Computing Hydrogen Ion Survival Probability: Academy Student, Graduate Student, and Faculty Experiences

David Monismith – Independent Researcher
Yixiao (Icy) Zhang – Undergraduate Student, Cornell University
John Shaw – Associate Professor, Northwest Missouri State University
Himadri Chakraborty – Associate Professor, Northwest Missouri State University

Outline

- Project Overview
- Academy Student Involvement
- Graduate Student Involvement
- Faculty work completed by Drs. Chakraborty, Monismith, and Shaw
- Discussion of current results
- Future progress

Project Overview

- Introduction
- Physics
- Parameter Sweeps
- Discussion of XSEDE startup allocation

Introduction

- Simulating firing a Hydrogen Ion at a Metal Surface.
- Goal: compute the survival probability of the ion to determine quantum effects.
- Interested in quantum effects on nanostructured surfaces, i.e. catalytic, magnetic, ferroelectric, mechanical, optical, and electronic effects.
- Desired findings: technological potential, suitability as templates for “designer surfaces”.
- For example large-scale fabrication of functional atomic or molecular assemblies [22-27]

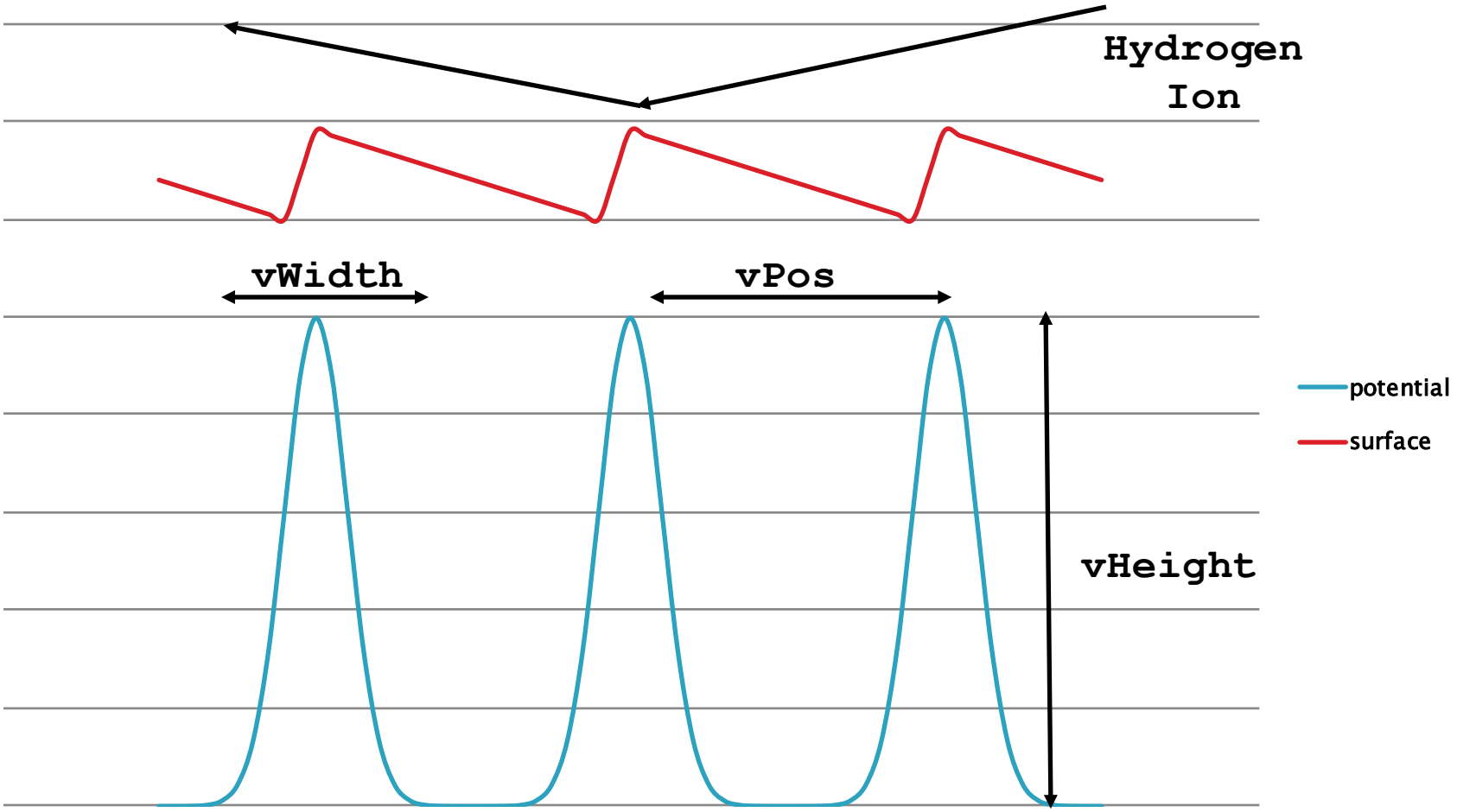
Project Goals

- Run multiple instances of the cn2Ds program with varying parameters in a reasonable amount of time.
- Understand how parameter variations affect ion-surface interactions [22-27,41].

Definitions

- Stampede – Supercomputer at Texas Advanced Computing Center (TACC).
  - Consists of 6416 nodes.
- Node – one computer (similar to a very fast desktop).
  - One node on stampede has 16 CPU-cores and 32 GB RAM.
  - Most (about 5500) have one Xeon Phi Accelerator.
- SUs – service units.
  - One service unit on stampede is equivalent to one CPU-core-hour [43,44].
- OpenMP – Open Multiprocessing – a tool for parallelizing computer code using threads (lightweight running programs).
  - OpenMP uses thread-based parallelism, which allows for parallelization on one node[35].
  - Maximum theoretical speedup on one Stampede node is ~16x using only the standard CPU cores [44].

Firing the Hydrogen Ion at a Surface



The Kronig–Penny potential and the vicinally stepped metallic surface it models

[22–27,41]

Computational Methodology

- The trajectory of the ion is modeled in two different ways
  - Using a broken straight line, and
  - Using the Biersack-Ziegler trajectory
- The electronic wave function is modeled using a solution of the time-dependent Schrödinger equation with Hamiltonian:

$$i\hbar \frac{d}{dt}\Phi(\vec{r},t) = H\Phi(\vec{r},t)$$

$$H = H' + H_{free}$$

$$H' = -\frac{1}{2}\frac{d^2}{dz^2} + V_{ion} + V_{surf}$$

$$H_{free} = -\frac{1}{2}\frac{d^2}{dx^2}$$

- The Crank-Nicolson method is used to solve the Hamiltonian.

