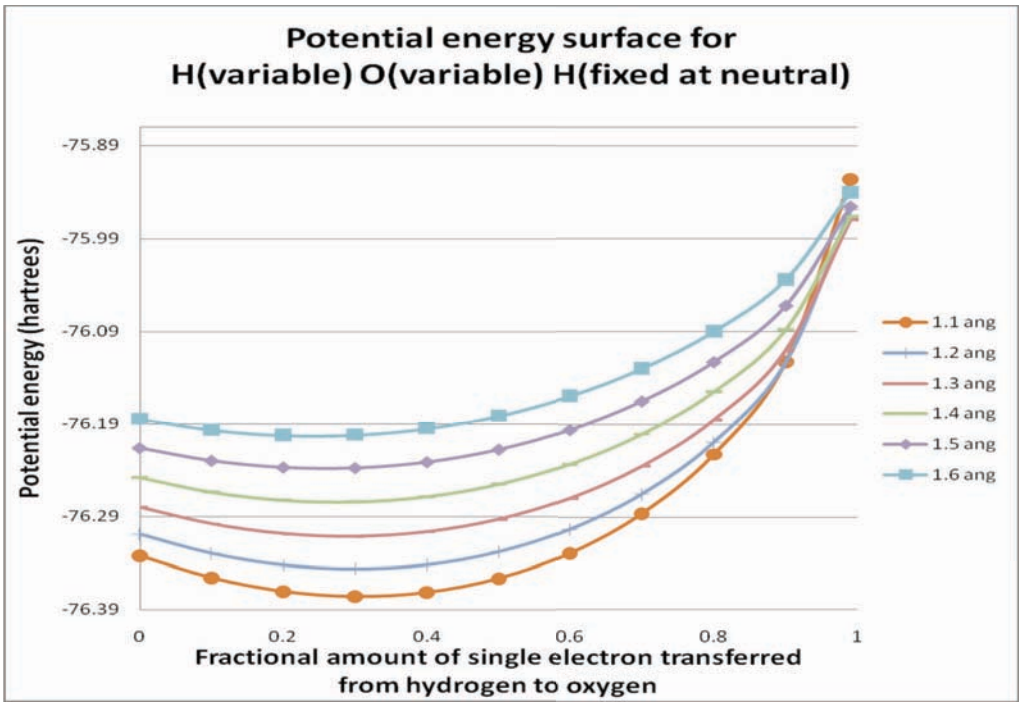
Hence if we were to write out the specific variational function for this charged state the expiation coefficient, $C_1$, would be 1 and all the other expansion coefficients would be zero.

Each point in the middle of the graph has been calculated using CDFT for that particular charge distribution.

Finally it is important to see that the graph's curvature is smooth. Such a shape is a sign that the molecule is close if not to equilibrium.




Graph B: The smooth curvature of this graph signifies that the molecule is highly interactive at any of the charge distributions




Graph C: By varying the bond length between the oxygen and hydrogen atoms we are able to map the plot numerous potential energy surfaces with the same charge distribution.

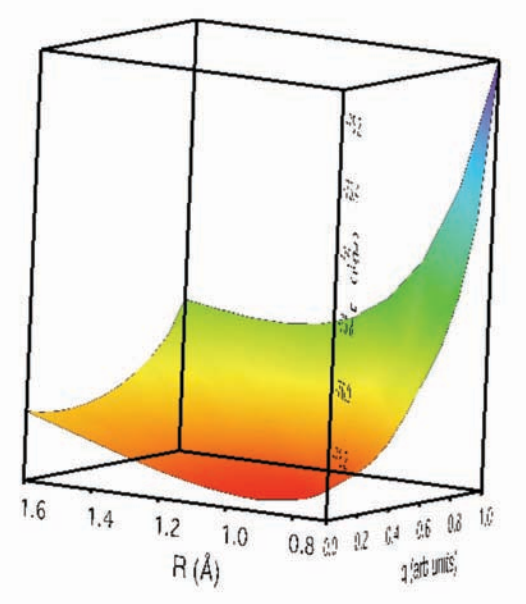
Turning our attention to graph C we see six variations of the geometry on the water molecule. The bond length between the oxygen and the hydrogen atoms has been extended,

symmetrically, from 1.1 angstroms to 1.6 angstroms. We can see that as the bond length is extended, beyond its normal optimized length, each consecutive extension plots higher potential energy surfaces. This is because we are forcing the water farther and farther from equilibrium.

Examining the right side of graph B, we see a cluster of end points. This tells us that when the charge distribution is close to +-0, the water molecule feels disassociated. In other words if we were to extend the bond lengths of the water molecule to 100 angstroms, the potential energy would be close to the potential energies plotted on the right of the graph.

Lastly in graph C there is a trend where the farther the bond lengths get from the equilibrium geometry the less smooth the curves become.

Graph D depicts a two dimensional model of our potential energy surface here we are varying bond length and charge distribution. This graph is a composite of the graphs B and C.




Graph D: This 2D plot is a vitalization of the general variational function with respect to bond length and charge distribution between the first hydrogen and Oxygen atoms.

## V. Future work

The next step is to construct a potential energy surface for any configuration of water geometry. After we are able to do this we will attempt at modeling the interaction between two water molecules. This is an ambitious goal and there is still a great deal of work to do. In the distant future we will attempt at modeling crystalline solids.

[1] I. Levine. Quantum Chemistry. Boston: Allyn and Bacon, Inc. 1974.
[2] S. M. Valone, J. Li, and S. Jindal, Int. J. Quant. Chem. **108**, 1452-64 (2008).
[3] Q. Wu, and T. Van Voorhis, Phys. Rev. A **72**, 024502 (2005).
[4] P. Hohenberg, W. Kohn, Phys. Rev. Vol. **136**, 3B (1964).
[5] W. Kohn, and L. J. Sham, Phys. Rev. Vol. **140**, 4A (1965).

# Passive Radiography Image Analysis

Sara R. Goltry
*The College of Idaho, Caldwell, Idaho*

Hanna Makaruk
*Mentor, Physics Division, P-21, Los Alamos National Laboratory, New Mexico*

Passive radiography is a novel technique with a growing number of applications. Cosmic-ray muon radiography allows for investigation of the inner structure of pyramids, volcanoes, industrial structures, and potentially also big cargo containers for border security applications. This work is connected with the application of passive gamma radiography in detecting nuclear threats and radioactive contaminants in urban areas. This report describes a procedure for image analysis for source recognition, and development of software automating this procedure.

## INTRODUCTION

Passive imaging techniques are applied in a variety of areas: archeology, geology, civil engineering, homeland security, even oceanography. Using cosmic-ray muon detectors, Alvarez et al [1] were able to determine that Chephren's second pyramid at Giza, unlike its neighboring pyramids, does not contain any large chambers above the King's chamber. The use of muon radiography enabled researchers to reach this conclusion without necessitating destructive and costly excavations into the pyramid's unexplored regions. In Japan, Tanaka et al [2] used muon radiography to safely make measurements on an active volcano. Passive acoustic tomography, a technique that is similar in many respects to passive radiography, uses naturally occurring ocean noise such as aquatic mammal calls and has been explored as a new way of gathering marine data without the noise pollution of sonar [3]. Such methods eliminate the biological and environmental hazards associated with active imaging techniques.

As access to nuclear technology spreads, the need to develop quick and reliable means of detecting nuclear threats becomes ever more pressing. This work is focused on development of an algorithm for automated radiation source recognition and location that will be implemented in a mobile radiation detection unit and will use passive radiography to pinpoint the location of radioactive sources. The apparatus consists of a set of Compton and coded aperture type of detectors placed in a truck. It is planned to passively record radiation data about the surrounding area, and then analyze that data to determine the existence and location of potential nuclear threats. Coded aperture imaging uses a mask with a pattern of holes placed in front of the detection plate. See Figure 1. The unique pattern of shadow and light cast on the detection plate is used to recreate an image of the source using 3D geometry [4]. Compton imaging uses a pair of gamma ray detectors in series to measure the energy and location of incident and Compton scattered gamma rays. This information is used to extrapolate three-dimensional spatial information about the source [5]. Combined, these two methods provide high resolution images for gamma radiation over a broad energy spectrum [6].

The focus of this summer project is writing software procedure instrumental in automated source detection from Compton and coded aperture images.

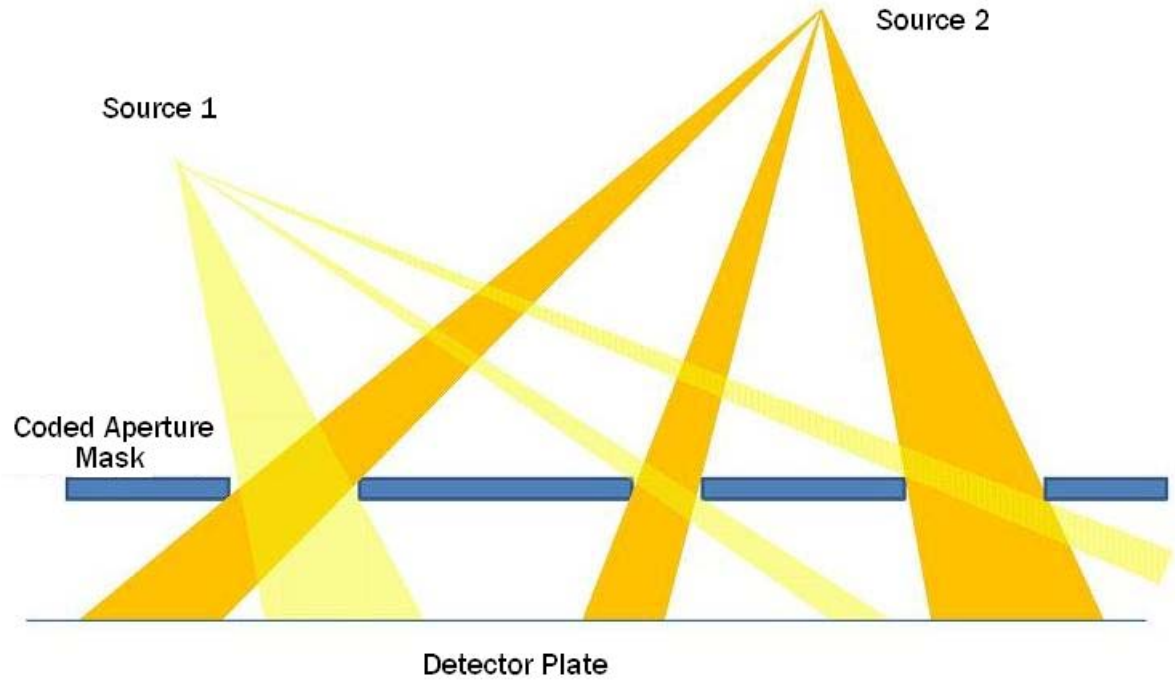



Fig. 1: 2D schematic of a coded aperture device. Source location is determined based on the shadows cast over the detection plate.

**METHODS**

The images we worked with are computer simulations of a test source superimposed over a typical background. They include files for 1 to 59 seconds of source exposure time with the background.

We start by establishing a source signature, which differentiates it from typical background. Software prototyping in Mathematica was used for image analysis. The developed algorithm, after optimization and testing, was translated into C++ for efficiency and compatibility with the main project code.

Image analysis shows that the typical size, shape, and intensity gradient of a source do not distinguish it from a typical background fluctuation. There are only two remaining measured characteristics which may serve for source identification, namely differences in expected energy spectrum and relative intensity. Background fluctuations have broad spectra, while a contaminating source spectrum is dominated by one or at most a few spectral lines. A source is expected to have a relatively narrow spectrum band. This is why subtraction of different

spectrum energy bands from each other for the same image may suppress the background without changing a source image in any significant way.

Since soil and bedrocks composition causes background radiation levels to vary wildly from one geographical area to another, absolute radiation intensity cannot serve in source detection unless software is tailored to a specific geographic region with known, consistent background radiation level. Relative intensity is therefore the characteristic of primary concern, and all images are rescaled to a 0-255 intensity scale.

Figure 2 shows example source and background images in two-dimensional gray scale. The top image simulates 20 seconds of exposure to the test source, while the bottom image simulates exposure to just background radiation. Figure 3 shows three-dimensional plots of the same data, where intensity is measured on the z-axis. The three-dimensional plot of the source image contains one peak which is much more prominent than the others, while the analogous plot of the background image contains several peaks of similar maximum height.

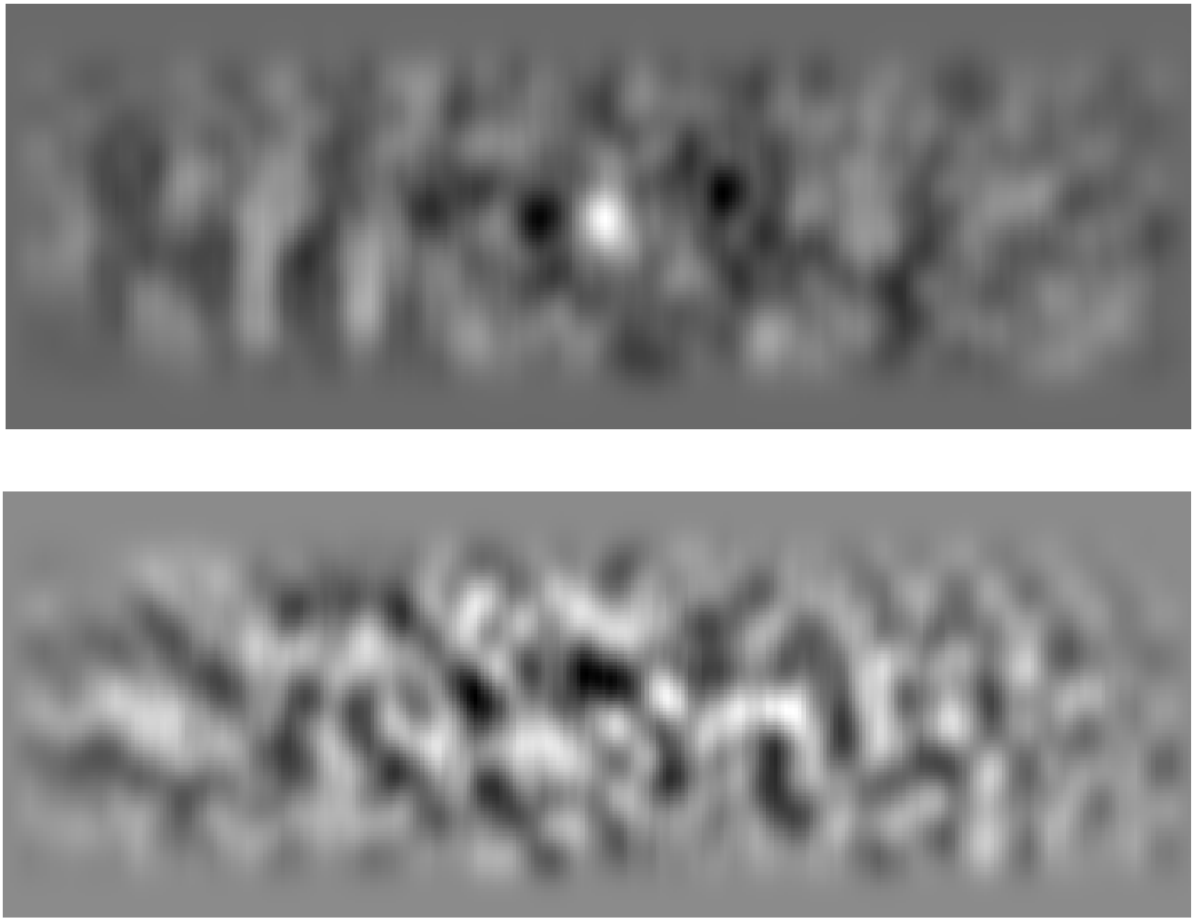


Fig. 2: Computer simulated test images. Top image includes a source in the center, bottom image represents background only.

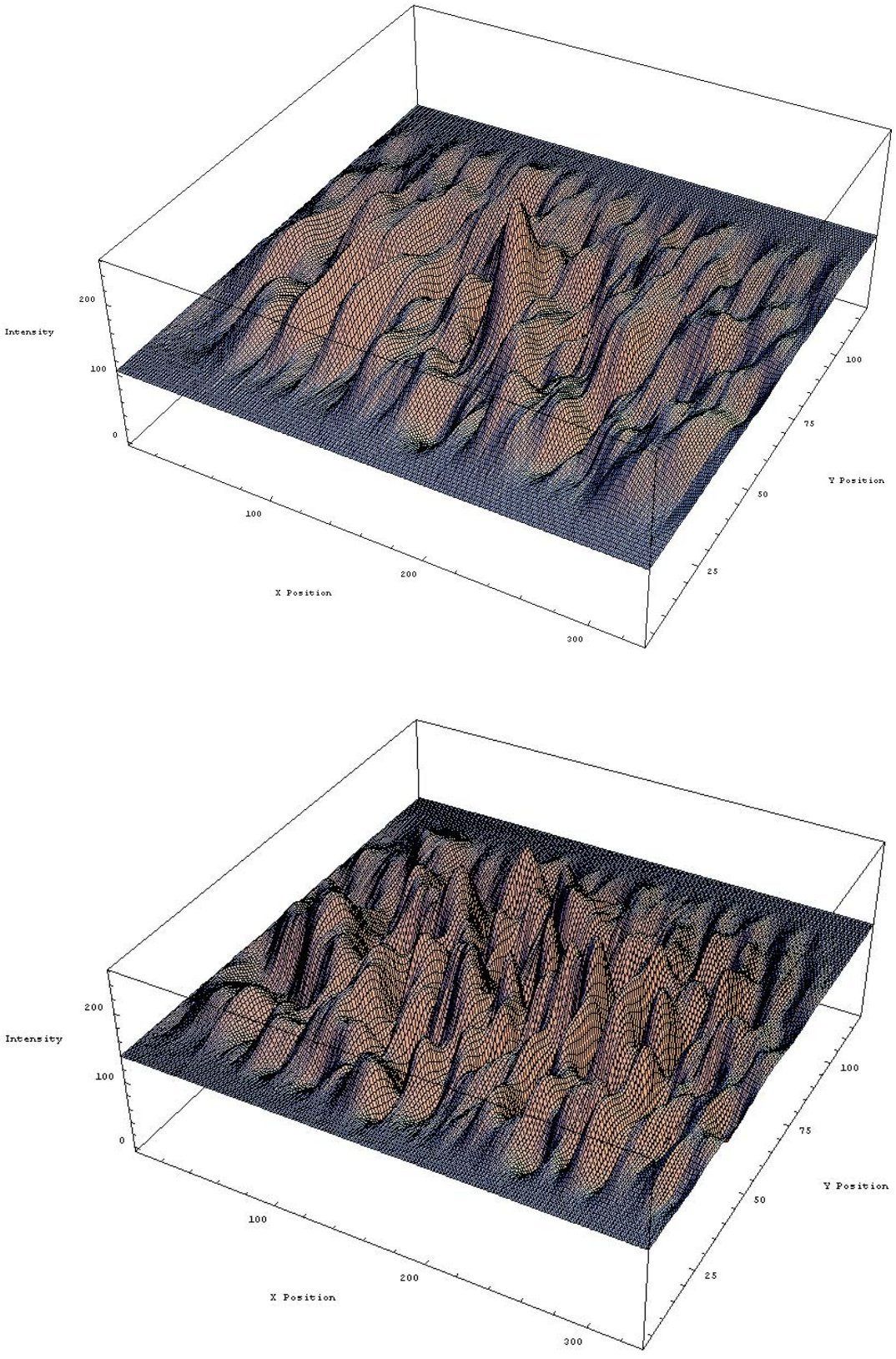



Fig. 3: 3D plots of intensity for the same test images. Source is located in the center of the top image.

It is important to know not only intensity of the background but also the typical intensity difference between peaks in this background. The program has a threshold parameter inserted. This threshold parameter has to be tuned to the background characteristics in such way that for all the backgrounds, multiple peaks are still visible above the threshold. A source which stands up more from the maximum of the background than peaks of the background differ between each other can be identified. This means that characteristics of the background determine which source strengths can be identified. For detection of relatively weak sources, preprocessing of the background, which decreases its intensity and also smoothes the differences between background peaks, is needed.

After thresholding, all pixels with values equal to or above the threshold are given a value of 1 (true), while those with values below the threshold are reassigned a value of 0 (false). Each image from the image set on which this program was developed is thresholded for a range of values from 215 to 255 to determine which thresholds produce the optimal source and background recognition. Figure 4 shows the same source and background images after a threshold intensity of 225 is applied. The source image exhibits only one white spot. Because the images have been rescaled to account for the relative intensity, the background image contains multiple white regions, as its peaks do not significantly differ in the intensity.

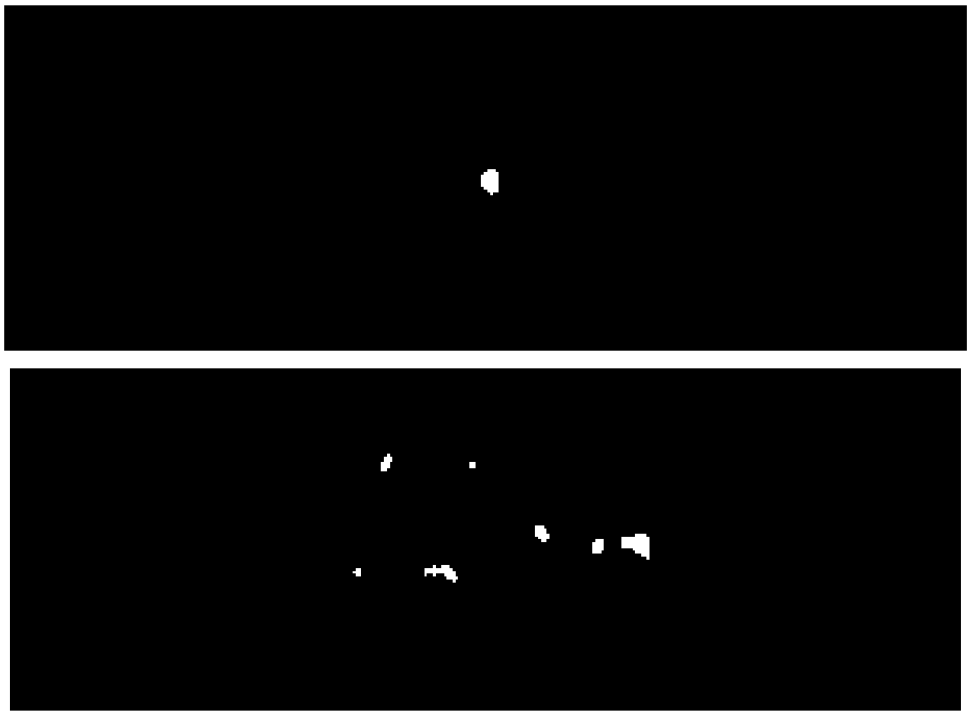


Fig. 4: Simulated images after a threshold intensity of 225 has been applied. The top image shows a localized source above the threshold; the bottom image shows non-localized background noise.

To quantify the spread of white (1 valued) pixels from the mean source position, the variance of the location of the white pixels for each image is calculated.

$$Variance = \frac{1}{image\ height * image\ width} \sqrt{\sum_{all\ white\ pixels} [(x_i - x_{mean})^2 + (y_i - y_{mean})^2]}$$

A low variance denotes localized activity above the threshold, while high variance indicates background noise. For images such as ours with dimensions of 333 by 120 pixels, a variance of 0.7 corresponds to a small test source located 50 to 100 feet away. It is assumed that most sources will be of this type. In case of changes in pixelization of the image, or in case of significant differences in the geometry of the scene, the cut-off variance has to be tuned. Although larger sources with very high variance "blind" the detectors when they are nearby, the mobility of the detection unit allows recognizing them at a distance. As the truck approaches from far away, large sources will appear as small points and have small variance. A cut-off variance of 0.7 is therefore used to signify the presence of a source. For images with variance less than or equal to this cut-off, the source location is given as the mean x- and y-position of the white pixels.

In the case of our example images, the source image has a variance of 0.0864 and the source location in pixels is (167.8, 58.3). The background image has a variance of 2.31 and is correctly identified as background

## RESULTS

The algorithm was tested on the whole set of simulated Compton and coded aperture images which were serving for its development. See Figure 5.

For coded aperture images containing source exposure 20s and more, correct source identification is in the range of 95% for all tested thresholds and achieves 100% for thresholds of 224 and up.

For background images the correct identification was 100% up to the threshold 229 and then diminished rather quickly for higher thresholds. Evidently the higher thresholds are already in the range of typical differences between background peaks. This is why they cannot be used for source identification for this type of background.

As for coded aperture images with 15s and 10 s source projection correct source recognition was 90% starting from the threshold 220.

In case of source identification the program produces its location. There are two sets of coordinates for each source: one is the location of the maximum intensity, and the second is mean of the coordinates for all the pixels over the threshold. Coordinates calculated in both ways differ less than one pixel in each direction from each other and from the source location.

For Compton images the parameters need to be tuned as both the pixelization and character of the background changes, but the procedure stays the same.

As simulations performed in this project continue to evolve to more closely resemble experimental data, therefore, the source recognition algorithm needs to be tuned to the specifications of the background radiation. Suppression of the background in the preprocessing is needed when attempting to recognize sources weaker than the criteria discussed before.

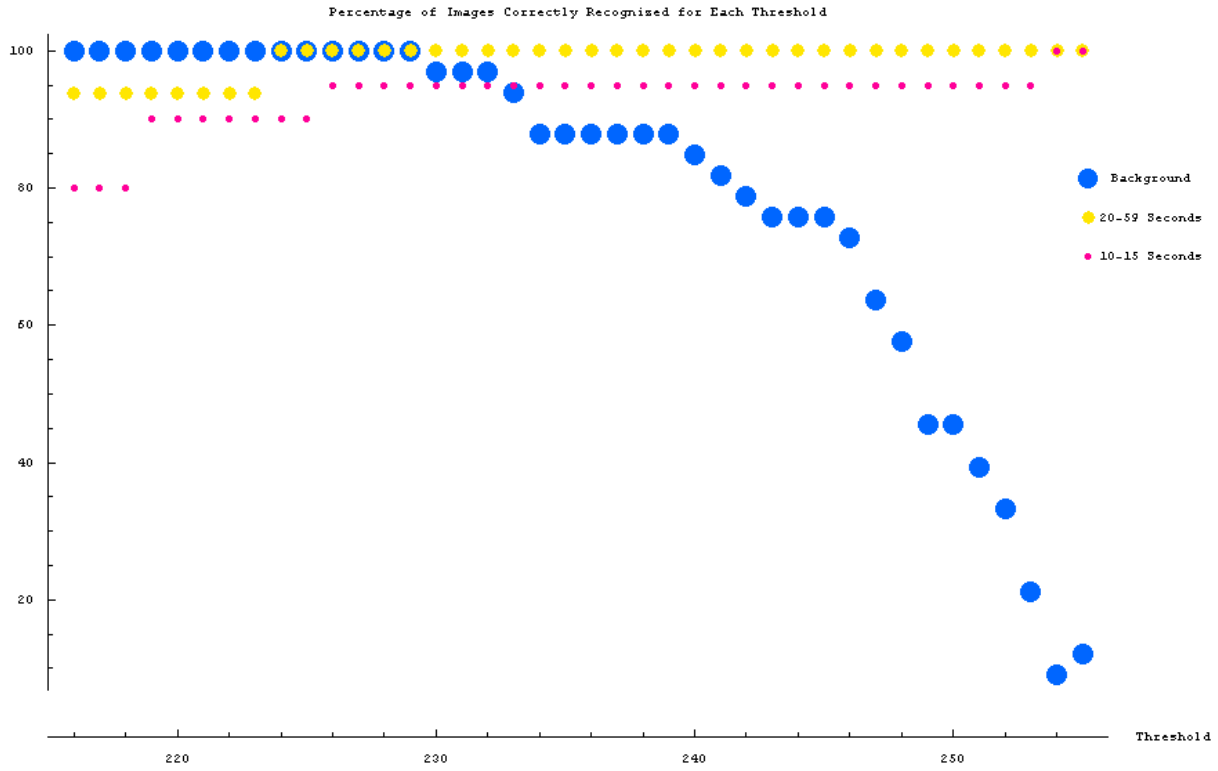



Fig. 5: Percentage of correctly identified images versus the threshold value used. Below the threshold of 230, all background images are correctly identified, but the recognition rate deteriorates rapidly for higher thresholds. 93.75% and 100% of 20-59 second source exposure images are accurately recognized for threshold ranges of 215-233 and 224-255, respectively. Simulated 10-15 second source exposure images are correctly identified 95% of the time for thresholds of 226-253, but achieve 100% recognition only for very high thresholds.

**ACKNOWLEDGEMENTS**

I would like to thank James Colgan and Norm Magee, the organizers of the Los Alamos Summer School in Physics. They went out of their way to make this summer a great experience. I would also like to thank my mentor Hanna Makaruk, Peter Volegov, Shawn Tornga, Larry Schultz, Mark Wallace, and the rest of the SORDS team for all the assistance they provided along the way.

## REFERENCES


[1] Luis W. Alvarez, et al, Science, New Series **167**, 832-839 (1970)
[2] H.K.M. Tanaka, et al, Nuclear Instr and Methods in Phys Research A **575**, 489-497 (2007)
[3] C. Gervais, et al, Journal of the Marine Biological Association of the U.K. **87**, 5-10 (2007)
[4] T. M. Cannon, E. E. Fenimore, Opt Eng **19**, 283-289 (1980)
[5] Gary W. Phillips, Nuclear Instr and Methods in Phys Research B **99**, 674-677 (1995)
[6] M. Forot, et al, Nuclear Instr and Methods in Phys Research A **567**, 158-161 (2006)

# The Effects of Coulomb Impurities and the Coulomb Pseudogap in Graphene

Robert Hembree[1]

[1] *Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*

(Dated: Printed August 11, 2008)

In this paper the effects of coulombic interactions on the density of states in graphene are explored. It is found that in the case of suspended graphene where poor electron screening is present and as a result the Coulomb pseudo gap exists in the presence of an impurity, that the coulombic interactions affect on the density of states is far stronger than the density of states caused by Dirac dispersion. In the coulomb effect on the density of states, which is strongly dependent on the dielectric constant and electron screening, is 21-30 times greater than the density of states arising form the Dirac dispersion. Scanning Tunneling Microscopy (STM) can be used to measure the Density of States and observe this effect.

## I. INTRODUCTION

The semiconducting industry is a multibillion dollar per year industry.[1] A semiconductor is a material whose conductivity is somewhere between that of a conductor and that of an insulator. This conductivity can be controlled through introduction of new states to the band gap. The band gap is the distance between the valence band and the conduction band. These new states can be added to the band gap through the introduction of impurities, called dopants (see figure 2). A method of controlling these impurities is a method of controlling the conducting properties of the semiconductor. Therefore it is important that we understand the effects of adding these impurities so that we may intentionally use the impurities to affect the material in a controlled manner.

Graphene, a newly discovered and the first and only observed two-dimensional crystal, has some interesting electronic properties.[2–9] Graphene is a zero gap semiconductor where the linearly dispersing valence and conduction band touch at the corners of the Brillouin zone(BZ).[2,10] Because of this, the electrons in Graphene behave like massless Dirac Fermions with the speed of light being replaced with the Fermi velocity, $v_f \approx \frac{c}{300}$. This also causes density of states to dissappear to zero at the corners of the BZ.[11] This also allows for a high charge carrier mobility on the order of $10^4 - 10^5 \frac{cm^2}{Vs}$, ten times higher than that of silicon.[12,13] This high charge carrier mobility, even at high charge densities creates a potential for application in nanoelectronics[14]. This property can be used to improve processing power of computer chips.

The absence of an energy gap[2] creates numerous opportunities to control the transport properties in graphene. This creates a situation in which there are no electrical switches, an essential feature used in silicon in the prduction of transistors.[15]. But. new band states can be introduced by using narrow ribbons of graphene.[16] What we would like to be able to do is introduce new band states into graphene that we can control. It is possible to introduce new states to the insulating gap.[4] Futhermore the states in the band gap can be controlled externally by the field effect.[17]

This energy gap arises from density of state calculations. The density of states is the number of states available to a particle for a given energy, that is, number of states per area per energy. In general the number of states will be equal to the number of particles[18]. In crystalline structures however, this is not true. In theses structures the atoms are close together causing the wave functions to overlap allowing the nuclei and electrons to have coulombic ineractions[18]. This overlap also causes the energy spectrum to split resulting in banding. When calculating the density of states in one of these systems a situation arises where there is a minimum gap between the valence and conduction bands in the density of states around specific points in the Brillouin zone. This gap is called the band gap. In graphene this point occurs at the corners of the Brillouin zone, and the minimum gap distance is zero (Note that these bands are not overlapping, but overlap can occur). From this, numerous electrical and optical properties can be determined for the given crystal. Therefore, to understand these properties, it is important to understand the density of states of a system

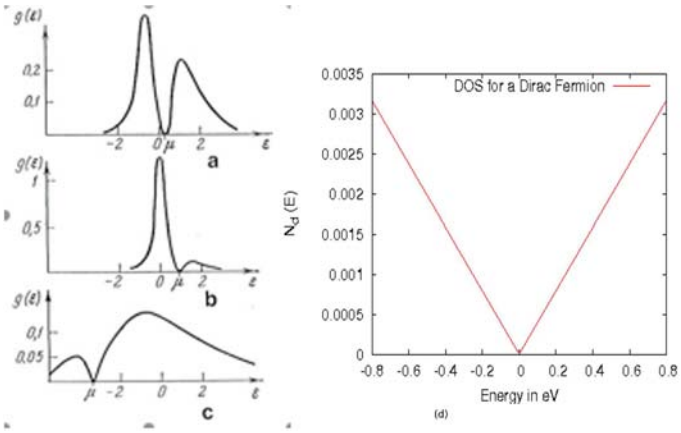



FIG. 1: (a)-(c) The DOS for the coulomb potential. Note how the DOS drops to zero at or near the fermi level. This is the Coulomb Pseudogap[19](d) The DOS for Dirac Dispersion, $\frac{4|E|}{9\pi t^2 a^2}$ where t is the hopping parameter and a is the lattice constant.

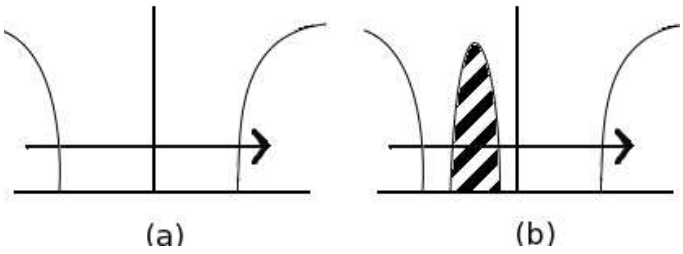



FIG. 2: (a) The original band gap. (b) The band gap with newly created states from introducing impurities.

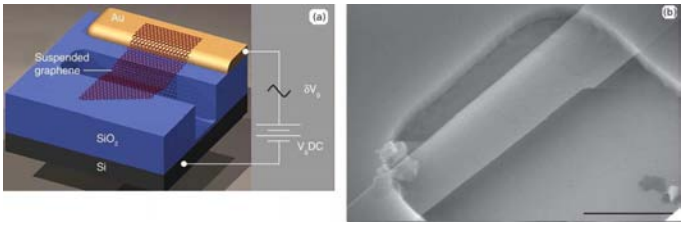



FIG. 3: (a) Schematic of a Suspended Graphene system[20] (b) An image of Suspended Graphene[20].

so that these properties may be explored in a theoretical manner.

In order to study the effects that introduced impurities have on the band gap we need some way to measure the density of states. We can treat these introduced impurities as Coulomb charged impurities. From this we will see as a result of poor screening the emergence of a Coulomb Pseudogap in the density of states (DOS)[19]. In the case of a graphene layer on a substrate, the substrate assists in the electron screening and masks any further unscreened effects on the graphene.[10] To prevent these effects we can create a suspended graphene which allows us to see the Coulombic charge impurities as unscreened or poorly screened. From this we see that the Coulombic impurity induced DOS on a two dimensional sheet of graphene is linear with respect to energy with the equation:

$$N_c(E) = \frac{2|\tilde{E}|\kappa^2}{e^4} \tag{1}$$

Because of this one deeds to compare the Coulomb interactions with the DOS due to the Dirac Points[10], which is:

$$N_D(E) = \frac{|E|}{\pi(\hbar v_f)^2} \tag{2}$$

From this we can take a ratio of $N_c/N_D$ and see that the Coulombic impurity affects the DOS 21 to 33 times more thant the DOS due to the Dirac points. From this we can study the DOS as a function of the dielectric constant dependent on the energy.

## II. THE DENSITY OF STATES CAUSED BY COULOMBIC IMPURITY INTERACTIONS

In this section we derive the equation for the DOS caused by coulombic impurity interaction between the electron donors and acceptors. The purpose of this calculation is to compare the results of the coulombic DOS with those caused by the Dirac dispersion at the node points in the BZ. To show this the argument of[10,19] will be repeated.

The DOS caused by the Coulombic correlation effects has 2 humps in its dependence on energy (see fig 1). This is caused by the DOS vanishing at the Fermi level. The area where the DOS drops to zero is called the Coulomb Pseudogap. A similar effect is seen at the Dirac points in graphene where the DOS vanishes[2].

To begin the calculations let us consider a randomly arranged system with 3 types of particles. The first type of particle is a positively charged donor. These have donated their electrons to acceptors. The second type of particle is the neutral donor. The third type of particle is an acceptor which always contains an electron and is negatively charged at the temperatures under considerations. That temperature is what ever temperature is required by the system to be at the minimum energy state, in this case that is the critical temperature $T_c$. We also assume that the system is in thermodynamic equilibrium and all impurities are fixed in space. The distribution of the particles is determined by the condition of minimum electrostatic energy. I also assume that the electron screening potential is low. The energy is given by the equation:[19]

$$H = \frac{e^2}{\kappa}\left[\frac{1}{2}\sum_{k}^{don}\sum_{k\prime\neq k}^{don}\frac{(1-n_k)(1-n_{k\prime})}{r_{kk\prime}} - \sum_{k}^{don}\sum_{i}^{acc}\frac{1-n_k}{r_{ik}}\right] + \frac{e^2}{\kappa}\left[\frac{1}{2}\sum_{i}^{acc}\sum_{j\neq i}^{acc}\frac{1}{r_{ij}}\right]$$

Where $n_k$ , the occupation number, is 0 if the particle is ionized and 1 if the particle is a neutral donor. In this equaton the first term is comes from the coulombic repulsion between the donor particles, the second term comes from the Coulombic attraction between the donors and the acceptors, and the final term comes fromt he natural repulsion between the acceptors. Here it can already be seen that a gap can arise in the energy of the system if the energies of the repulsions match the energy of the attractions.