Computational Methodology

- For fixed distances between the ion and surface numerical propagation over *t* is

$$\Phi(\vec{r}, t + \Delta t : D) = \exp(-iH(D)\Delta t)\Phi(\vec{r}, t : D)$$

- And the Crank-Nicolson scheme is applied by solving a tridiagonal matrix to obtain an approximate solution to the formula above at every time step over a grid.
- The approximation is as follows:

$$\exp(-iH\Delta t) \approx \exp(-iH_{free}\frac{\Delta t}{2})\exp(-iH'\Delta t)\exp(-iH_{free}\frac{\Delta t}{2})$$

Simulation with cn2Ds

- Code developed over 15 years in Fortran IV and 77.
- Recently updated to Fortran90.
- Designed to simulate surface interaction of a Hydrogen Ion.
- Serial code was used to successfully produce 2D results for firing a H- ion at a surface of metal atoms via the Crank-Nicolson method.
- Empirical results show the program is compute bound.
- Runtimes on local resources range from days to weeks for a single cn2Ds instance.
- Used the `ifort` compiler (v13) with level 3 optimization, which allows for the use of SIMD instructions.

[22-27,35,40,41]

cn2Ds Algorithm – Initialization

cn2Ds Pseudocode

Perform initialization steps including:

- **Construct an equidistant time mesh**
- **Define projectile trajectory**
- **Construct an equidistant spatial mesh**
- **Identify initial coordinates**
- **Add translation phase factors to the initial wave function**
- **Build a tri-diagonal matrix for the Crank Nicolson propagator**

[22-27,41]

cn2Ds Algorithm – Time Loop

```
Start the time loop
   During iteration 1:
      Check the norm of the
        initial packet and get the
        center position.
      Copy the initial wave
        function into a 2D array
        for use in the
        autocorrelation integral.
   End iteration 1
```

cn2Ds Algorithm – Time Loop, Continued

During iterations 2 to t:
    Add translation phase factors.
    Apply the Crank Nicholson method for one timestep for Φ_x.
    Apply the Crank Nicholson method again for the same time step for $\Phi_z + V(x,z)$.
    Apply the Crank Nicholson method a third time for the same time step for Φ_x.
    Include absorption at the grid boundary.
    Print wave function and check the norm.
End iterations 2 to t.
End time loop.

Parameter Sweep

- In order to understand the affect of parameter variations on ion-surface interactions, it is necessary to run the program with varying parameter sets.
- A single node on the Stampede supercomputer is sufficient to run multiple serial instances of cn2Ds.
- A batch script generator was written for use with the TACC Launcher – a tool to perform parameter sweeps on a supercomputer [36].
- Successfully able to execute one job per CPU core on Stampede to perform several modest parameter sweeps using 5 to 6 nodes per sweep [41].

Academy Student Involvement

- Student Involved – Yixiao (Icy) Zhang
  - A Northwest Academy Student
- Understanding the project.
- Learning about the command line and parallelism.
- Performing OpenMP updates to the code.
- Experience with Missouri Academy of Science presentation.

Understanding the Project

- Meetings with Drs. Chakraborty and Shaw to understand the Physics of the project.
- Virtual attendance at TACC ICERT REU from June 2 – 5, 2014.
  - Covered C, Fortran 90, logging on to TACC resources, and more.
  - Converted cn2Ds code to Fortran 90 from Fortran 77 and Fortran IV shortly after this meeting.
- Meetings with Dr. Monismith to review OpenMP directives.
- Attendance at graduate student meetings to understand data management.

Parameters

Parameter Names	Description	Value Range
vWidth	Width of the potential	1, 2, 3
vHeight	Height of potentials	0, 0.027, 0.054, 0.081
parV	Parallel velocity	0.20 to 1.0, 0.05 step size
vPos	Distance between adjacent steps	5, 7, 9, 11, 13
normV	Normal velocity	0.03 (originally 0.01 to 0.08), 0.01 step size
metalType	Types of metals used in simulation	Li (110), Cu(100),Cu(111), Au(100), Au(111), Pd(111)
trajectory	Ion trajectory	Biersack-Ziegler or Broken Straight Line
DCS	Distance of Closest Approach	1, 2 (atomic units)
Impact	Location where the ion strikes the surface	Center or Peak

- Listed above are the parameters that were planned for variation in the cn2Ds simulation.
- There are possible 16320 parameter variations.
- A file consisting of over 20 lines was used to provide all parameter values to the simulation.

Parallelization

- Runtimes for serial code (one thread) on Stampede often exceeded the maximum wall time for a program (48 hours).
- Thus, code was updated to include parallelism via OpenMP directives.
- OpenMP Modifications to cn2Ds performed from 2013 to 2015 and were successfully integrated into existing code.
- Parameter sweeps were modified to work with one process (and thus one parameter set) per node and up to 16 threads using OpenMP parallelism without use of the Xeon Phi accelerator.
- Benchmarking was performed with 16 threads per job.
- Jobs were executed on Stampede supercomputer using the a "startup" allocation (50,000 SUs) through NSF XSEDE.
- The "startup" allocation provided enough CPU time to sweep across initial parameter variations for Cu100 and expended the entire allocation.

[22-27,35,40,41,43,44]

Implementing Parallelization

- cn2Ds is computationally intensive within the time loop during the three applications of the Crack-Nicholson method.
- Each of three nested loops updates the 2D matrix, Φ, to perform different parts of the Crank-Nicholson time step computation.
- Parallelization was implemented within each of the three loops, as each depends upon the previous application of the Crank-Nicholson method.
- Each nested loop, contains **independent** inner loop computations.
- These were parallelized to allowing for maximum speedup for these sections to be equivalent to the number of available processors.
- OpenMP `parallel for` directives were used to parallelize each of the three Crank-Nicholson loops.

[22-27,35,40,41]

OpenMP Parallelization Example

```
!$OMP PARALLEL DO PRIVATE(kx,psiin,psiout)
do kz=1,nzstep
  do kx=1,nxstep
    psiin(kx)=psi(kx,kz) ! load vector
  enddo
  call cn1Dpropx(lowerx,diagx,upperx,psiin, &
&    psiout,nxstep)
  do kx=1,nxstep
    psi(kx,kz)=psiout(kx)  ! unload vector
  enddo
enddo
!$OMP END PARALLEL DO
```

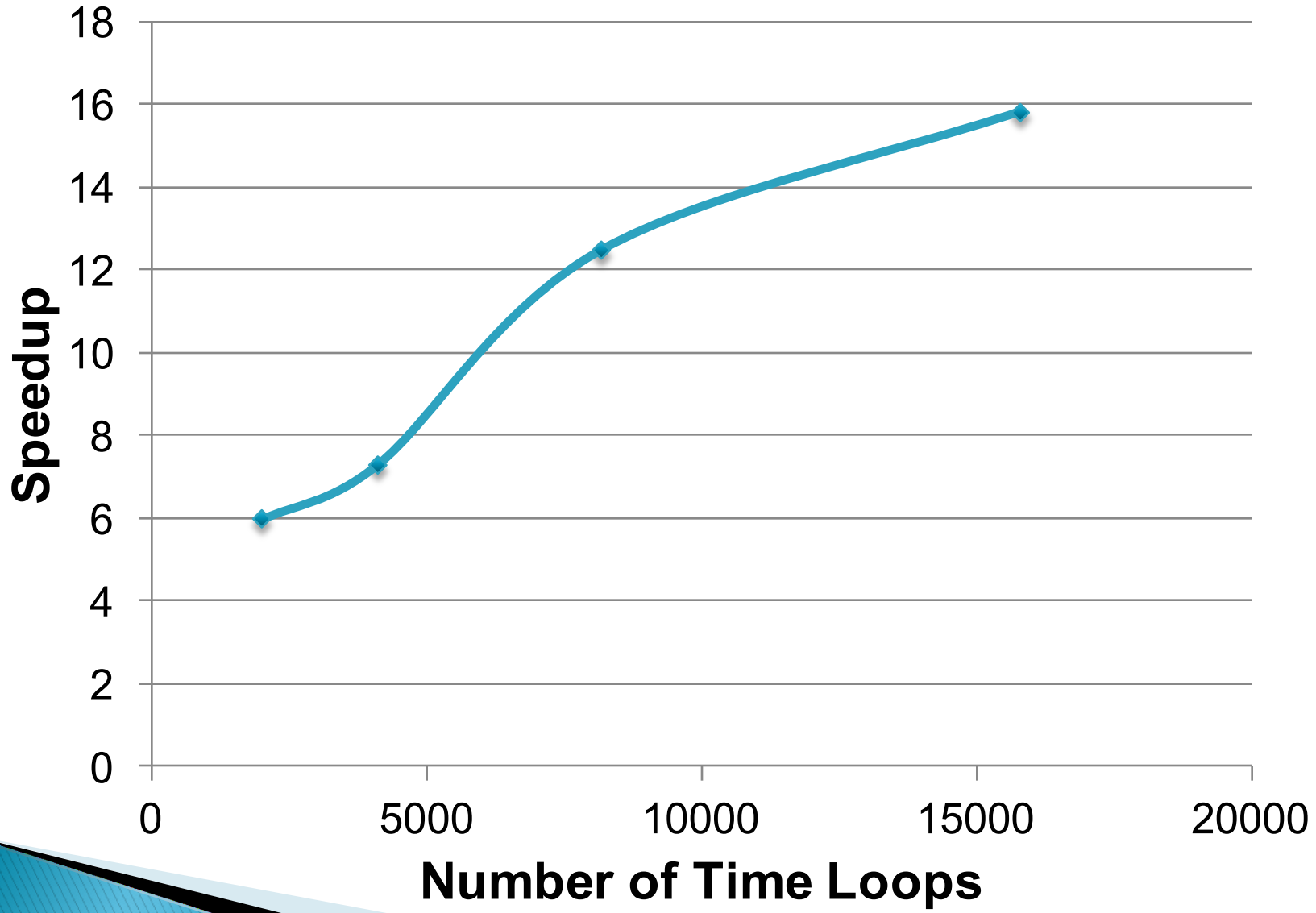
[22-27,35,40,41]

Run Times for Different Values of normV for Li110

normV	# of time loops	Parallel Run Time	Serial Run Time	Speedup
0.08	2013	0:29:07	2:54:29	5.99
0.04	4099	1:21:53	9:57:18	7.29
0.02	8176	3:35:09	44:43:59	12.47
0.01	15804	12:53:51	203:47:18 (estimated)	15.80 (estimated)

- Parameter values were vPos = 25, vHeight = 0.054, vWidth = 1, parV = 0.3.
- The time step was 0.25 atomic units.

Parallelized cn2Ds Speedup



Parameter Sweep

- Parameter Sweep – running the same code many times with many different parameters for each run.
- Run multiple instances of the cn2Ds program with varying parameters in a reasonable amount of time.
- Understand how parameter variations affect ion-surface interactions.

Project Experiences

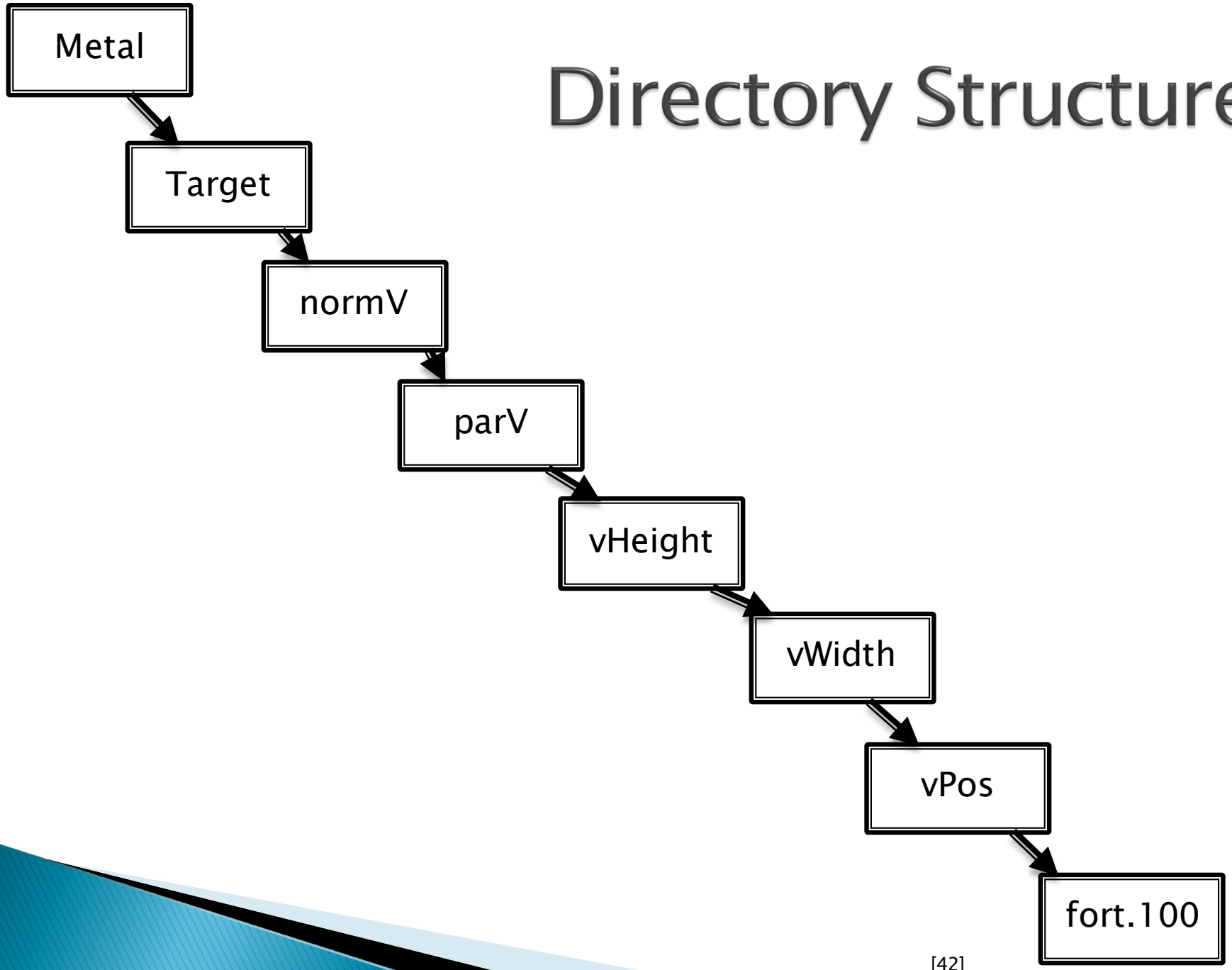
- Learned about Schrödinger's equation and how it is used in the cn2Ds code.
- Learned about directive-based parallelism and worked on national supercomputing resources.
- Presented results at the Missouri Academy of Science in May 2015.