The set of $\{n_k\}$ can be found by minimizing the problem with a fixed number of electrons $\sum_k n_k$. Needless to say this is a very difficult problem. So, instead we use techniques from density functional theory. To this end we must introduce some new variables. First is:

$$\tilde{H} = H - \mu\sum n_k \tag{3}$$

where $\mu$ is the fermi level(chemical potential) as determined by the neutrality conditions. While $\tilde{H}$ gives the energy of the whole system, we are actally interested only in the single electron energies $\epsilon_i$ rather than the occupation numbers. The reason being is that we are interested in is the DOS of a single electron. To simplify the equations the energy will be measured from the fermi level. To this end we introduce the term $\tilde{\epsilon} \equiv \epsilon_i - \mu$. This substitution will give the energy equation for a single electron as:

$$\tilde{\epsilon} = \frac{e^2}{\kappa}\left(-\sum_{k}^{don}\frac{1-n_k}{r_{ik}} + \sum_{j}^{acc}\frac{1}{r_{ij}}\right) - \mu \tag{4}$$

From this equation we can see why the Coulomb pseudogap arises. In this equation the energy of an electron will equal zero when the repulsive term between the acceptors is sufficient to cancel with the attractive terms and then chemical potential. When the energy falls to zero so must the DOS. This coulombic repulsion between the electrons is responsible for a gap in the energy. The energy minimum is effectively pinned to the chemical potential! Again finding $\{n_k\}$ and $\{\epsilon_i\}$ is incredibly difficult. Each of the variables is dependent on each of the other variables. In order to solve this the system is first minimized with respect to one arbitrary $n_k$, then with respect to two, ...etc. It will be found that this results in the fermi distribution where $n_i = 0$ for $\tilde{\epsilon}_i > 0$ and $n_i = 1$ for $\tilde{\epsilon_i} < 0$. These previous calculations have set up the distribution of our system in the ground state. We can now use this information to calculate the energy needed to begin exciting electrons the system. From this we will be able to determine the geometry of the system and the geometries energy dependence. This can then be used to determine the effect of energy on the DOS.

Now we begin the next approximation. In this we will consider the narrow band from $E = \mu - \frac{\tilde{\epsilon}}{2}$ to $E = \mu + \frac{\tilde{\epsilon}}{2}$. In this band I am concerning my self with the transfer of one 1 electron from 1 filled donor i to an empty donor j. With this we can find the energy required to transfer the electron. This can be calculated from the difference in the configuration energies. We only need to concern ourselves with the particles that we are changing because all of the other particles are constants of the action on the system. Their configuration energy is the same before and after the electron exchange. So let us step through the process of this calculation. First we remove the electron from the donor to infinity. This requires an energy $\epsilon_i < 0$. When the electron is removed we create a positive hole in the system. This will make bringing the electron back to the next state easier. So now we bring the electron into the previously unoccupied state $\epsilon_j$ In total this jump causes and energy change of:

$$\Delta_{ij} = \epsilon_j - \epsilon_i - \frac{e^2}{\kappa r_{ij}} > 0 \tag{5}$$

The energy must positive because we must increase the energy of the system to bring the system from ground level to the newly excited state. This inequality gives rise to the coulomb gap in the DOS.

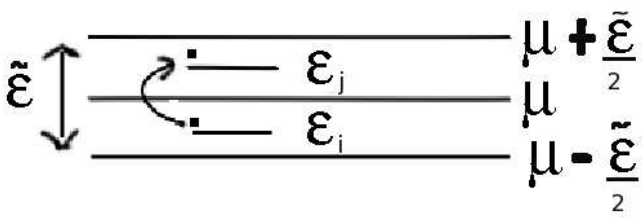



FIG. 4: This is a graphical representation of the jump across the fermi level being described.

Now we derive the minimum distance between these energy jumps. To do whis we solve the previous equation for $r_{ij}$. This result is:

$$r_{ij} > \frac{e^2}{\kappa(\epsilon_j - \epsilon_i)} \tag{6}$$

Now let us make the smallest possible jump in the system, from $E = \mu - \frac{\tilde{\epsilon}}{2}$ to $E = \mu + \frac{\tilde{\epsilon}}{2}$.This is the smallest jump because it is the exchange of only one electron. The distance from $\epsilon_i$ to $\epsilon_j$ reduces to $\tilde{\epsilon}$(see fig 4) and we are left with a final equation of:

$$r_{ij} > \frac{e^2}{\kappa\tilde{\epsilon}} \tag{7}$$

Now let us use this result to calculate the impurity density of states in 2 dimensions. ( The three dimensional case is analagous in calculation, just replace the area with the volume.) We will consider a system with impurities randomly distributed throughout the area under consideration. The impurity density is given by the equation:

$$n_{imp} = \frac{\#imp}{area} \tag{8}$$

Now as a part of the approximation we will assume that the system is homogeneous, and use that to shrink the area under consideration to include only 1 impurity. The area in this case is given by $r_{min}^2$. So substituting in our value of $r_{ij}$ we get :

$$n_{imp} = \frac{\kappa^2\tilde{\epsilon}^2}{e^4} \tag{9}$$

The density of states is given by $g(\tilde{\epsilon}) = \frac{dn(\tilde{\epsilon})}{d\tilde{\epsilon}}$, resulting in

$$\frac{2|\tilde{\epsilon}|\kappa^2}{e^4} \tag{10}$$

where $\kappa$ is an almost constant function of energy which is approximately between 4 and 5. We can now use the result of this section to compare the DOS of the Coulomb pseudogap to the Dirac Dispersion DOS.

## III. COMPARISON OF THE COULOMB DOS AND THE FERMI DIRAC DOS

The purpose of the derivation was to compare the two DOS's. From this we can determine which term will dominate any observations of the DOS that may be made. So:

$$\frac{g(\tilde{\epsilon})}{N(\tilde{\epsilon})} = \frac{\left(\frac{2\kappa^2|\tilde{\epsilon}|}{e^4}\right)}{\left(\frac{|\tilde{\epsilon}|}{\pi(\hbar v_f)^2}\right)} \tag{11}$$

Recall that $\frac{e^2}{\hbar c} = \frac{1}{137}$ and note that $\frac{e^2}{\hbar \frac{c}{300}} = \frac{300}{137}$. Substitute this in to get:

$$\left(\frac{137}{300}\right)^2 2\pi\kappa^2 \cong 20.965 - 32.758 \tag{12}$$

Because of such a difference in the amplitude we expect the the Coulomb Pseudogap might be observed if it is indeed in graphene.

## IV. CONCLUSION

This result shows that the coulombic forces dominate in the situation where the screening potential is low, i.e. in the case of suspended graphene. This tells us that the coulombic affects will dominate any DOS measurements made. This opens up a way to measure the DOS as a function of $\kappa$ which is a function of the energy. Using STM we may get a direct measure of the DOS.

The question arises of how to actually measure this affect. There are numerous experimental difficulties that must be overcome. The first difficulty is that the Coulomb dispersion and the Dirac dispersion are caused by two fundamentally different phenomenon. The Dirac dispersion occurs at an energy level such that $\mu$, the fermi level, is zero. it occurs at this point even when there are impurities which would shift the fermi level. The problem arises in that the Coulombic dispersions, which occur from impurities, are centered around the non-zero fermi level. This can be compensated for by adjusting the bias voltage to re-center the fermi level back to zero. This will allow us to have everything on the Dirac dispersion. The next difficulty arises because we are trying to measure two linear functions. It becomes very difficult to differentiate between the two. Because the Coulomb pseudogap is strongly dependent on the dielectric constant and electron screening we have a way to compensate for this problems. The first method is changing the substrate. This will alter the dielectric constant and cause a variation in the slope of the Coulombic DOS. The other method is to actually change the bias voltage. This will shift the Coulomb pseudogap's position and create strange distortions in the DOS. From these it should be possible to actually observe the Coulomb pseudogap.

## V. ACKNOWLEDGMENTS

This paper was supported by the Los Alamos Summer School, The University of New Mexico, . I would also like to greatly thank Alexander V. Balatsky and Hari P. Dahal for all of the help and guidance that they have given to me through this summer in creation of this paper. If I did not have their help then this paper would not have happened.

[1] G. M. Scalise (2008).
[2] P. R. Wallace, Phys. Rev. **71**, 622 (1947).
[3] A. H. C. Neto, F. Guinea, N. M. R. Peres, K. S. Novoselov, and A. K. Geim, *The electronic properties of graphene* (2007), URL `http://www.citebase.org/abstract?id=oai:arXiv.org:0709.1163`.
[4] S. Y. Zhou, G. H. Gweon, J. Graf, A. V. Fedorov, C. D. Spataru, R. D. Diehl, Y. Kopelevich, D. H. Lee, S. G. Louie, and A. Lanzara, NATURE PHYS. **2**, 595 (2006), URL `doi:10.1038/nphys393`.
[5] E. McCann and V. I. Fal'ko, Physical Review Letters **96**, 086805 (pages 4) (2006), URL `http://link.aps.org/abstract/PRL/v96/e086805`.
[6] K. S. Novoselov, A. K. Geim, S. V. Morozov, D. Jiang, Y. Zhang, S. V. Dubonos, I. V. Grigorieva, and A. A. Firsov, Science **306**, 666 (2004).
[7] T. O. Wehling, A. V. Balatsky, M. I. Katsnelson, A. I. Lichtenstein, K. Scharnberg, and R. Wiesendanger, Physical Review B (Condensed Matter and Materials Physics) **75**, 125425 (pages 5) (2007), URL `http://link.aps.org/abstract/PRB/v75/e125425`.
[8] J. Nilsson, A. H. C. Neto, F. Guinea, and N. M. R. Peres, Physical Review Letters **97**, 266801 (pages 4) (2006), URL `http://link.aps.org/abstract/PRL/v97/e266801`.
[9] E. McCann, Physical Review B (Condensed Matter and Materials Physics) **74**, 161403 (pages 4) (2006), URL `http://link.aps.org/abstract/PRB/v74/e161403`.
[10] A. V. Balatsky (2008), notes.
[11] J. Nilsson, A. H. C. Neto, N. M. R. Peres, and F. Guinea, Physical Review B (Condensed Matter and Materials Physics) **73**, 214418 (pages 10) (2006), URL `http://link.aps.org/abstract/PRB/v73/e214418`.
[12] A. Akturk and N. Goldsman, Journal of Applied Physics **103**, 053702 (pages 8) (2008), URL `http://link.aip.org/link/?JAP/103/053702/1`.
[13] J. H. Chen, C. Jang, S. Xiao, M. Ishigami, and M. S. Fuhrer, NATURE NANOTECHNOLOGY **3**, 206 (2008), URL `doi:10.1038/nnano.2008.58`.
[14] K. S. Novoselov and et al., Science **306**, 666 (2004).
[15] M. I. Katsnelson, K. S. Novoselov, and A. K.S. Geim, Nature Phys. **2**, 620 (2006).
[16] M. Y. Han, B. Özylimaz, Y. Zhang, and P. Kim, Phys. Rev. Lett. **98** (2007).
[17] J. Nilsson and A. Castro Neto, APS Meeting Abstracts pp. 28010–+ (2007).

[18] C. Kittel, *Introduction to Solid State Physics* (John Wiley and Sons, Inc, 1996), 7th ed.

[19] B. Shklovskii and A. Efros, *Electronic Properties of Doped Semiconductors*, no. 45 in Springer Series in Solid-State Sciences (Springer-Verlag, 1984).

[20] J. S. Bunch, A. M. van der Zande, S. S. Verbridge, I. W. Frank, D. M. Tanenbaum, J. M. Parpia, H. G. Craighead, and P. L. McEuen, Science **315**, 490 (2007), http://www.sciencemag.org/cgi/reprint/315/5811/490.pdf, URL `http://www.sciencemag.org/cgi/content/abstract/315/5811/490`.

# Probing the Quark Gluon Plasma at the Large Hadron Collider at CERN

Katelyn Hessler
Lehigh University, Bethlehem, PA 18015

August 12, 2008

## Abstract

Physicists have been getting closer and closer to figuring out what happened during and after the Big Bang. Currently, we know that within $10^{-20}$ seconds after the Big Bang there existed a state of matter called a quark-gluon plasma with a temperature of trillions of degrees. The extreme temperature and density of this matter allowed quarks and gluons to act independently of each other; something which does not occur in normal matter, as the strong force keeps quarks and gluons confined within hadrons. However, we can now recreate this quark-gluon plasma in the laboratory using heavy ion colliders, such as the Large Hadron Collider at CERN. We can probe the quark gluon plasma by searching for events that contain a $Z^0$ boson and a jet of other particles. We know that the $Z^0$ does not interact via the color charge, but that the particles in the jet do interact via the color charge. We can then use the $Z^0$ as a tagging particle and to figure out how the jet has been changed by the interactions in the quark-gluon plasma. We will then compare the properties of the these modified jets to those of jets in proton-proton collisions. We can use any differences that we find to learn about the properties of the quark-gluon plasma. We will studying this by running Monte Carlo simulations of such events and looking at the angular and rapidity distributions of the $Z^0$ and the jet.

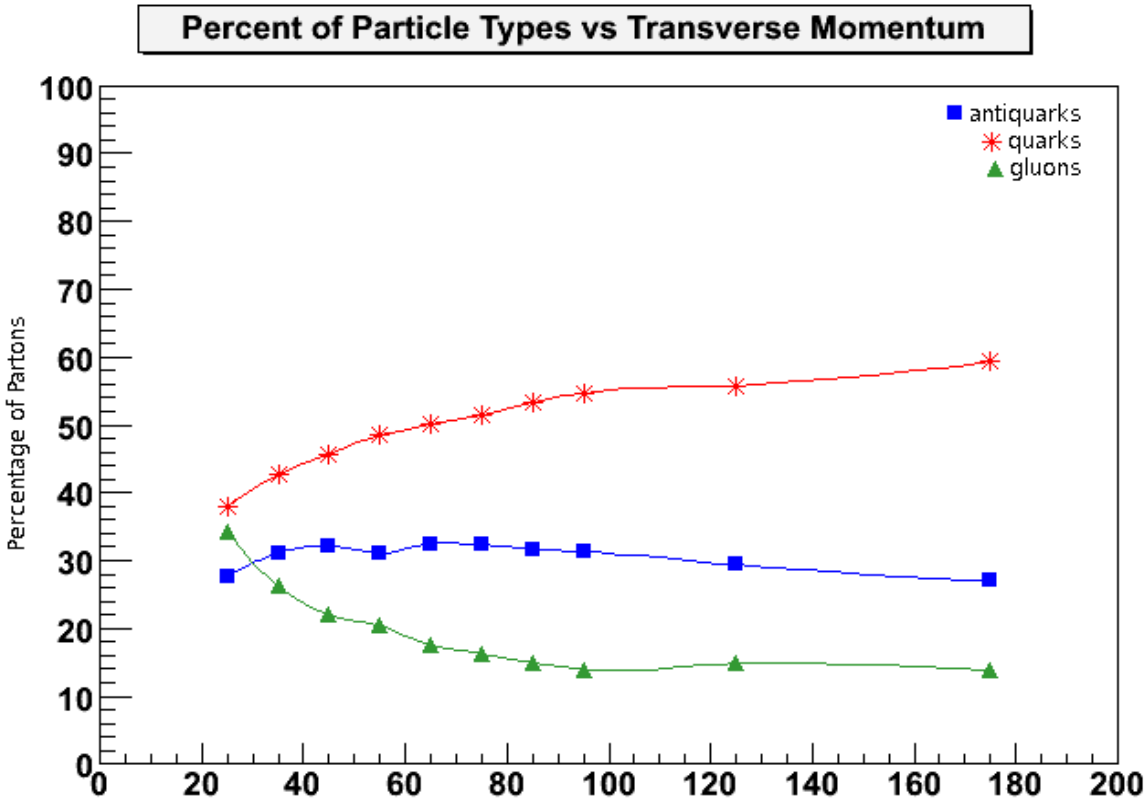
# 1 Introduction

In the current universe, conditions are such that all quarks and gluons are confined into hadrons. Such particles cannot be isolated, as they are confined by the strong force, which becomes stronger as distance increases between two particles. However, physicists have been able to recreate a state of matter in the Relativistic Heavy Ion Collider (RHIC) at Brookhaven called a quark gluon plasma(QGP) by colliding two gold nuclei. They believe that this was the state in which the universe existed $10^{-20}$ seconds after the Big Bang. This "plasma" consists of quarks and gluons which are not confined to color-neutral hadrons, which creates a strong color field. This is not to be confused with a traditional plasma, which consists of electrons which have been stripped from nuclei to create a strong electromagnetic field, though the two have some analogous features. Now that we know that it is possible to recreate this QGP within particle accelerators, we would like to know more about it's properties. We plan to study the QGP by looking at proton-proton collisions within heavy ion collisions at the Large Hadron Collider (LHC) at CERN in Switzerland. We will look at one possible result of the proton-proton collision—a color-neutral $Z^0$ boson and a jet of color-charged particles—inside of the QGP created by the heavy ion collision and use the analysis of such events to determine properties of the QGP.

# 2 Methods and Tools

This summer, we ran simulations under a software package called PYTHIA to simulate thousands of proton-proton collisions inside of the QGP that would give us the $Z^0$ and jet result. PYTHIA is a program that can generate simulate high-energy physics events such as the proton-proton collisions we are interested in. PYTHIA contains models and theory for such many high-energy physics events. However, our team is only interested currently in the simulations of proton-proton collisions with the aforementioned result. We can specify this within PYTHIA, which allows us to only look at these interactions, instead of having to sort through a multitude of different interactions that are not particularly relevant to what we are currently trying to study.

PYTHIA also allows us to specify the number of events we would like it to run. After running this number of events, it generates an output file in the form of a .root file. This has all of the information about the energy and direction of the particles resulting from the collision in the form of what is called an ntuple by ROOT users. ROOT is a physics workstation program that allows scientists to read in data from this ntuple form and manipulate the data into plots, histograms and other graphs that allow us to visually analyze the data given to us by PYTHIA. ROOT can be used for many purposes, but our main focus at the moment is to use it to analyze the PYTHIA simulation data.


Percent of Particle Types vs Transverse Momentum
Percentage of Partons
antiquarks
quarks
gluons
0
10
20
30
40
50
60
70
80
90
100
0
20
40
60
80
100
120
140
160
180
200

# Plasma Temperature and Density Dependence for Collisional Cross Section Calculations

Jakob Kotas
*Cornell University, Ithaca, NY 14853*

James Colgan
*T Division, Los Alamos National Laboratory, Los Alamos, NM 87545*

August 15, 2008

## Abstract

In modeling dense plasmas, the probability of an electron-ion or photon-ion collision is calculated and expressed using a cross section. For most applications, each collisional process is treated as independent of the ambient conditions of the plasma medium. Our research aims at introducing plasma temperature and density dependence into the cross section calculations to give a more accurate depiction of the occurrent collisions. At high densities, the presence of other ions nearby effectively partially screens the target ion as seen by the incoming particle. The LANL atomic collision codes GIPPER and ACE have been modified to include the static Debye-Hückel screening factor (which is a function of both temperature and density.) Two types of collisions have been considered to date: electron-impact ionization and electron-impact excitation. Results will be discussed and comparisons made with previous work where possible.

## I. Introduction

Plasmas can occur in a large variety of temperatures and densities– from the cold, sparse plasma of the Earth's ionosphere to the relativistic plasmas of pulsar magnetospheres.[1] The focus of our research has been on hot, dense (high-energy) plasmas. Interest in high-energy plasmas began in an attempt to understand the interiors of stars; now, these plasmas are studied for their use in inertial confinement fusion (ICF), short wavelength lasers, and short pulse X-ray sources.[2] An important distinction between the plasmas in cosmic bodies and those in the laboratory is that the interiors of stars (apart from convection zones) can be approximated as being in local thermodynamic equilibrium (LTE), which are more straightforward to model. Laboratory plasmas are too short-lived to achieve this state, and models must take into account all specific atomic processes that can occur.[3] There are five important processes that determine ionization balance: collisional excitation, collisional ionization, autoionization, photoexcitation, and photoionization. Reaction formulae for the processes are given in Table 1. Our research has dealt with collisional excitation and collisional ionization.

[1] Murillo and Weisheit, 4-5.
[2] *Ibid.*, 5.
[3] *Ibid.*, 6.

| Table 1. The main atomic processes that influence ionization balance in plasmas. A represents an ion, $e^-$ an electron, $\gamma$ a photon.[4] | |
|---|---|
| Collisional excitation | $e^- + A_{il} \rightarrow e^- + A_{im}$ |
| Collisional ionization | $e^- + A_{il} \rightarrow 2e^- + A_{jm}$ |
| Autoionization | $A_{il} \rightarrow e^- + A_{jm}$ |
| Photoexcitation | $\gamma + A_{il} \rightarrow A_{im}$ |
| Photoionization | $\gamma + A_{il} \rightarrow e^- + A_{jm}$ |

The probability that a particular atomic process will occur is given using a cross section, which has units of area. The cross section is defined as the probability of observing a scattered particle in some quantum state per unit solid angle if the target has a flux of one incident particle per surface unit. Classically the idea of a "cone of observation" is used, with the target particle taking up some finite area within the cone. The larger the cross section, the more likely the event is to occur.

$$\frac{d\sigma}{d\Omega} = \frac{Scattered\ flux\ per\ unit\ solid\ angle}{Incident\ flux\ per\ unit\ surface\ area} \tag{1}$$

$$\sigma = \int d\Omega \frac{d\sigma}{d\Omega} \tag{2}$$

For our calculations, a partial wave approximation is used. The wavefunction for each free electron is assumed to be:

$$\varphi(\vec{K}, \vec{r}) = \sum_{l,m} a_{lm} Y_{lm}(\theta, \phi) F_l(K, r)\,, \tag{3}$$

where $l$ and $m$ are the orbital and azimuthal quantum numbers, and $Y$ and $F$ are the angular and radial solutions to the Schrödinger equation. The radial Schrödinger equation is given by:

$$\left[-\frac{1}{2}\frac{d^2}{dr^2} + \frac{l(l+1)}{2r^2} + V(r)\right] F(r) = \frac{K^2}{2} F(r)\,, \tag{4}$$

where $V(r)$ is the potential and $K$ is the momentum of the free electron.

There are several common choices for $V$. Perhaps the simplest is the Plane Wave Born (PWB) Approximation, which takes $V(r) = 0$. In this approach, the effect of the target ion on the incident electron is ignored. This approximation is not bad for high energies and hydrogenic wavefunctions, but for our purposes PWB does not suffice.

[4] Abdallah and Clark.

The Coulomb Approximation improves on PWB by taking into account the effect of the nuclear charge on the incident electron. In the Coulomb Approximation,

$$V(r)=\frac{-Z}{r} \quad , \tag{5}$$

where $Z$ is the effective nuclear charge. The Distorted Wave Approximation goes further by including a term for the effect of the target ion electrons as well:

$$V(r)=-\frac{Z}{r}+V_{DW}(r) \, , \tag{6}$$

where the $V_{DW}(r)$ term approximates the potential due to target electrons.[5] We used the DW approximation for our calculations.

## II. Method

Most models that consider excitation and ionization do not take into account the effect of plasma temperature and density on collisional cross sections. For a cool, sparse plasma, ignoring temperature and density dependence is a reasonable approximation, as ions are far enough apart that they do not feel one another significantly. However, in a hot, dense plasma, nearby ions have a non-negligible contribution to each other's potential, and the target ion is effectively partially screened as seen by the incident electron. We account for this effect by using the Debye-Hückel screening radius:

$$\Lambda=\sqrt{\frac{T_e}{4\pi N_e}} \, , \tag{7}$$

where $T_e$ is the temperature and $N_e$ is the electron density of the plasma. The screening radius appears in a screening exponential decay factor of the form:

$$e^{\frac{-r}{\Lambda}} \tag{8}$$

We took:

$$V_{DW}(r)=V_{HXR}^{N-1}(r) \tag{9}$$

for both our direct and exchange DW potentials, where $V_{HXR}^{N-1}(r)$ is an ($N$ – 1) electron Hartree with local exchange potential.[6]

---

[5] Märk and Dunn, 4.

[6] Cowan, 198.

Our calculation could be refined further by using the Close-Coupling (CC) calculation, which takes into account more quantum mechanical effects. However, considering the much greater computing time needed for CC, we did not use it in our calculations.

In practice, the Debye-Hückel screening factor (henceforth DHSF) must be included in the radial matrix element:

$$R^{\lambda}(k_e l_e, k_f l_f, nl, k_i l_i) = \int_0^{\infty} dr_1 \int_0^{\infty} dr_2 \frac{r_{lower}^{\lambda}}{r_{higher}^{\lambda+1}} e^{\frac{-r_2}{\Lambda}} P_{k_e l_e}(r_1) P_{k_f l_f}(r_2) P_{nl}(r_1) P_{k_i l_i}(r_2) \text{ , (10)}$$

where $r_{lower}$ and $r_{higher}$ are min($r_1$, $r_2$) and max($r_1$, $r_2$), respectively; as well as in the potential terms for the Schrödinger equation for the incident and scattered (but not ejected) electrons:[7]

$$\left[ -\frac{1}{2}\frac{d^2}{dr^2} + \frac{l(l+1)}{2r^2} - \frac{Z}{r} + V_{HXR}^{N-1}(r) \right] P_{kl}(r) = \frac{K^2}{2} P_{kl}(r) \text{ (ejected)} \quad (11)$$

$$\left[ -\frac{1}{2}\frac{d^2}{dr^2} + \frac{l(l+1)}{2r^2} - \frac{Z}{r} e^{\frac{-r}{\Lambda}} + V_{HXR}^{N-1}(r) e^{\frac{-r}{\Lambda}} \right] P_{kl}(r) = \frac{K^2}{2} P_{kl}(r) \text{ (inc. \& scat.)} \quad (12)$$

The reason we assume the potential of the ejected electron to be unscreened is that we assume the plasma is dense enough to influence the electron-ion interaction, but not enough to affect electrons on the target ion. The radial matrix element for the excitation calculation is similar to the RME for ionization given in equation (10).

In order to implement these changes, existing LANL atomic collision codes were altered. GIPPER, which calculates collisional ionization cross sections, was modified first. Then ACE, which calculates collisional excitation cross sections, was altered.

## III. Results

The first benchmark of our results was comparison with a paper by M.S. Pindzola, et al. In that paper, cross-section data is given for electron-impact ionization of $Ne^{4+}$ ($Ne^{4+}$ → $Ne^{5+} + e^-$) and $Au^{47+}$ ($Au^{47+}$ → $Au^{48+} + e^-$). Our results agreed with theirs. For ionization of $Ne^{4+}$, we had no more than a 5% error. For ionization of $Au^{47+}$, we had a 5 to 10% error. The shapes of the curves matched very well, and peak cross sections occurred at approximately 320 eV and 5800 eV for $Ne^{4+}$ and $Au^{47+}$, respectively. Pindzola uses a DW method which is similar to ours. The 5-10% difference in our results could be explained by the fact that Pindzola uses $V_{DW}(r) = V_{HXR}^{N}(r)$ for the incident and scattered potentials, whereas we use $V_{DW}(r) = V_{HXR}^{N-1}(r)$ for all three electrons. Note that all graphs referenced from other papers can be found in the appendix.

[7] Pindzola, et al.

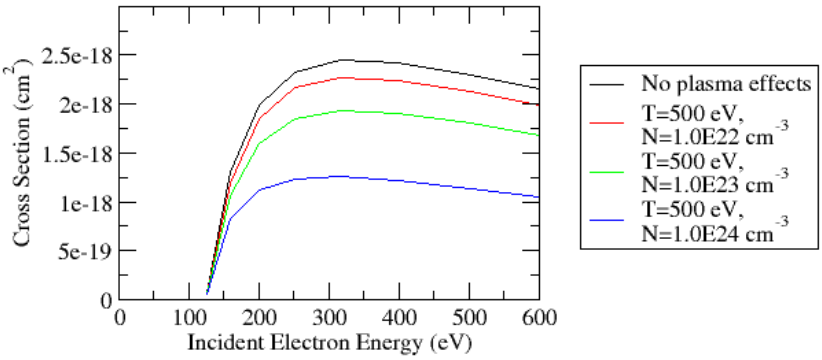



Figure 1. $Ne^{4+}$ ionization, cf. Fig. 9.

From our graphs, we see that the cross section goes to zero at the ionization potential, which occurs at 125.07 eV for $Ne^{4+}$ and 2551.7 eV for $Au^{47+}$. The cross section reaches a peak and then has a long tail which extends toward infinity. We also notice that when plasma temperature and density dependence is included, the cross section always decreases from the "baseline" (no plasma effects) curve. Lastly, as density increases, the cross section decreases, and this effect is most noticeable for smaller ions.

Next we did a run of $Au^{47+}$ ionization again, this time with the temperature and density parameters used in the paper by Wu, et al. Not surprisingly, our graph had a similar shape to the run of $Au^{47+}$ ionization with the Pindzola parameters.

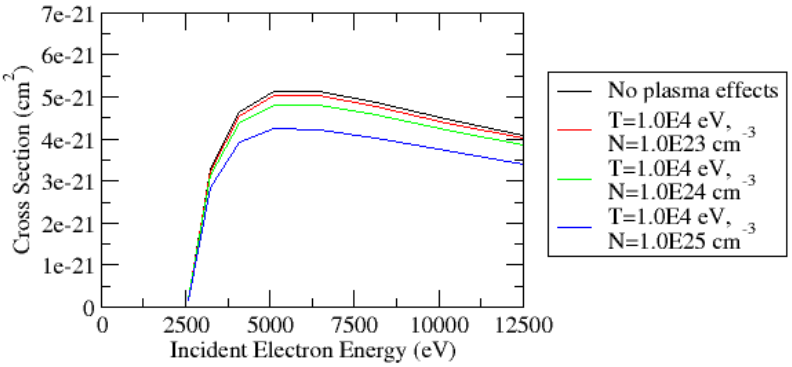



Figure 2. $Au^{47+}$ ionization, cf. Fig. 10.

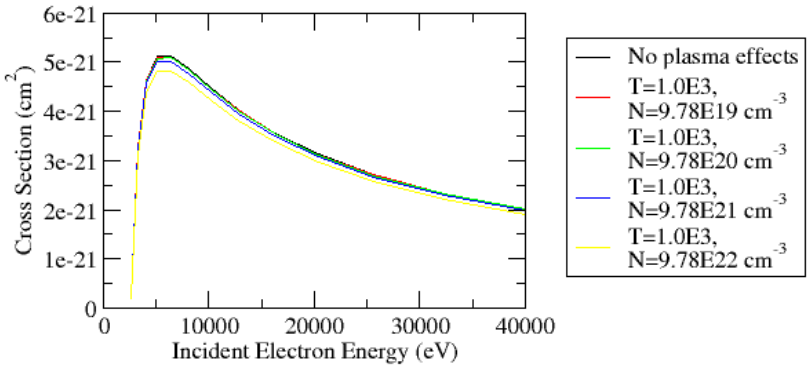



Figure 3. $Au^{47+}$ ionization, cf. Fig. 11.

The Wu paper, however, gave a significantly different set of results. First, our baseline calculation was off from theirs by a factor of two. Second, we show that increasing density decreases the cross section for all incident electron energies, whereas Wu, et al. have the opposite trend. Both of these discrepancies were noted by Pindzola. Pindzola noticed that by turning off the DHSF in the radial matrix element and turning on the DHSF in the radial Schrödinger equation for the ejected electron, such a trend could be reproduced. Even so, however, the temperature and density effects were much smaller than seen by Wu. We altered the code to be able to turn off each factor independently, and tried a run using the parameters in Pindzola's explanation. Our results were the same as those described by Pindzola.

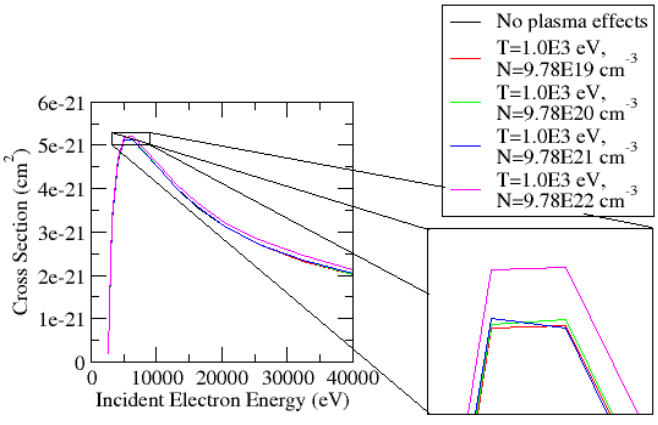



Figure 4. $Au^{47+}$ ionization with DHSF turned off in radial matrix element and on in radial Schrödinger equation, cf. Figs. 10, 11.

For our final test of the modified GIPPER program, we did a run of H ionization, which was carried out in a paper by Bornath, et al. Coming in to this run, we did not expect our results to match theirs very closely. One notable difference between the two calculations is that Bornath includes plasma effects in the wavefunctions for both the bound and free orbitals, whereas we only alter the free orbitals. Also, Bornath is looking at H(2s) ionization, not ground state ionization. On a more fundamental level, we have approached the situation from an atomic physics standpoint whereas Bornath is starting with plasma physics. Essentially, we are solving the Schrödinger equation first and then adding plasma effects. Bornath starts with plasma and is not clear on the details of the actual atomic scattering processes– for example, it appears Bornath uses plane waves for free electrons, but the paper does not explicitly state so. Nevertheless, we felt it would be informative to try the run anyway.

Our results did differ from Bornath's significantly. Our baseline calculation matched well; however, we show decreased cross section with increasing density, whereas Bornath reports the opposite. Nevertheless, we do not discount our data for the reasons mentioned above.

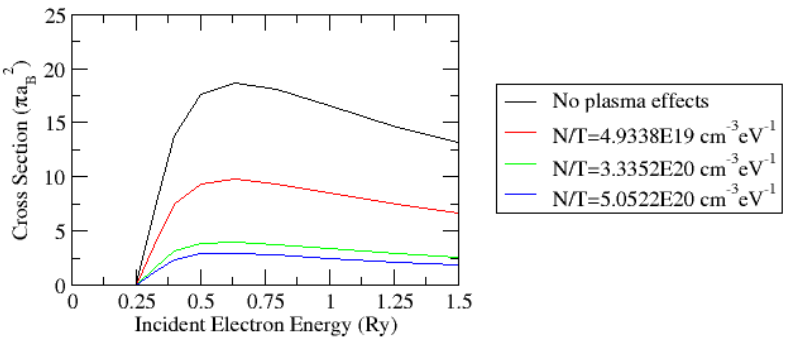



Figure 5. H ionization, cf. Fig. 12. (Ry = 13.6 eV; $a_B$ = 5.292E-9 cm)

To begin our analysis of our modified ACE code, we did a run of $Ne^{9+}$ excitation, which was studied in a paper by Whitten, Lane, and Weisheit (with 1s → 2s, 1s → 2p, and 2s → 2p transitions) and Pindzola, et al. (with the 1s → 2p transition only.) Once again, our results are quite similar to Pindzola's. For the 1s → 2s and 1s → 2p transitions of the Whitten run, our graphs match reasonably well, with approximately 15% error on both curves (with and without plasma effects.) More importantly, our trends are identical, showing that plasma effects decrease the cross section over all incident electron energies. The 2s → 2p transition is farther off; however, considering the very low excitation energy of the transition (0.3 eV compared to 1022 eV for the other two transitions,) and the fact that plane wave convergence for the 2s → 2p transition is slow, there is more room for discrepancy.

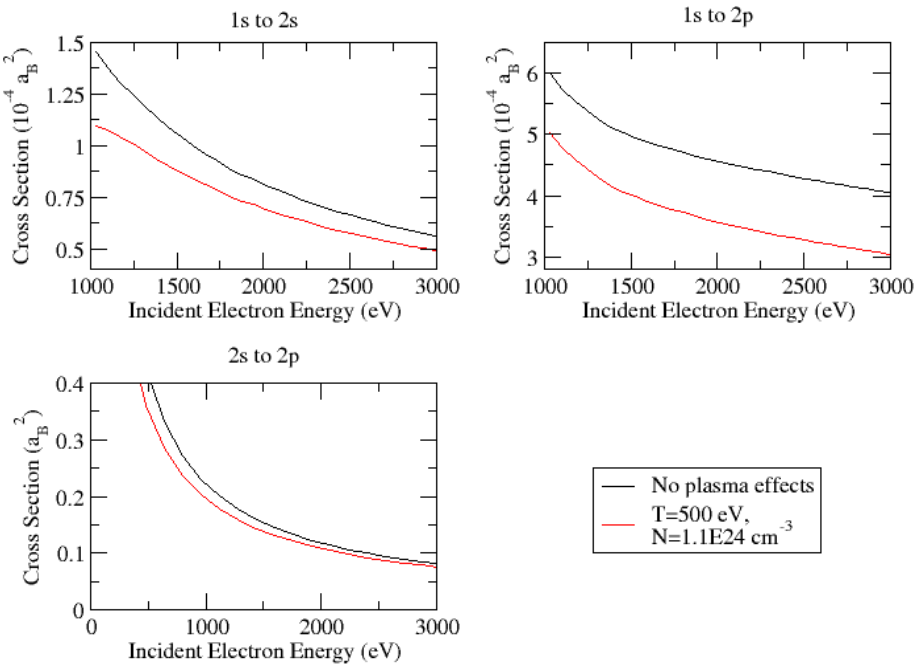



Figure 6. $Ne^{9+}$ excitation, cf. Figs. 13, 14.

Finally we did a run of $He^+$ excitation, which was also done by Whitten. For this run, we expected at best mediocre agreement again, as Whitten used the CC calculation mentioned earlier, while we did not.

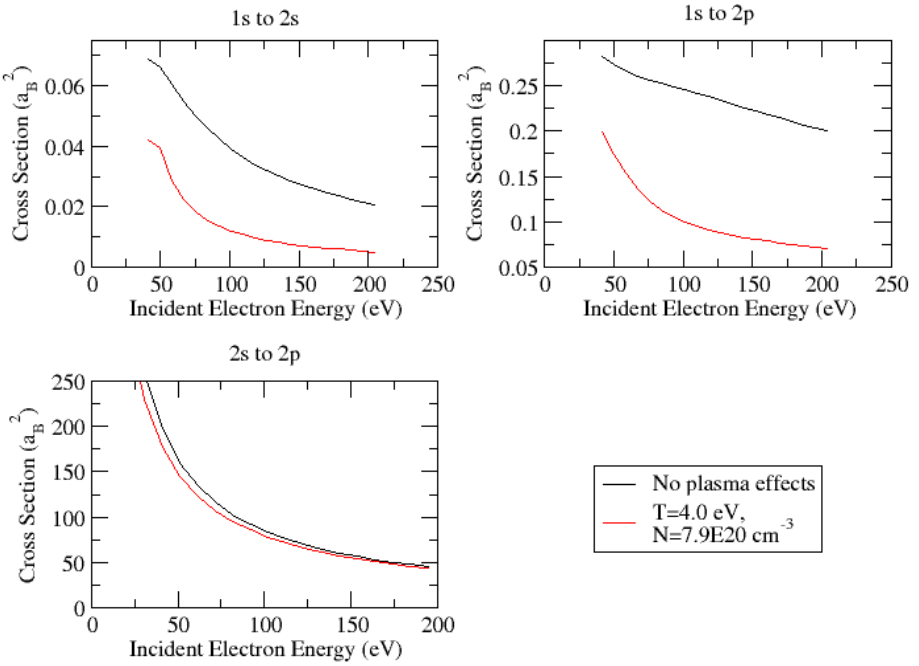



Figure 7. $He^+$ excitation, cf. Fig. 15.

Our results did not match terribly well numerically, but again, the trends were similar. As was the case with $Ne^{9+}$, our results for the 1s → 2s and 1s → 2p transitions agreed better than the 2s → 2p transition, for the same reason as before. Notably, we did notice that our results matched theirs more closely for higher incident electron energy. We attribute this observation to the decreased importance of quantum effects at higher energies.