Graduate Involvement

- Students involved:
- Anusha Manne
- Deekshith Narsina
- Tejaswini Dingari
- Rishi Varma Kosuri
- Sowmya Chowdary
- Vasundhara Pidikiti
- Dedeepya Salla
- Vandana Nidasanametla

- Graduate students learned about Schrödinger's equation, Survival Probability, Fortran, Command Line Tools, and Stampede.
- They developed two tools for this project
  - A backend database program used to upload survival probability data from Stampede to asok.nwmissouri.edu
  - A graph generator that would extract and displaying data from asok on a desktop or laptop

Directory Structure



Front End Graph Generator

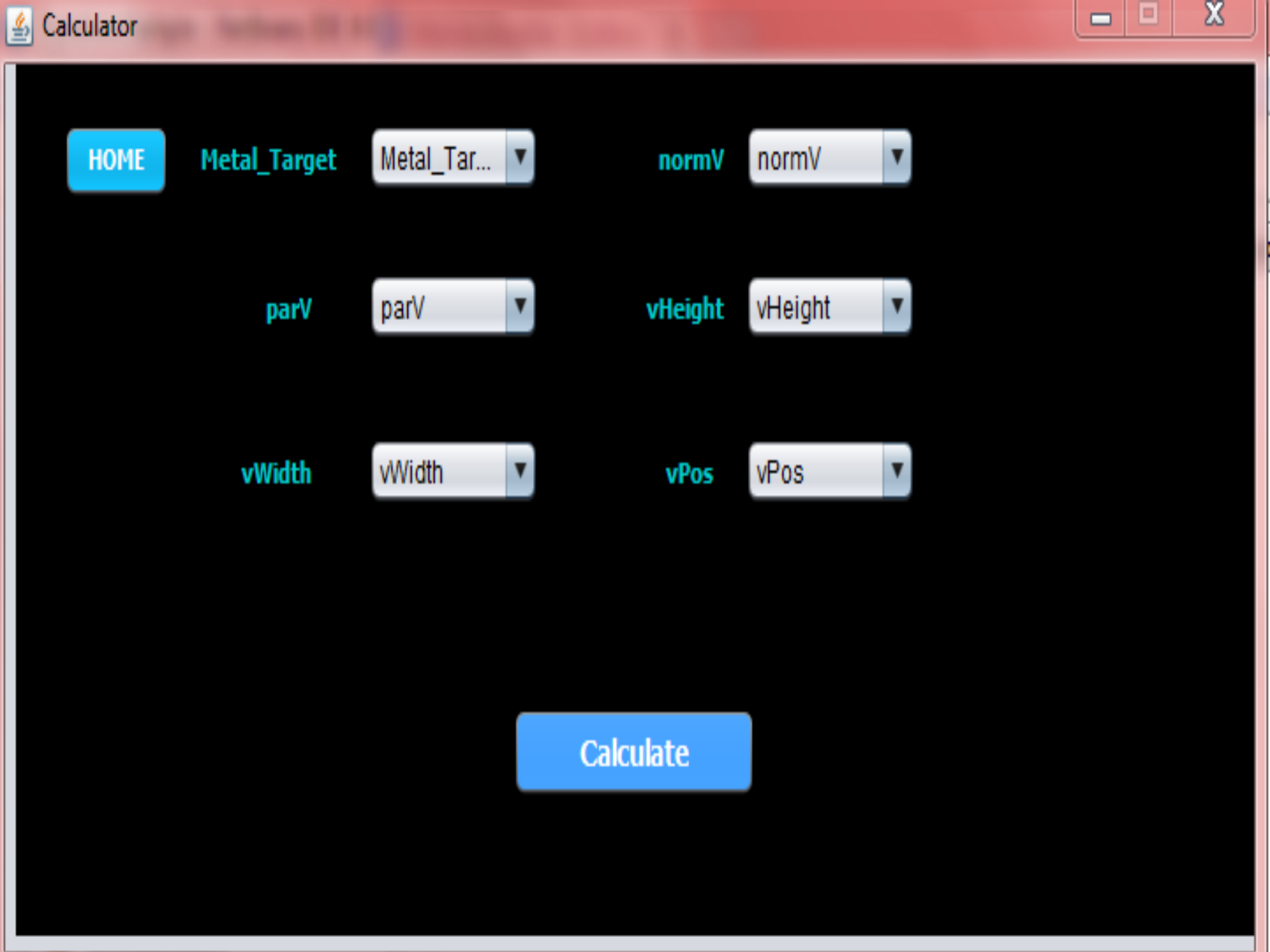
- Java Swing GUI
- MySQL connection via JDBC to asok.nwmissouri.edu
- Graph display via JasperReports

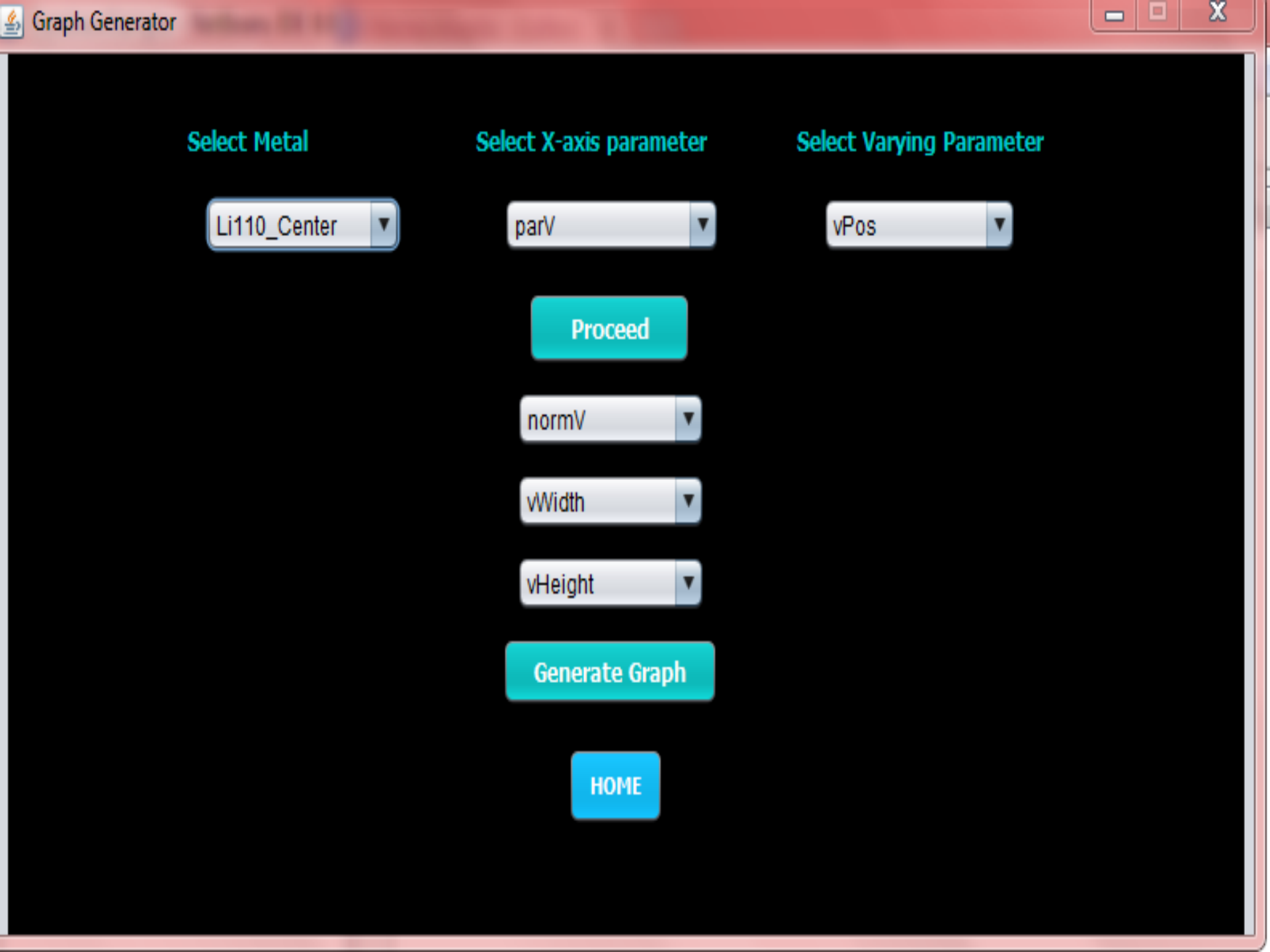
Hydrogen Ion Survival Probability

Hydrogen Ion Survival Probability

Calculator

Graph Generator


Calculator
HOME
Metal_Target
Metal_Tar...
normV
normV
parV
parV
vHeight
vHeight
vWidth
vWidth
vPos
vPos
Calculate


Graph Generator
Select Metal
Select X-axis parameter
Select Varying Parameter
Li110_Center
parV
vPos
Proceed
normV
vWidth
vHeight
Generate Graph
HOME

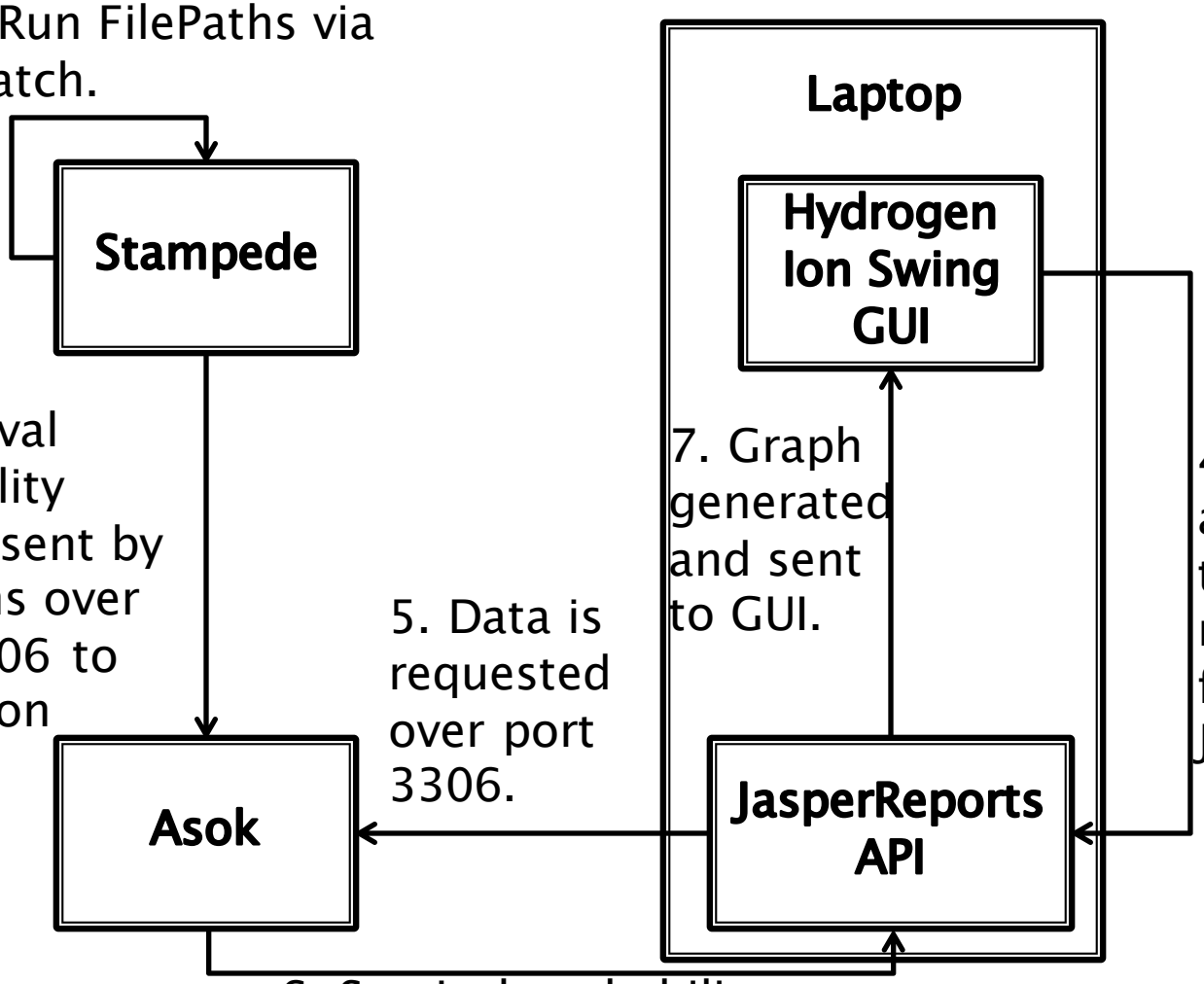
Back End Tools

- FilePaths – a Java program and related batch script to traverse directories and upload data created after running a cn2Ds parameter sweep [42].