Once GIPPER and ACE were successfully modified, an effort was made to change the way input files were read into the two programs. Originally, each run of GIPPER or ACE could only be done for a specific temperature and density. Our goal was to be able to put in a range of temperatures and densities, and GIPPER or ACE would iterate over an entire grid of values to return one complete output file. We could then input this file into ATOMIC, another LANL program which calculates rate equations and spectra. While it is our eventual goal to be able to run multiple temperatures and densities at once, we put this task on hiatus due to time constraints; instead, we chose to focus on finishing a full ATOMIC run.

ATOMIC uses the output files from both GIPPER and ACE, as well as CATS, another LANL atomic collisional physics program which we did not alter. We were interested in seeing how the modified ionization and excitation cross sections would affect a plot of radiated power loss (henceforth RPL), which is calculated by ATOMIC. We did calculations for a Li plasma so that we could compare with the results of Loch, et al.

First, we did the baseline calculation at a density of $10^{14}$ $cm^{-3}$ and a temperature range of $0.25\ eV \leq T \leq 100\ eV$, the same parameters used in the Loch paper. Then we ran GIPPER and ACE and fed the results into ATOMIC. Our output was nearly identical to the baseline curve, meaning that the altered cross sections had virtually no effect on RPL. However, considering that $10^{14}$ $cm^{-3}$ is a relatively low density, the DHSF has little effect and we would not expect the RPL to stray very far from the baseline.

In an attempt to obtain results that differed significantly from the baseline, we increased the density to $10^{16}$ $cm^{-3}$, $10^{18}$ $cm^{-3}$, and $10^{20}$ $cm^{-3}$. Disappointingly, we still saw practically no difference from the baseline. Runs with a density of $10^{24}$ $cm^{-3}$ produced nonsensical results for our ionization and excitation cross sections, leading us to believe that at such extreme densities, the static screening approximation breaks down and the DHSF can no longer accurately model the cross section. See Bornath, et al. for a more thorough discussion on the limitations of screening factors.

At $10^{22}$ $cm^{-3}$, we observed an increase in RPL with temperature and density effects taken into account on the GIPPER and ACE runs. This increase ranges from essentially zero at a temperature of 0.25 eV to around 60% at a temperature of 25.0 eV. However, these results must be taken with a grain of salt as the baseline and temperature/density-dependent cases were calculated with slightly different configurations. In the future we would like to do a run with identical configurations to be more certain of this trend.

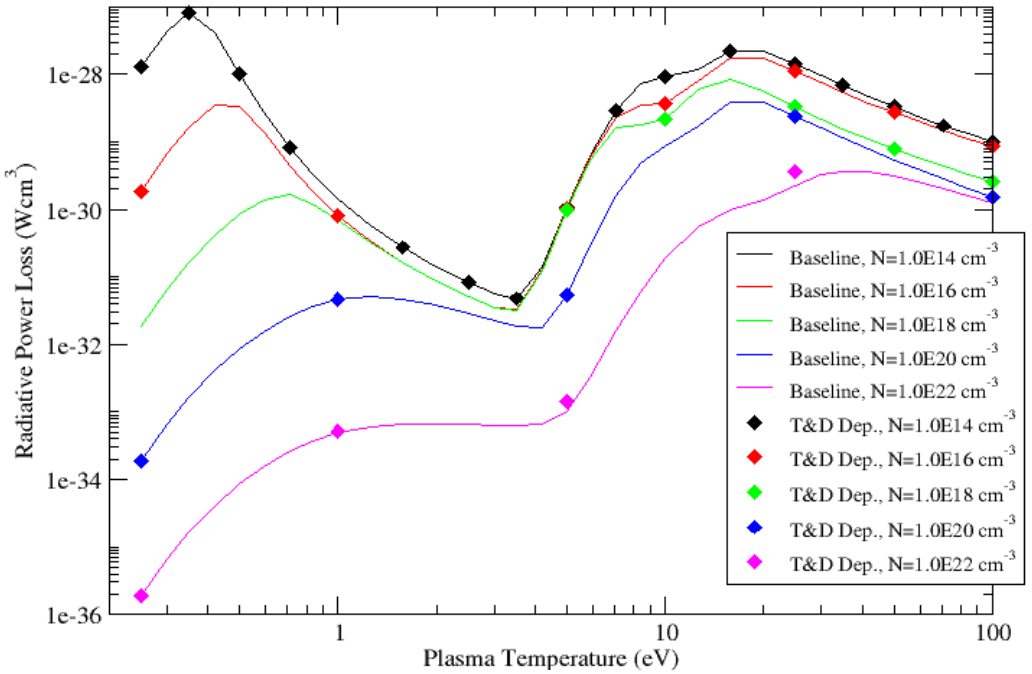



Figure 8. Radiated Power Loss for Li plasma, cf. Fig. 16.

# IV. Conclusions

We have successfully integrated temperature and density effects into the LANL atomic collisional codes GIPPER and ACE. While published data can vary widely in the assumptions made and calculations used, we have come to very good agreement with the collisional excitation and collisional ionization results given in Pindzola, et al. We have had reasonable agreement with some other papers using similar methods, such as the $Ne^{9+}$ results in Whitten, Lane, and Weisheit. Finally, we have had poor agreement with some other papers, such as Bornath, et al. However, considering the discrepancy in calculation methods, agreement would not be expected and does not necessarily discount our results. Furthermore, for those papers with which we agreed poorly, we feel we have solid arguments as to why our results differ.

Using the updated GIPPER and ACE codes, it is now possible to retrieve cross sections for collisional excitation and collisional ionization for any ion at any temperature and density (within the hot, dense plasma assumption.) In future research, ATOMIC can be used to calculate rate equations and spectra for these plasmas.

## V. Acknowledgments

I would like to thank James Colgan both for being my mentor and working behind the scenes of the Summer School. Without his patience, dedication, and remarkable knowledge of physics, I would not have been able to do research this summer.

Additionally, I would like to thank Chris Fontes for giving me suggestions on programming; Norm Magee for taking care of the logistics of the Summer School; and Sally Seidel for being our UNM liaison.

I thank UNM, NSF, and LANL for funding our research.

Finally, thank you to the other students in LASS 2008. You made my summer a memorable one.

## VI. References

Abdallah, J. and Clark, R.E.H. (1990) *Theoretical Atomic Physics Code Development V: Rate Equations for Plasmas in Non Local Thermodynamic Equilibrium.* Los Alamos National Laboratory.

Bornath, Th., et al. (1997) Dynamical Screening Effects on Rate Coefficients for Dense Plasmas. *Journal of Quantitative Spectroscopy and Radiative Transfer.* v. 58, n. 4-6, pp. 501-8.

Cowan, R.D. (1981) *The Theory of Atomic Structure and Spectra.* Berkeley: University of California Press.

Iglesias, C.A. and Lee, R.W. (1997) Density Effects on Collisional Rates and Population Kinetics. *Journal of Quantitative Spectroscopy and Radiative Transfer.* v. 58, n. 4-6, pp. 637-43.

Loch, S.D., et al. (2004) Collisional-Radiative Study of Lithium Plasmas. *Physical Review E.* v. 69, n. 066405.

Märk, T.D. and Dunn, G.H. (1985) *Electron Impact Ionization.* Vienna: Springer-Verlag.

Murillo, M.S. and Weisheit, J.C. (1997) Dense Plasmas, Screened Interactions, and Atomic Ionization. *Physics Reports.* v. 302, n.1-65.

Pindzola, M.S., et al. (2008) Electron-Impact Ionization of Atoms in High-Temperature Dense Plasmas. *Physical Review A.* v. 77, n. 062707.

Whitten, B.L., Lane, N.F., and Weisheit, J.C. (1984) Plasma-Screening Effects on Electron-Impact Excitation of Hydrogenic Ions in Dense Plasmas. *Physical Review A.* v. 29, n. 2, pp. 945-52.

Wu, Z.Q., et al. (2002) Plasma Effects on Electron Impact Ionization. *Journal of Physics B.* v. 35, pp. 2305-11.

# VII. Appendix

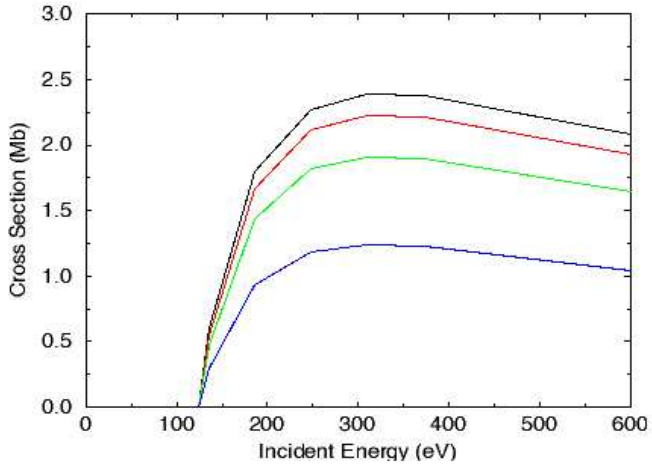



Figure 9. $Ne^{4+}$ ionization. Temperature and density for each curve correspond to those given in Fig. 1. (Mb = $10^{-18}$ $cm^2$) (Pindzola, et al., 3.)

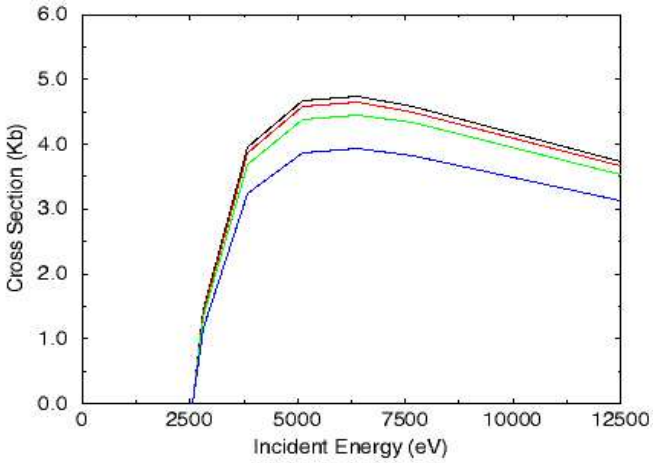



Figure 10. $Au^{47+}$ ionization. Temperature and density for each curve correspond to those given in Fig. 2. (Kb = $10^{-21}$ $cm^2$) (Pindzola, et al., 3.)

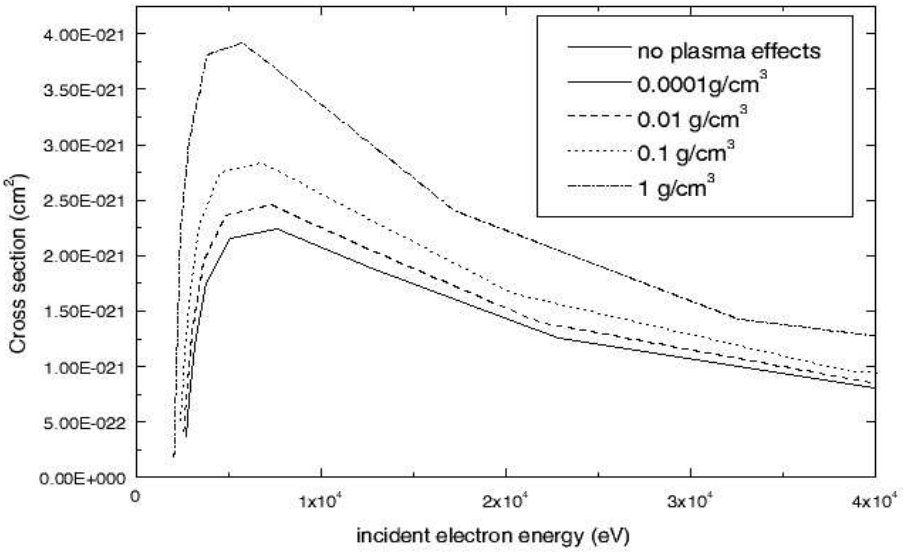



Figure 11. $Au^{47+}$ ionization. Temperature and density for each curve correspond to those given in Fig. 3. (Wu, et al., 2309.)

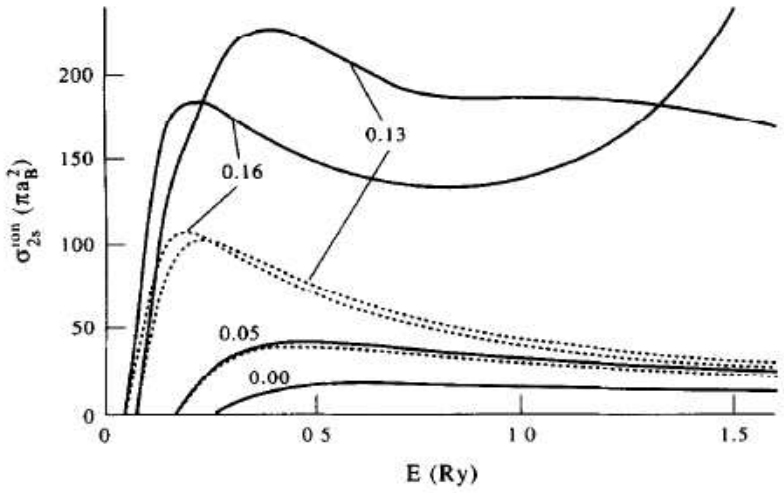



Figure 12. H ionization. Temperature and density for dotted-line curves correspond to those given in Fig. 5. Solid lines use a dynamic (not static) approximation for the scattering potential. (Bornath, et al., 505.)

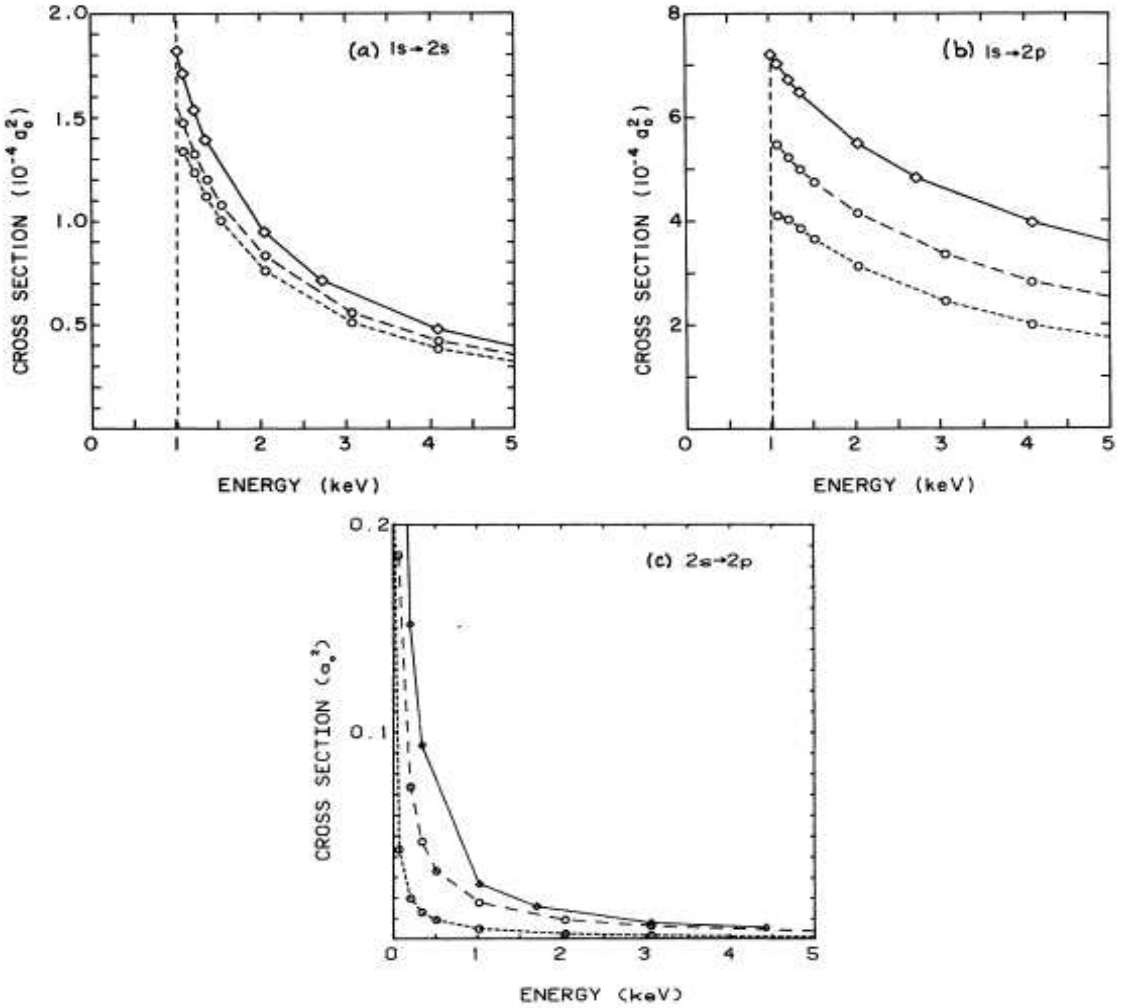



Figure 13. $Ne^{9+}$ excitation. Temperature and density for dashed lines correspond to those given in Fig. 6. Dotted lines represent an ion sphere calculation (as opposed to DHSF.) (Whitten, Lane, and Weisheit, 949.)

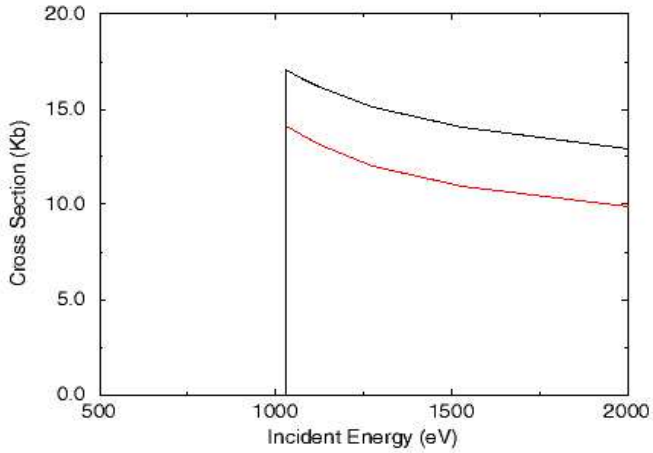



Figure 14. $Ne^{9+}$ excitation, 1s → 2p. Temperature and density correspond to those given in Fig. 6. (Pindzola, et al., 2.)

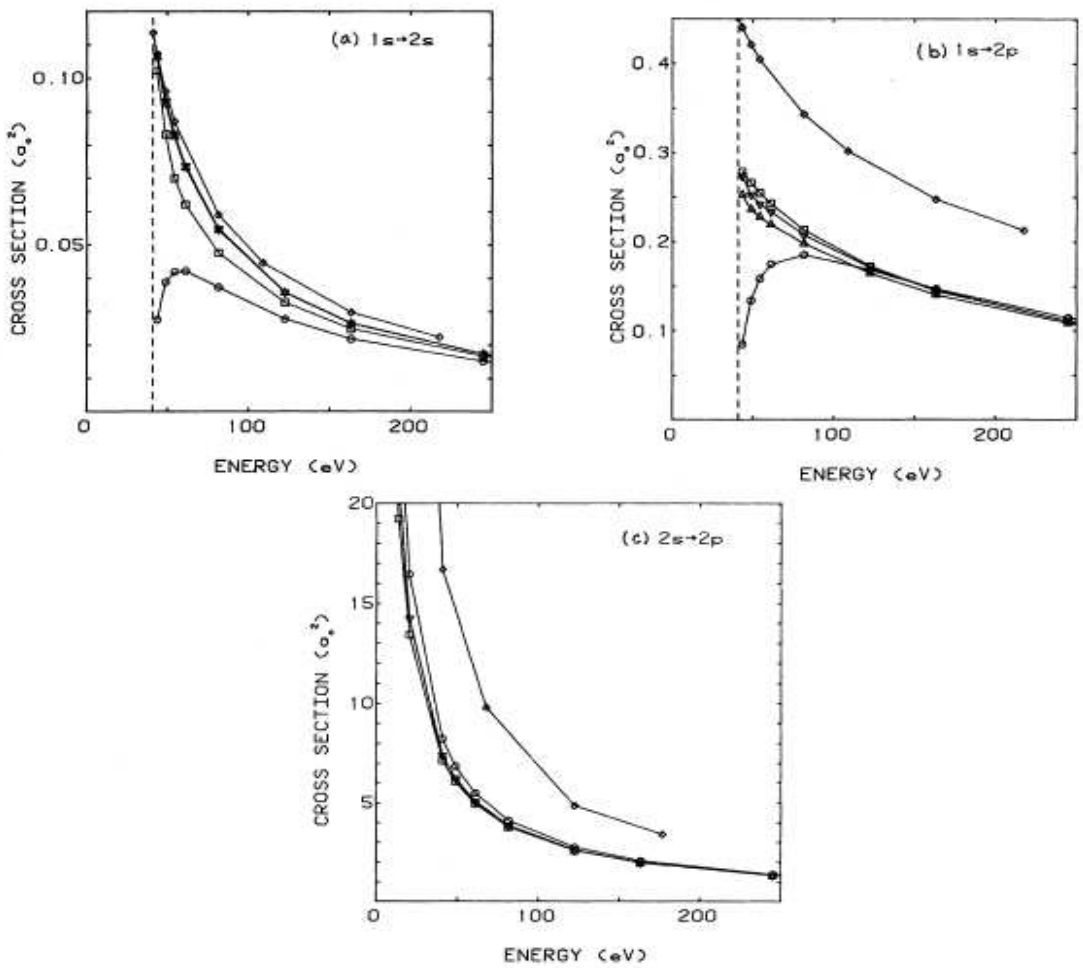



Figure 15. $He^+$ excitation. Temperature and density for downward-pointing triangle points correspond to those given in Fig. 7. Other points represent calculations using various approximations other than DW. (Whitten, Lane, and Weisheit, 948.)

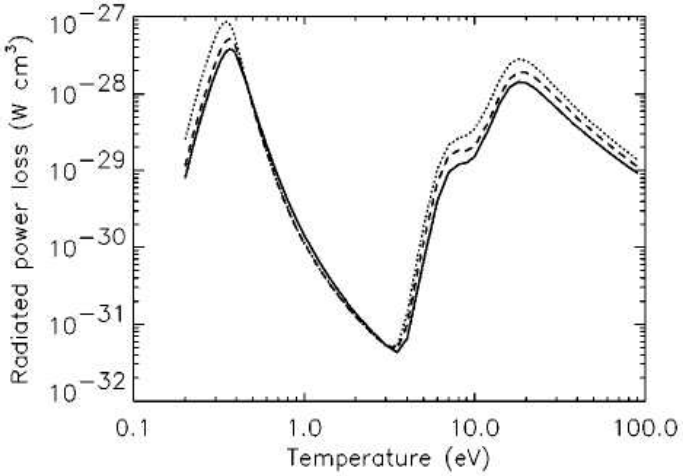



Figure 16. Radiated power loss for Li plasma at $10^{14}$ $cm^{-3}$. Dashed line represents ATOMIC DW results. (Loch, et al., 13.)

# A NEW DEFINITION OF INFORMATION AND ITS IMPLICATIONS

Darrell K. Lim
*California State University, Fullerton*
T-4
Hans Ziock

*Los Alamos National Laboratory, Los Alamos, New Mexico* (EES-6)

INFORMATION (noun): Anything that is...

1) Actionable and expressed in physical form

2) Able to make and sustain more of itself

3) Can be stored

4) Is readable

5) Is non-randomly selected

Explaining information is very difficult in that it usually results in a wide variety of circular definitions. Many popular definitions explain information in the context of how it's transferred, obtained, or what other words to which it's synonymous. The problem with these definitions is that it makes it difficult to truly understand what information is and to figure how it originates. It's important to acquire a more operational and distinguishable definition of information, seeing how it has become increasingly more important in today's society, hence this era's title: “The Information Age”.

One of the many ways the new definition of information can be understood and possibly used would be to model a simple program that implements as much of the definition as possible. I have developed a code that in a simple and loose manner incorporates the constraints that form part of the definition. These are the first steps in achieving a code that would evolve and learn.

So far, a simple source code is all I have accomplished. As I build upon this simple source code, however, I hope to discover more things about the new definition of information than I could easily conceptualize through mere thought. These new insights will form part of the presentation.

I'm working on implementing the elements of the definition to my code in a more robust form. If successful, I will have achieved the first steps in a code that would be able to learn, adapt and survive in a given environment. In the long run, this code could eventually assist the study of artificial intelligence.

## 1 INTRODUCTION

The objective in this project can be thought of in three interrelated parts. The first and most basic part deals with finding possible errors or misinterpretations in the definition of information, shown above, and

correcting them.

The second part involves a biological application for the definition in man-made living protocells. Ironically, the idea for this project actually originated from the now completed protocell assembly which attempts to produce an artificial self-assembling nano-scale system. This biological part of the project aimed to construct a fully operational self-replicating nanomaterial using a bottomm-up method, contrary to the previously-used top-down method. Possible applications for this 5nm protocell would be implementing it into a computer so that the computer would be able to "fix" itself when damaged much the same way as skin heals itself from scratches. "Future generations of protocells might be useful for concentrating nuclear waste, metabolizing toxins, recognizing and neutralizing bio-agents, and addressing a host of other issues related to the security, human health, environment, energy, materials, and predictive science programs of the Laboratory." [1]

The third part involves a machine-based application for the definition through computer algorithms. The third part can be further broken down into two main objectives. First, by writing an algorithm using the definition of information, it would be easier to find flaws (if any exist) or necessary additions to the definition. For example, one can discover that random selection is indeed necessary (definition part 5) to "get the ball rolling" and create a strand of information that can successfully make and sustain more of itself, thus being able to read itself and store its code. It is trivial that a physical piece of information would soon become obsolete from thermal noise (kT) if it does not obtain the ability to copy or maintain itself.

Second, a more useful strategy in writing an algorithm for the definition of information is its possible application in the expanding field of evolutionary computing. More specifically, thinking about the three components of a working system: inputs, outputs and the internal model connecting the two; the program that simulates the definition of information can extrapolate an unknown output from known inputs and known models [3].

## 2 HYPOTHESIS AND PREPARATION

### 2.1 The Complete Definition

Though the definition displayed above is sufficient for the purpose of getting a feel for the criteria for something to be called information, it lacks a solid explanation. That being the case, the following is the official definition [2]:

1) The selection must take place among actionable alternatives and hence the information must be actionable. Information that is not (directly or indirectly) expressed in some physical form cannot be selected.

2) The criterion of selection is directly (and/or indirectly) simply whatever makes and sustains more of "itself", namely the information, e.g., DNA, RNA (i.e. XNA).

    a. Note that information makes more information and if it does not, the information will eventually be lost due to thermal noise that by thermodynamics always "pushes" one back to the equilibrium state.

b. Making more information may happen by further selecting among a finite set of possibilities, bits, or it may imply selecting from a greater number of possibilities, or bits. The advantage of a greater number of possibilities or bits to the growth of information is self-evident, and hence, we define our system of storage as open-ended.

c. Selecting and making more information locally reduces the entropy.

3) In order to usefully select information, information must be stored (written); otherwise there is no way to decide that it has been selected.

a. Storage also implies the possibility of multiple copies and hence "more of itself".

4) In order to make information actionable, information must be read.

a. Note that XNA reads itself.

5) The selection is made by the environment.

a. Initially this was purely the physical environment.

b. Life itself has become the most important/challenging part of the environment.

c. This in turn makes information the key environmental feature today.

d. This leads to a continual push or selection for more and better information.

e. This creates open-endedness, an underlying "property" of life.

### 2.2 Issues in Speculation

In order to test each component of the definition, it is beneficial to take multiple fronts; in our case, biological and computational. As mentioned before, my particular role in this project is to test the definition using computational methods. The search for the root meaning of information requires a great amount of thought and difficult questions are bound to pop up.

1) **Question:** Is one bit of information in theory sufficient for life (biological or artificial) to begin? If so, what other preconditions must the non-thermal environment provide?

**Speculation:** It is arguable that the origin of information could have been nothing more than a single binary digit: that some bit (1 or 0) may have been more adaptive to the given environment than its counterpart. This guess, however, would more than likely violate the definition because a single bit of code would by no practical means have the potential to read, store, or reproduce itself. If that is the case, then some initial violations in the 5$^{th}$ part of the definition of information must be made to start the life process of a code strand. As mentioned above, a random unguided natural process seems necessary to get a possible combination of units that can eventually fulfill the requirements. The transition from the simplest form of code and life (being perhaps one bit of binary code) to a reproducible, readable, storable and actionable strand of information seemingly is a thermally driven generation of possibilities followed by a down-selection process; this statistically high improbability of forming "information" is made reasonable with enough trials in that the high improbability can be overcome.

2) **Question:** Is truly new knowledge only gained by trial and error?

**Speculation:** A similar question would be "Is information truly new if we use previous information to deduce it?" Apparently one can think of a string 1s and 0s as already existing information and applying this already existing string to an even bigger string would be considered new because it may benefit the larger string against the environment. However, determining whether it would indeed be beneficial would only be possible by testing it. Even more speculative, strands of code can exist in the universe and the act of considering and understanding these pieces of potential information would be considered newly acquired information if found to be useful. It can also be put in terms of language: the English alphabet consists of 26 letters, yet many of their potential combinations can be thought to produce new information such as words, sentences, paragraphs and articles. However again, they would only seemingly be deemed to represent information if they prove to be useful, which again can only determined by trial and error.

3) **Question:** What implications do the answers to issue (2)) have on artificial intelligence and machine learning?

4) **Question:** What implications do the criteria associated with our definition and the general open-endedness aspect have for artificial intelligence and machine learning?

5) **Question:** Can one begin to explore some of these issues in a computer/computer algorithm?

**Speculation:** This last three issues were the prime motivation for my part of the project in dealing with computers. Artificial Intelligence is still far beyond my capabilities in computer programming, but the use of Evolutionary Algorithms (EA) would seemingly be more obtainable as of the near future and more practical in the short run. The reason for that is EA explicitly makes use of the method of trial and error to obtain an optimal solution to a certain situation given the known inputs and relevant model. Instead of the initial inputs being intelligently picked it is common to put in random data and use several thousand or even millions of generations of parent selection, recombination or mutation, and survivor selection to get a global optimum.

# 3 PROCEDURE AND OBSERVATION

With the time allotted during this summer, I was able to create three complete workable programs, each more complicated and fulfilling of the definition than the previous.

## 3.1 First Program

The first program implemented the 1st, 3rd, and 5th parts of the definition of information. In detail, this program takes an initialized parent strand (z-strand) of 10 digits all set to zero and creates ten offspring from it while choosing one random digit in each offspring to manipulate in the range of 0-9. In the manipulation process, that one digit either increments, decrements or stays the same as the parent strands' digit. After that, one offspring (or the

parent strand itself) is selected based on the sum of its 10 digits: the highest sum gets selected. If there are more than one offspring with the same highest value, the program just selects the first one it evaluates. The selected strand then becomes the parent strand and the whole process loops over again until a maximum sum is reached, predictably, 9 9 9 9 9 9 9 9 9 9. The output of the program's execution is shown below:

*Generation: 1*

**zstrand: 0 0 0 0 0 0 0 0 0 0**

astrand: 0 0 0 0 0 0 1 0 0 0 bstrand: 0 0 0 0 0 0 0 0 0 0

cstrand: 0 0 0 1 0 0 0 0 0 0 dstrand: 0 0 0 0 0 0 0 0 0 0

estrand: 0 0 0 1 0 0 0 0 0 0 fstrand: 0 0 0 0 0 1 0 0 0 0

gstrand: 0 0 0 0 1 0 0 0 0 0 hstrand: 0 0 0 0 0 0 0 0 0 0

istrand: 0 0 0 0 0 0 1 0 0 0 jstrand: 0 0 0 0 0 0 0 0 0 0

Selection is: Strand A Selection is: Strand C Selection is: Strand E

Selection is: Strand F Selection is: Strand G Selection is: Strand I

Selection Value is: 0.10

*Generation: 79*

**zstrand: 9 7 9 7 5 9 9 4 9 9**

astrand: 9 7 9 7 5 9 9 4 9 9 bstrand: 8 7 9 7 5 9 9 4 9 9

cstrand: 9 7 9 7 6 9 9 4 9 9 dstrand: 9 7 9 7 5 9 9 4 9 9

estrand: 9 7 9 7 5 9 9 4 9 9 fstrand: 9 7 9 7 5 8 9 4 9 9

gstrand: 9 7 9 7 5 8 9 4 9 9 hstrand: 9 7 8 7 5 9 9 4 9 9

istrand: 9 7 9 7 5 9 9 4 9 9 jstrand: 9 7 9 7 5 9 9 4 9 9

Selection is: Strand C

Selection Value is: 7.80

*Generation: 103*

**zstrand: 9 9 9 9 9 9 9 8 9 9**

astrand: 9 9 9 9 9 9 9 8 9 9 bstrand: 9 9 9 9 9 9 9 8 9 9

cstrand: 9 9 9 9 9 9 9 9 9 9 dstrand: 9 9 8 9 9 9 9 8 9 9

estrand: 9 9 9 9 9 8 9 8 9 9 fstrand: 9 9 9 9 9 9 9 8 9 9

gstrand: 9 9 9 9 9 9 9 9 9 9 hstrand: 9 9 9 9 9 9 9 8 8 9

istrand: 9 9 9 9 9 9 9 7 9 9 jstrand: 9 9 9 9 9 9 9 8 9 9

Selection is: Strand C Selection is: Strand G

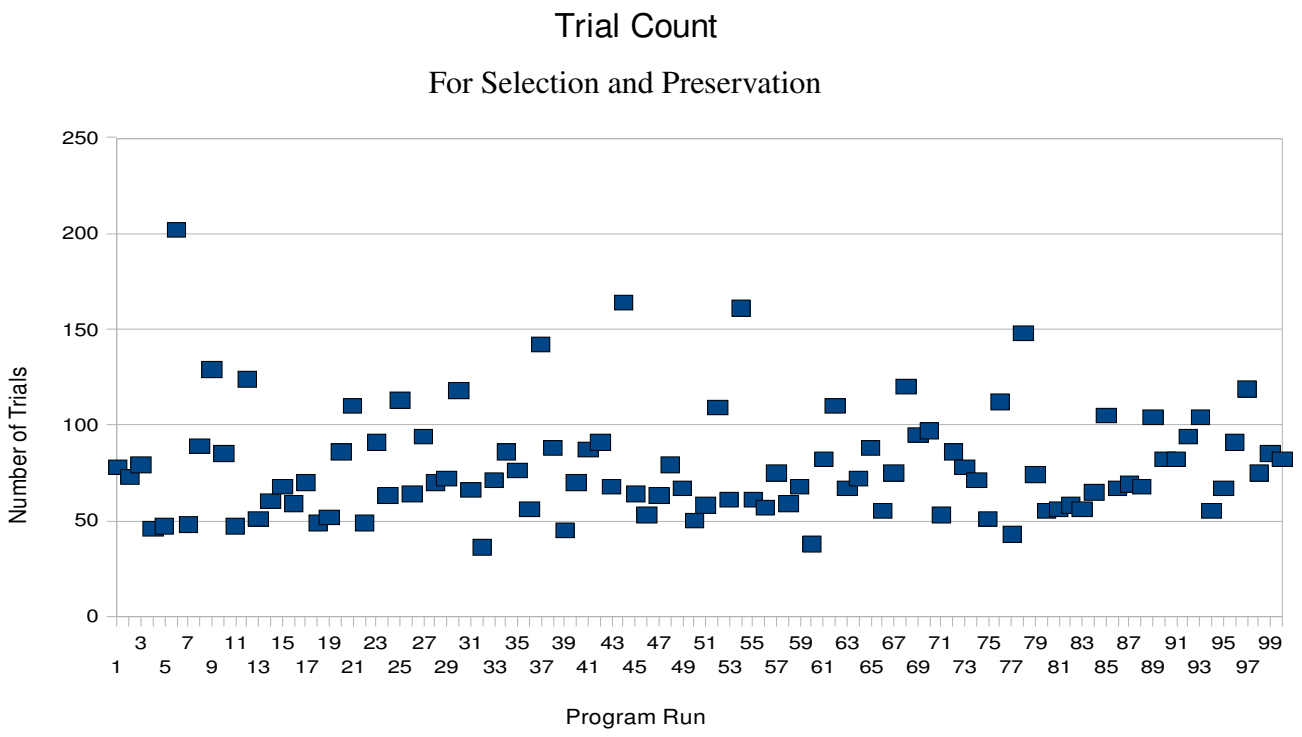
Selection Value is: 9.00

The program examined/fulfilled the 1st part of the definition because the integers in each digit of each strand is expressed by a collective state of many logic gate outputs in the computer; by the same concept in bistable multi-vibrators, the 3rd part is also fulfilled; lastly, the selection process in this program is non-random, fulfilling the 5th part of the definition as well.

### 3.2 Second Program

The second program seemed to stray off the topic of self-correcting entities, but did pave the way to my third program which took this idea to a more complicated level. This program does not have a parent data strand create multiple mutated copies of itself, but it does preserve favorable qualities and easily discards everything else; it also makes it easier to analyze the basic properties of selection processes. In detail, this program initializes what will become the "living strand" with all digits at zero. The "correct strand" is initially set to a predetermined combination in which if the "living strand" matches it, will execute a certain operation. Getting the "living strand" to match the "correct strand" proceeds via a randomization process involving the "randomization strand". Each generation the "randomization strand" will dish out a string of thirty random numbers between -9 and 9. If the right number in the right digit of the "randomization strand" matches that of the "correct strand", then the "living strand" will take and preserve that number in its correct digit place.

This simple display of selection and preservation demonstrates its effectiveness in evolution, such as that of information. With thirty digits, each with 19 possible values (-9 to 9, inclusive), getting the correct combination that mimics the "correct strand" with out the use of selection and preservation would theoretically take 19^30 trials. With the use of selection and preservation, however, the number of randomization trials gets drastically reduced to an average (over 100 program executions) of 78.71. As shown in the scatter plot below, the density of points lie between 50-100 trials for a perfect match of 30 digits with 19 possible integers per digit.

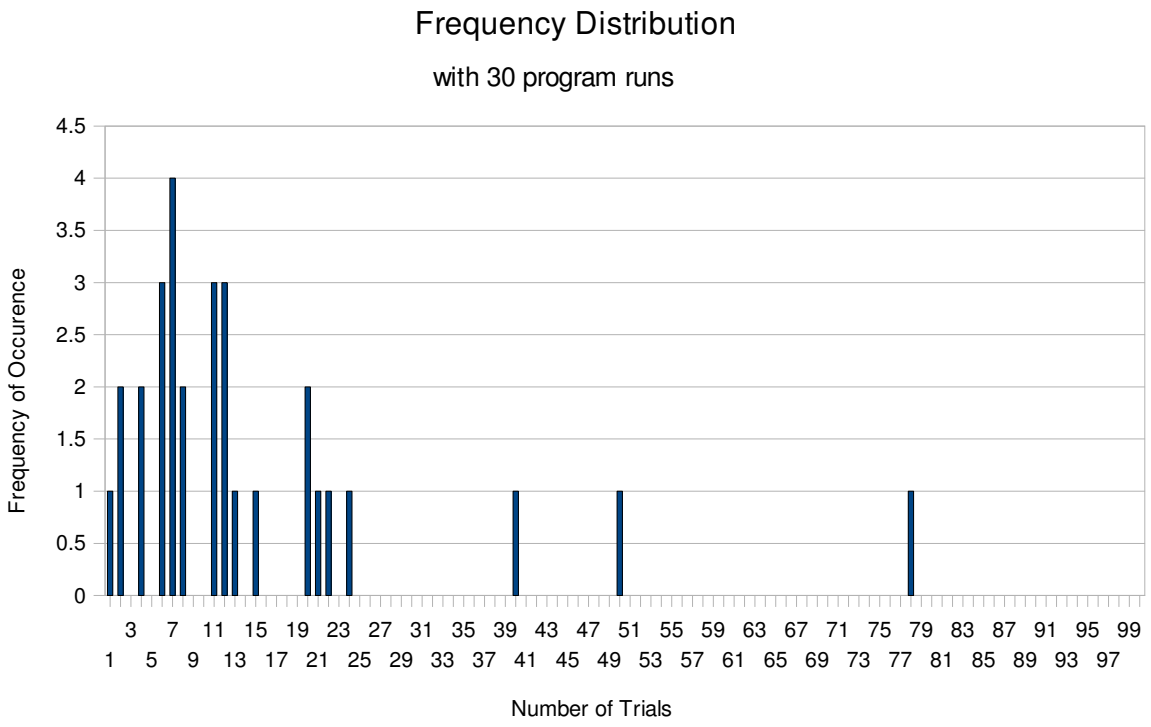


Graph 1 – A scatter plot of the Number of Trials for all 30 digits of the "living strand" to match that of the "correct strand" for 100 program executions.

In theory, 78.71 would be the number of trials for the last digit (of all 30) to match the "correct strand" in a randomization process. Intuitively, 19 trials should be enough to yield a correct combination of 30 digits, since each digit has to only get the correct number once to be preserved and the majority of digits would yield a correct number by 19 trials. However, for 30 different digits in which one needs to get the correct number (in a range of 19) the program would only terminate if all 30 got it correct.

Of the 30 digits, on the first trial I would expect to get 30/19 correct, leaving me with 30 - 30/19 to get right on the next trial which is, again, completely random. On the next trial I would expect to get (30-(30/19))/19 correct digits leaving me with 30 – 30/19 - (30-(30/19))/19 to get right on the next trial. Continuing this down, on each subsequent trial I will on the average get fewer correct digits numbers. In fact, when I only have one remaining digit to get right, it would take me an additional 19 trials on average. The above sum will be far more than 19 as you can (and do) easily see.

Taking a different approach, looking at a frequency distribution for the number of trials it takes to get one digit to be the correct number out of 19 possible values, the highest value for the number of trials is roughly 78, as observed from 30 executions from the program. Note that 30 executions of randomizing one digit in a range of 19 is the same as taking 30 randomized digits in a range of 19 as each of the 30 digits can be thought of and treated separately.




Graph 2 - A frequency distribution of the number of trials needed to get one digit to match a randomized range from -9 to 9 (19 possible values).

Shown below, is a random summarized execution of the program. Interestingly, yet intuitively, one realizes from looking at the output data that the improvement of the “living strand” from selecting and preserving correct integers decelerate exponentially as the trials reaches its end. Obviously this deceleration would be due to fewer opportunities for correction (explained above) as the preserved integers in their correct order does not change after its placed. Coincidently, I observe this phenomenon in all three of my programs, each using the “select and preserve” method, though in a different manner.

Generation: 1

Randomization Strand: 9-51302-3063-347-88-5-5-2-439-7-76-4-7-2534

Living Strand: 00000000000000000000000000000

Correct Strand: -94-84-64-54-44-234-54-34-13404-7514-5423

Generation: 2

Randomization Strand: 9280-7-59-115-77-466-977-5-70-5-73-9-52-19-5

Living Strand: 00000000000000000000-70000000

Correct Strand: -94-84-64-54-44-234-54-34-13404-7514-5423

Generation: 48

Randomization Strand: -9-2-5-3-2-2554-264-13-32-52-8901-6-97871-5-1

Living Strand: -94-84-64-5404-204-54-34-13404-7510-5423

Correct Strand: -94-84-64-54-44-234-54-34-13404-7514-5423

Generation: 83

Randomization Strand: 76-40-22-2-1-484-561-8-17992-7-189-3-3-9415

Living Strand: -94-84-64-5404-234-54-34-13404-7514-5423

Correct Strand: -94-84-64-54-44-234-54-34-13404-7514-5423

Generation: 84

Randomization Strand: -8-2-2-6465-7-8-975-81-7-76985-2-9-6-47-3277-6

Living Strand: -94-84-64-54-44-234-54-34-13404-7514-5423

Correct Strand: -94-84-64-54-44-234-54-34-13404-7514-5423

Execute Operation!

### 3.3 Third Program

Though these two programs implement (very loosely) the 1st, 3rd, and 5th part of the definition of information, they still lack sophistication to reproduce on their own (2nd definition part) and to be able to read/evaluate itself for selection (4th part). My third program aimed to make a population of data strands (strings of numerical characters that hold meaning and gives it abilities) that, based on their code, are able to do multiple tasks. The first task that came to mind and that is undoubtedly important in any individual life form is reproduction. I figured that in the future I or someone working with my program will want to add a few more simple or complicated qualities that each data strand can potentially have, so I made my program easily manageable for modification. Since this project would be substantially more complicated that the previous two, there are obviously more things to consider.

#### 3.3.1 Difficulties

An underlying difficulty is understanding the nature of information and how it may have originated so my program can mimic this nature as closely as possible. Obviously, a possibly closest mimic one could get to the actual nature of information by means of machine would be to produce a code that could ultimately write itself. More explanation will be given later when I briefly discuss my attempt to do just that in my fourth project element.

The rest of the decision making process was no easier. It is important to realize the many factors that go into evolutionary computing of all types: representation, evaluation/fitness function, population, parent selection mechanism, variation operators, survivor selection mechanism, initialization and termination conditions to name most. For each situation, each component of an evolutionary algorithm (EA) should be considered and carefully selected so as to not make the program converge too quickly, leaving the termination result a “local optimum” instead of a desired “global optimum”, or too slowly, leaving the program taking too long to agree on a single “optimum”. Specifically, one would have to consider binary representation or integer representation, recombination or mutation, stable population or increasing population, etc.

Even beyond that, there are multiple types of mutations and recombinations for both integer representations and binary representations. For example, in binary mutations there is “bit flipping”, which simply takes a digit and flips it (1 to 0 or 0 to 1). In contrast, in binary recombinations there is the “one-point crossover” case for which one takes a randomly selected point in both parents and switches the part after that point to create two different offspring. There is also the “multi-point crossover” case, in which one randomly selects a number of crossover points for both parents and alternates between trading and keeping the parts between the crossover points to create two different offspring and finally there is the “uniform crossover” case in which considers each digit of an offspring and uses a probability value to decide which of the two parents to copy from.

Again, depending on the situation and conditions of the problem you are trying to solve, one option may be better than another. However, each option has its flaws and biases. The one-point crossover is good for strands of code that require a certain correct combination of it's digits to operate, but that crossover technique has a tendency to split around the middle part of the strand and almost never on the sides and the ends. Uniform crossovers equally consider every part of a data strand, but it would not be a good option for strands with sensitive combinations. If one would want to use an integer

representation instead of a binary representation for the set of data strands, the mutation and recombination options become a lot more complicated (depending on your problem condition).

### 3.3.2 Details of Program Operation and Procedure

Taking all those caveats into careful consideration, I figured the best way to eventually construct an optimized program for my objective is to use the method of “build and modify”. Indeed, the essential properties of evolution comes from multiple cycles of modification. Relying on that comforting ideal, I've decided to partake the following options:

| | |
|---|---|
| Representation | A binary string whose length that is randomly selected (between 0 and 1000) |
| Recombination | Uniform crossover of two parents of the same strand length |
| Recombination Probability (RP) | Depends on the code, where a 7 digit binary value (from 1 to 127) relates directly to the percentage probability of recombination between two parents (if value is over 100%, then it is read as 50%) |
| Recombination Crossover Probability (RCP) | Depends on the combined values of the two parents' individual Recombination Crossover Probability. They both individually obtain a value of the RCP the same way they obtain the value of the RP (through a 7 digit binary value) |
| Mutation | Binary inversion (1s become 0s and 0s become 1s) |
| Mutation Probability (MP) | Depends on the code, where a 7 digit binary value (from 1 to 127) relates directly to the percentage probability of successful mutation (if value is over 100%, then it is read as 50%) |
| Mutation Digit Probability (MDP) | Depends on the individual's code. It obtains a value of the MDP the same way it obtains the values of the MP (through a 7 digit binary value) |
| Parent Selection | If the parent(s) possess the correct binary sequence, then the parent(s) will possess the qualities to reproduce through either mutation or recombination |
| Survival Selection | The number of working parts in a data strand based on a Fitness Value scoring method |
| Initial Population Size | 100 randomized data strands (where each digit in each strand is randomly picked to be either 1 or 0) |
| Maximum Population Size | 1000 (one-thousand) |
| Maximum Strand Size | 1000 digits |
| Number of Offspring | Depends on the combination of the binary sequences for a strand |
| Initialization | 100 randomized data strands with length that is randomly selected (between 0 and 1000) |
| Termination Condition | 10 selection cycles |

Table 1: Details on the components and operations of my program.

My program will initialize, evaluate, select and reproduce (make exact or manipulated copies). Then, it will restart the cycle over again (except the initialization part) until a termination condition is met. I will represent the concept of "information" with a single string of binary numbers. As mentioned above, this string can be from 0 to 1000 digits long and altogether there will be 1000 strands (representing 1000 individual living entities). Depending on the combination of its binary sequence (e.g. 10011101001), it will be able to apply certain qualities that are defined by a list of "selection sets". "Selection Sets" (shown below) are strings of binary sequences of a certain length (much like the data strands), depending on which selection set, that a given data strand in the population will have to match in order to receive that quality to which the "selection set" pertains. Altogether there are eight selection sets (two main selection sets with three sub-sets underneath each). More detail on these "selection sets" is given in the following table and procedure:

| **Abilities:** | **Number of digits a data strand has to match:** | **Selection Set Name:** |
|---|---|---|
| Reproduce through recombination | 10 | RR |
| Recombination probability | +7 where the binary value is the percentage; and if over 100%, then = 50% | RP |
| Uniform crossover probability | +7 where the binary value is the percentage; and if over 100%, then = 50% | RCP |
| Number of recombination trials | +(0-100) where the number of digit matches is the number of trials | RT |
| Reproduce through asexual mutation | 10 | RM |
| Mutation probability | +7 where the binary value is the percentage; and if over 100%, then = 50% | MP |
| Mutation digit probability | +7 where the binary value is the percentage; and if over 100%, then = 50% | MDP |
| Number of mutation trials | +(0-100) where the number of digit matches is the number of trials | MT |

Table 2: Selection Set details (the procedure below will explain more about this).

1) Four of the eight selection sets (RR, RP, RT, RM, MP, MT) are randomized (meaning that each digit of a selection set is randomly put to either 1 or 0). The length of each selection set has been predetermined and is given in Table2 shown above. For instance, the selection set RR has 10 randomized digits and selection set RT has 100 randomized digits. The reason four of the eight selection sets (RP, RCP, MP and MDP) are not included in the randomization process is because their criteria for a data strand to obtain their quality depends solely on the binary value (i.e. 0001101 = 13) that the data strand holds. For instance, if a part of a data strand has 0110001, then the RCP or MDP (depending on what you are looking at) will be 49.

2) An initial population (in this case 100) of data strands are created. Each data strand is composed of a random number of digits in which its maximum is 1000.

3) Each of the 100 data strands are then randomized (meaning each digit of a data strand is randomly put to either 1 or 0). For example: a data strand of length 10 COULD be randomized to look like this: 0010111010 or some other combination of 1s and 0s.

4) Next, the fun begins. Each data strand of the initial population will be evaluated individually. For each evaluation, my program will first check to see if the data strand has the qualifications to reproduce through recombination (meaning the data strand will possibly be able to find a mate and create offspring). If the data strand happens to have the qualifications for that, then the program will then check to see the data strand's probability of reproduction through recombination (RP) (how likely it will successfully create offspring), the data strand's crossover probability (RCP) (the chance that each digit of its offspring will be the same binary number as the data strand being evaluated) and the data strand's number of trials (RT) (the number of chances it gets to mate). Notice that the number of trials and the data strand's recombination probability go hand-in-hand; the more trials, the less of a probability value needed, and vice versa. If the data strand has more than one complete match of the RR selection set, then the program will randomly take just one match of all RR matches and evaluate the associated selection sets (RP, RCP, RT).

5) After the program finishes evaluating the data strand's recombination abilities, it puts it to use. If the data strand happens to have the ability to recombine with a mate, the program first randomly finds a mate of the same strand length. Depending on the data strand's recombination probability (RP) and its number of given trials (RT), it may or may not produce an offspring. If the data strand and its mate does produce an offspring, the program takes into account both the data strand and its mate's recombination crossover probability (RCP). What happens is the program normalizes the two RCPs. The normalization of two numbers is their ratio equivalence in which if summed together would equal 100. For example, if the data strand has an RCP of 68 and its mate has an RCP of 25, then the normalization of the two, respectively, would be 73.12 and 26.88. Depending on these ratios, the chances of each digit of the data strand's offspring to possess the same binary value as the data strand would be roughly 73% and 27% for the data strand's mate.

6) Next up is the program's evaluation to see if the same data strand has the qualifications to mutate. The program goes on to evaluate its mutation abilities much the same way it evaluates its recombination abilities. It first checks whether or not the strand has the right combination (matches the RM selection set) to reproduce through mutation (meaning a data strand can asexually make a copy of itself with possible variations in the offspring). After that, again, it will check the data strand's probability of mutation (MP), the data strand's digit probability (MDP) (the chance of occurance that each digit of its offspring will invert rather than stay the same) and the data strand's number of trials (MT). Again, if the data strand has more than one complete match with the RM selection set, then the program will randomly choose just one match and then evaluate its associated selection sets (MP, MDP, MT).

7) Also much the same way, the program will then implement the data strand's mutation qualities, if it has any. If the data strand has the qualities to mutate, the program will then give it an MT number of trials, each with a probability MP. Mind you, this is the same thing that happened with the implementation of the same data strand's recombination abilities (if any). Next, if the production of an offspring is successful, then the program uses the data strand's digit probability (MDP) to decide whether or not each digit of the data strand's offspring will invert or stay the same as the digit of the data strand.

8) It is quite obvious that eventually, with reproduction going on with each individual data strand in an initial population, that the maximum population will be reached and no more "room" will be available. This is where the heart of the program takes place: competition. There are many types of competition, but the one that I liked was called Tournament Selection (usually for unknown populations or systems in which one cannot guess the population at a single given time). When the maximum population is reached, the program will randomly pick two data strands in the population pool and evaluate them both according to their "fitness value" (a value given to each individual data strand that quantitatively measures its abilities defined by how many selection sets the data strand matches or optimizes). Here is how the "fitness value" for an individual data strand gets measured: 1. The data strand gets 100 points for every RR (ability to recombine) it possesses and every RM (ability to mutate) it possesses.

   2. For the one randomly selected RR or RM (if the data strand has more than one), the data strand will get additional points correlating to the randomly selected RR and/or RM's associated selection sets ((RP, RT) or (MP, MT)). Notice that RCP and MDT are not included here because their values are usually neither beneficial nor deleterious to a given data strand. So, the data strand will get additional points correlating to the binary values of the RP and MP, which is pretty much the same as the percentage probability of reproducing through mutation or recombination (which ever the data strand possesses).

   3. As for the number of trials (RT and MT), the data strand will get additional points correlating to the number of trials times 10. So if the data strand possesses 3 trials for mutation and 4 trials for recombination, then for this part, the data strand has gained 70 points. The two fitness values for the two randomly selected data strands are then normalized (the term “normalized” explained above) and the yielded values determine the probabilities of one winning over the other. Say if the normalized fitness values for two data strands are 75 and 25, then the data strand with the normalized fitness value of 75 has a 75% chance of winning and being selected to survive. This competition process may go on for as long as I want. So if I decide to have it go for 10 cycles, then altogether 10 competitions will take place and 10 losing strands will be deleted.

9) After 10 competitions, the program will randomly select one survivor (who competed) to implement its reproduction abilities to possibly create new offspring and fill up the vacancies in the population from the deleted strands. If one survivor didn't fill up all the vacancies, then the program will loop around and select another survivor (who competed) to possibly finish filling up the vacancies.

10)After the population reaches its maximum again, the cycle of competative

selection goes again until a termination condition is reached in which the program will terminate.

### 3.3.3 Analysis of the Program

Given all of these circumstances, about 1 in every 10 runs of the program actually yields surviving strands. The rest terminate shortly after the randomization of the initial population.

As predicted, though still questionable (based on my particular program) the Fitness Value of the surviving strands gradually increased. Sometimes, however, a lower Fitness Value strand fills the vacancies after competition and the total population's Fitness Value goes down as a whole. Following is a sample of a few Fitness Values taken in undetermined intervals during a run of 250 generations:

| Analyzed Generation (of 250 generations) | First Common Fitness Value | Second Common Fitness Value | Third Common Fitness Value |
|---|---|---|---|
| 1st | 0 | 393 | 190 |
| 2nd | 171 | 393 | 418 |
| 3rd | 393 | 428 | |
| 4th | 190 | 393 | |
| 5th | 428 | 383 | 418 |
| 6th | 393 | 428 | |
| 7th | 190 | 428 | 393 |
| 8th | 393 | 495 | 428 |
| 9th | 393 | 428 | |
| 10th | 428 | 393 | 190 |
| 11th | 418 | 428 | 393 |
| 12th | 383 | 418 | |
| 13th | 428 | 393 | |
| 14th | 428 | 393 | |
| 15th | 418 | 383 | |
| 16th | 393 | 383 | |
| 17th | 393 | 418 | 428 |

Table 3 – The Fitness Values of a few winning strands who competed. This was taken over 17 undetermined generations spread out over a total of 250 generations. Some cells are empty due to lack of variation in the observed Fitness Values.

The downfall, and possibly a predicting factor, in this program is the lack of diversity. One type of strand (determined by its length) takes over a good portion of the population very quickly because the majority of all initialized strands (100) have a fitness value of 0, which reproduces nothing. Other than that, I expected a little variety in Fitness Value for successive generations due to mutations, however the values seem to converge fairly quickly to a local optimum. The convergence factor was one of the warnings that I considered before starting this program, as mentioned in 3.3.1 (Difficulties). Since this was my first trial though, all that is needed is constant modification to prevent homogeneous population outcomes. The first modification should be replacing uniform crossover recombination with one-point crossover. As mentioned earlier, uniform crossover ensures equal chances that each portion of a data strand gets mutated. However, it also falls short for strands whose digit combinations are both large and sequence-sensitive, which is the case for my program. The second improvement should be fixing the mutation operators so there can be more diversity in data strand types, and thus a better chance of reaching a global optimum.

### 3.4 Fourth Program

Unfortunately, my fourth project was not a complete success, largely due to the little time that remained in the summer when this effort was undertaken. The objective of this program was to create a potentially working source code by having a data strand randomly put together a string of command fragments. Depending solely on what the string of command fragments is, that command string will either fail to compile or will compile and execute. As mentioned earlier in this paper, this is potentially the best possible way of implementing the definition of information. It is, most importantly, as open-ended as the Python language that I was using. Any possible command or collection of commands that the data strand automatically constructed will be implemented, whether or not I knew about the command before. The problem with attempting to construct this type of program is the issue with the data strand defining and implementing its own code. Say, in order for the data strand to copy itself, it must first obtain the code to do so. But what is the code that says “Copy me!”? In order to copy something, such as an array, you must first define it before or in the *for* loop:

```
input:
      Array1 = [1,2,3,4,5]
      Array2 = []
      for i in Array1:
            Array2.append(i)
output:
      Array2 == [1,2,3,4,5]
```

With the time allotted after the third project, I was unable to figure out how to define an actual source code and duplicate that source code without changing it first. Once you've defined the source code to duplicate, you've changed that source code and thus, would have to redefine it. This would create an infinite loop of regression. That being the case, the other option that was considered was to create two files of the exact same source code: one for analyzing and

implementing; the other for being the code that gets implemented. For instance, I can have a file with a code that was randomly constructed from command fragments and does absolutely nothing because of the incorrect placement of these fragments; the file then automatically copies itself into another file of the same type. After that, the code somehow analyzes itself and implements whatever it analyzes, whether it be to duplicate something or to delete it, to the newly copied file. The problem, again, with that is that I would have to give pre-programmed cues to implement whatever the code has, which brings nothing new to this program that hasn't been done before in my first three programs. The big problem that I was trying to overcome that I hadn't in my previous three programs was to have the program be able to read itself and manipulate itself accordingly without my assistance. Making a workable program that can read itself will require more research.

## 4 CONCLUSION

For this summer, I was able to test the elements of the proposed new definition of information, albeit very loosely. As mentioned earlier in this project, information was expressed in physical form by the collective states of countless logic gate outputs inside the computer. Information was also able to sustain and make more of itself by implementing the qualities of recombination and mutation defined by the selection sets. Information was able to be stored in memory through an array defined in the program. The stored information was readable to humans (only) through each data strand's binary combination: the qualities of each data strand gets printed on the console during each execution. And lastly, the selection of each initially randomized data strand was certainly non-random as it kept the fittest and deleted the less so.

## 5 REFERENCES


[1] S. Rasmussen, J. Bailey, J. Boncella, L.Chen, S. Colgate, G. Collis, M. Declue, D. Dhaval, S. Grosser, G.Jarvinen, Y. Jiang, C. Knutson, S. Maurer, J. Maxwell, P.-A. Monnard, F. Mouffouk, J. Schneider, A. Shreve, B. Travis, P. Weronski, W. Woodruff, J.Zhang, X. Zhou, H. Ziock (2004). Protocell Assembly Project. LDRD proposal 20050064DR. Retrieved July2008 from Hanz Ziock at LANL.

[2] Ziock, Hans-Joachim; Colgate, Stirling & Jungman, Gerard (in progress). A New Definition of Information and its Implications. LDRD proposal 20090404ER. Retrieved 23June2008 from Hanz Ziock at LANL.

[3] Eiben, A. E. & Smtih, J.E. (1998). Introduction to Evolutionary Computing. Verlag, Berlin, Heidelberg, New York: Springer.

# Modeling the Emission from the Supermassive Black Hole in the Galactic Center Using GRMHD Simulations

Kyle W. Martin,[1] Siming Liu,[2] Chris Fragile,[3] Cong Yu,[4] and Chris L. Fryer [2,5]

## ABSTRACT

Sagittarius (Sgr) A* is a compact radio source at the Galactic center, powered by accretion of fully ionized plasmas into a supermassive black hole of $\sim (3-4)\times 10^6 M_\odot$. However, the radio emission cannot be produced through the thermal synchrotron process in a gravitationally bounded flow (Liu & Melia 2001). General relativistic magneto-hydrodynamical (GRMHD) simulations of black hole accretion show that there are strong unbounded outflows in accompany with the accretion. With the flow structure around the black hole given by GRMHD simulations, we investigate whether thermal synchrotron emission from these outflows may account for the observed radio emission and discuss the implications of this study on these GRMHD simulations and possible production of non-thermal particles by this source. We find that simulations producing relatively high values of plasma $\beta$ cannot produce the radio flux level without exceeding the X-ray upper limit set by *Chandra* observations through the bremsstrahlung process. The predicted radio spectrum is also harder than the observed spectrum both for the one temperature thermal model and a simple nonthermal model with a single power-law electron distribution. Since higher frequency emission is produced at smaller radii, the electron temperature needs to be lower than the gas temperature near the black hole to reproduce the observed radio spectrum. A more complete modeling of the radiation processes, including the general relativistic effects and transfer of polarized radiation, will give more quantitative constraints on physical processes in Sgr A* with the current multi-wavelength, multi-epoch, and polarimetric observations of this source.

---

[1]Physics Department, University of New Mexico, Albuquerque, NM 87131; kmartin1@unm.edu

[2]Los Alamos National Laboratory, Los Alamos, NM 87545; liusm@lanl.gov

[3]Physics & Astronomy, College of Charleston, Charleston, SC 29424; fragilep@cofc.edu

[4]National Astronomical Observatories/Yunnan Observatory, Chinese Academy of Science, Kunming, Yunnan 650011, China; yccit@yahoo.com.cn

[5]Physics Department, The University of Arizona, Tucson, AZ 85721; fryer@lanl.gov

*Subject headings:* acceleration of particles — black hole physics — Galaxy: center — plasmas — radiation mechanisms: thermal, non-thermal

## 1. Introduction

Sagittarius (Sgr) A*, the compact radio source at the Galactic center, is powered by accretion of a supermassive black hole of $\sim (3-4)\times 10^6 M_\odot$ (Schödel et al. 2002; Ghez et al. 2004) in the prevailing stellar winds in the Galactic center region (Melia 1992). The bolometric luminosity of Sgr A* is more than 9 orders of magnitude lower than the corresponding Eddington luminosity, suggesting a radiatively inefficient accretion flow (Melia et al. 2001; Yuan et al. 2003). The radiative cooling processes therefore may be ignored while studying the dynamics of the accretion flow near the black hole, which simplifies the dynamical equations and the corresponding numerical algorithms significantly. Several general relativistic magneto-hydrodynamical (GRMHD) codes have been developed over the past few years to study the structure of non-radiative accretion flows near black holes quantitatively (De Villiers et al. 2003; Gammie et al. 2003; Anninos et al. 2005). Given the extensive observations available for this supermassive black hole, it provides perhaps the best opportunity to study the physical processes in the strong gravity near black holes with GRMHD simulations (Falcke et al. 2000; Huang et al. 2008).

The observed linear polarization and variability of the millimeter to near-infrared (NIR) emissions suggest that they are produced by the synchrotron process near the black hole (Aitken et al. 2000; Melia et al. 2000; Genzel et al. 2003; Eckart et al. 2004, 2006). The longer wavelength emissions are circularly polarized (Bower et al. 2001) with weak linear polarization observed during frequent outbursts or flares (Zhao et al. 2004; Yusef-Zadeh et al. 2007), which in combination with the continuity of the centimeter to sub-millimeter spectrum suggests a synchrotron origin for the longer wavelength emissions as well (Falcke et al. 1998; An et al. 2005). However, the longer wavelength flux densities are less variable than the millimeter and sub-millimeter flux densities, indicating emissions from relatively larger radii (Falcke & Markoff 2000; Liu & Melia 2001; Herrnstein et al. 2004). Indeed, the millimeter and higher frequency emissions have been explained reasonably well with an accretion torus within $\sim$ 10 $r_S$ of the black hole, where $r_S$ is the Schwarzschild radius (Melia et al. 2000; Goldston et al. 2005; Liu et al. 2007); And high spatial resolution VLBI observations do show that the intrinsic size of the millimeter emission region is within 20 $r_S$ in radius (Bower et al. 2004; Shen et al. 2005). While the general relativistic (GR) effects are important for these shorter wavelength emissions originating near the black hole (Huang

et al. 2008), the longer wavelength emissions are produced beyond $\sim 10\ r_S$ so that the GR effects are insignificant and one may readily use the GRMHD simulations to model the observed emission without using the sophisticated GR ray-tracing radiation transfer codes of polarized synchrotron emission (Broderick & Loeb 2005).

High resolution X-ray observations with the *Chandra* space telescope shows that the quiescent X-ray excess from the direction of Sgr A* is extended and has ion emission lines in the spectrum, suggesting that the emission is mostly produced at large radii near the capture radius of the accretion flow by a plasma of a few keV in temperature (Baganoff et al. 2001, 2003; Xu et al. 2006). X-ray flares are routinely observed from the direction of Sgr A* (Belanger et al. 2006). Their spectra can be fitted with a featureless power law (Porquet et al. 2003), and they appear to be always accompanied by NIR flares, which are produced near the black hole (Eckart et al. 2004, 2006; Yusef-Zadeh et al. 2006, 2008). The correlation between the NIR and X-ray flares indicates a synchrotron self-Comptonization (SSC) origin for the X-ray flares (Markoff et al. 2001; Liu & Melia 2001), though the X-ray flares may also be produced via the bremsstrahlung process enhanced by instabilities related to the accretion process (Liu & Melia 2002; Tagger & Melia 2006).

The quiescent state X-ray emission from the inner accretion flow therefore must be lower than the observed flux level. X-ray emission can be produced through both the SSC and bremsstrahlung process. The observed high radio flux level and low upper limit of X-ray flux suggest that the radio emission cannot be produced via thermal synchrotron emission in a bounded flow (Liu et al. 2001). It therefore has to be produced by unbounded outflows and/or by nonthermal populations of relativistic electrons. Given the non-radiative nature of these simulations, strong outflows and winds appear to be inevitable, especially for highly spinning black holes (Noble et al. 2006; Hawley et al. 2007). In this paper, we investigate whether emission from these simulated accretion flow can count for observations of Sgr A* self-consistently.

We carried out 2-dimensional (2D) GRMHD simulations with the Cosmos++ and HARM codes (Gammie et al. 2003; Anninos et al. 2005) assuming axis-symmetry of the accretion flow. The axis of the accretion disk therefore must be aligned in the equatorial planes of spinning black holes. The length scale is determined by the black hole mass. Since no radiative cooling is included, the density is scaled linearly with the mass accretion rate, and the gas temperature and plasma $\beta$, defined as the ratio of the gas pressure to the magnetic field pressure, are independent of the black hole mass and accretion rate. GRMHD simulations give the gas density, pressure, and magnetic field. The gas temperature can be derived from the equation of state. The mass of the supermassive black hole in Sgr A* is well-measured by observations of stellar orbits around the black hole (Schödel et al.

2002; Ghez et al. 2004). One therefore needs to adjust the accretion rate, black hole spin, and electron distribution to fit the observed spectrum of Sgr A*. We consider both a one temperature and a two-temperature model of electrons in the thermal synchrotron emission scenario. For non-thermal synchrotron models, the electron distribution is assumed to follow a steep power-law with a low energy cutoff.

In § 2 we discuss the methods that we used to model the emission spectra from the simulated accretion flows. A detailed discussion of our results and the implications are given in § 3.

## 2. Modeling the Emission Spectrum

We run the Cosmos ++ code over a simulation domain of 218 $r_S$ in radius. Initially, there is an equilibrium torus centered near 100 $r_S$. Under the influence of the strong gravity of the central black hole, the torus evolves and the magneto-rotational-instability sets in and induces the accretion process (Balbus & Hawley 1991). Specific attributes of the accretion disk were examined and the magnetic field, temperature, and particle density profiles are shown in figure 1. Because the simulation was run until the dynamical time near 100 $r_S$, the initial torus structure has not relaxed and there is a second high dense region of particles at $\sim 100\ r_S$. If the simulation were allowed to run longer this region of high density would be pulled closer to the black hole.

We first consider the simplest case, where electrons and ions reach thermal equilibrium throughout the simulation domain, so that the electron temperature is given by the gas temperature. The synchrotron emission coefficient at frequency $\nu$ is given, respectively, by (Pacholczyk 1970; Mahadevan et al. 1996; Liu et al. 2006)

$$\mathcal{E}_\nu(\nu) = \frac{\sqrt{3}e^3}{4\pi m_e c^2} B\, n\, x_m\, I(x_m)\,, \tag{1}$$

where

$$x_m = \frac{\nu}{\nu_c} \equiv \frac{4\pi m_e c\ \nu}{3eB\gamma_c^2}\,, \tag{2}$$

$$I(x_m) = 4.0505 x_m^{-1/6}(1 + 0.40 x_m^{-1/4} + 0.5316 x_m^{-1/2}) \times \exp(-1.8899 x_m^{1/3})\,, \tag{3}$$

$$\gamma_c = \frac{k_{\rm B} T_e}{m_e c^2}\,. \tag{4}$$

The magnetic field, gas density, and electron temperature are indicated by $B$, $n$, and $T_e = T$, respectively, where $T$ is the gas temperature, and $e$, $m_e$, $c$, and $k_{\rm B}$ are the elementary charge unit, electron mass, speed of light, and the Boltzmann constant, respectively. The left and

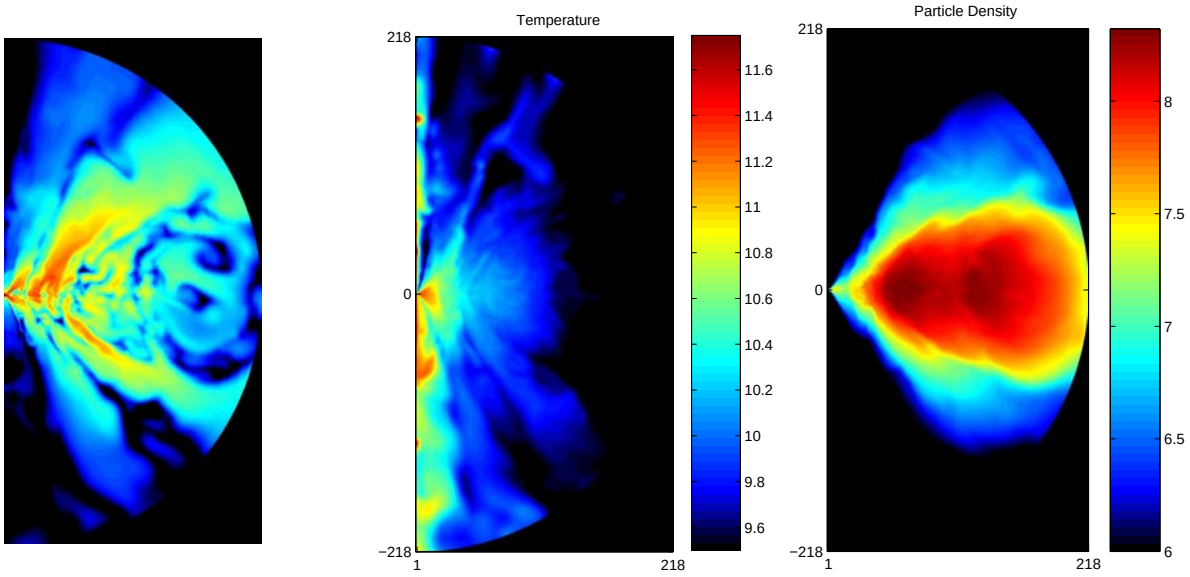



Fig. 1.— *Left:* The magnetic field in units of Gauss in a logarithmic color scale for the simulation with the Cosmos++. *Middle:* Same as the left panel but for the temperature in Kelvin. *Right:* Same as the left panel but for the particle number density in $\mathrm{cm}^{-3}$. The outermost high density region corresponds to the position of the initial torus. If the simulation were to run longer this region of high density at about $100r_S$ would merge with the inner high density region.

middle panels of figure 2 show the profiles of $\mathcal{E}$ at 1.0 and 100 GHz, respectively. The corresponding absorption coefficient $\kappa_\nu$ and the opacity $\tau$ along the light of sight of the accretion disk is given, respectively, by

$$\kappa_\nu = \frac{\mathcal{E}_\nu}{2\gamma_c m_e \nu^2}\,, \qquad \tau = \int \kappa_\nu \, dl\,, \tag{5}$$

where the integration of $\tau$ is along the light of sight from the observer into the source region.

Then the observed flux density from the accretion flow (Melia et al. 2001)

$$F_\nu(\nu) = \frac{1}{D^2}\int I_\nu \, ds\,, \tag{6}$$

where

$$I_\nu = \int \mathcal{E}_\nu exp(-\tau)\, dl\,,$$

is the specific intensity and $D$ is the distance to the Galactic Center and the integration of $F_\nu$ is over the projected area of the source. For a face-on disk with the observer located

at $z = D$, the radio spectra are shown in the left panel of figure 3 for several values of the density normalization. These values are chosen so that the model spectra embrace the observed flux densities. The red line with a 10 times higher density profile than that shown in figure 1 gives the best fit to the observed spectrum.

However, the high gas density implies strong X-rays via the bremsstrahlung process from the two torii evident in the density profile of figure 1 The bremsstrahlung emission coefficient is calculated by (Rybicki & Lightman 1979):

$$\mathcal{E}_\nu = 8.5 \times 10^{-39} n^2 T_e^{-1/2} \exp(-h\nu/k_{\rm B}T_e)\,, \tag{7}$$

where $h$ is the Planck constant. The right panel of figure 2 shows the bremsstrahlung emission coefficient at $\sim 4.1$ keV and the corresponding emission spectra are also indicated in the left panel of figure 3. The bremsstrahlung emission exceeds the observed upper limit for all the three density normalization. This simple model therefore cannot produce the observed radio flux level without exceeding the X-ray upper limit.

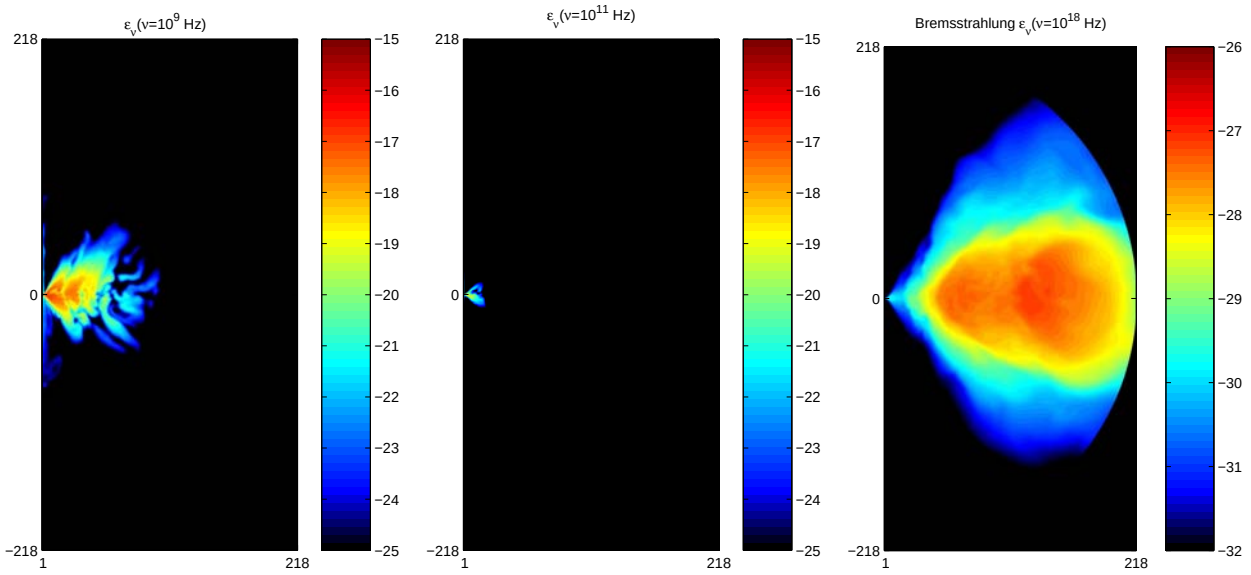



Fig. 2.— *Left:* The synchrotron emission coefficient in the plasma co-moving frame at 1.0 GHz in the c.g.s. units. *Middle:* Same as the left panel at 100 GHz. *Right:* The bremsstrahlung emission coefficient at $10^{18}$Hz, which corresponds to $\sim 4.1$ keV in energy.

It is well-known that nonthermal synchrotron emission is more efficient than thermal synchrotron. We next study whether a simple nonthermal model with a single power-law electron distribution can account for the radio flux without violating the X-ray upper limit.