SurvivalProbabilities
metal_target
normV
parV
vHeight
vWidth
vPos
prob

Workflow Diagram

1. Generate results with TACC Launcher and cn2Ds.
2. Run FilePaths via sbatch.



3. Survival Probability Data is sent by FilePaths over port 3306 to MySQL on Asok.

5. Data is requested over port 3306.

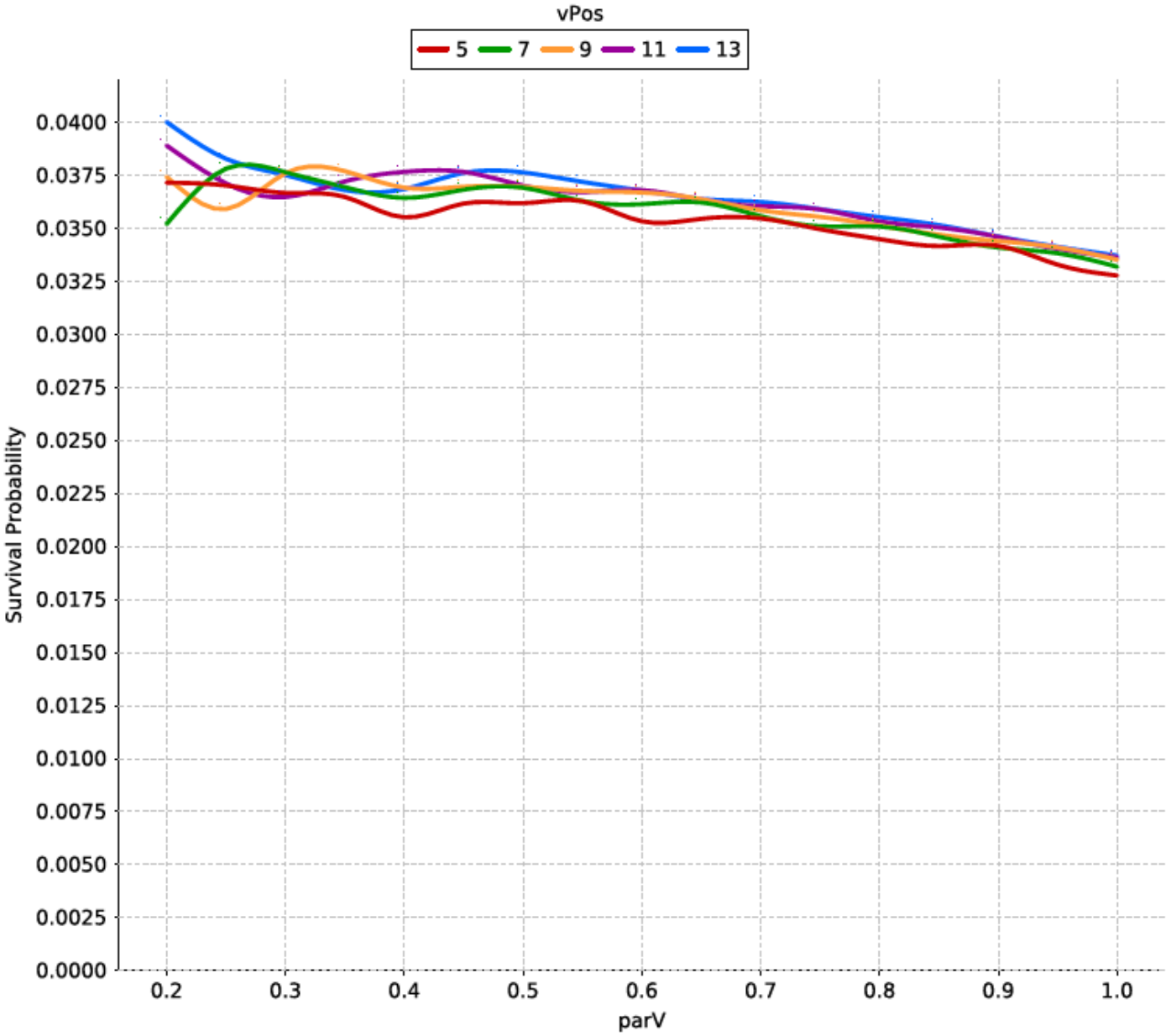
7. Graph generated and sent to GUI.

4. User requests a graph through the GUI and request is forwarded to Jasper.

6. Survival probability data is sent to Jasper.

[36-39,42]

Example Graph



Graduate Student Involvement Recap

- Students developed two tools for use on this project
  - Backend tool to allow for database connectivity and storage.
  - GUI to allow for data retrieval and display of graphs.
- Graduate students learned about Linux command line tools, data management, GUI design, and usability.

Faculty Involvement

- XSEDE XRAC Allocation
- Running large parameter sweeps
- Profiling and optimizing the cn2Ds code
- Review of updated results

XRAC Allocation

- Wrote a proposal describing the project and its physics and a scaling and describing how computational resources would be used in Dec. 2014.
- Submitted Jan. 2015
- Requested 1.5M SUs to perform a parameter sweep of significant size.
- Awarded 624K SUs in March 2015.

Parameter Sweep Updates

- After parallelization only one instance of the program could execute per node.
- Launcher scripts were updated to reflect this and to run ten parallelized jobs per node in a serial fashion making use of the maximum wall time – 48 hours.
- Most jobs took less than 4 hours, so we chose ten total jobs per node as a “safe” number.
- Scripts were written to allow for multiple batches to be launched at the same time.
- The limiting factor was the number of jobs – a maximum of 50 active jobs can be running at once on Stampede [41].

Updates for Xeon Phi

- A smaller allocation and queue limits provided motivation to improve cn2Ds runtimes.
- The cn2Ds code was updated for use with the Xeon Phi accelerator.
- Half of the independent tridiagonal matrix computations were offloaded in each of three loops where the core computations were performed.
- Computations were performed asynchronously [40,41].

Xeon Phi Example Code

```
psisignal3 = 975
!dir$ offload begin target (mic:0) in(<input parms here>)
  inout(<inout parms here>) signal(psisignal3)
!$OMP PARALLEL DO PRIVATE(kx,psiin,psiout) SCHEDULE(guided)
do kz=1,xeonNZStep
  do kx=1,nxstep
    psiin(kx)=psi(kx,kz) ! load vector
  enddo
  !dir$ attributes offload:mic ::cn1Dpropx
  call cn1Dpropx(lowerx,diagx,upperx,psiin,psiout,nxstep)
  do kx=1,nxstep
    psi(kx,kz)=psiout(kx)  ! unload vector
  enddo
enddo
!$OMP END PARALLEL DO
!dir$ end offload
```

[22-27,35,40,41]

Example Code Continued

```
!$OMP PARALLEL DO
PRIVATE(kx,psiin,psiout)
SCHEDULE(guided)
! Repeat inner loop code on prev.
slide
!$OMP END PARALLEL DO
!dir$ offload_wait target(mic:0)
wait(psisignal3)
```

[22-27,35,40,41]

Runtime Improvement

normV	Number of Time Loops	Xeon Phi Run Time	Parallel Run Time	Serial Run Time
0.08	2013	00:18:59	00:29:07	02:54:29

- Completed prior to code optimization.
- Parameter values were vPos = 25, vHeight = 0.054, vWidth = 1, parV = 0.3.
- The time step was 0.25 atomic units [41].

cn2Ds Improved Runtime

- Profiling was performed with gprof.
- Identified and removed redundant code from pot_surf_2d_h.
- Tried different percentages of offloaded computations.
- Tried different numbers of Xeon Phi threads.
- Used a Biersack Ziegler trajectory and parameters vHeight = 0.027, vWidth = 2.0, vPos = 25, normV = 0.01, parV = 0.05.

gprof Results

```
Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  Ks/call  Ks/call  name
 26.68   9506.92  9506.92        1     9.51    17.60  MAIN__
 19.56  16476.99  6970.07
  __libm_error_support
 16.66  22413.45  5936.45                             exp.L
 13.06  27066.99  4653.55 2807680531   0.00     0.00  pot_surf2d_h_
  6.43  29359.22  2292.22 28776855     0.00     0.00  cn1dpropx_
  5.50  31320.13  1960.91                          __intel_ssse3_rep_memcpy
  4.06  32767.42  1447.30                             __svml_cexp2_e9
  3.18  33899.82  1132.39  1128246     0.00     0.00  cn1dprop_
  1.69  34502.63   602.82                             log.L
```

Notice that a significant amount of time is spent in pot_surf2d_h

Runtime Results

normV	Number of Xeon Phi Threads	Lines of data produced per second
0.01	60	2.475
0.01	120	2.039
0.01	240	1.6903

normV	Percentage of computations offloaded to Xeon Phi	Program Runtime
0.01	33%	3.20hrs
0.01	50%	3.18hrs
0.01	66%	4.30hrs

Additional Code Optimizations

- Additional attempts at optimization were made.
- Replaced several mathematical statements with faster counterparts.
  - E.g. Pre-computed divisions, multiplication instead of exponents, etc.
- Attempted Golden Section search for best amount of computations to offload.
  - Started with a range of 25% to 83.3% of offloading and narrowed down to the result that produced the best run time for one time loop.
  - Not very effective.
- Runtime was reduced to 2:57:54

PSC Greenfield Results

- Ran cn2Ds on Greenfield with 240 cores on Cu111, Center, vHeight=0.027, vWidth = 2.0, vPos=11, normV=0.03, parV = 1.0
  - Runtime was approximately 7 minutes.
- Dr. Shaw modified cn3D code to include optimizations.
- Dr. Monismith added OpenMP statements to cn3D to run on PSC Greenfield using Au111.

CPU Cores	Speedup	Program Runtime
1	1	02:11:23 (est.)
120	25.11	00:05:14
240	38.42	00:03:31

Future Work

- Continued work (debugging) on 3D code.
  - Currently results are dissimilar to the 2D results.
- Continued work on profiling and optimization
  - Recently discovered that removal of Xeon Phi code after optimizations had been completed resulted in 5x speedup on Stampede.
- Production of “videos” of firing the ion at the surface.
  - Much more data intensive.

References

- [1] "Engineering atomic and molecular nanostructures at surfaces"; J.V. Barth, G. Costantini, and K. Kern, *Nature* **437**, 671 (2005).
- [2] "Nanostructured surfaces: Challenges and frontiers in nanotechnology"; F. Rosei, *J. Phys.: Condens. Matter* **16**, S1373 (2004).
- [3] "Onset of catalytic activity of gold clusters on titania with the appearance of nanometallic properties"; M. Valden, X. Lai, and D.W. Goodman, *Science* **281**, 1647 (1998).
- [4] "Bonding of gold nanoclusters to oxygen vacancies on rutile TiO_2(110)"; E. Wahlstroem, N. Lopez, R. Schaub, P. Thostrup, A. Ronnau, C. Afeich, E. Laegsgaard, J.K. Norskov, and F. Besenbacher, *Phys. Rev. Lett.* **90**, 026101 (2003).
- [5] "Size- and support-dependency in the catalysis of Gold"; M. Haruta, *Catal. Today* **36**, 153 (1997).
- [6] B. Hammer and J.K. Norskov, in *Advances in Catalysis* (Elsevier, Amsterdam, 2000), Vol. 45, p. 71.
- [7] "Electron transmission through molecules and molecular interfaces"; A. Nitzan, *Annu. Rev. Phys. Chem.* **52**, 681 (2001).
- [8] "The signature of chemical valence in the electrical conduction through a single atom contact"; E. Scheer, N. Agrait, J.C. Cuevas, A.L. Yeyati, B. Ludoph, A. Martin-Rodero, G.R. Bollinger, J.M. van Ruitenbeek, and C. Urbina, *Nature* (London) **394**, 154 (1998).
- [9] "Beyond the surface atlas: A roadmap and gazetteer for surface symmetry and structure"; S.J. Pratt and S.J. Jenkins, *Surf. Sci. Rep.* **62**, 373 (2007).
- [10] "Electronic states at vicinal surfaces"; A. Mugarza and J.E. Ortega, *J. Phys.: Condens. Matter* **15**, S3281 (2003).
- [11] "Electronic wave function at vicinal surfaces: Switch from terrace to step modulation"; J.E. Ortega, S. Speller, A.R. Bachmann, A. Mascaraque, E.G. Michel, A. Naermann, A. Mugarza, A. Rubio, and F.J. Himpsel, *Phys. Rev. Lett.* **84**, 6110 (2000).
- [12] "Nanopatterning the electronic properties of gold surfaces with self-organized superlattices of metallic nanostructures"; C. Didiot, S. Pons, B. Kierren, Y. Fagot-Revurat, and D. Malterre, *Nature Nanotech.* **2**, 617 (2007).
- [12a]"Spatial imaging of two-dimensional electronic states in semiconductor quantum wells"; K. Suzuki, K. Kanisawa, C. Janer, S. Perraud, K. Takashina, T. Fujisawa, and Y. Hirayama, *Phys. Rev. Lett.* **91**, 136802 (2007).
- [12b]"Wave-function mapping of InAs quantum dots by scanning tunneling spectroscopy"; T. Maltezopoulos, A. Bolz, C. Meyer, C. Heyn, W. Hansen, M. Morgenstern, and R. Wiesendanger, *Phys. Rev. Lett.* **98**, 196804 (2003).
- [13] "Fermi surfaces of the two-dimensional surface states on vicinal Cu(111)"; F. Baumberger, T. Greber, and J. Osterwalder, *Phys. Rev. B* **64**, 195411 (2001).
- [14] "Step-induced one-dimensional surface state on Cu(332)"; F. Baumberger, T. Greber, and J. Osterwalder, *Phys. Rev. B* **62**, 15431 (2000).
- [15] "Image potential states on periodically corrugated metal surfaces"; J. Lei, H. Sun, K. W. Yu, S. Louie, and M.L. Cohen, *Phys. Rev. B* **63**, 045408 (2001).
- [16] "Band gap effect on H^- ion survival near Cu surfaces"; B. Bahrim, B. Makarenko, and J.W. Rabalais, *Surf. Sci.* **594**, 62 (2005).
- [17] "Charge exchange in Li scattering in Si surfaces"; Y. Yang and J.A. Yarmoff, *Phys. Rev. Lett.* **89**, 196102 (2002).
- [18] "Role of the 2D surface state continuum and projected band gap in charge transfer in front of Cu(111) surface"; T. Hecht, H. Winter, A.G. Borisov, J.P. Gauyacq, and A.K. Kazansky, *Phys. Rev. Lett.* **84**, 2517 (2000).
- [19] "Interaction time dependence of electron tunneling processes between an atom and a surface"; L. Guillemot and V.A. Esaulov, *Phys. Rev. Lett.* **82**, 4552 (1999).
- [20] "Electron transfer in the interaction of Fluorine and Hydrogen with Pd(100): The case of a transition metal versus jellium"; E. Sanchez, L. Guillemot, and V.A. Esaulov, *Phys. Rev. Lett.* **83**, 428 (1999).