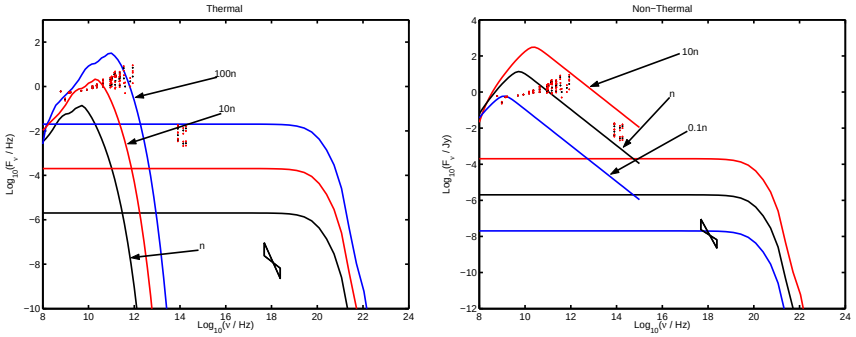



Fig. 3.— *Left:* Thermal synchrotron and bremsstrahlung emission spectra for the one temperature model. The blue, red, and black lines correspond to the particle density in figure 1, 10, and 100 times this density, respectively. As expected, the bremsstrahlung flux level scales with the density squared. The radio to sub-millimeter data was obtained from observations over the past decade. The scattering of the millimeter and sub-millimeter data is caused by the variability of the source. The NIR flux densities are for flares observed during the past few years and therefore should be considered as upper limit for the "quiescent" state emission from the accretion flow. The butterfly is given by *Chandra* observations of the quiescent state X-ray emission, which appears to be dominated by thermal emission at large radii. Thus it also should be interpreted as an upper limit for X-ray emission from the small scale accretion flow. All the bremsstrahlung spectra exceed this upper limit for X-ray emission. *Right:* Same as the left panel but for the non-thermal model with a single power-law electron distribution with a low energy cutoff. The synchrotron spectra are showed up to $10^{15}$ Hz. The density normalization is indicated in the figure. Compare with the thermal models in the left panel, the nonthermal model is more efficient in the synchrotron emission process. However, to match the observed radio flux level, the bremsstrahlung X-ray flux still exceeds the observed upper limit.

The electron distribution is given by

$$N(E) = N_0 E^{-p} \quad \text{for} \quad E > E_0\,, \tag{8}$$

where $E_0$ is the low energy cutoff. In our calculations we chose $p = 3$ to avoid significant NIR emission in the quiescent-state. As the $p$ increases the non-thermal emission model became less efficient and closer to the thermal model. $N_0$ and $E_0$ can be calculated with

$$\int_{E_0}^{\infty} N_0 E^{-p}\, dE = n\,, \tag{9}$$

$$\int_{E_0}^{\infty} N_0 E^{-p+1}\, dE = 3nk_{\rm B}T\,. \tag{10}$$

The corresponding emission and the absorption coefficients are given respectively by (Pacholczyk 1970)

$$\mathcal{E}_\nu = c_5(p) N_0 [B \sin\alpha]^{(p+1)/2} \left(\frac{\nu}{2c_1}\right)^{(p-1)/2}, \tag{11}$$

$$\kappa_\nu = c_6(p) N_0 [B \sin\alpha]^{(p+2)/2} \left(\frac{\nu}{2c_1}\right)^{-(p+4)/2}, \tag{12}$$

where $\alpha$ is the angle between the magnetic field and the line of sight and $c_i$ are defined in Pacholczyk (1970). The right panel of figure 3 show the spectra for several density normalizations, where the synchrotron spectrum is cut off artificially at $10^{15}$Hz and the bremsstrahlung spectra is given by the thermal formula equation (7). Compared with the thermal model, more radio emission is produced for a given density normalization. However, the bremsstrahlung emission still exceeds the observed upper limit to produce the observed radio flux level.

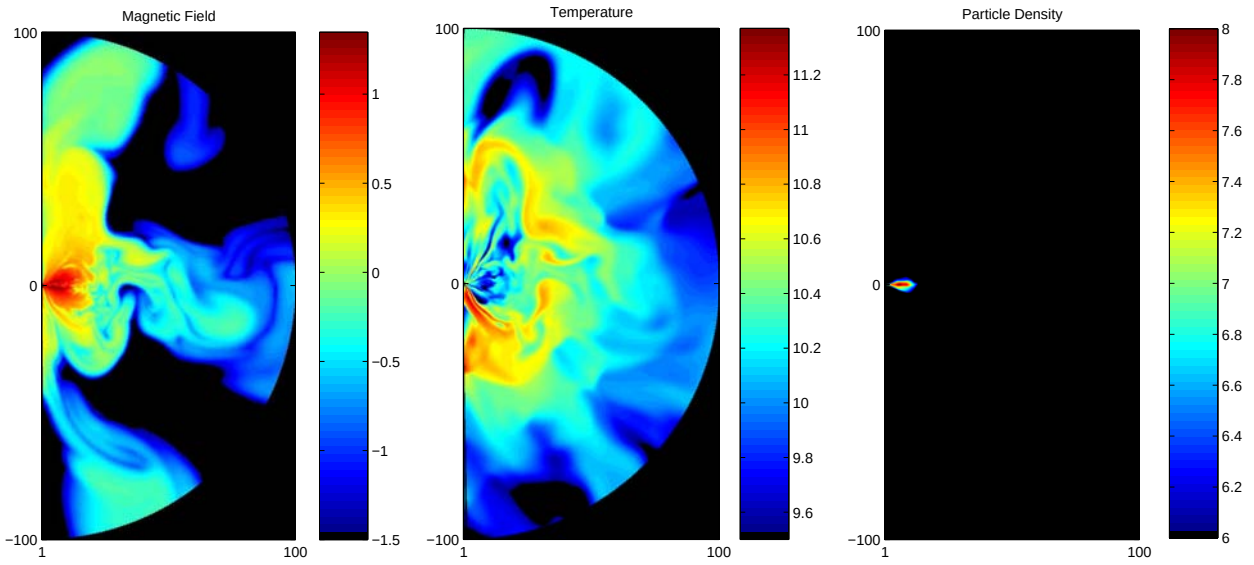



Fig. 4.— Same as figure 1 but for a simulation with the HARM with a zero black hole spin.

While the outer torus of the simulation is expected to merge with the inner torus if the simulation is run for a long enough time, the inner torus is already relaxed at the present epoch of the simulation. Therefore a longer run may remove the bremsstrahlung emission from the outer torus, we don't expect the overall bremsstrahlung X-ray flux to decrease by more than a factor of 2. It is therefore challenging to produce the observed radio flux level from Sgr A* without exceeding the X-ray upper limit with this simulation. The high X-ray

flux as compared with the radio flux is a direct consequence of the high values of plasma $\beta$ given by this simulation.

$$\beta = \frac{P_{gas}}{B^2/8\pi}\,, \tag{13}$$

where $P_{gas}$ is the gas temperature. For this simulation, $\beta$ is always greater than 100 except in the polar regions, where the GRMHD simulation is not reliable and we have excluded these regions while obtaining the above emission spectra. The bremsstrahlung emission is determined by the gas temperature and density. The synchrotron emission also depends on the magnetic field. For high values of $\beta$, the magnetic field is weak. It is therefore difficult to produce significant radio emission without making X-rays through the bremsstrahlung. Simulations that are capable of producing lower values of $\beta$ are needed to overcome this challenge. We also noticed that the radio spectra of both the thermal and non-thermal model were too hard to fit the observed spectrum. Decreasing the value of plasma $\beta$ alone may not explain the observations.

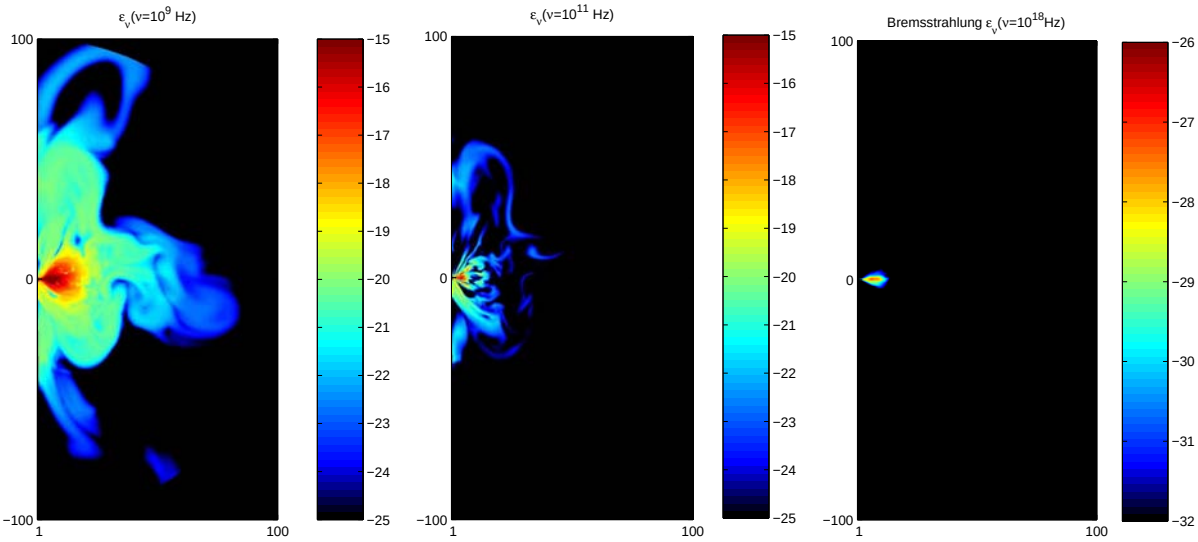



Fig. 5.— Same as figure 2 but for the simulation of figure 4.

Errors in the divergence of B are handled differently in the Cosmos++ than the HARM code (Gammie et al. 2003; Anninos et al. 2005). The latter uses a contrained transport schemes for evolving the induction equation whereas Cosmos++ does not. As a consequence, the HARM code appears to be able to produce lower values of plasma $\beta$. We ran the HARM code in a domain of 100 $r_S$ in radius with the torus initially located at $\sim$ 10 $r_S$. The simulated disk structure in a late time when the initial torus is completely relaxed are shown in figure 4. The simulation produces more than 10 times stronger magnetic

field with comparable density maximum. And figure 5 gives the corresponding emission coefficient maps. Although the synchrotron emission may be comparable to that in figure 4, we expect much less bremsstrahlung X-rays due to the compactness of the torus in this latter simulation. The left panel of figure 6 shows the spectra of the thermal model with several density normalizations. The bremsstrahlung X-ray emission is far below the observed upper limit. However, the radio spectra are still not flat enough to fit the observed spectrum. Since higher frequency emission is produced at relatively smaller radii, where the magnetic field is strong and the gas temperature is high. The fact that the model predicted spectrum is harder than the observed spectrum suggests that emission from small radii needs to be suppressed to produce a softer spectrum. The synchrotron cooling time of electron at smaller radii is shorter. It is possible that the electron temperature is actually smaller than the gas temperature at small radii. We model the electron temperature with

$$T_e = \left(\frac{r}{R}\right)^{\delta} T\,, \tag{14}$$

where $R$ is the radius, where electrons reach thermal equilibrium with the ions and $\delta$ is a free parameter determined by the electron heating and cooling processes.

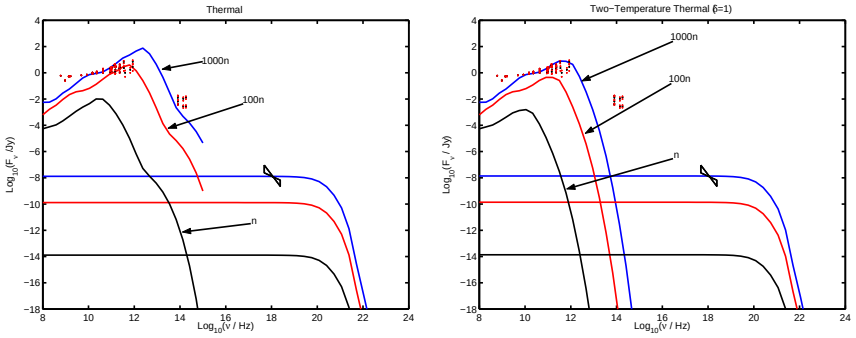



Fig. 6.— *Left:* The same as the left panel of figure 3 but for the simulation of figure 4. *Right:* The same as the left panel except that electron temperature is lower than the gas temperature toward small radii as given by equation (14) with $R = 10\ r_S$ and $\delta = 1.0$.

There was no need to apply the two-temperature model to the first simulation because it would decrease the amount of NIR and radio emissions without decreasing the Bremsstrahlung emission values, we would still have the same problem of exceeding the upper limits imposed by Chandra on x-ray emission. We applied this two-temperature model to the second simulation and calculated the thermal emission spectrum. Figure 6 shows the results. Again we had the same problem, the magnetic field was too weak to produce the observed results. When we did achieve the correct shape the bremsstrahlung emission began to break the upper bound.

## 3. Conclusions

Sag A* is a radio source powered by accretion into a black hole. Modeling this radio source at the galactic center provides us with knowledge into the dynamics of black holes and helps us further understand the plasma physics of the accretion torus. We hope to one day have a simulation that correctly reproduces the observed radiation. With what we learned in this project and the ground work that we laid it should not be too hard to figure out what is missing from the current models.

In this paper none of the simulations that we used were completely correct in matching the observed emission spectrum. Although we were able to use the two temperature model to correctly match the shape of the observed radio and NIR emissions we had the reoccurring problem of exceeding the upper-bound imposed by Chandra in the x-ray emission spectrum. The modeling of the second simulation appears promising and seems that if we examine the non-thermal synchrotron emission we will be able to correctly match the observed spectra without exceeding the upper bound on the x-ray emission. We also discovered that the magnetic fields produced in the simulations needed to be quite high in relation to the particle density of the accretion disc, meaning that the accretion disc $\beta$ factor must be quite low, less then one hundred. If we add spin to the current HARM code the magnetic field inside of the accretion disc will increase and it would be easier to match the observed spectra without exceeding the x-ray upper bound. We also did not incorporate the Doppler effect, we suspect that this would slightly alter our final spectrum but again not drastically enough to fix the underlying problem. We have also not yet included the arbitrary line of sight or taken into account polarization, again this should not effect our results drastically by is needed for a more complete analysis of the model. Future more complete models would include all of these effects.

## 4. Acknowledgments

I would like to thank my mentor Siming Liu for guiding me throughout the duration of this project. I would like to also thank James and Norm for all of their help during this summer. I would finally like to thank UNM, Los Alamos and Sally Siedel for giving me the opportunity to participate in this great program.

## REFERENCES


Aitken, D. K., et al. 2000, ApJ, 534, L173

An, T., Goss, W. M., Zhao, J. -H., Hong, X. Y., Roy, S., Rao, A. P., & Shen, Z.-Q. 2005, ApJ, 634, 49

Anninos, P., Fragile, P. C., & Salmonson, J. D. 2005, ApJ, 635, 723

Baganoff, F. K., et al. 2001, Nature, 413, 45

Baganoff, F. K., et al. 2003, ApJ, 591, 891

Balbus, S. A., & Hawley, J. F. 1991, ApJ, 555, L83

Bélanger, G., Goldwurm, A., Melia, F., Ferrando, P., Grosso, N., Porquet, D., Warwick, R., & Yusel-Zadeh, F. 2005, ApJ, 635, 1095

Bower, G. C., Wright, M., Falcke, H., & Backer, D. C. 2001, ApJ, 555, 103

Bower, G. C., Falcke, H., Herrnstein, R. M., Zhao, J.-H., Goss, W. M., & Backer, D. C. 2004, Science, 304, 704

Broderick, A. E., & Loeb, A. 2005, MNRAS, 363, 353

De Villiers, J., & Hawley, J. 2003, ApJ, 589, 458

Eckart, A., et al. 2004, A&A, 427, 1

Eckart, A., et al. 2006, A&A, 450, 535

Falcke, H., & Markoff, S. 2000, AA, 362, 113

Falcke, H., Melia, F., & Agol, E. 2000a, ApJ, 528, L13

Falcke, H., Goss, W. M., Matsuo, H., Teuben, P., Zhao, J.-H., & Zylka, R. 1998, ApJ, 499, 731

Gammie, C. F., McKinney, J. C., & Tóth, G. 2003, ApJ, 589, 444

Genzel, R. et al. 2003, Nature, 425, 934

Ghez, A. M., et al. 2004, ApJ, 601, L159

Ghez, A. M., et al. 2005, ApJ, 635, 1087

Goldston, J. E., Quataert, E., & Igumenshchev, I. V. 2005, ApJ, 621, 785

Hawley, J. F., Beckwith, K., & Krolik, J. H. 2007, ApSS, 311, 117

Herrnstein, R. M., Zhao, J.-H., Bower, G. C., & Goss, W. M. 2004, ApJ, 127, 3399

Huang, L., Liu, S., Shen, Z. Q., Cai, M., Li, H., & Fryer, C. L. 2008, ApJ, 676, L119

Liu, S., & Melia, F. 2001, ApJ, 561, L77

Liu, S., & Melia, F. 2002, ApJ, 566, L77

Liu, S., Petrosian, V., Melia, F., & Fryer, C. L. 2006, ApJ, 648, 1020

Liu, S., Qian, L., Wu, X., Fryer, C. L., & Li, H. 2007, ApJ, 668, L127

Mahadevan, R., Narayan, R., & Yi, I. 1996, ApJ, 465, 327

Markoff, S., Falcke, H., Yuan, F., & Biermann, P. L. 2001, 379, L13

Melia, F. 1992, ApJ, 387, L25

Melia, F., Liu, S., & Coker, R. 2000, ApJ, 545, L117

Melia, F., Liu, S., & Coker, R. 2001, ApJ, 553, 146

Nobel, S. C., Gammie, C. F., McKinney, J. C., & Del Zanna, L. 2006, ApJ, 641, 626

Pacholczyk, A. G. 1970, Radio Astrophysics (San Francisco: Freeman), 86

Porquet, D., et al. 2003, A&A, 407, L17

Rybicki, G. B., & Lightman, A. P. 1979, Radiative Processes in Astrophysics (John Wiley & Sons), 160

Schödel, R., et al. 2002, Nature, 419, 694

Shen, Z.-Q., Lo, K. Y., Liang, M.-C., Ho, T. P., & Zhao, J.-H. 2005, Nature, 438, 62

Tagger, M., & Melia, F. 2006, ApJ, 636, L33

Xu, Y. et al. 2006, ApJ, 640, 319

Yuan, F., Quataert, E., & Narayan, R., 2003, 598, 301

Yusef-Zadeh, F., et al. 2006, ApJ, 644, 198

Yusef-Zadeh, F., Wardle, M., Cotton, W. D., Heinke, C. O., & Roberts, D. A. 2007, ApJ, 668, 47

Yusef-Zadeh, F., Wardle, M., Heinke, C., Dowell, C. D., Roberts, D., Baganoff, F. K., & Cotton, W. 2008, ApJ, 682, 361

Zhao, J. H., Herrnstein, R. M., Bower, G. C., Goss, W. M., & Liu, S. 2004, ApJ, 603, L85

# Needs-Based Agent Activity Generation by Means of Genetic Algorithms

Julia Scheevel, Rice University
Christof Teuscher, CCS-3, Mentor

**Abstract**

A recurring problem in agent-based modeling is the optimization of the daily activity schedule of an agent given a set of possible activities. This optimization question becomes even more pressing as populations of agents increase to large population scales and long durations (simulation of 1 million or more agents for one year). In a break from the purely utility-motivated approach that has been favored in the past, we propose to model an agent s behavior based on fulfilling a set of basic but competing needs: sleep, hunger, health, happiness, money, and social interaction. This needs-based approach is more closely derived from first principles of human behavior, and so may better resemble human decision making processes. We model each need as a numeric value that decreases with a function over time and increases by doing activities that fulfill each need. Using genetic algorithms, we optimize the parameters of each need function for a moving 24-hour simulation period; the solutions will be used to obtain the initial agent simulation parameters. With the addition of temporal constraints on the activities, we obtain daily schedule results in agreement with the utility-based schedules. Future work will incorporate further refinement of the activity selection mechanism through the implementation of reinforcement learning; future extension of the travel capabilities of the agents will also more closely model everyday human behavior. The ultimate goal will be to incorporate this technique into a large-scale agent simulation.

## 1 Introduction and Background

Human agent activity scheduling and schedule optimization have been long-standing problems in agent simulations [5], requiring a reliable method for creating reasonable and repeatable schedules. As simulations move to large scales, with populations of millions and timescales on the order of years, the need for an adaptable yet capable schedule generation scheme becomes inescapable. This problem has been addressed adequately in the past using utility maximization models, in which activities are assigned numeric values quantifying their usefulness to the agent [3, 5]; however, we propose a new model that uses a set of basic but competing needs to generate behavior; our model is especially promising due to

the anticipated ease of integration into large-scale, multi-faceted agent simulations. From this needs-based model, we may more easily incorporate agents behavior into extremely dynamic scenarios such as disaster situations.

The agents we model may be classically defined as computational systems that exist in a dynamic environment. The agents can perceive and interpret the environment, then uses the obtained information to act autonomously in order to accomplish a specified task or goal [8]. The agents that we model have an environment defined by a set of basic and competing needs (such as sleep and hunger); the agent knows the state of its needs at any given time during the simulation, then chooses activities to execute so as to address and fulfill those needs. Over the course of several generations , or iterations, through the genetic algorithm, the agent acts so as to construct the best schedule, defined by criteria that we will introduce later on.

However, given that agent simulations have already successfully produced daily schedules with utility-based schemes, this begs the question as to why the problem of activity scheduling should be readdressed with our needs-based approach. Instead of working from an abstract concept such as utility of activities , we begin with first principles of human behavior - at the most basic level, human behavior is motivated by the fulfillment of a set of essential needs, such as sleep, hunger, and maintenance of health. This bottom-up perspective of the needs-based approach provides a much more intuitive way of viewing the scheduling problem. In addition, it has the side benefit of providing a very easy way to modify agent behavior to reflect changes in the environment or responses to a specific scenario based on personality type; by changing the way the needs interact with each other, the behavior of the agent is fundamentally changed. We believe that this quality will allow our model to scale more easily than existing techniques. By implementing this scheduling approach in large-scale agent models, we hope to more realistically model the effects of external and internal factors on agent behavior.

## 2 Algorithm Architecture

Our algorithm is radically different from other activity generation schemes in that our final schedule is the result of emergent behavior of satisfying needs rather than the direct output of a utility maximization. To achieve this behavior, our model hinges on the time-dependence of so-called needs functions. Each of our basic needs (sleep, hunger, health, happiness, money, and social interaction) is modeled as a numeric value according to a monotonically decreasing function. The value of a need will continue to decrease over time until an activity is completed that addresses that need in some respect. The relative values of the needs are responsible for determining activity priority, and thus the schedule. Qualitatively then, the ideal schedule is one that maintains the needs values at consistently high levels throughout the scheduling period; in practice, the optimum schedule works to satisfy the highest priority need at every opportunity.

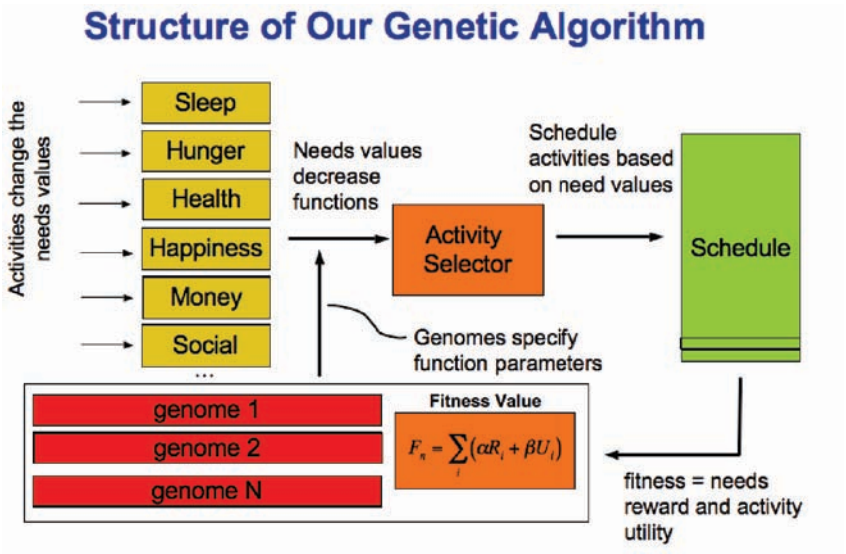



Figure 1: Summary of the genetic algorithm structure

An overview of our algorithm is shown in Figure 1. We evolve the parameters of the needs functions in the genetic algorithm, and thus affect the activity scheduling and schedule utility. Selection of the next activity is decided at the completion of the previous one. The choice of activity is governed by the priority of each need. Priority is determined by ordering the needs values in ascending order; the need with the lowest value is taken to be the most urgent. Thus, by altering function parameters, we can change the relative priorities of needs in a given individual solution. A subset of activities is selected that satisfy both the chosen need and the time constraints of the schedule. One of these activities is chosen with a certain probability according to how well is satisfies the need and added to the schedule. The increase to the needs function and the schedule utility is calculated, and the relevant needs functions are shifted in time such that now the value of the function at the stop time of the activity is equal to the newly recalculated needs value. Scheduling continues for an interval specified by the user. The "fitness" of each schedule is determined by its aggregate reward; schedules with higher fitness values are then "better." We select a number of top individuals for survival to the next generation, then generate a population of offspring schedules to be reinserted into the population and reevaluate the schedule utilities. More specific descriptions of the aspects and processes of the algorithm follow.

## 2.1 The Needs Functions

The needs functions are the central feature of our approach. Each need is modeled as a monotonically decreasing function in time; the lower the value of the function, the more

urgent the need is. This reflects the property of human needs that they become more and more urgent if they are not addressed in a timely manner, i.e. if a person does not sleep, he becomes exhausted, and if he does not eat, he starves. In order to address these needs, then, each activity that we schedule has a vector of weights that describes how well it addresses each need; this is discussed further in Section 2.2. By completing activities, the agent increases its needs values, i.e. addresses its basic needs.

We specifically choose the function

$$u(t) = \frac{u_{max}}{(1 + e^{\beta(t-\alpha)})^{\gamma}} \tag{1}$$

to represent each need. We scale $u(t)$ by a factor of 1000 to put it in the same order of magnitude as the utility values taken from [5]. The following three parameters for this function are represented by the algorithm s genome:

- $\beta$ - Determines the steepness of the downturn; defined on $[1 \times 10^{-6}, 0.0005]$ Low values produce shallow curves; high values produce steeper ones.
- $\alpha$ - Determines the timescale at which the function downturns sharply; defined on [0, 86400] (in seconds). Lower values give shorter timescales.
- $\gamma$ - Determines the sharpness of the curve; defined on [0.001,1]. Lower values give more gentle curves.

At the beginning of the algorithm, we randomly initialize the parameters within their specified bounds, and the values are evolved over the course of the simulation.

The quantity $u_{max}$ is calculated specifically for each set of $\{\beta\ \alpha\ \gamma\}$, such that $U(t = 0) \equiv 1000$. From equation 1, $U_{max}$ then is (omitting the scaling factor of 1000):

$$u_{max} = (1 + e^{-\beta\alpha})^{\gamma} \tag{2}$$

The advantage of this profile for the needs functions is the S-shaped curve is a realistic representation of most physical needs. After completing an activity that addresses a need, there is only a small marginal decrease in the needs value until the timescale (set by $\alpha$) for addressing that need has been reached. For example, a need function for sleep can be explicitly engineered such that the marginal decrease in the need value does not become significant until approximately 18 hours after waking. In addition, the allowable range for parameter variation permits a wide range of profiles, with approximately a step-function at one extreme and a constant function at the other. Virtually any monotonically decreasing profile can be explicitly created, if so desired. Example needs functions generated from a 24 hour simulation are show a wide variety of profiles in Figure 2.

As a side note, our formula for the needs functions is, in fact, the inverse of a formula used to define utility functions for activities in the utility-based paper [5]. Much as the

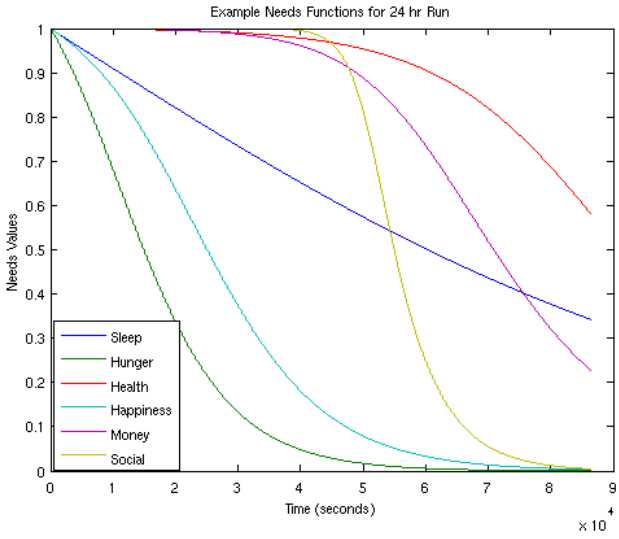



Figure 2: Example Needs Functions from a 24-hour Run

values of the needs are assigned numeric values, the activities are assigned a utility value dependent upon the duration of the activity, according to a monotonically increasing function. In this respect then, our needs functions are in effect the *disutility* functions of not completing activities.

As mentioned before, the three parameters for each need function are randomly initialized in a genome, the object to be operated upon by the genetic algorithm. In this case, the genome is a matrix containing the parameter vectors for each member in the candidate solution population; we use real representation for each component of the genome. In fact, two copies of this genome are initialized so that one may be a static reference during the current generation of solutions, and the other may be modified to reflect changing needs values, as will be explained in Section 2.3.

## 2.2 Activity Selection

As mentioned in the algorithm overview, activities are chosen probabilistically from the subset of activities that address a high-priority need. In our algorithm, we use a simple set of 12 activities, labeled with a numerical ID, that have an associated vector of weights specifying how well they address each of the six basic needs. Our set of activities, taken mostly from [1], are Idle, Sleep, Personal Care including breakfast (Pcare), Morning Work (Work1), Lunch, Afternoon Work (Work2), Dinner, Inside Leisure (Leis_I), Outside Leisure (Leis_O), Medical care (Mcare), Daily Shopping 1 (Dshop1), and Daily Shopping 2 (Dshop2). The activities, their vectors of needs weights, and their utility values are shown

| ID | Activity | Sleep | Hunger | Health | Happiness | Money | Social | Utility |
|---|---|---|---|---|---|---|---|---|
| 0 | Idle | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | Sleep | 0.9 | | 0.1 | | | | 200 |
| 2 | Pcare | 0.05 | 0.5 | 0.3 | -0.15 | | | 130 |
| 3 | Work1 | | | | | 0.8 | 0.07 | 325 |
| 4 | Lunch | 0.03 | 0.9 | 0.04 | 0.03 | | 0.01 | 135 |
| 5 | Work2 | | | | | 0.8 | 0.07 | 325 |
| 6 | Dinner | 0.03 | 0.9 | 0.04 | 0.03 | | 0.03 | 160 |
| 7 | Leis_I | | | | 0.8 | | | 60 |
| 8 | Leis_O | | | | 0.8 | | 0.4 | 60 |
| 9 | Mcare | 0.05 | | 0.3 | -0.15 | | | 100 |
| 10 | Dshop1 | | | | 0.3 | -0.3 | | 100 |
| 11 | Dshop2 | | | | 0.3 | -0.3 | | 100 |

Table 1: Activity Weights and Utilities

in Table 1.

In addition, each activity is subject to time constraints that specify earliest possible start, latest possible start, earliest possible end, and latest possible end (epS, lpS, epE, and lpE, respectively). These constraints, as well as the base duration of each activity, are listed in Table 2. The duration of each activity may vary by up to $\pm 15$ minutes from the base duration indicated; the amount of variation is randomly generated. We choose approximately fixed durations since this is the only way to schedule and execute activities in sequence. We do not employ a scheduling queue, where durations (and therefore, activity utilities) are variable and can be adjusted in the scheduling window. Such an implementation is left for future work.

Once the relative priorities of the needs have been determined from comparing the needs values at a specific time $t$, the activity selector first finds the highest-priority need and determines the subset of activities that have a positive weight for that need. This subset is further reduced to those activities that can be started at evaluation time $t$ and that can be completed at $(t + BaseDuration)$, according to their respective time constraints. The weights of the remaining activities for the selected need are compared; an activity is then selected probabilistically, with probability proportional to these weights. If the subset of activities is empty, the activity selector repeats this process for the second-most urgent need, and so on, through all six needs.

If no activity has been scheduled after examining all the needs, the selector schedules the activity Idle. Though the base duration for Idle is 0 seconds, it is subject to random duration variation for 0-15 minutes; this ensures that the simulation time progresses in order to end the scheduling deadlock.

| ID | Activity | Base Duration | epS | lpS | epE | lpE |
|---|---|---|---|---|---|---|
| 0 | Idle | 0 | 0 | 86400 | 0 | 86400 |
| 1 | Sleep | 27600 | 75600 | 3600 | 7200 | 32400 |
| 2 | Pcare | 5400 | 18000 | 84600 | 21600 | 86400 |
| 3 | Work1 | 12900 | 18000 | 37800 | 36000 | 43200 |
| 4 | Lunch | 6000 | 39600 | 46800 | 43200 | 50400 |
| 5 | Work2 | 18300 | 43200 | 54000 | 57600 | 79200 |
| 6 | Dinner | 7500 | 61200 | 72000 | 72000 | 79200 |
| 7 | Leis_I | 10200 | 25200 | 86400 | 75600 | 86400 |
| 8 | Leis_0 | 10200 | 25200 | 70200 | 27000 | 72000 |
| 9 | Mcare | 5700 | 28800 | 66600 | 30600 | 68400 |
| 10 | Dshop1 | 3900 | 25200 | 70200 | 27000 | 72000 |
| 11 | Dshop2 | 3900 | 25200 | 68400 | 28800 | 72000 |

Table 2: Time Constraints (in seconds)

### 2.3 Updating the Needs Functions

Once the next activity has been selected, the effect on the needs functions must be calculated. Completing an activity benefits (or in some cases, harms) each need for which the activity has a non-zero weight. In order to reflect the new needs values and correctly predict how those needs will behave in the future, it is necessary to realign the needs functions. For example, if the agent has a hunger needs value corresponding to when the function is in a period of high marginal decrease, then hunger is rapidly becoming more and more urgent. If the agent eats a meal in order to come away with a very high hunger value, then hunger will not need to be addressed immediately afterwards; the need has been satisfied. This could be reflected in the hunger function by shifting it along the time axis to the plateau region; a value in the plateau section of the function, where there is a low marginal decrease in the hunger value, now corresponds to the time of completion of the meal. In this plateau region, the need will most likely have low priority (due to the high need value), and it will change priority much more slowly for a period of time. This situation is illustrated in Figure 3 with the Original and High Gain curves at the example time $t = 40000$ seconds. If, on the other hand, the agent only ate a snack, the increase in the needs value would most likely be insufficient to shift the function out of the section of rapid decline; even though the needs value would increase and priority might decrease, the marginal decrease of the value would still be large, so it would take much less time for the hunger need to return to high priority status again. Comparison of the Original and Low Gain curves demonstrates demonstrates this scenario.

To achieve the appropriate shift, the quantity $\alpha$ is recalculated so that at $t_f = t + duration$, the function value is equal to the updated need value $nV$; the parameters $\beta$ and

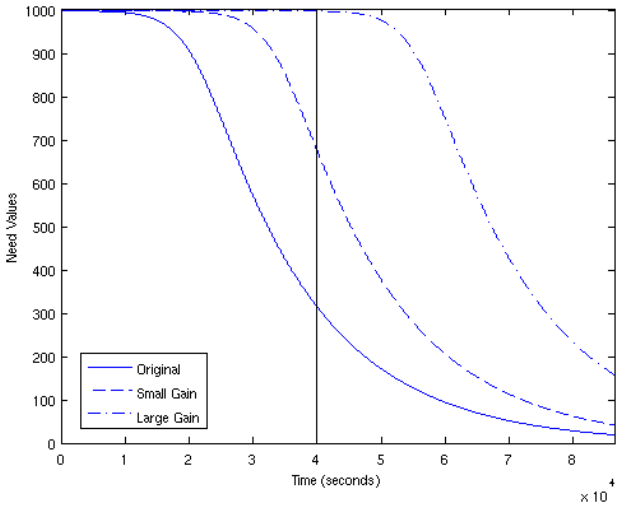



Figure 3: Shift of needs function

$\gamma$ remain static once initialized. The new $\alpha'$ is determined analytically by

$$\alpha' = t_f - \frac{1}{\beta} \ln \left( \left( \frac{u_{max}}{nV} \right)^{\frac{1}{\gamma}} - 1 \right) \tag{3}$$

Steps are taken to ensure that $\alpha' \in \Re$ and that it is a number that can be represented by the computer. This new time constant is then stored in the second copy of the parent chromosome, the one that was designed to be modified. This precludes the need for the algorithm to maintain some sort of memory storage object for the chromosome; the needs values are always evaluated at the current simulation time with no knowledge of previous timesteps.

Which functions are shifted during this realignment is dependent on the vector of weights that each activity has, as was shown in Table 1. Functions with non-zero elements are shifted to reflect the higher (or lower) needs value; the unaffected functions are left alone. Shifting continues for the duration of the schedule.

## 2.4 Determining the Fitness of Each Schedule

After the process of activity selection and realignment is complete, each member in the population of candidate solutions has a schedule for the specified length of time. To determine the fitness of each schedule (and consequently, which schedule is best ), we use a weighted sum of the needs rewards (determined by the weights shown in Table 1) and the activity s utility. There is one such sum for each activity in the schedule, so the aggregate

fitness value for the *nth* schedule is given by

$$\begin{aligned} F_n &= \sum_i \left(\alpha\left(\rho R_{n,i}\right) + \beta U_{n,i}\right) \\ (1 &= \alpha + \beta) \end{aligned} \tag{4}$$

The scalar $\rho$ is only present to put the rewards $\{R_{n,i}\}$ on the same scale as the utilities $\{U_{n,i}\}$. We obtain satisfactory results with $\alpha = 0\,5$, $\beta = 0\,5$, and $\rho = 250$. Effects of variation of these parameters will be discussed in Section 3.1.

### 2.5 Creation of Offspring and Reinsertion

After the fitness values for the parent generation have been determined, we probabilistically select 90% of the population according to the fitness values to help determine the offspring population. We then shuffle the genetic material with a multi-point crossover and mutate the components of each individual genome at a rate of $\frac{1}{18}$, or one over the total number of variables in each genome (6 needs functions with 3 parameters = 18 variables controlled by the genome). With this rate, we mutate one gene per individual per generation on average. The recombined and mutated chromosomes comprise the offspring genomes, which we then evaluate according to the fitness function above and compare to the parents fitness values. We choose to keep the top 10% of the parent population for survival to the next generation, and we replace the remainder with the entirety of the offspring population. We repeat this for 1000 generations, when the entire system has converged on an optimal schedule.

## 3 Schedule Results and Discussion

We obtain promising results from our needs-based scheme, both in short-term and long-term schedules. While short-term schedules demonstrate good agreement with schedules produced with existing schemes, the long-term schedules are promising because of their steady-state behavior. This behavior has the potential for application to a wide variety of simulated scenarios.

### 3.1 Effects of Parameter Tuning

Though we use a needs-based approach, we have not entirely done away with the concept of activity utility. As we found out, a mix of both the needs rewards and the activity utility is necessary to prevent the algorithm from incorrectly maximizing either quantity, as is described by Equation 2.4. We discovered that when schedule fitness was defined solely by the reward or by the utility, the algorithm would quickly discover which activities were

| Day | Activity | Begin Time | End Time | Need ID | Priority |
|---|---|---|---|---|---|
| 001 | Sleep | 00:00:00 | 07:32:47 | Sleep | 1 |
| 001 | Leis_O | 07:32:47 | 10:26:35 | Happiness | 1 |
| 001 | Pcare | 10:26:35 | 11:54:09 | Hunger | 1 |
| 001 | Lunch | 11:54:09 | 13:43:16 | Hunger | 1 |
| 001 | Leis_O | 13:43:16 | 16:39:42 | Happiness | 1 |
| 001 | Pcare | 16:39:42 | 17:59:51 | Hunger | 1 |
| 001 | Dinner | 17:59:51 | 19:51:12 | Hunger | 1 |
| 001 | Leis_I | 19:51:12 | 22:29:09 | Happiness | 1 |
| 001 | Pcare | 22:29:09 | 23:50:11 | Hunger | 1 |
| 001 | Sleep | 23:59:48 | 07:47:05 | Sleep | 1 |

Table 3: A 24-hr schedule, $\alpha = 1$

| Day | Activity | Begin Time | End Time | Need ID | Priority |
|---|---|---|---|---|---|
| 001 | Sleep | 00:00:00 | 07:36:48 | Sleep | 1 |
| 001 | Work1 | 07:36:48 | 11:10:04 | Money | 1 |
| 001 | Pcare | 11:10:04 | 12:35:39 | Sleep | 1 |
| 001 | Mcare | 12:35:39 | 13:55:48 | Sleep | 1 |
| 001 | Work2 | 13:55:48 | 18:51:06 | Money | 1 |
| 001 | Dinner | 18:51:06 | 20:45:07 | Sleep | 1 |
| 001 | Pcare | 20:45:37 | 22:06:57 | Sleep | 1 |
| 001 | Pcare | 22:06:57 | 23:35:34 | Sleep | 1 |
| 001 | Sleep | 23:35:34 | 07:29:11 | Sleep | 1 |

Table 4: A 24-hr schedule, $\beta = 1$

the most efficient (highest ratio of benefit to duration) and evolve the needs functions so that the needs that those activities addressed needed to be fulfilled more often than all the others. This would lead to schedules with a disproportionately high occurrence of only one or two activities. In the case where $\alpha = 1$ (all needs reward), the culprit activities tended to be shopping and leisure (Dshop1, Dshop2, Leis_I, Leis_O); for $\beta = 1$ (all utility), the overused activities were the two personal care activities (Pcare and Mcare). Examples of both types of schedules are included in Tables 3 and 4.

Although we developed a needs-motivated algorithm, the concept of utility is not entirely out of place when scheduling activities; after testing ($\alpha$ $\beta$) pairs of (0.25, 0.75), (0.5, 0.5), and (0.75, 0.25), we found that a half-and-half mixture of the needs reward and the activity utility was best suited to producing normal daily schedules. Values of $\alpha = 0\ 5$ and $\beta = 0\ 5$ are used for the remainder of the schedules.

| Day | Activity | Begin Time | End Time | Need ID | Priority |
|---|---|---|---|---|---|
| 001 | Sleep | 00:00:00 | 07:35:51 | Sleep | 1 |
| 001 | Work1 | 07:35:51 | 11:15:59 | Money | 1 |
| 001 | Lunch | 11:15:59 | 12:58:47 | Hunger | 1 |
| 001 | Dshop2 | 12:58:47 | 13:52:08 | Happiness | 1 |
| 001 | Work2 | 13:52:08 | 18:45:41 | Money | 1 |
| 001 | Dinner | 18:45:51 | 21:00:52 | Hunger | 1 |
| 001 | Leis_I | 21:00:52 | 23:59:48 | Happiness | 1 |
| 001 | Sleep | 23:59:48 | 07:47:05 | Sleep | 1 |

Table 5: A 24-hr needs-based schedule ($\alpha = 0\ 5\ \ \beta = 0\ 5$)

### 3.2 Short-Term Schedules

With the weighting of $\alpha = 0\ 5$, $\beta = 0\ 5$, and $\rho = 250$ as mentioned before, we obtain extremely reasonable 24-hour results. The agents follow a very normal sequence of activities; a 24-hour schedule is given in Table 5. In this schedule, the agent wakes up; goes to work in the morning; eats lunch and goes shopping during the lunch break; returns to work; goes home for dinner; perhaps reads before bed during inside leisure time; and finally sleeps again - all in all, a reasonably typical day. The column Need ID identifies which need was addressed by the activity being executed, and the Priority column indicates the priority ranking of the need in Need ID. Since every need identified in this schedule has a rank of 1, we confirm that our algorithm functions by identifying and addressing the most urgent needs of the agent.

#### 3.2.1 Comparison to Utility-Based Decision Schemes

In Table 6, we show a 24-hour schedule derived from a utility-based scheduling method in a 2001 paper [5]. The schedule is exactly the same as our 24-hour schedule, with the exception of an inserted personal care (Pcare) activity before work. Given this high level of agreement between the two methods, we are confident that the needs-based method will provide reasonable schedules, on par with those produced by utility-based methods. However, we anticipate that the needs-based method will have the added benefit of being able to interface more easily with other facets of in the agent model, such as personality and family units.

Unfortunately, there is no evidence in the literature of utility-based schedules generated for longer than 24 hours, so we have no basis of comparison for schedules of longer duration. However, we hope that given the agreement of the very short schedules, the longer ones may be evaluated on their reasonableness.

| Activity | Duration | Begin Time | End Time |
|---|---|---|---|
| Sleep | 448 | 23:22 | 6:50 |
| Pcare | 73 | 6:50 | 8:03 |
| Work1 | 207 | 8:03 | 11:30 |
| Lunch | 72 | 11:30 | 12:42 |
| Dshop | 48 | 12:42 | 13:30 |
| Work2 | 296 | 13:30 | 18:26 |
| Dinner | 132 | 18:26 | 20:38 |
| Leis_O | 164 | 20:38 | 23:22 |

Table 6: A 24-hr utility-based schedule

### 3.3 Longer-Term Schedules

For longer schedules, we observe also observe the emergence of reasonable schedule behavior; however, it appears on a different timescale than for the shorter schedules. Whereas with short schedules, the normal behavior is present immediately, for longer schedules (ones that are five days or longer), this behavior only begins to appear after at least two days of scheduling have been completed. However, once present, the scheduled activities (and more precisely, the order in which needs are addressed) appear in a very regular, repetitive cycle. This indicates to us that the schedule essentially reaches a steady state on longer timescales. Much like a driven physical oscillator exhibits a transient oscillation state that transitions into a steady state, our needs-driven schedules exhibit chaotic behavior that decays into normalcy. We show the transient and steady state of a single 10 day simulation in Tables 7 and 8, respectively.

In the transient state, one day s schedule bears no resemblance to another s, and activity times are completely different from day to day. More indicative of the instability than the activity sequence (since activities are chosen probabilistically, after all) is the order in which needs are addressed. Even then, the only constant is that sleep is the highest priority at the end of the day. Upon inspection of the steady state schedule, however, the situation is completely different. When the schedule is not idle, the needs repeat in exactly the same order, with the small exception of an inserted activity for happiness at the end of Day 9. Even the activities are nearly exactly the same, and the start and end time for each activity differ only by minutes from day to day. Examination of the needs values (Figure 4) corroborates this behavior. The time in between each major dark blue peak (for sleep) is almost exactly 24 hours, and the behavior in between them is very similar, with spacing nearly constant from day to day. The agent eats late in the day at the same time each day (green); he goes to work twice a day as expected (purple); and he does something for his happiness around the middle of the day (light blue). Not only is the order of relative priorities of needs constant during the steady state, but so are the needs

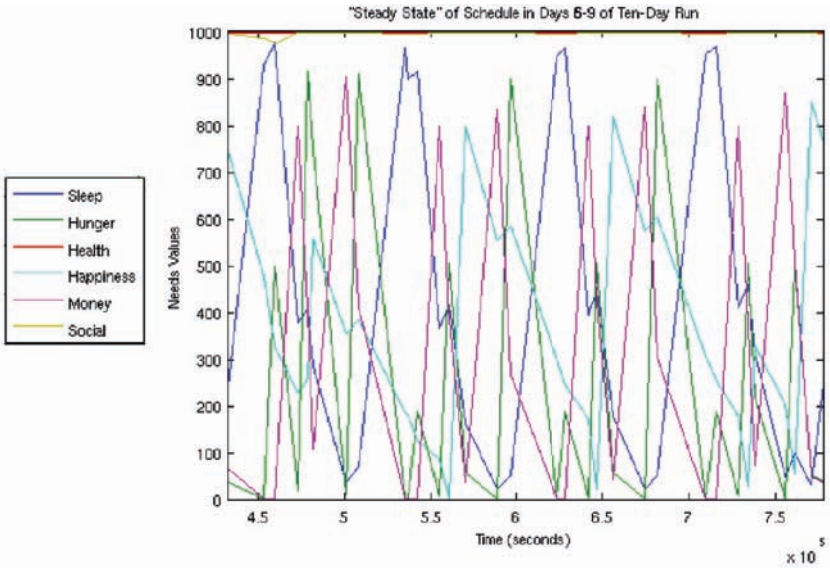



Figure 4: Needs values during steady state shown in Table 8

values at corresponding times of day, as one would expect from having nearly exactly the same daily activity schedule.

The existence of these two distinct states in the schedule implies the exciting possibility that in disaster scenarios, we may capture a population's initial reaction to the event, then later observe the agents adjusting to the new scenario. When confronted with a new situation or obstacle, humans tend to exhibit altered and non-repetitive behavior as they search for the best way to cope with the changes in their environment. For example, in the case of disaster, panic may ensue; in the far less dramatic case of a long-term road blockage, drivers will test out new routes. As the new state becomes the status quo, chaotic behavior dies down and a new "normal" behavior emerges. This behavior is exactly what we see with our longer needs-based schedules. We may take advantage of this accommodation of "perturbations" by defining any sort of event as temporarily altering the defining parameters of the needs functions (the set of $\beta, \alpha, \gamma$ ) so that we are altering behavior from its most fundamental level. The resulting behavior would then enter into a transient state (the initial reaction), then return to a steady state (the adjusted behavior).

# 4 Future Work and Conclusions

Though our algorithm gives satisfactory initial results, there are several aspects of the model that would benefit from refinement. At the moment, the activity selector uses extremely simplistic criteria to arrive at a scheduling decision; by choosing activities only

| Day | Activity | Begin Time | End Time | Need ID | Priority |
|---|---|---|---|---|---|
| 001 | Sleep | 00:00:00 | 07:29:52 | Sleep | 1 |
| 001 | Pcare | 07:29:52 | 08:59:01 | Hunger | 1 |
| 001 | Dshop1 | 08:59:01 | 9:49:55 | Happiness | 1 |
| 001 | Pcare | 09:49:55 | 11:07:59 | Hunger | 1 |
| 001 | Leis_O | 11:07:59 | 14:06:17 | Happiness | 2 |
| 001 | Work2 | 14:06:17 | 18:58:52 | Money | 1 |
| 001 | Dinner | 18:58:52 | 21:13:31 | Hunger | 1 |
| 001 | Sleep | 21:13:31 | 04:53:48 | Sleep | 1 |
| 002 | Idle | 04:53:48 | 05:08:26 | Health | 6 |
| 002 | Pcare | 05:08:26 | 06:35:42 | Hunger | 2 |
| 002 | Work1 | 06:25:42 | 10:14:08 | Money | 1 |
| 002 | Pcare | 10:14:08 | 11:52:41 | Hunger | 1 |
| 002 | Leis_O | 11:52:41 | 14:38:58 | Happiness | 1 |
| 002 | Work2 | 14:38:58 | 19:50:05 | Money | 1 |
| 002 | Dinner | 19:50:05 | 21:45:56 | Hunger | 1 |
| 002 | Sleep | 21:45:56 | 05:20:57 | Sleep | 1 |
| 003 | Pcare | 05:20:57 | 06:43:12 | Hunger | 2 |
| 003 | Work1 | 06:43:12 | 10:04:46 | Money | 1 |
| 003 | Pcare | 10:04:46 | 11:34:57 | Hunger | 1 |
| 003 | Dshop2 | 11:34:57 | 12:33:34 | Happiness | 1 |
| 003 | Work2 | 12:33:34 | 17:36:06 | Money | 1 |
| 003 | Pcare | 17:36:06 | 19:02:33 | Hunger | 1 |
| 003 | Leis_I | 19:02:33 | 21:54:12 | Happiness | 1 |
| 003 | Sleep | 21:54:12 | 05:34:55 | Sleep | 1 |

Table 7: Transient state in a 10 day schedule

| Day | Activity | Begin Time | End Time | Need ID | Priority |
|---|---|---|---|---|---|
| 006 | Pcare | 05:53:02 | 07:37:37 | Hunger | 1 |
| 006 | Work1 | 07:37:37 | 11:25:21 | Money | 1 |
| 006 | Lunch | 11:25:21 | 25:50:50 | Hunger | 1 |
| 006 | Dshop1 | 12:50:50 | 13:55:39 | Happiness | 1 |
| 006 | Work2 | 13:55:39 | 19:08:18 | Money | 1 |
| 006 | Dinner | 19:08:18 | 21:12:39 | Hunger | 1 |
| 006 | Sleep | 21:12:39 | 04:41:53 | Sleep | 1 |
| 007 | Idle | 04:41:53 | 04:44:08 | Health | 6 |
| 007 | Idle | 04:44:08 | 04:53:10 | Health | 6 |
| 007 | Idle | 04:53:10 | 05:06:23 | Health | 6 |
| 007 | Pcare | 05:06:23 | 06:37:24 | Hunger | 2 |
| 007 | Work1 | 06:37:24 | 10:08:04 | Money | 1 |
| 007 | Pcare | 10:08:04 | 11:37:43 | Hunger | 1 |
| 007 | Leis_O | 11:37:43 | 14:18:07 | Happiness | 1 |
| 007 | Work2 | 14:18:07 | 19:28:54 | Money | 1 |
| 007 | Dinner | 19:28:54 | 21:38:38 | Hunger | 1 |
| 007 | Sleep | 21:38:38 | 05:15:38 | Sleep | 1 |
| 008 | Pcare | 05:15:38 | 06:37:06 | Hunger | 2 |
| 008 | Work1 | 06:37:06 | 10:12:52 | Money | 1 |
| 008 | Pcare | 10:12:52 | 11:34:10 | Hunger | 1 |
| 008 | Leis_O | 11:34:10 | 14:12:49 | Happiness | 1 |
| 008 | Work2 | 14:12:49 | 19:22:10 | Money | 1 |
| 008 | Dinner | 19:22:10 | 21:34:10 | Hunger | 1 |
| 008 | Sleep | 21:24:10 | 05:11:33 | Sleep | 1 |
| 009 | Pcare | 05:11:33 | 06:48:42 | Hunger | 2 |
| 009 | Work1 | 06:48:42 | 10:18:08 | Money | 1 |
| 009 | Pcare | 10:18:08 | 11:15:00 | Hunger | 1 |
| 009 | Dshop1 | 11:51:00 | 13:02:57 | Happiness | 1 |
| 009 | Work2 | 13:02:57 | 17:58:57 | Money | 1 |
| 009 | Pcare | 17:58:57 | 19:27:20 | Hunger | 1 |
| 009 | Leis_I | 19:27:20 | 22:13:30 | Happiness | 1 |
| 009 | Sleep | 22:13:30 | 06:05:31 | Sleep | 1 |

Table 8: Steady state in a 10 day schedule

on the basis of which need is most urgent and can be addressed at a specific moment, we effectively implement a winner-takes-all strategy. Compromise schedules, i.e. schedules that forgo a more urgent need in favor of a less urgent one because the end benefit is higher, are therefore not possible with the current set-up, and thus the behavior of planning for the future is not currently a feature of our agent model. We plan to refine the activity selector with the inclusion of reinforcement learning, where the agent learns to recognize recurrent scenarios and their possible outcome(s) through a system of rewards and penalties. In a related but separate scheme, we could borrow a strategy from needs-based animat (animal agent) modeling where lower priority needs are able to make recommendations to the highest priority need about which activity to choose [2]. Both plans would teach the activity selector to consider future consequences of actions.

In addition, we also plan to incorporate agent-agent interactions, or effectively, communication. For humans, the actions of a close network of neighbors (i.e. family and friends) can dramatically impact behavior; for example, in a household setting, only one agent may have to complete an activity in order for the need to be satisfied for all members of the household, as would be the case for grocery shopping. Evolving the behavior of agents within small groups will help us more realistically model responses within large-scale simulations. In addition, we are currently working to assign personality types to agents to affect their responses to events and their interactions with other agents.

Also needed for modeling truly realistic behavior will be granting the agents travel abilities. The cost of travel between activities can counteract the utility of completing an activity, so generated daily activity schedules cannot be entirely complete until this has been taken into consideration. The inclusion of both transportation and communication will be essential for full integration into infrastructure simulations.

By treating agent activity schedules as the result of emergent behavior due to a complex system of the interactions of needs functions, we create what we anticipate will be a much more flexible method for generating activity schedules than the utility-based method that has been previously favored. Through use of genetic algorithms, our bottom-up architecture reliably produces results that agree well with those of existing techniques. However, our novel approach will be able to more easily integrate and reflect both internal and external influences on agent behavior than existing approaches by allowing these influences to change the root cause for behavior (the needs functions) rather than the outward signs of it (the activities).

## 5 Acknowledgments

I would like to thank Christof Teuscher for his invaluable help and encouragement this summer. Thanks also to Stephan Eidenbenz and the other members of CCS-3, as well as to the National Infrastructure Simulation and Analysis Center at Sandia National Laboratory. Special thanks to the Los Alamos Summer School, a joint program between Los Alamos

National Laboratory and the University of New Mexico, and to its organizers, especially James Colgan and Norman Magee, for organizing this opportunity and for helping us with life in general in Los Alamos.

For more information on our work, please see our project website:
http://www-int.c3.lanl.gov/twiki/bin/view/Main/ActivityGeneration (LANL Only)

## References

[1] Arentze, T., H. Timmermans, 2003, Modeling learning and adaptation processes in activity-travel choice, Transportation **30**, 37-62.

[2] Blumberg, B., 1994, Action selection in Hamsterdam: Lessons from ethology, *Proceedings of the 3rd International Conference on the Simulation of Adaptive Behavior*, MIT Press, Brighton, MA.

[3] Charypar, D., K. Nagel, 2004, Generating complete all-day activity plans with genetic algorithms, Transportation **32**(4), 369-397.

[4] Chipperfield, A., P. Fleming, H. Pohlheim, C. Fonseca, **19**—, Genetic Algorithm Toolbox User s Guide, v1.2, http://www.shef.ac.uk/content/1/c6/03/35/06/manual.pdf

[5] Joh, C., T. Arentze, H. Timmermans, 2001, Understanding activity scheduling and rescheduling behaviour: Theory and numerical illustration, GeoJournal **53**, 359-371.

[6] Kebair, F., F. Serin, C. Bertelle, 2008, Agent-Based Perception of an Environment in an Emergency Situation, arXiv:0804.0558v1.

[7] Li, J., U. Aickelin, 2003, Explicit Learning: an Effort towards Human Scheduling Algorithms, *The 1st Multidisciplinary International Conference on Scheduling: Theory and Applications MISTA*, 240-241.

[8] Maes, P., 1995, Artificial Life Meets Entertainment: Lifelike Autonomous Agents, Communications of the ACM **38**(11), 108-114.

[9] Marchal, F., K. Nagel, 2005, Computation of location choice of secondary activities in transportation models with cooperative agents. *Applications of Agent Technology in Traffic and Transportation*, M. (Walliser et al., Eds.) Birkhauser Basel, 153-164

[10] Pribyl, O., K. Goulias, 2005, Simulation of Daily Activity Patterns. *Progress in Activity-Based Analysis*, (H Timmermans, Ed.) Elsevier Ltd, 43-65.

[11] Yang, Z., K. Tang, X. Yao, 2008, Large scale evolutionary optimization using cooperative coevolution, Information Sciences **178**, 2985-2999.

# THEORETICAL SHOCK BREAKOUT MODELING FROM TYPE-II SUPERNOVA SIMULATIONS

David Stalla

*Dept. of Physics, Lewis University, Romeoville, IL 60466*

*T−06, Theoretical Astrophysics, Los Alamos National Laboratory*

Christopher Fryer

*CCS−2, Computational Physics and Method, Los Alamos National Laboratory*

## Abstract

Recent observations have opened a new window into a vital component of type-II (core collapse) supernovae, the shock breakout. As the inner stellar material rebound off of the dense core remnant, photons become entangled in the dense outer envelope. The particles proceed through a decreasing density gradient, driving the envelope outward, and eventually breaking free and accelerating through the surface. While still a highly disputed theory, shock breakout is becoming a topic of great interest as new technology is allowing us to observe it directly. Our aim is to run simulations using Los Alamos National Laboratory's (LANL) computational facilities in order to model shock breakout and attempt to note its effects.

## I. STAR EVOLUTION AND CORE COLLAPSE

A star is formed as gravitational instability collapses a molecular cloud comprised mainly of Hydrogen. The decrease in gravitational potential energy yields an increase in temperature, contributing to the formation of the newborn stars core. A stars core is an immense powerhouse of energy. Our sun, rather small and cool relative to other stars, radiates over $3.6 \times 10^{26}$ Watts of power; by comparison, the United States consumes roughly $3.23 \times 10^{12}$ Watts according to Elert(2001). This power is achieved through nuclear fusion, where the immense pressure and temperature encourage the fusion of the Hydrogen into Helium, following the overall reaction:

$$4^1H \rightarrow^4 He + 2e^+ + 2\gamma + 2\nu_e \tag{1}$$

This process dominates a stars existence, lasting for about 90% of its lifetime. For this reason, it is known as the stars main sequence. Stars with a mass under around 9 $M_\odot$ will not progress much further than the main sequence, and will eventually burn out into a white dwarf. Larger stars, however, undergo a much more fantastic death.

An inherent property of stellar evolution is the tremendous outward pressure generated by the thermal energy released in its series of nuclear reactions. It is this pressure that allows the core to remain in hydrostatic equilibrium against the star's immense gravity. Eventually, however, the star will exhaust its supply of Hydrogen. Without fuel, the reactions cease to propagate, causing the core to collapse upon itself. This creates sufficient pressure and temperature to ignite the the stored Helium to fuse into Carbon. This new reaction halts the collapse, creating a modified core composed of fusing Helium surrounded by a layer of Hydrogen. This process, where a core collapse caused by fuel deficiency is interrupted by the ignition of heavier elements to fuse, is repeated several times. The number of different reactions is dependant on the size; smaller stars, upon collapse, will reach a point where the pressure is insufficient to drive the next reaction, and the star dies. Stars about 0.4 to 1 $M_\odot$, including ours, will progress past Hydrogen burning onto Helium to Carbon burning, but no further. In somewhat larger stars, 5 to 9 $M_\odot$, the pressure is enough to begin fusing Carbon into Oxygen.

In each case, the collapse causes a much higher pressure and temperature than previously seen which are in turn offset by the energy released through the fusion of much more massive nuclei. This creates a multi-layer core that, in the case of the biggest stars (¿9 $M_\odot$),

is comprised of Hydrogen, Helium, Carbon, Neon, Oxygen, Silicon, and Iron. Each new reaction, while more energetic than the last, is also much shorter lived. For example, while the Hydrogen fusion process may last for billions of years, the Silicon reaction lasts for only a matter of days. The vital component of each reaction is the binding energy, the energy that holds the nuclei together. At the smaller elements, the binding energy is larger than the gravitational, halting the collapse and reigniting the core. With the fusion of more massive atoms, however, the binding energy decreases. Ultimately, the binding energy of Iron as it fuses into Nickel yields less energy than needed to stave off the gravitational collapse of the core. At this point, only the degeneracy pressure of the electrons remains to prevent the total collapse. However, with a star this massive, the core soon exceeds the Chandrasekhar limit and may no longer sustain itself, resulting in possibly the most cataclysmic event in the universe: a type-II supernova.

## II. SUPERNOVA AND SHOCK BREAKOUT

The core falls in on itself at close to a quarter of the speed of light. The density at the center rises to immense proportions, allowing protons and electrons to combine via inverse Beta decay, yielding neutrons and electron neutrinos. The neutrinos, uninhibited by the stellar matter, radiate away from the core, transporting away energy which hastens the collapse. The neutron-neutron repulsion at the center of the core, with a density comparable to that of the nucleus, becomes great enough to halt the collapse. However, the surrounding matter continues to fall. This material rebounds off of the cores remnant and is ejected outward. This produces a shockwave that ejects the contained heavy elements, a process often sufficient to stall the explosion within the outer core. Meanwhile, the center continues to merge protons and electrons, and the emitted neutrinos energize the shockwave into motion. In many stars, the core remnant forms the basis for what is to become a neutron star, while in extraordinarily large stars (greater than 20 $M_{\odot}$), the core is unable to sustain itself and collapses into a black hole.

As with all of the universes processes, the star's death heralds new life. In many cases, as it is believed to be for our Sun, the shockwave acts as the impetus for the collapse of giant molecular clouds in star forming regions. Supernovae are responsible for a majority of the distribution of the principle heavy elements as the core's stellar matter is ejected outward in

a massive shockwave. Before the shockwave can escape, however, it faces the task of driving through the star's dense outer envelope. Rather than escaping directly through star, the photons of the shock are deflected following a Brownian-type motion, essentially traveling at a radial velocity that is much slower than their true velocity. While initially a steep density gradient, the photons move from the diffusion regime to the free-streaming regime and their radial velocity once again approaches the speed of light. Once the radiative diffusion time becomes shorter than the time for the shock to emerge (breakout) from the envelope, a soft X-ray burst emerges. This breakout precursor to the ejection of a majority of the stellar material has been long since predicted, but only recently observed. As the shock propagates outward, it heats and drives the envelope before dissipating out the surface. Shock breakout may take anywhere from few minutes to a few days, dependent upon a number of factors, including the size of the star, density of the outer envelope, and any surrounding stellar wind.

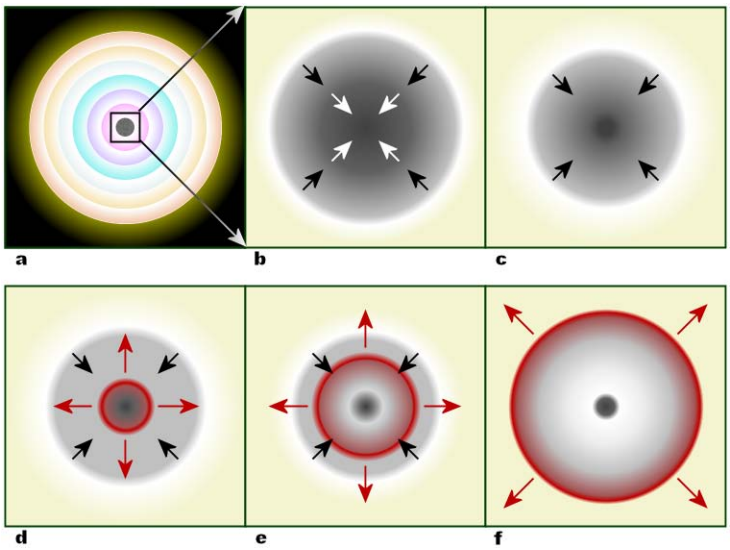



FIG. 1: (a) Nuclear fusion of heavier and heavier elements creates a multi-layered core, whereupon (b) the Iron core collapses. At the very center, pressure compresses the core into an immensely dense remnant. (c) The surrounding matter continues to fall inward, (d) rebounding off of the core remnant and creating an outward bound shock (red). (e) As the shock encounters the dense outer envelope, it may begin to stall. Re-invigorated by the emitted neutrinos from the remnant, (f) the shock blasts outward through the envelope into space.

## III. SIMULATION DESCRIPTION

Successfully modeling supernovae and their associated shockwave breakouts is a topic of great interest in the astrophysical world as of late. Theoreticians such as Stirling Colgate have developed models as early as 1972, but observations have been impossible to due the nature of the mechanism and the lack of suitable technology. However, recent advances have allowed us to detect the earliest signs of impending supernova, gaining us insight as never before. For the first time, researchers like Gezari et al.(2008) , Quimby et al.(2007), and Schawinski et al.(2008) are able to provide observable data. A quick search turns up 23 papers on this topic in this year alone, as of the date of this publication. For this project, we used LANL's specialized radiation hydrodynamic code RAGE (Radiation Adaptive Grid Eulerian) to run spherical one-dimensional supernova simulations, studying the output in the hopes of comparing this theoretical data to the observed. The code uses an input deck outlining the initial conditions in each of a preset amount of cells in the grid, listing the distance from the solar center, the velocity and density of the escaping material, as well as the energy and radiation energy contained within. My particular input deck was that of a 15 solar mass star as drafted by Weaver and Woosley(1993). This code is exceptionally computationally intensive and therefore makes use of LANL's extensive computing capabilities, specifically the FLASH supercomputer. The simulation runs in timesteps of a few tenths of a second, outputting to dump files that would compile data in increments of 10,000 timesteps. These dump files may then be graphed in order to visualize the simulation. We used the plotting program xShow for of its exceptional compatibility with RAGE and its ability to graph consecutive dumpfiles in contour and line plot animations.

## IV. SIMULATION DATA

The initial grid was set to 10,000 cells at $4 \times 10^7$ m each; an enormous simulation space, but one the explosion would exceed nonetheless. As it turned out, the grid was surpassed at about the 37th dump file, corresponding to t = 103000 s, a little over a day in the supernova. At this point, I would need to construct new a new input deck based on the final conditions of the simulation. The dump files are a warehouse of scrambled information; the cells are interspersed with parent and daughter cells, and not in any easily recognizable sequence. In

addition, the initial input deck was written to account for a larger grid than was used in the first set of simulations. Any cells outside of the initial grid size from the first input deck would need to be added to the new input deck. To create an input deck of this magnitude by hand would take weeks. Instead, I wrote a rather simple program to generate an input deck outlined by the following algorithm:

```
Read data from dump file
     Determine if cell has next smallest radius
          Determine if cell has no daughter cells
               Number cell and output radius, velocity, density, energy, and
               radiation energy.
     Repeat for all cells in dump file
Read data from input deck
     Find first cell with radius greater than previous grid size
          Number cell and output radius, velocity, density, energy, and
          radiation energy.
     Repeat for all cells in input deck
Read material information from dump file
     Determine if cell has next smallest radius
          Determine if cell has no daughter cells
               Number cell and output percentage of each element
     Repeat for all cells in dump file
Read material information from input deck
     Find first cell with radius greater than previous grid size
          Number cell and output percentage of each element
     Repeat for all cells in input deck
```

This code was put to use when, at t = 108000 s, the shockwave had exceeded the initial boundaries. A new input deck was generated from t = 103000 s and the boundary conditions enlarged by a factor of one hundred. After this first remap, a graph was constructed to ensure the code had worked properly. To our dismay, there was an anomaly in the data (see Figure 2). The left part of the graph shows the data from the simulation with the rightmost part, just before the boundary, being solar wind effects added onto the outside of the star. While

originally attributed to the remap code, it became apparent through graphs of the dumpfiles themselves (see Figure 3) that the cause lay in a discontinuity where the two parts meet. To offset this, I chose an earlier timestep, roughly 56000 s, (see Figure 4) where the simulation data had not yet reached the wind effects, and replaced the current wind effects with the original set from the initial input deck. This resulted in a much smoother and continuous data set (see Figure 5).

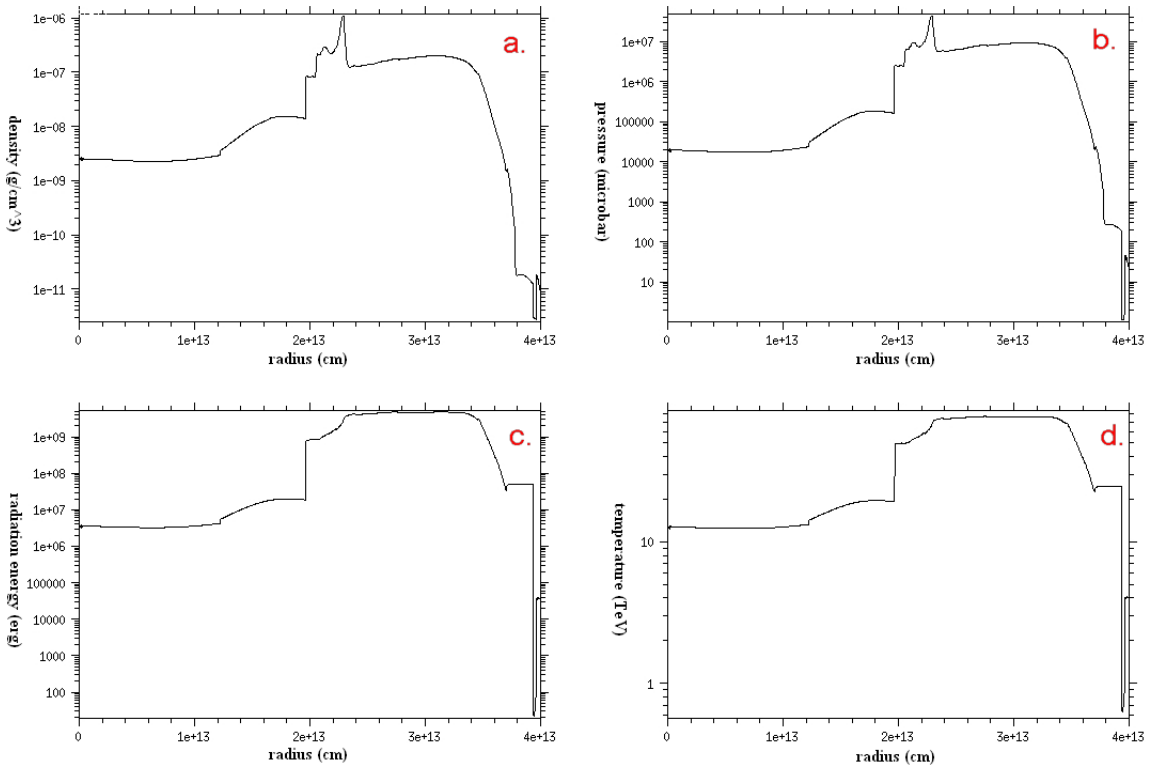



FIG. 2: The graphs of log density vs. radius (a), log pressure vs. radius (b), log radiation energy vs. radius (c), and log temperature vs. radius (d), plotted from the simulation run from the newly generated input deck, all feature a sharp discontinuity shortly before the edge of the star.

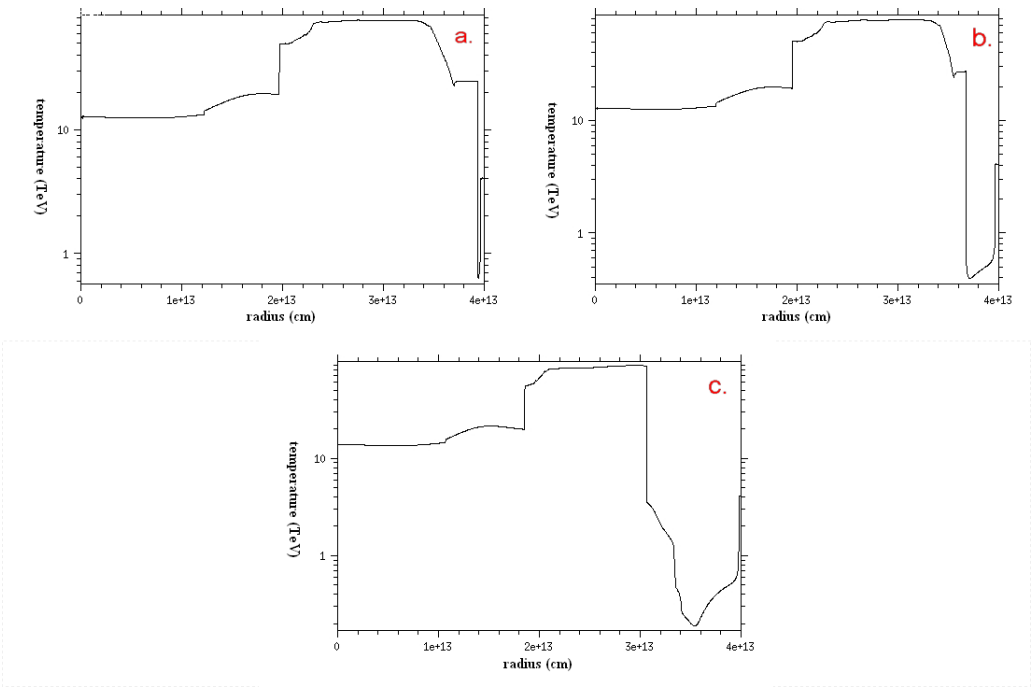



FIG. 3: The log temperature vs. radius plots of t=103000 s (a), 94400 s (b), and even 74000 s (c) show the discontinuity persists well before remapping.

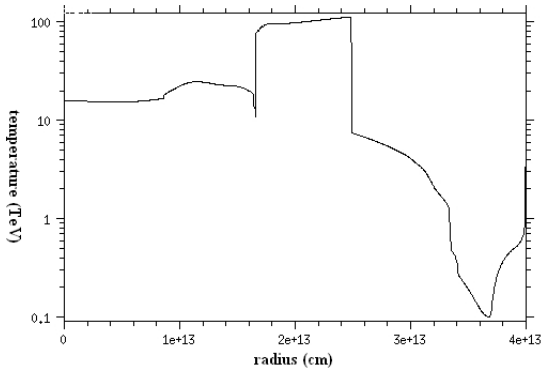



FIG. 4: The log temperature vs. radius plot of t = 56000 s shows a shockwave that has not yet reached the solar wind effects and suffered the subsequent distortion.

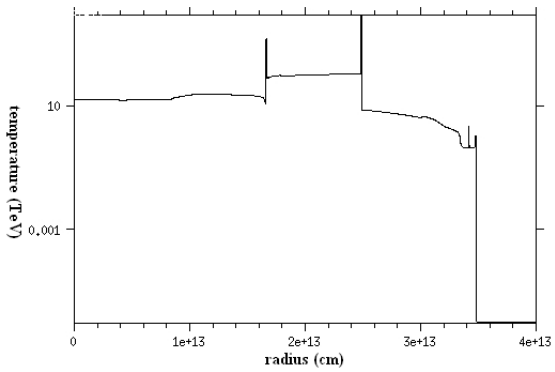



FIG. 5: The resultant log temperature vs. radius plot with the solar wind effects readjusted.

The original intent of the project was to run the simulation out to about a year of the star's time, a process that would require several re-mappings of the grid, and study the various light curves emitted. As a precursor to the project, there was a significant learning curve in the usage and syntax for the systems I used, such as the queue network for the supercomputer. Time constraints forced us to shift focus to the study of shock breakout, a process that occurs much earlier in life of a supernova. While the true shock breakout doesn't occur in our simulation until around t = 75000 s (roughly 21 hours), the effects leading up to it manifest much earlier.

## V. SHOCK BREAKOUT EFFECTS

The effects of shock breakout are most evident in graphs of speed vs. radius. Figure 6 shows the speed of the supernova in the earlier stages of development. The shock initially moves with at very high velocity, but is immediately slowed by the star's dense material. As the shock (right spike) continues to be slowed outward, it begins to drive the stellar material (left spike) outward. Of note on the far right is the solar wind which is moving at a constant velocity outside of the stellar envelope.