References

- [21] "Relevant effects of localized atomic interactions and surface density of states on charge transfer in ion-surface collisions"; F. Bonetto, M.A. Romerio, E.A. Garcia, R. Vidal, J. Ferron, and E.C. Goldberg, *Europhys. Lett.* **80**, 53002 (2007).
- [22] "Resonant neutralization of H^- near Cu surfaces: Effects of the surface symmetry and ion trajectory"; H. S. Chakraborty, T. Niederhausen, and U. Thumm, *Phys. Rev. A* **70**, 052903 (2004).
- [23] "Effects of the surface Miller index on the resonant neutralization of hydrogen anions near Ag surfaces"; H. S. Chakraborty, T. Niederhausen, and U. Thumm, *Phys. Rev. A* **69**, 052901 (2004).
- [24] "On the effect of image states on resonant neutralization of hydrogen anions near metal surfaces"; H. S. Chakraborty, T. Niederhausen, and U. Thumm, *Nucl. Instrum. Methods B* **241**, 43 (2005).
- [25] "Band-gap-confinement and image-state-recapture effects in the survival of anions scattered from metal surfaces"; A. Schmitz, J. Shaw, H. S. Chakraborty, and U. Thumm, *Phys. Rev. A*. **81**, 042901 (2010).
- [26] "Image potential states of single and multi-walled carbon nanotubes"; M. Zamkov, H. S. Chakraborty, A. Habib, N. Woody, U. Thumm, and P. Richard, *Phys. Rev. A* **70**, 115419 (2004).
- [27] "Time-resolved photoimaging of image-potential states in carbon nanotubes"; M. Zamkov, N. Woody, B. Shan, H. S. Chakraborty, Z. Chang, U. Thumm, and P. Richard, *Phys. Rev. Lett.* **93**,156803 (2004).
- [28] "Image potential states of metal surfaces: Binding energies and wave functions"; E.V. Chulkov, V.M. Silkin, and P.M. Echenique, *Surf. Sci.* **437**, 330 (1999).
- [29] "Model pseudopotential for the (110) surface of fcc noble metals"; S.S. Tsirkin, S.V. Eremeev, and E.V. Chulkov, *Surf. Sci.* **604**, 804 (2010).
- [30] "Wave packet study of H^- decay in front of a metal surface"; V.A. Ermoshin and A.K. Kazansky, *Phys. Letts. A* **218**, 99 (1996).
- [31] "Lifetime of excited electronic states at surfaces: Comparison between the alkali/Cu(111) systems"; A.G. Borisov, J.P. Gauyacq, E.V. Chulkov, V.M. Silkin, and P.M. Echenique, *Phys. Rev. B* **65**, 235434 (2002).
- [32] "Probing adsorbate state lifetime with low energy ions"; J. Sjakste, A.G. Borisov, and J.P. Gauyacq, *Phys. Rev. Lett.* **92**, 156101 (2004).
- [33] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipes: The Art of Scientific Computing* (Cambridge University Press, Cambridge, 2007).
- [34] "Refined universal potentials in atomic collisions"; J.P. Biersack and J.F. Ziegler, *Nucl. Instrum. Meth.* **194**, 93 (1982).
- [35] OpenMP Architecture Review Board (2011), *OpenMP Application Programmer Interface Version 3.1*, Retrieved from http://www.openmp.org/mp-documents/OpenMP3.1.pdf
- [36] "Launcher: A Shell-based Framework for Rapid Development of Parallel Parametric Studies"; L.A. Wilson and J.M. Fonner, *Proceedings of the Extreme Science and Engineering Discovery Environment: Engaging Communities (XSEDE14)*, Atlanta, GA, USA, July 2014.
- [37] Jaspersoft Corporation (2014), JasperReports Library: Open Source Java Reporting Library, Retrieved from http://community.jaspersoft.com/project/jasperreports-library
- [38] Oracle Corporation (2014), Java SE 7 Swing APIs and Developer Guides, Retrieved from http://docs.oracle.com/javase/7/docs/technotes/guides/swing/
- [39] Oracle Corporation (2014), MySQL 5.6 Reference Manual, Retrieved from http://dev.mysql.com/doc/refman/5.6/en/
- [40] Intel Corporation (2014), User and Reference Guide for the Intel Fortran Compiler 15.0, Retrieved from https://software.intel.com/en-us/compiler_15.0_ug_f
- [41] "Computational Simulations of Electronic Motions and Excitations in Nanostructured Surfaces by Ion Surface Charge"; Y. Zhang, J. Shaw, D. Monismith, and H. Chakraborty, *Missouri Academy of Science 51st Annual Meeting, Missouri Western State University,* St. Joseph, MO, Apr. 18, 2015.
- [42] "Graduate Directed Project: Hydrogen Ion Survival, Final Presentation"; A.Manne, D. Narsina, T. Dingari, R.V. Kosuri, V. Nidasanametla, S. Chowdary, V. Pidikiti, D. Salla, Northwest Missouri State University, Maryville, MO, March 2015.
- [43] XSEDE (2015), XSEDE User Portal Allocations Policy , Retrieved from https://portal.xsede.org/allocation-policies .
- [44] TACC (2015), TACC Stampede User Guide, TACC User Portal, Retrieved from https://portal.tacc.utexas.edu/user-guides/stampede .

Acknowledgement

- This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1053575. Specifically, it used TACC Stampede and the Greenfield system at the Pittsburgh Supercomputing Center (PSC).
- The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this paper.
  URL: http://www.tacc.utexas.edu
- This research was supported in part by NSF Grant Number PHY-1413799, “Photon Impact Ionization of Fullerene and Endofullerene Molecules: Cross Sections, Resonances, and Time Delays”