Figure 7 shows the shock encountering the less dense outer envelope. The inner material continues to move steadily outward, while the shock accelerates immensely. The high rate of acceleration is notable due to the fact that the graphs show such a large spike in consecutive dumpfiles.

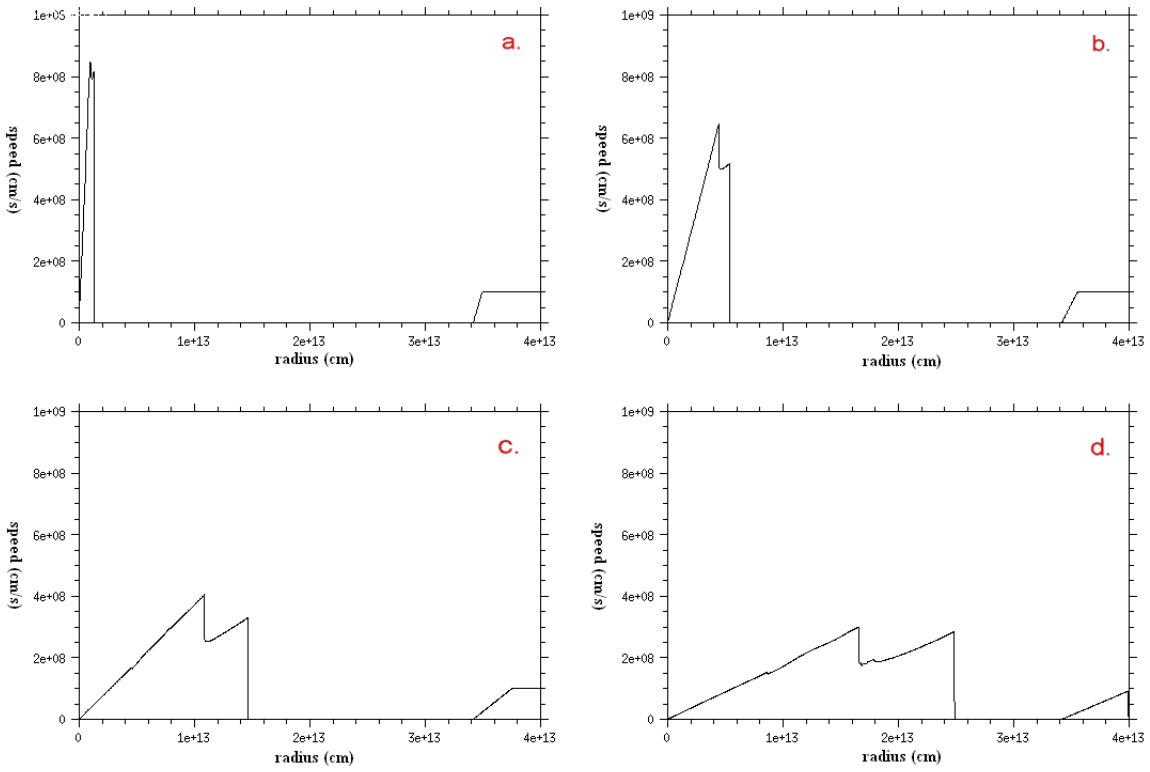



FIG. 6: Speed vs. radius plots for t = 1020 s (a), 6790 s (b), 26700 s (c), and 56000 s (d).

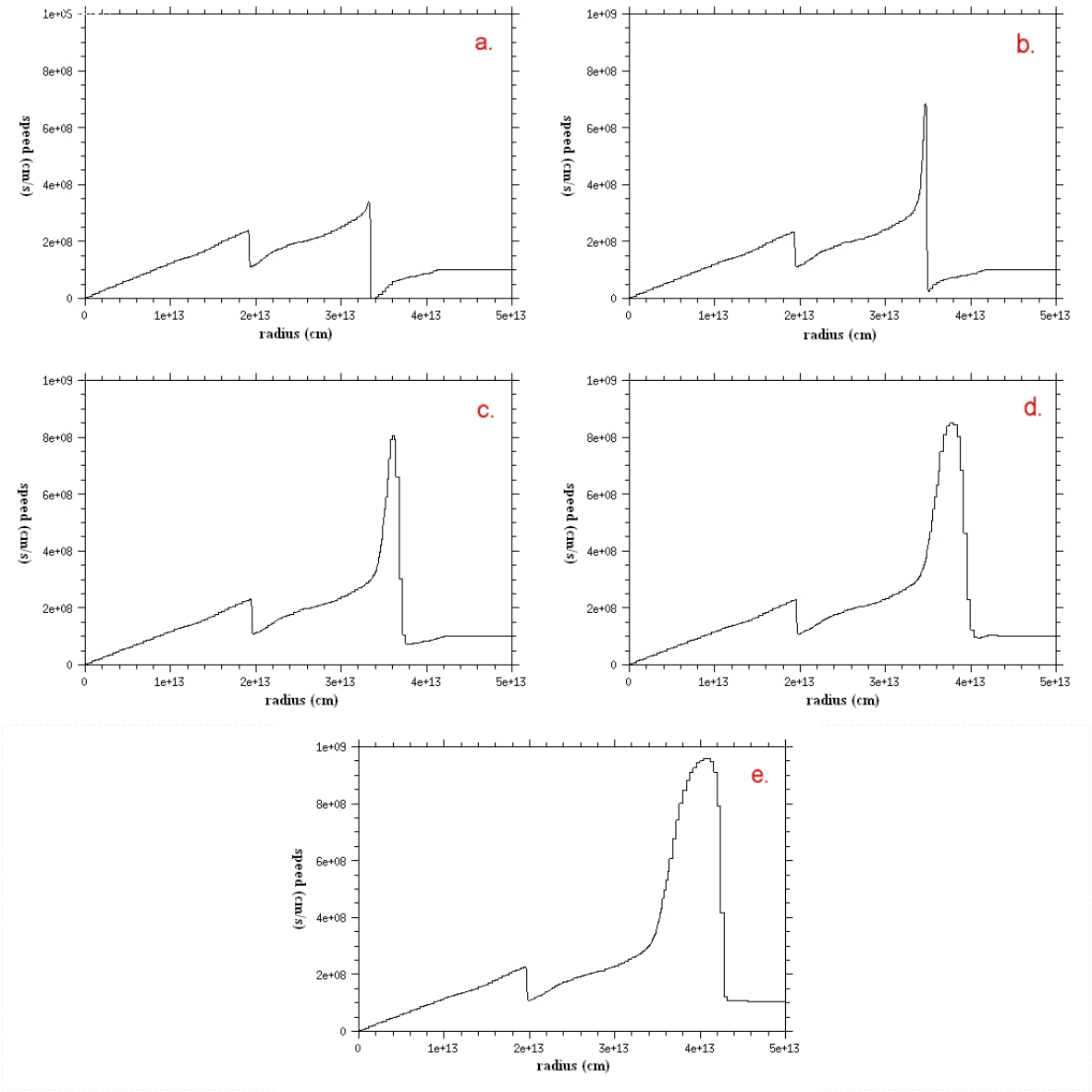



FIG. 7: Speed vs. radius plots for t = 59400 s (a), 62900 s (b), 66500 s (c), 70200 s (d), and 74000 s (e).

Graphs of density vs. radius also yield interesting results The initial density distribution (Figure 8) behaves as one would expect. As the shock rebounds, a majority of the material is falling inward, creating a gradient very dense towards the center and decreasing rapidly outward. However, as early as the second dumpfile, the shock (rightmost spike) acts somewhat as a mixing agent, dispersing the highly dense material down the gradient. The shockwave propagates outward, followed by a tail of highly dense stellar matter that follows

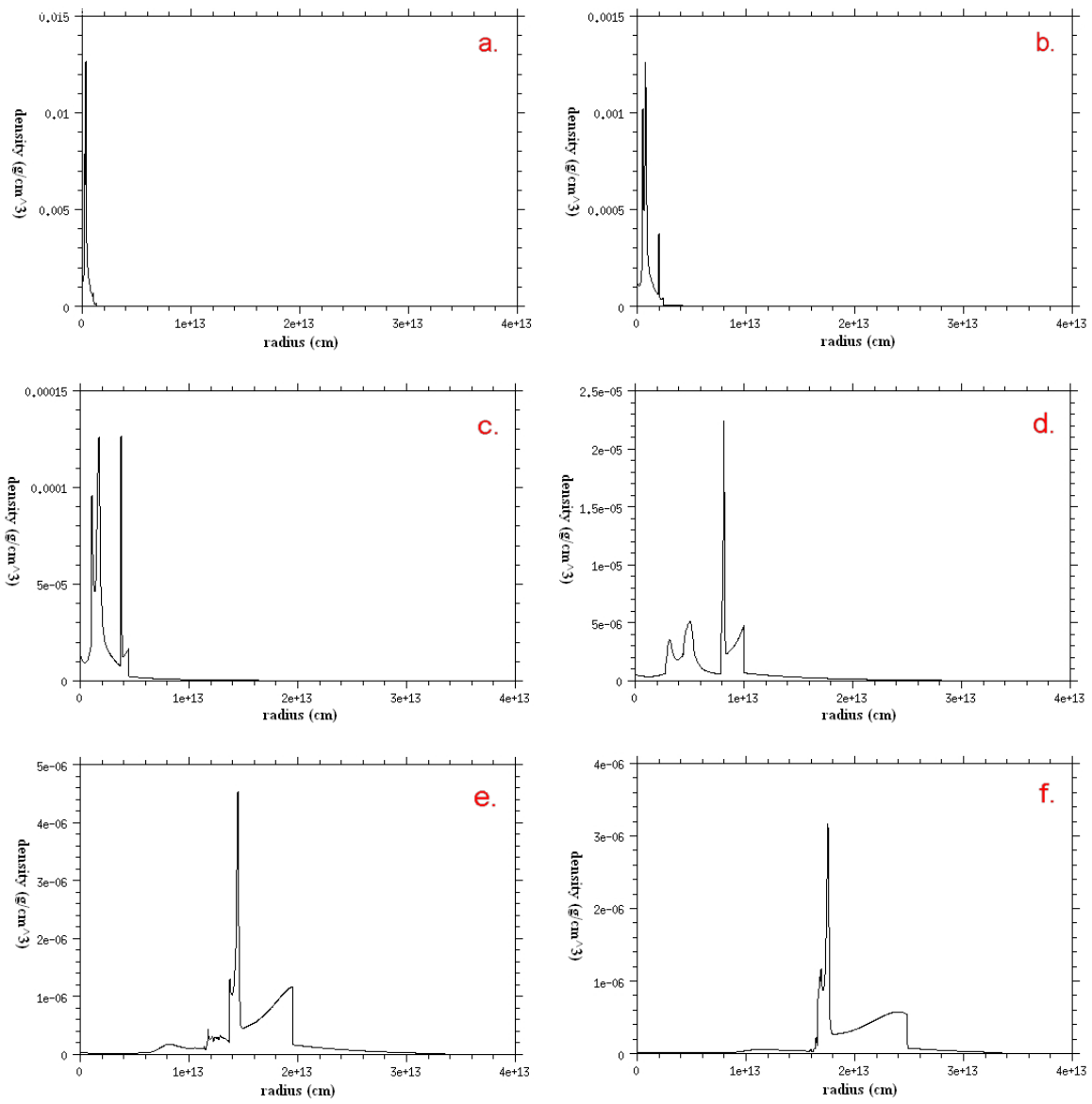



FIG. 8: Density vs. radius plots for t = 1020 s (a), 2390 s (b), 5300 s (c), 15600 s (d), 5600 s (e), and 40300 s (f).

out of the star and is ejected into space.

Unlike the graphs of speed vs. radius which increase in intensity later in the explosion, the density vs. radius graphs feature an opposite effect. As the shockwave continues to mix the stellar material, the graph becomes less dynamic (Figure 9). This is due to the fact that the shock passes through a decreasing gradient, facing less resistance. This creates a more stable environment, less prone to the rapid and chaotic changes as shown in the earlier

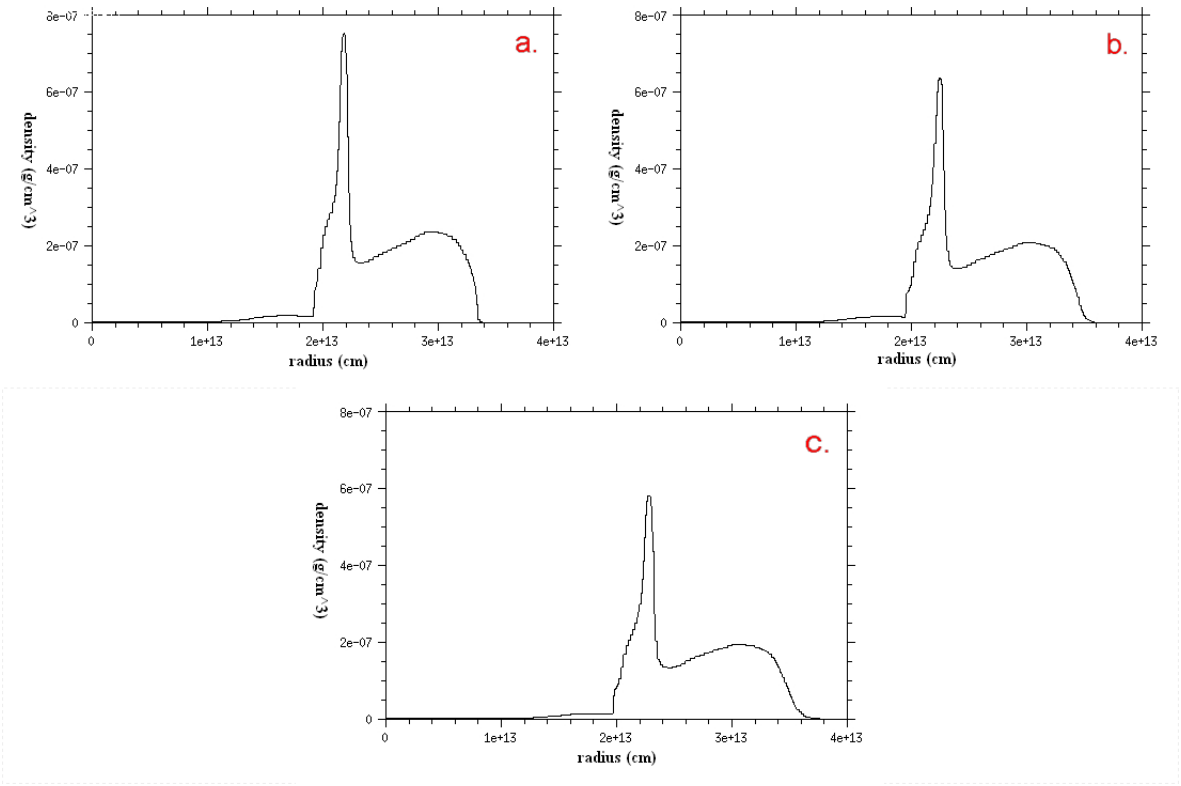



FIG. 9: Density vs radius plots for t = 59400 s (a), 66500 s (b), and 74000 s (c).

graphs. There is very little change, save for a slight decrease as the material continues to disperse.

As noted earlier, the shock travels through the dense stellar matter and is hindered as the photons are scattered. This creates a dispersion of high radiation energy. However, further out, there is much less interference, yielding a much lower amount of radiation (Figure 10). Many other supernovae simulations feature a much less diverse radiation pattern. In larger stars such as the one I used, much of the nickel in the core remains stationary in the center, accounting for a significant source of radiation at very low radius throughout.

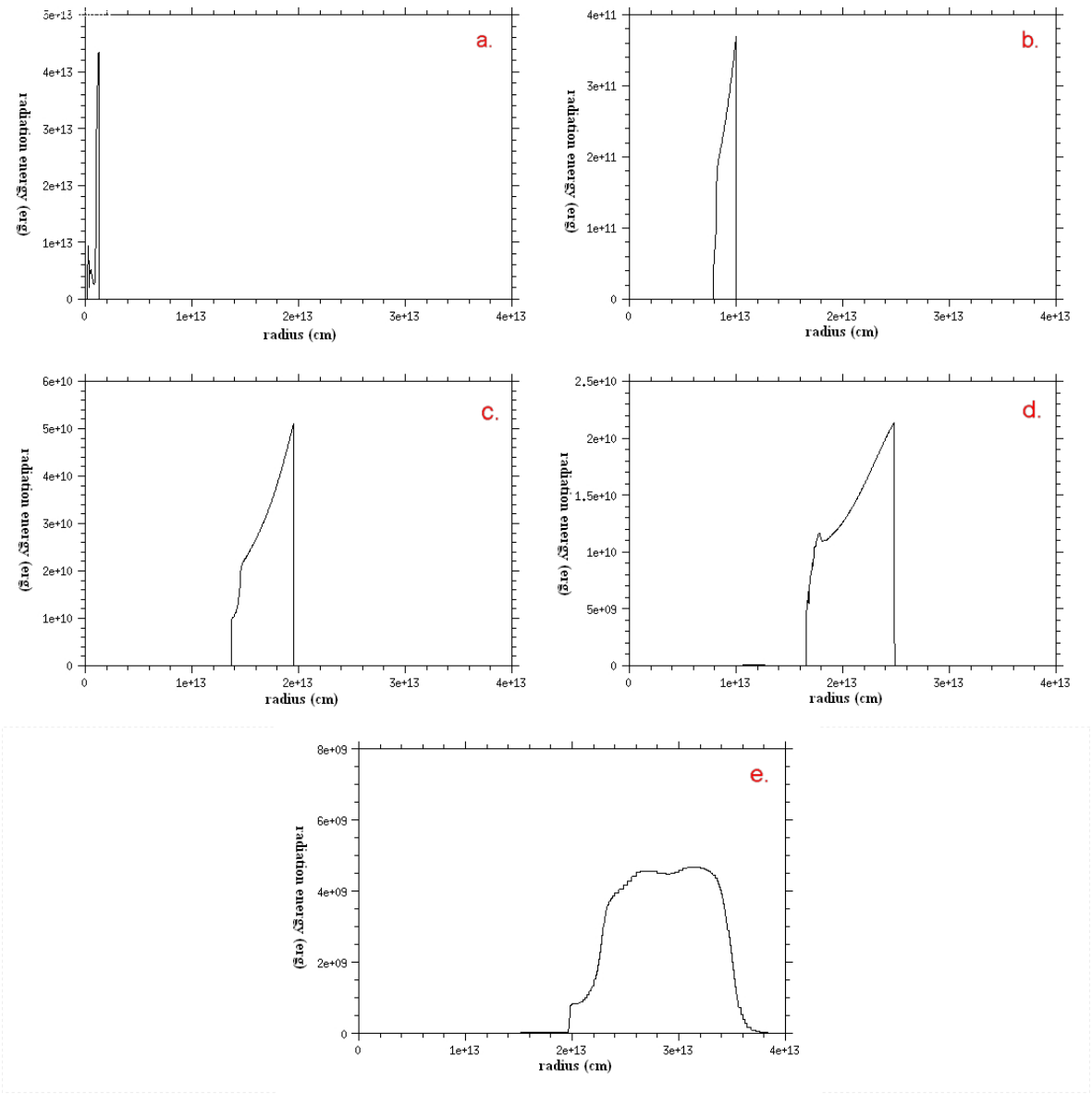



FIG. 10: Radiation energy vs. radius plots for t = 1020 s (a), 15600 s (b), 40300 s (c), 56000 s (d), and 74000 s (e).

## VI. CONCLUSION

From the simulation data it is apparent that the RAGE code is quite capable of handling this particular model, save for a few minor adjustments. Even such a small aspect of the explosion like shock breakout is expounded upon in great detail. The next foreseeable progression, then, would be to eventually run the simulation in 2 dimensions, something most machines don't have the computing capabilities to process. However, the advent of today's supercomputers, such as LANL's teraflop Roadrunner, allow for many exciting new possibilities.

## VII. ACKNOWLEDGEMENTS

I would like to thank my mentor, Chris Fryer, for providing an excellent research opportunity and for being open and encouraging. I would also like to extend my gratitude to Gabe Rockefeller, for being a fountain of knowledge regarding all things computing as well as remaining extremely patient through a barrage of emails, and to Aimee Hungerford, whose name I dropped gratuitously while in the process of requesting access. Special thanks to James Colgan and Norm Magee for their continual support and involvement in both our projects and personal lives. Thanks to Sally Seidel and the Univeristy of New Mexico for the experience, and to NSF for funding the program. I would also like to take the opportunity to acknowledge my professors at Lewis University, Leonard Weisenthal, Joseph Kozminski, and Charles Crowder; without them, I wouldn't be here.

**REFERENCES**


Blinnikov, S., S.E. Woosley et al., "Observable Effects of Shocks in Compact and Extended Presupernovae", Astrophysical Journal, arXiv:astro-ph/0212569v2 (2003)

Colgate, S.A. & C. Fryer, "Supernova theory - the search for a robust mechanism. The Wetherill Prize lecture", Physics Reports 256, 5-21 (1995)

Colgate, S.A., C.R. McKee & B. Blevens, "Relativistic Shock Propagation and Search for Electromagnetic Pulses from Supernovae", Astrophysical Journal 173, L87-L91 (1972)

Elert, Glenn, "Power Consumption of the United States", The Physics Factbook, http://hypertextbook.com/facts/2001/KatyHo.shtml (2001)

Ensman, L. & A. Burrows, "Shock Breakout in SN 1987A", Astrophysical Journal 393, 742-755 (1992)

Falk, S.W., "The Interaction of Mantle and Envelope in Type II Supernova Models", Bulletin of the American Astronomical Society, Vol.10, 425 (1978)

Gezari, S. et al., "Probing Shock Breakoutwith Serendipitous GALEX Detections of Two SNLS Type II-P Supernovae", Astrophysical Journal, arXiv:0804.1123v4 (2008)

Hall, R.J., "Modeling Supernovae with PHEONIX, The PHEONIX Project, University of Hamburg", http://www.hs.uni-hamburg.de/EN/For/ThA/phoenix/supernova.html (2006)

Janka, H.T. & E. Müller, "Neutrino-driven Type-II supernova explosions and the role of convection", Physics Reports 256, 135-156 (1995)

Quimby, R. et al., "SN 2006bp: Probing the Shock Breakout of a Type II-P Supernova", Astrophysical Journal, arXiv:0705.3478v1 (2007)

Schawinski, K et al., "Supernova Shock Breakout from a Red Supergiant", Astrophysical Journal, arXiv:0803.3596v3 (2008)

Schwarzchild, B., "X-ray outburst reveals a supernova before it explodes", Physics Today, August 2008, 21-23 (2008)

Weaver, T.A., & Woosley, S.E., "Nucleosynthesis in massive stars and the $^{12}C^{16}O$ Reaction Rate", Physics Reports 227, 65-96 (1993)

# A Model of Synchrony in the Primary Visual Cortex

**David vanMaanen**
**Garret Kenyon, Mentor, P-21**

## Abstract

Computational models of the visual system have matched human performance in a number of task. However, some of these 'state of the art' models exclude the more basic features in the brain, such as spiking dynamics. These dynamics can lead to oscillations which has been recently shown in a primate primary visual cortex. A new model of V1 hopes to achieve these oscillations, using lateral co-circular inhibitory and excitatory connections with curvature restrictions, and show them to be object dependent. It also hopes to show that these can be used to recognize arbitrary closed curvatures. The preliminary results show that the clutter and object in a circle and clutter problem are not completely object dependent. It will be compared to experimental, when available, to determine how well it model the primary visual cortex.

## Introduction

Computational neuroscience is proving not only to give us more information about the brain, but also gives algorithms for computers to start to operate like neural networks and perform well in tasks that are usually done exceedingly better by humans than computers. An area where this becomes very apparent is in the area of visual cognition, especially object detection and recognition. Available computer algorithms cannot perform nearly as well as a human can in identifying and classifying objects. Or at least that was the case before a group at the Massachusetts Institute of Technology (MIT) developed a model of the visual cortex, which followed a hierarchical model of layered simple and complex cells, which would loose spacial resolution for a greater feature resolution. The model matches human in speed of sight tasks, but it's best performance of 90% correct in a simple animal or no animal task still leaves a lot to be desired if such an algorithm should be implement in computer technology for security or military systems[1]. There is also a great space of neuron interactions which the model does not take into account such as spiking dynamics and feed-back interactions. It is possible that spiking dynamics together with feed-back loops can cause synchronization and lead to the detection of objects.

The neurons modeled throughout this paper can be thought of as modified resistance-conductance(RC) circuits. Their membrane has a certain capacitance of charge and this is 'leaked' through the 'resistors' made through various protein canals. If left without input, they enter into an exponential decay just like an RC circuit. If this is all there was to neurons, we would have very unexciting brain and they would follow a completely linear development. However, the neurons suffer a non-linearity at a certain membrane potential called the threshold potential. When a potential is applied across the membrane at the threshold level, the membrane potential immediately drops to zero and the cell releases a signal known as a spike. The spike travels down pathways known as axions and reach inter-cellular connections known as synapses. The pre-synaptic(or sending) membranes releases transmitters which travel to the post synaptic(or receiving) membrane which rise or lowers the potential of the receiving cell depending on which transmitter was received. The transmitters sent by excitatory cells raise the potential of the post synaptic cell, while the transmitters sent by inhibitory cells lower the potential of the post synaptic cell. There is also a set of connections exclusive to inhibitory cells known as the gap junctions, which connect the membranes of two cells via a protein. This protein allows for the exchange of electrical signals and is a pathway for inhibitory cells to excite each other.

Individual neurons are relatively easy to model and pretty well understood. However networks of neurons are very complex. Mostly in the fact that connections between neurons in any sort of wiring diagram can usually form what appears to be spaghetti thrown onto the paper. Those trying to model

these networks must conform to a few well accepted facts about these connections. The neurons form groups that complete a similar task called layers. These layers can be grouped by their functionality. In the visual system, the largest portion of the primate brain, these layers can be further divided into two streams, the ventral and the dorsal. The dorsal stream handles the question of where the objects are in the scene. The ventral stream, which the MIT model and this paper are focused on, handles the question of what. It must be able to recognize and identify objects. The layers form a hierarchy, which is the basis for the MIT model[1].

The hierarchical structure in the MIT model is based on the basic structure from Hubel and Wiesel, which repeats layers with the same basic pooling operations. These layers are called simple and complex layers. The simple layers find a set of features and positions while the complex layers generalize the positions while maintaining the features. This process is repeated and has the effect of increasing the number of features, while decreasing the spacial resolution, which maintains a reasonable population of neurons throughout the hierarchy. The model treats these layers as strictly feed-forward and non-spiking model, which means that neurons of one layer take their input and calculate scalar numbers and pass that to the neurons of the next without any other interactions. This exclusion in the model is reflected in it's 90% result. This works well when compared to a speed of sight task, which gives the subject a very short period of time to make a decision on a picture[1].

The entire visual system begins with an input layer at the retina in the eye. The signals from the retina are passed to the LGN, which acts as a relay and sends signals to the rear of the brain to the primary visual cortex (V1), from here the visual pathway goes deep into the cortex levels until it reaches the inferior temporal cortex (IT) where whole objects are detected and identified. Probably the most studied and understood of all of these is V1, perhaps due to it's location on the outside of the brain. It is very firmly known that this layer(for the simple cells) detects orientations and positions of lines in the visual field. There have also been recent discoveries of synchronizations of these neurons.

The neurons in V1 as with every layer is known to have lateral connections between cells in the same layer. In V1, the connections between neurons can be described using a theory from Zucker and Sahar called co-circularity. This means that points which are lie one circle of any radius will have a stronger weight than those off the circle[2]. This combined with the well known features and new discoveries of synchronizations in V1 makes it a good candidate test models of synchronous neuron activity[3].

## Methods

The model is based on the original PetaVision, which is an excitatory model for V1 that ran in the world record petaflop run on RoadRunner. It included only excitatory cells in V1 with lateral connections. First the cells are stimulated as though stimulated by electrodes. The cells that represent orientations and positions of lines in an image with a circle and clutter are the ones stimulated. The cells that are stimulated then connect to other cells through the co-circular weighted lateral connections, which fade over distance and are further restricted to a certain curvature. These connections weights between sending cell, j, and receiving cell, i, are given by (1).

$$w_{ji} = G_d(d_j i)\, G_\kappa(\Delta\kappa_j i)(G_\theta(\chi) - I) \tag{1}$$

The $G_d$ , $G_\kappa$ , and $G_\theta$ terms are the distance, co-circularity, and curvature Gaussian functions,

$$G_d(d_{ji}) = e^{(\frac{d_{ji}^2}{\sigma_d^2})} \tag{2}$$

which normalize the weight term to one for the strongest connection. The I term is used to limit the co-circular weights to only a certain top percentage of connections.

The operand of the Gaussian for distance is the usual euclidean distance using 'pixel' coordinates, in which each 'pixel' contains a line. The operand of the Gaussian for co-circularity is computed by taking the difference between the orientation of the receiving cell and a line at the position of the

receiving cell which is tangent to the same circle as the sending cell. The target orientation for the receiving cell is given by (2).

$$\theta_{target} = \tan^{-1}\left(\frac{-\Delta x \sin(\theta_j) + \Delta y \cos(\theta_j)}{\Delta y \sin(\theta_j) + \Delta x \cos(\theta_j)}\right) + \theta_j \tag{3}$$

where $\theta_j$ is the angle of orientation of the sending cell and $\Delta x$ and $\Delta y$ are the changes in pixel coordinates. The operand for the curvature Gaussian is the difference between the target curvature, which is set for each cell arbitrarily by multiples of the sigma term in the curvature Gaussian. This ensures that all curvatures are covers at least through the Gaussian function. The overlap at the ends attempt to ensure a rough equality among all curvatures. The curvature of this circle is given by

$$\kappa_{target} = \frac{2(\Delta y \cos(\theta_j) - \Delta x \sin(\theta_j))}{d_{ji}^2} \tag{4}$$

where $r$ is the radius of the co-circle and $d$ is the euclidean distance.

These weights are then summed for the receiving cell to arrive at a partial potential input,

$$\Phi_i = \sum_{j=1}^{N-1} a w_{ji} f_j \tag{5}$$

where $a$ is an arbitrary constant and $f$ is either 1 or 0 depending on whether or not the sending cell fires. This means that only weights from firing cells have an effect on the partial potential input. These partial potentials are used to calculate the potential of the receiving cell where the potential is the solution to

$$\frac{dV}{dt} = -\left(\frac{V - (\Phi - \Phi' + z)}{\tau}\right) \tag{6}$$

where $\Phi$ is the input from excitatory cell, $\Phi'$ is the input from inhibitory cells, and $z$ is the random noise factor. Equation (6) is the equation of an RC circuit with a input voltage. The solution can be approximated for the purposes of this model by

$$V_{n+1} = V_n - \frac{\Delta t}{\tau}\left(V_n - (\Phi_n + \Phi'_n - z_n)\right) \tag{7}$$

where $V_n$ is the potential for time step $n$ and $\Delta t$ is the size of the time step.

Equation (7) is the same inhibitory cell except for the additional input term from gap junctions and different constant for each input. The main difference for the weights of all inhibitory connections (gap junction and inhibition of inhibitory and excitatory cells) is that there is no curvature selectivity, so there is no curvature Gaussian factor in the weight equation drops out. Also all the constants are independent, especially the arbitrary a factor in (5).

The model works by repeating these equations for every time step. The potential and firing events are updated and recorded for analysis purposes. The usual run uses 200 time steps, which is approximately 200 milliseconds in the time frame of the neurons.

**Results and Discussion**

The model is to be compared against human performance in a number of tasks. Particularly in finding an amoeba, or an anamorphic closed 2D surface, in statistically similar clutter. The task will be a speed of sight task for human subjects. The computer simulation will run on the same data and the comparison will give a metric for how well the model compares to the actual neural system in the brain. Currently the model is being tested in 'toy' problems were the figure and clutter positions are known exactly and analysis can be performed with this knowledge. One such problem is a simple circle in clutter as shown in figure 1.

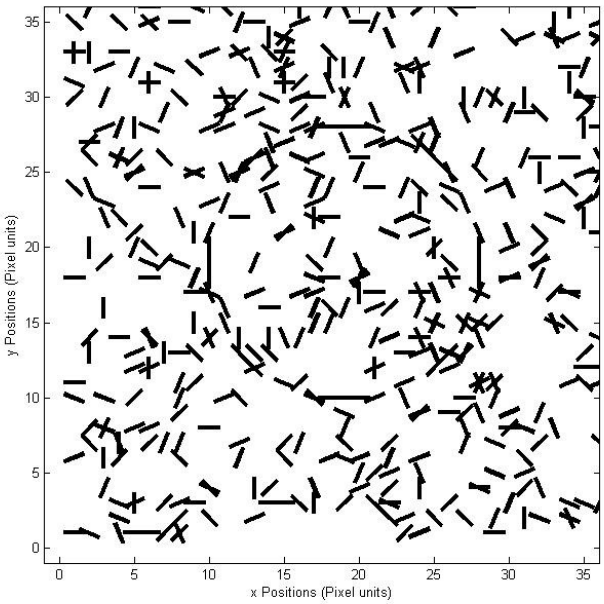



*Figure 1: Circle and clutter problem. The input for the model stimulates the neurons corresponding to the orientations and positions in the above graph.*

As shown in previous runs of the exclusively excitatory V1 model on which this code is based, the firing rate of the circle is much greater on average than that of the clutter and all other neurons as acted upon by noise. This is seen in figure 2.

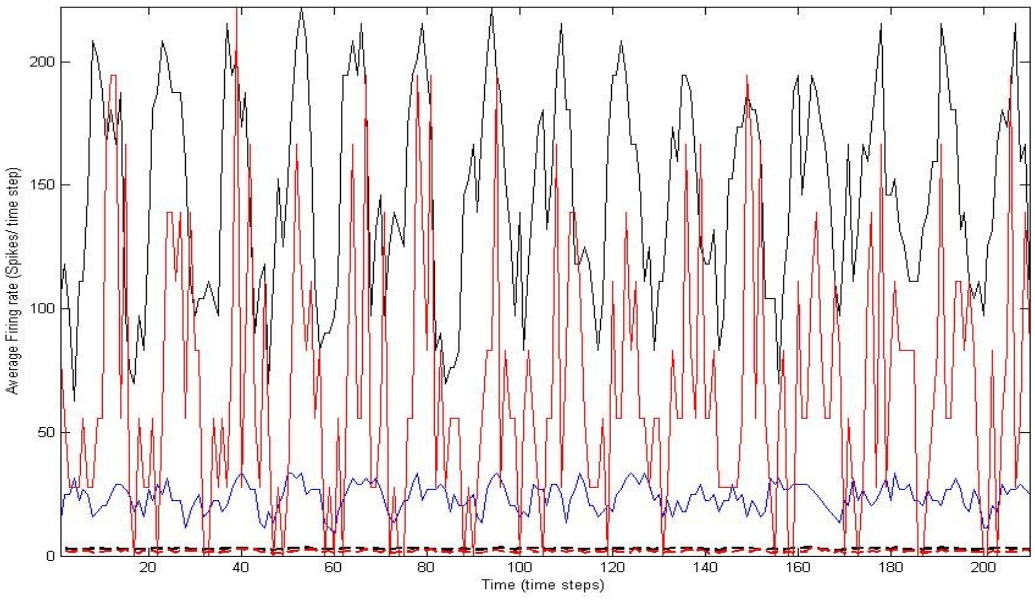



*Figure 2: Histogram. The solid black line represents the cells of the circle. The solid red line represent the inhibitory rates on the circle, while the blue represents the firing rate of the clutter neurons. The black and dotted red lines are the excitatory and inhibitory background firing rate.*

The analysis should show that neurons are oscillating due to inhibition. The frequency and strength of these oscillations can be shown through a power function, which is found as the Fourier transform of the firing rate function in figure 2. The power function determines the spectrum frequencies that are present in the signal. These amplitudes are normalized by the DC power of the system, which gives a maximum amplitude of 1, which would only occur for a single perfect oscillation and occur as a delta function at that oscillation. The greater the amplitude is, the stronger the oscillation is at that frequency. A well defined peak would show strong oscillations at a given frequency. Figure 3 shows that the oscillations of the neurons representing the clutter and the circle have a common strong frequency.

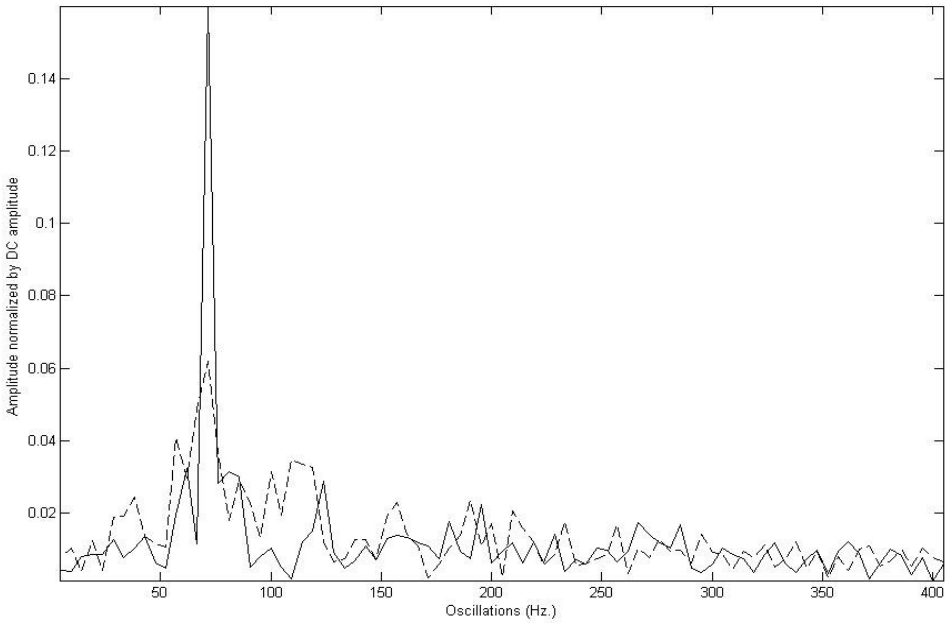



*Figure 3: The Power Spectrum. The solid line is the power spectrum of cells on the circle. The dotted line is the power spectrum of cells on the clutter.*

It also shows that the oscillations in the circle are much stronger than those in the clutter. This means that model has a preference towards oscillation on neurons representing the closed curvature of the circle. This is shown in the reconstruction of the image using the peak of the power spectra for each cell in figure 4. However, the dominant frequency of these oscillations is the same as that of the clutter which makes separation of these signals difficult for computational models.

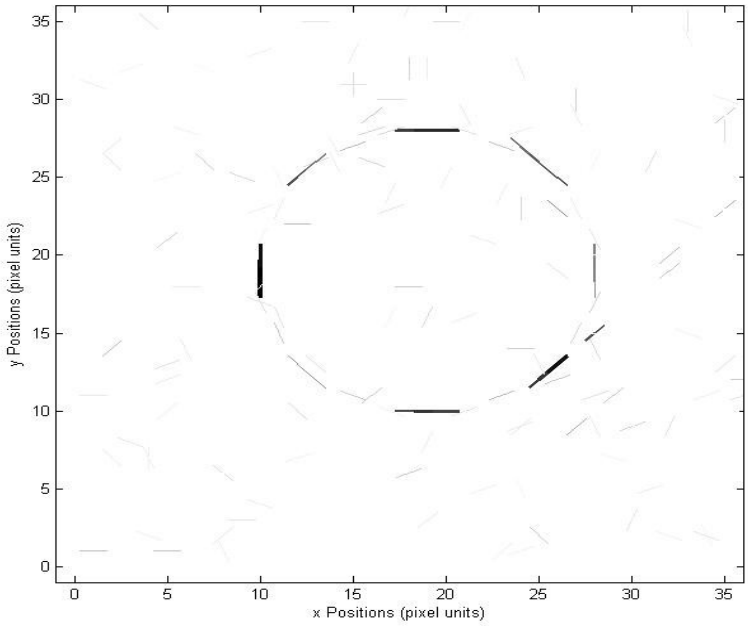



*Figure 4: Output of Cells. This firing rates are reflected in the analysis for excitatory cells only.*

To separate these oscillations, the relative phases of these oscillations can be analyzed. If the hypothesis holds then the neurons representing the circle should bind in their oscillations and be out of phase with the neurons representing the clutter. This would prove the hypothesis and allow the model to detect objects. The oscillations can be shown through an autocorrelation function and the phase tuning between the clutter and the circle can be see through a cross correlation function. The autocorrelation and cross correlation functions determine how correlated a function is with itself or with another function respectively. The closer either of these get to exact zero the closer that correlation is to being completely uncorrelated and the larger the oscillations in this function the more oscillatory that correlation is. This means that the results should show a close to exact zero cross correlation between circle neurons and clutter, which means they are completely out of phase. The autocorrelation for the neurons on the circle should show strong oscillations in comparison to the cross correlation. The autocorrelation and cross correlation functions are computed using the rate information in figure 2. These functions are shown in figure 5.

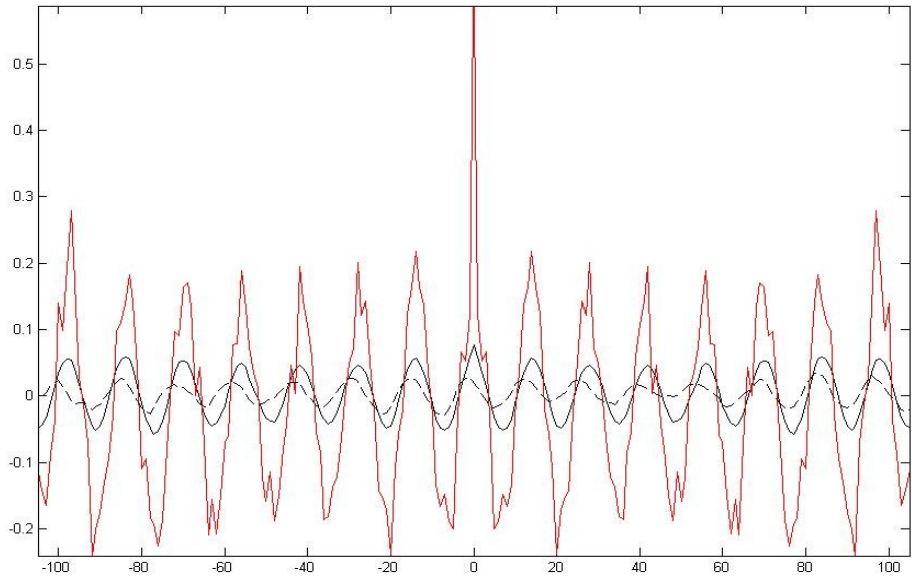



*Figure 5: Correlation functions for neurons representing the circle. Inhibitory and excitatory neurons on the circle are the solid red and black lines respecticely. The cross correlation between the neuorns on the circle and those representing the clutter is shown in the dashed black line. This shows that the circle neurons and clutter neurons are cross correlated.*

As the figure shows, there is a considerable cross correlation between the circle and clutter. This shows that the model is not as perfect as the hypothesis might hope. However, this does not mean the model is not biologically correct. So this maybe just a difficult task for a neural network to perform. Only experimental results from a human performance test can show whether or not this model is a good description of the primary visual cortex with respect to synchronization of neurons representing an object.

**Conclusion**

The model shows that there are oscillations in the V1 layer which agrees with recent experiments. The model fails at completely separating the clutter and circle elements in the circle and clutter problem. However, it is unknown if this will create a problem for future analysis of the amoeba task. This model's performance will be tested against human performance in a 'amoeba or no amoeba' task. This is the only way of determining how well it models the primary visual cortex in this task.

**References**

[1] Thomas Serre, Aude Oliva, and Tomaso Poggio. A feedforward architecture accounts for rapid categorization. *Proceedings of the National Academies of Science of the USA.* Retrieved from

http://www.pubmedcentral.nih.gov/articlerender.fcgi?tool=pubmed&pubmedid=17404214 on July 22, 2008.

[2] Ohad Ben-Shahar and Steven Zucker**.** Geometrical Computations Explain Projection Patterns of Long-Range Horizontal Connections in Visual Cortex. Retrieved from http://www.pubmedcentral.nih.gov/articlerender.fcgi?tool=pubmed&pubmedid=17404214 on July 22, 2008.

[3]P.E Maldonado, C. Babul, W. Singer, E. Rodriguez, D. Berger, S. Grün. Synchronization of Neuronal Responses in Primary Visual Cortex of Monkeys Viewing Natural Images. *Journal of Neurophysiology.*2008

# Double Photoionization of $H_2$: Double Slit Interference?

D. Weflen[1,2], D. A. Horner [2]

[1] *Department of Physics, Drake University, Des Moines, IA 50311*

[2] *Los Alamos National Laboratory, Theoretical Division, Los Alamos, NM 87545*

Experiments performed by Akoury *et al.* and Kreidi *et al.* recently claimed to have found double slit interference in the cross-section for double photoionization of $H_2$. We offer an alternate explanation, namely that the observed pattern is the result of using circularly polarized light.

## I. INTRODUCTION

Recent experiments presented in refs. [1] and [2] found a four-lobed pattern for extreme energy sharings in the cross section for double photoionization of $H_2$. Double slit-like behavior has been predicted and observed in the past for single ionization of diatomic molecules[3], and the authors [1] and [2] were trying to observe a similar phenomenon in the cross section for one-photon double ionization. In their measurements, a four-lobed pattern was observed when the cross section was examined at extreme energy sharings or framed in terms of the dielectron momentum, $\vec{k^+} = \vec{k_1} + \vec{k_2}$. We believe that this pattern is due to the presence of contributions from both the parallel and perpendicular polarization in circularly polarized light.

To investigate this, we solved the time-independent Schrödinger equation using the method Exterior Complex Scaling (ECS) [4] with a finite-element discrete-variable representation (FEM-DVR) basis. We used integral amplitude methods to generate observable cross sections.

In this paper we will present the calculated cross sections for both circularly and linearly polarized light, with photon energies between 130 and 400 eV. Then, in light of our theoretical results, we will present an interpretation of the experimental results. We will begin in section II by covering the numerical methods used to calculate the cross section, as well as the methods used to put those cross sections in terms of the dielectron momentum. In section III we will examine the results of our calculations, and compare them to the results of experiments [1] and [2]. We will also present cross sections at higher energies than were used in the experiments to look for evidence of actual double slit interference. We will then conclude and summarize our findings in section IV.

## II. THEORY

### A. Numerical Methods

For photoabsorbtion, the problem can be written as a driven Schrödinger equation

$$(E_0 + \omega - H)\Psi^{sc} = \mu\phi_0 \tag{1}$$

where $\mu$ is the dipole operator, $\mu = \epsilon \cdot (\vec{\nabla}_1 + \vec{\nabla}_2)$ and $\phi_0$ is the initial state wave function. We expand the wave function on a finite-elemet discrete-variable (FEM-DVR) basis [5] on a finite element grid to allow for fast and accurate calculation of the matrix elements.

The asymptotic boundary conditions for $\Psi^{sc}$ are extremely complicated and not known in a useful form. In order to treat the problem rigorously, we applied an exterior complex scaling (ECS) transformation[4]. In exterior complex scaling, the radial coordinates of both electrons are rotated into the complex plane beyond some radius $R_0$:

$$r \rightarrow \begin{cases} r & r \leq R_0 \\ R_0 + (r - R_0)e^{i\eta} & r > R_0. \end{cases} \tag{2}$$

The ECS transformation is shown schematically in Fig. 1.

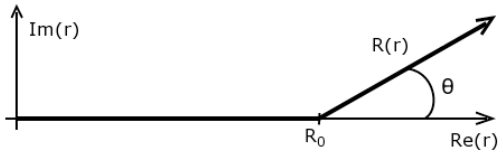



FIG. 1: ECS contour in the complex plane.

Under this transformation, outgoing functions decay exponentially on the complex part of the grid, while the real part of the grid maintains the physical solution. We are then left with simple boundary conditions.

Once the outgoing wave function has been evaluated, we need to some means to extract the scattering and ionization amplitudes. The traditional method of matching the outgoing wave function to the scattering amplitudes does not work in this case, as the wave function goes to zero asymptotically. Originally, this problem was solved by calculating the quantum mechanical flux,

$$\mathbf{F}_\rho(\hat{r}_1, \hat{r}_2, \Omega) = \frac{1}{2i}((\Psi^{sc})^* \nabla \Psi^{sc} - \Psi^{sc} \nabla (\Psi^{sc})^*), \tag{3}$$

through some surface on the real part of the grid, and then extrapolating. This, however, had a number of problems, particularly when treating very unequal energy sharings [4], so other methods were developed. Starting with the formal definition of the amplitude, we are able to use Green's theorem [4] and rewrite the amplitude as a surface integral, which is accurate and stable on finite grids.

$$f(\mathbf{k}_1, \mathbf{k}_2) = \frac{1}{2} \int_S (\Phi_{k_1} \Phi_{k_2} \nabla \Psi^{sc} - \Psi^{sc} \nabla \Phi_{k_1} \Phi_{k_2}) \cdot d\hat{S} \tag{4}$$

The $\Phi_k$ are "testing functions," and it is important to note that this is not a projection, but an expression that formally extracts the amplitude from the wave function. In our case, they are $H_2^+$ continuum eigen functions. Finally, to get the cross section we use the formula

$$\frac{d^3\sigma}{dE_1 d\theta_1 d\theta_2} = \frac{4\pi^2}{\omega c} \mathbf{k}_1 \mathbf{k}_2 |f(\mathbf{k}_1, \mathbf{k}_2)|^2. \tag{5}$$

### B. The Dielectron Momentum

Because the results of the experiments were expressed either as integrated over a range of energy sharings or a range of $k^+$, two programs to integrate the cross-section were developed. The first was a simple program to integrate over energy sharings, to compare the results of our calculations with the results of the experiments for extreme energy sharing. The second was significantly more complex, as $k^+$ is a vector sum, and is dependent on the directions of the electron momenta as well as the energy sharing. Also, since $k^+$ must be considered at many angles, a number of coordinate transformations must take place, since the scattering amplitudes and the cross section were a function of $k_1$ and $k_2$.

Getting the cross-section for a given $k^+$ consisted of integrating over all magnitudes of $k_1$ and $k_2$ that were not precluded from adding up to $k^+$ by the triangle inequality. A given $k_1$, $k_2$ and $k^+$ uniquely determine the angles involved, except for the azimuthal angle of $k_1$ and $k_2$ about $k^+$. This angle must be integrated over. The problem, then, is the following: for every given angle of $k^+$, integrate over a specified range of magnitudes of $k^+$, determine which $k_1$ and $k_2$ add up to every given $k^+$ and integrate over them, including the magnitudes and two angles for every $k_1, k_2$ pair.

The problem of generating the angles for $k_1$ and $k_2$ specified by a given energy sharing that add up to a given $k^+$ was approached by first solving the problem for the case where $k^+$ is aligned along the $z$-axis. This case can be solved quite easily; the polar angles can be generated from the

law of cosines, and the azimuthal angles of $k_1$ and $k_2$ have a difference of $\pi$. Specifically,

$$\cos(\theta_1) = \frac{(k^+)^2 + k_1^2 - k_2^2}{2k_1k^+} \qquad \cos(\theta_2) = \frac{(k^+)^2 - k_1^2 + k_2^2}{2k_2k^+} \qquad \phi_2 = \phi_1 \pm \pi. \tag{6}$$

Counterintuitively, the simplest way to make rotational transformations in spherical coordinates is to translate them into cartesian coordinates, transform them, and then transform them back into spherical coordinates. The cartesian vectors that satisfy our conditions, then, are

$$\vec{k_1} = \begin{bmatrix} k_1\sqrt{1-\cos^2(\theta_1)}\cos(\phi_1) \\ k_1\sqrt{1-\cos^2(\theta_1)}\sin(\phi_1) \\ k_1\cos(\theta_1) \end{bmatrix} \qquad \vec{k_2} = \begin{bmatrix} k_2\sqrt{1-\cos^2(\theta_2)}\cos(\phi_2) \\ k_2\sqrt{1-\cos^2(\theta_2)}\sin(\phi_2) \\ k_2\cos(\theta_2) \end{bmatrix} \tag{7}$$

Since the $\theta_i$ are fixed, all we need to do is loop over $\phi_1$. Then, after applying the rotation of $k^+$, we have

$$\vec{k_i'} = \begin{bmatrix} (x_i\cos(\theta_{k^+}) - z_i\sin(\theta_{k^+}))\cos(\phi_{k^+}) + y_i\sin(\phi_{k^+}) \\ -(x_i\cos(\theta_{k^+}) - z_i\sin(\theta_{k^+}))\sin(\phi_{k^+}) + y_i\cos(\phi_{k^+}) \\ x_i\sin(\theta_{k^+}) + z_i\cos(\theta_{k^+}) \end{bmatrix} \tag{8}$$

The transformed $k_1$ and $k_2$ were then converted back to spherical coordinates and used to gather the proper scattering amplitudes and calculate the cross-section.

The resulting program was not particularly efficient, so an effort was undertaken to optimize it. Shark, part of Apple's Xcode suite, was used as a profiler. Using this, we were able to determine that a particular subroutine, bodyamps, was responsible for much of the slowdown. This was because an if-test was used in the innermost loop as a replacement for calculating the bounds of the loop beforehand. This was changed, allowing the program to run in a fraction of its former time.

## III. CROSS SECTIONS

### A. Comparison with Experiment

The real portion of our grid was 40 $a_0$ (9 real elements) and the total grid was 90 $a_0$ (13 total elements). The complex scaling factor was 40°. Our results were obtained from a grid order of 19 and an $l_{max}$ of 7. These values result in an acceptable convergence (Fig. 2), but still less than ideal.

In Figs. 3-5 we show the results of our calculations and the experimental data for photon energies of 130 eV to 240 eV. In general, the agreement is quite good. The figures line up nicely, with rather small differences.

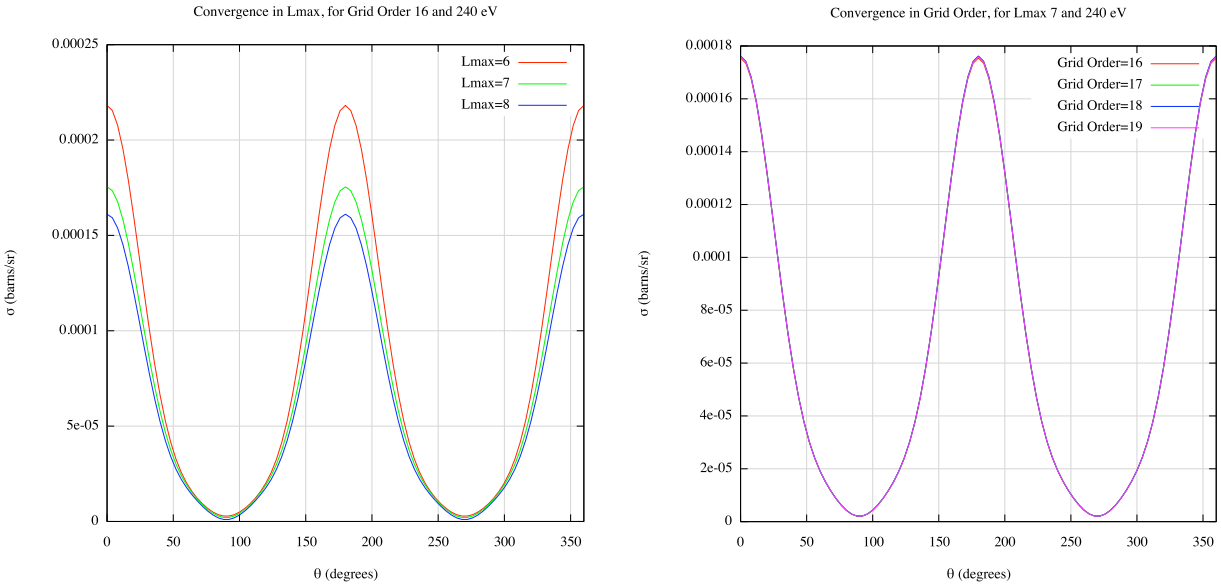



FIG. 2: Convergence in $l_{max}$ (left) and grid order(right).

A key observation in Figs. 3-5 is that the average of the cross sections for parallel and perpendicular polarization (blue curves) are nearly identical to the cross section for circularly polarized light. The only difference is a "twisting" in the circular polarization case. This is the crux of our interpretation that the four-lobed patten is the result of the fact that circular polarization contains equal amounts of parallel and perpendicular polarization. The twisting is caused by the difference in phase between the parallel and perpendicular amplitudes; the two cross sections tend to be similar because each tends to be large where the other is small, allowing for only minimal interference (the separate cross sections for linearly polarized light can be seen in Figs. 4 and 7). Thus the phase difference imparts only a small twist to the cross section, leaving it essentially the same as an incoherent mixture of the two linear polarizations. The interpretation is then changed: rather than demonstrating double slit interference, the experiment demonstrated the composition of circularly polarized light.

### B. Higher Energies

While the four-lobed pattern is most likely not double slit interference, that does not mean that double-slit like behavior is not present in the cross section for double photoionization of $H_2$. Specifically, we propose that the secondary lobes seen in the case of perpendicular polarization are in fact a result of double slit interference. These secondary lobes increase in prominence as photon

energy increases, as can be seen in Figs.5 through 7. The secondary lobes approximately satisfy

$$R\sin(\theta) = \lambda_e \tag{9}$$

the condition on the electron wavelength for constructive interference for a double slit. Relatively small lobes are even seen at 240 eV, the highest photon energy included in the experiment, and possibly could have been observed. They cannot be seen in the data, however, as the mixture with parallel component of circular polarization completely obscures the small lobes. Ironically, circular polarization was chosen to *avoid* double slit effects being obscured by the polarization (it was feared that the quasinode perpendicular to the polarization axis would hide the interference pattern). If the experiment were performed again, with linear polarization and possibly a higher energy, these secondary lobes, the sought-after double slit interference, could most likely be observed.

## IV. CONCLUSIONS

We have converged calculations of the cross section for one-photon double ionization of $H_2$. Our calculations of the cross section for double photoionization of $H_2$ agree with experiment, generating the four-lobed pattern found in refs. [1] and [2]. Comparison with the cross sections for parallel and perpendicularly polarized light leads us to conclude that the observed pattern is caused by the composition of circularly polarized light rather than double slit interference.

While the four-lobed pattern is not double slit interference, the authors of refs. [1] and [2] were not wrong to suspect the effect would occur, nor were they wrong regarding the pattern being strongest in $k^+$. Had the light source been linearly polarized rather than circularly polarized, double slit interference would have been observed.

---


[1] D. Akoury *et al.*, *Science* **318** 949 (2007).

[2] K. Kreidi *et al.*, *Phys, Rev. Lett.* **100** 133005 (2008).

[3] H.D. Cohen and U. Fano, *Phys. Rev.* **150** 30 (1966)

[4] C. W. McCurdy, M. Baertschy and T. N. Rescigno, *J. Phys. B* **37** (2004) R137-R187

[5] D. E. Manolopoulos and R. E. Wyatt, *Chem. Phys. Lett.* **152** (1988), 23

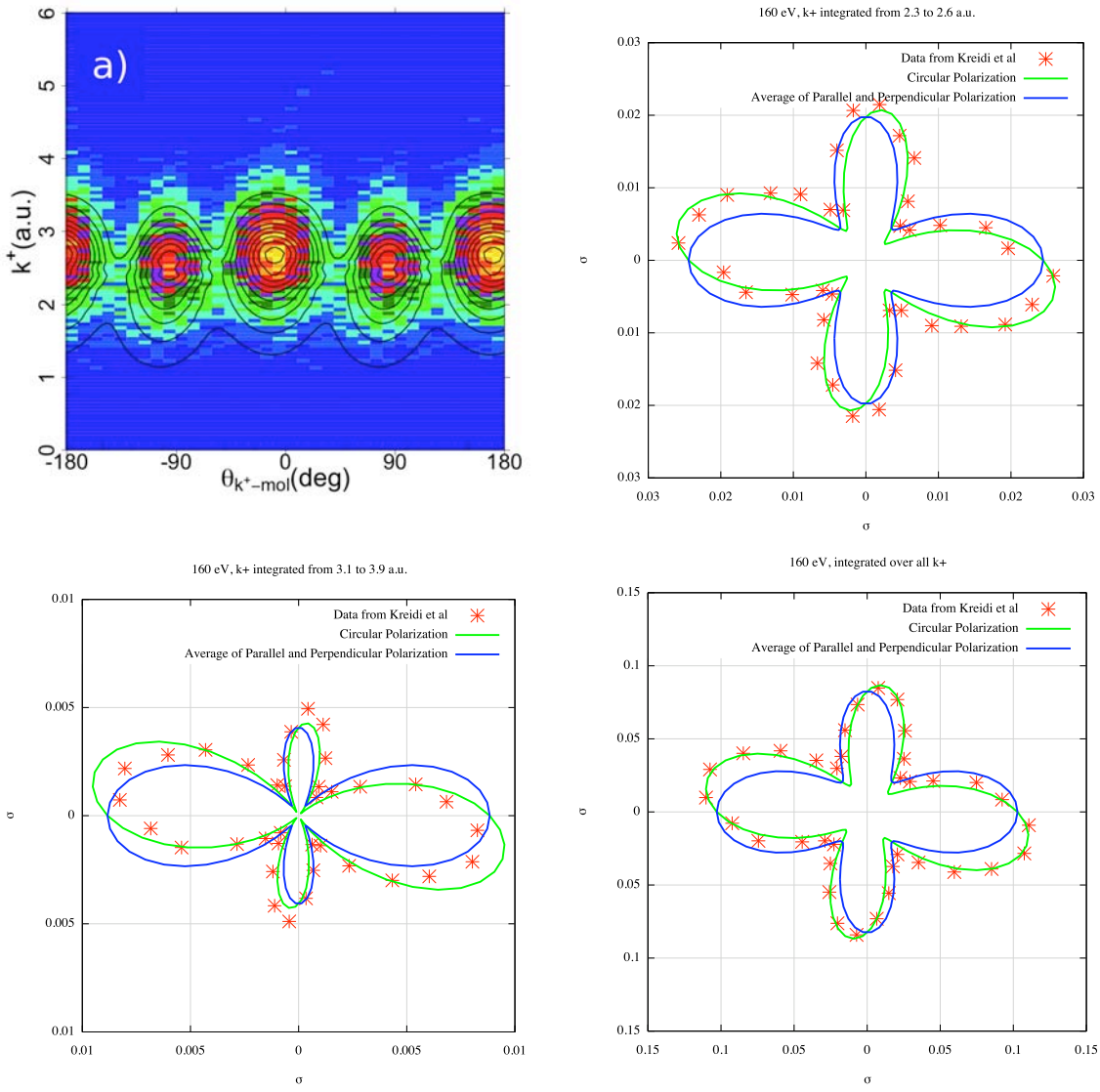



FIG. 3: A contour plot of the calculated cross-section superimposed on the data from [2] (top left) and several sections across it. This is intended to show that, in general, our calculations agree with experimental data.

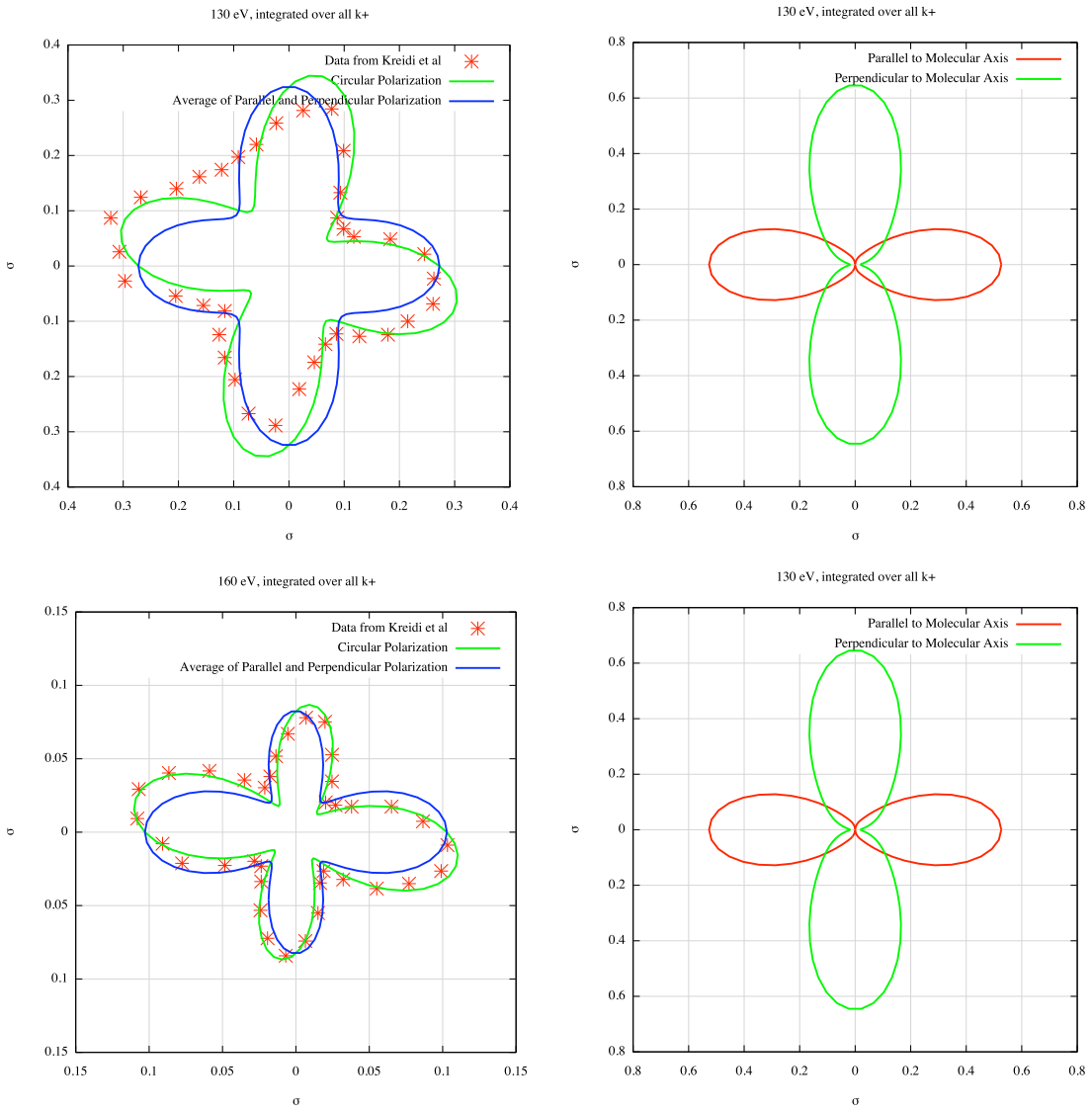



FIG. 4: A comparison of the data from [2], and the calculations at 130 eV (top row) and 160 eV (bottom row). On the left we compare our calculations of the cross section for circular polarization with experimental data and with the average of our calculations for polarization parallel and perpendicular to the molecular axis. This is done to show the similarity between the cross section for circular polarized light with the average of the cross sections for linearly polarized light. On the right are graphs of the cross section for parallel and perpendicular polarization. The cross section has units of barns/sr.

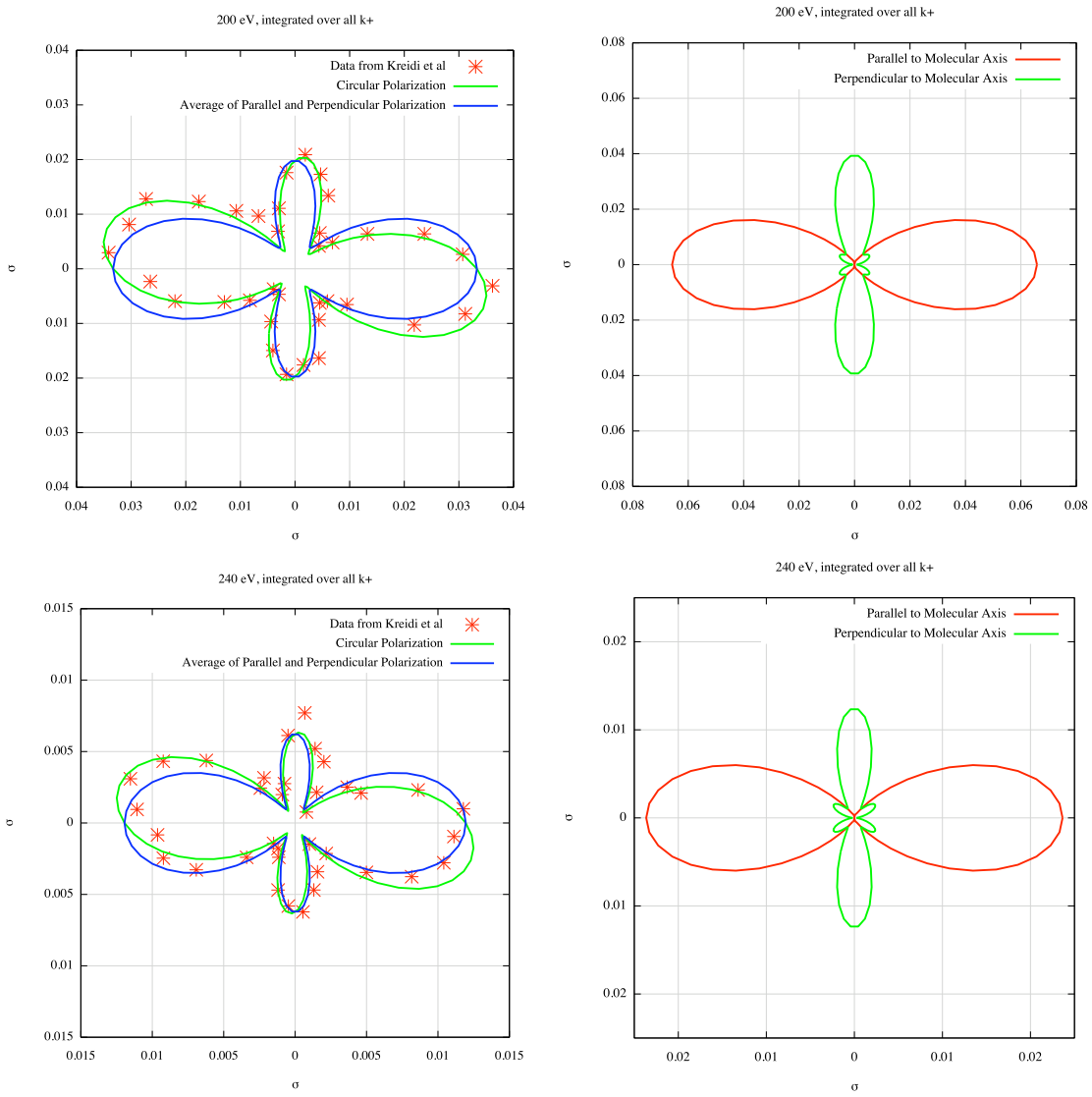



FIG. 5: Same as Fig. 4, but for 200 eV (top row) and 240 eV (bottom row). Notice the secondary lobes in the cross section for perpendicular polarization are already apparent at 240 eV.

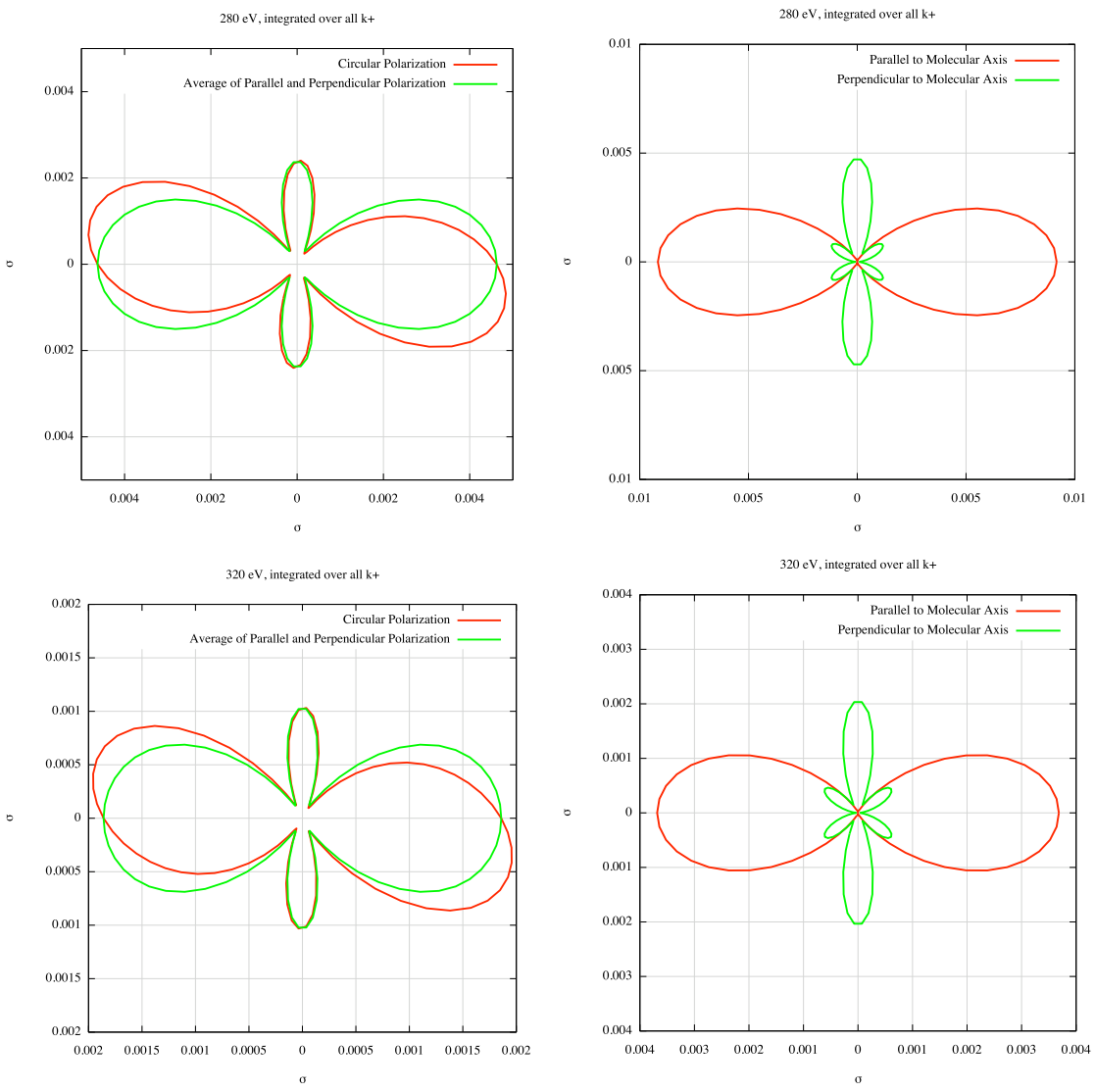



FIG. 6: Same as Fig. 4, but for 280 eV (top row) and 320 eV (bottom row). These energies are higher than any used in the experiment.

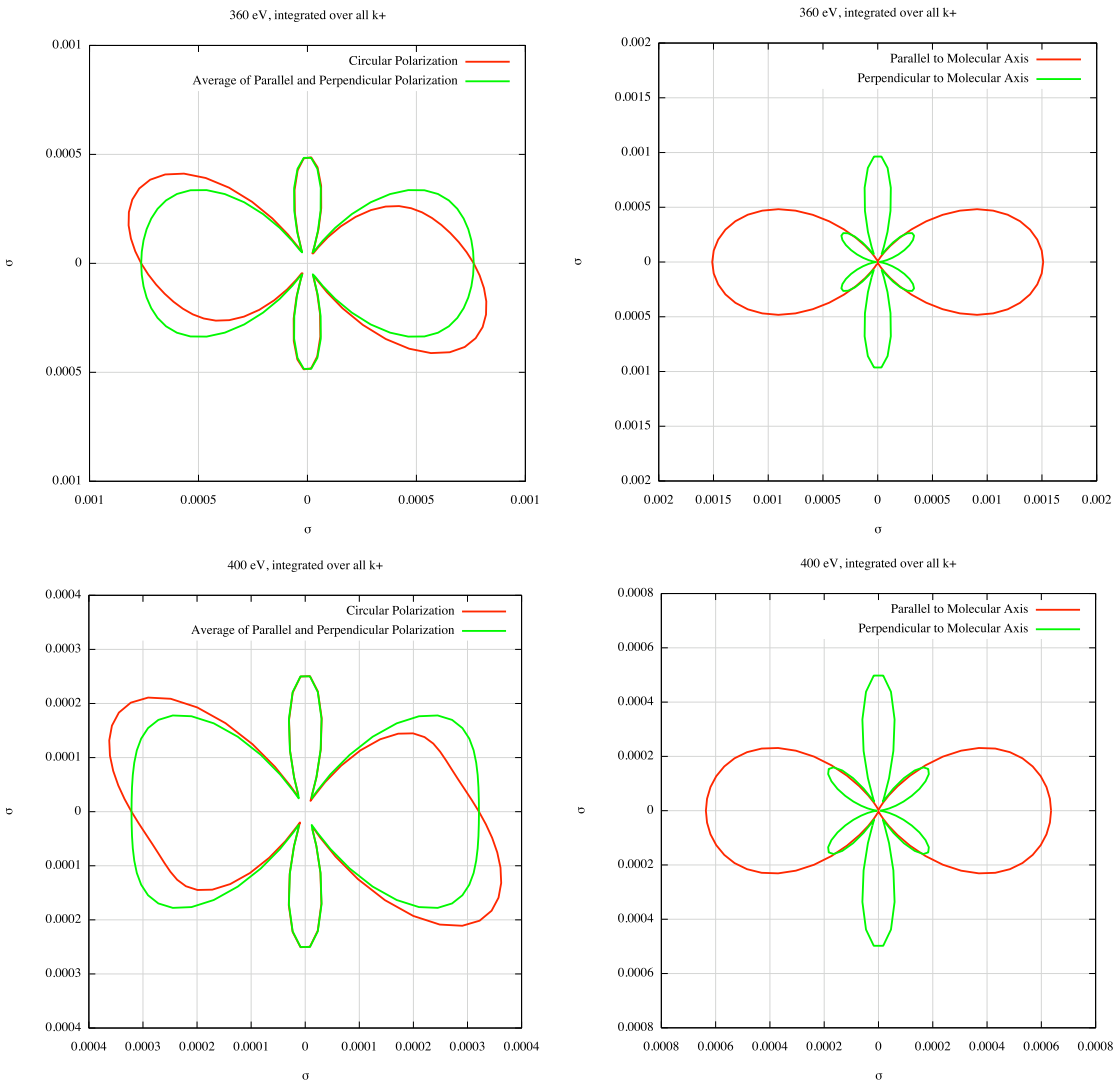



FIG. 7: Same as Fig. 4, but for 360 eV (top row) and 400 eV (bottom row).