

How do I simulate problem X?

Joan Adler, Yaron Artzi, Liron ben Bashat, Tzipora Yael Izraeli, Meital Kreif, Ido Lavi, Alexander Leibenzon, Adam Levi, Itai Schlesinger, Elad Toledano, Uria Peretz, Yonatan Weisler and Alon Yagil

Technion - IIT, Haifa, Israel, 32000

E-mail: phr76ja@tx.technion.ac.il

Abstract. We present reports on projects from the Technion Physics Department's annual Computational Physics class that are in, near or far away from its lecturer's personal expertise.

1. Introduction

Learning numerical techniques of relevance to their theoretical or experimental research project is a valid reason for a graduate student to take a Computational Physics class. Most universities are not able to provide separate computational classes geared towards each sub-discipline and so lecturers can find themselves supervising projects falling outside their personal research expertise.

At the Technion last fall some 20 graduate and final year undergraduate students took the Computational Physics class. It is relatively easy to find simulation projects loosely connected to the lecturer's research area, in that case atomistic simulations of condensed matter or statistical physics. However it is important that each student carries out a project that is relevant and interesting for them, and can be completed with the appropriate effort. When the class includes students from areas such as relativity, optics, plasma physics and types of quantum physics which fall outside the lecturer's expertise in either topic or algorithm (or both) this must be accounted for. In some cases their supervisors hope said projects will jumpstart a computational component in their research groups. We present projects from the fall 2012 class and describe our pitfalls and successes.

In the next section articles based on presentations of projects from the class in recent years will be described. Details of the course structure were presented there and do not need to be repeated. Then some educational projects, designed for class use in undergraduate courses at the Technion will be introduced. Three groups of research projects - in condensed matter/statistical physics, close to condensed matter/statistical physics, and those beyond the lecturer's personal research experience will then be presented. Not all projects will be described in detail here, but a full list can be found on the class website [1].

2. Previous presentations and generalities

A presentation about the class and its syllabus [2] explained the process whereby each student submits a research proposal halfway through the course in order to decide on topic, language, operating system, visualization tools and algorithms. Although the majority of projects



described here are from the class (70% of the class grade), a few are final undergraduate projects of a slightly larger scope, and in some cases project students are asked to attend the class lectures to widen their algorithmic knowledge. In [3] two successful class projects from earlier years are described, and in [4], [5] and [6] three projects closely related to the Computational Physics group's research interests are introduced. The last is from the recent class, but is an integral part of the Computational Physics group's research and thus was published separately.

In all projects submission as a website is a requisite step. Code has to be ready for download with help files, default parameters must be proposed and the project must run on a LINUX computer (generally either Red Hat or Ubuntu at the moment). When appropriate, windows versions can also be presented. Visualization is heavily emphasised and the website on the classroom server (known as PHELAFEL) is available for student websites. Since the Technion Computational Physics Group has developed a Mesa/Qt based C++ Atomistic Visualization code, known as AViz [7, 8, 9], this is recommended when appropriate, but for two-dimensional systems PGPLOT is often used and sometimes OpenGL is appropriate. When the visualization does not involve discrete objects, MATLAB, MATHEMATICA or gnuplot are used. The PGPLOT codes usually create the graphics while the code is being run, but most of the AViz visualizations involve preparation of data files which are then visualized post preparation. Several of the projects provide instructions on how to output results from different programs into the .xyz AViz input files. Each year some new codes/systems enter the mix; this year it was python, php and C# (C-sharp) as well as additional supplements to OpenGL such as GLEW and GLUT.

3. Educational projects

Three dimensional visualizations of hydrogen atom wavefunctions are very helpful for teaching Modern Physics or Quantum Mechanics classes. The concept of electronic density is hard to grasp. Earlier [10] 3d Hydrogen Schroedinger wavefunctions with binned color and rotating images on a website were helpful and used in Modern Physics class. This approach uses an "off-label" application of AViz, wherein the dot representation, originally created to enable quick selection of viewing angle etc for atomistic samples, is used to create a translucent effect whereby the sample's interiors are visible. Subsequent to the analoglyphic stereo enhancement to AViz described in [4, 11], these were redone. The new website also has improved explanations of the generation process. While two color glasses give even more insight, just the images on a white background are quite useful. This project used MATHEMATICA to generate the data and AViz to view it. Two examples are given in Figures 1 and 2, the former of the $n = 3, l = 1, m = 1$ orbital and the latter of the $n = 4, l = 3, m = 2$ orbital of the H atom. On the website all images can be rotated to further aid in depth perception.

Another concept of Quantum Mechanics that is difficult for students to grasp is the notion of energy levels of infinite wells and how bosons and fermions take their different states therein. Since they are going to sit in class with smartphones and tablets above or below the desk, we may as well encourage them to use these tools for something beyond Facebook. The energy level app enables them to enter some number of different particles (selected both from different preassigned choices and added "on the fly") into a well and see how many select each energy state. Following on the talk at CCP2013 a realtime test was made to introduce the Higgs particle described earlier in the day; it took less than 1 minute on an iPhone. This project is written in php and accessible from every smartphone and tablet that it was tried on. A typical output image is given in Figure 3.

To complete the trio of elementary Quantum Mechanics education projects, a website with downloadable executable code to simulate the Stern-Gerlach experiment was prepared. The code is OpenGL with GLUT and comes with executable for both Windows 7 and Ubuntu/RedHat. The former requires another file (provided on the website) to be downloaded to run, the latter

some installation, again with instructions on the website. There are also sample outputs and movies on the website. The code shows atoms released on the left of a variable magnetic field that they pass through on their way to a screen on the right. Figure 4 shows how atoms with classical spins simply cover some region of the screen more or less uniformly. Figure 5 shows how spins quantized with $\pm 1/2$ spins form two lines, exactly as on the famous postcard [12] Gerlach sent to Niels Bohr in 1922.

The final project in this section is for a somewhat more advanced course, and concerns quantum scattering preconfigured for the Yukawa potential. It has a good interactive interface, and is python based, but is a little less straightforward to run for non-experts than the above three projects. Images can be seen at [1] and the link thereon.

4. Condensed matter/statistical physics projects

The simplest project in this group is one concerned with percolation on fractals using C++ to create the datafiles and AViz to visualize the results. Percolation, fractals and the Mersenne twister random number generator are introduced and the code is well documented. One result for the Sierpinski square pyramid is shown in Figure 6.

Two projects have been carried out using the LAMMPS [13] molecular dynamics code. Both include general instructions as well as Technion-specific files, which are probably also useful for other parallel machines that use the pbs queueing system. One concerns visualization of defects in graphene, and presents an additional C++ code to indentify defects as well as instructions on extracting the .xyz AViz input file from LAMMPS. The second was an exercise in creating results with LAMMPS that verify a research groups' earlier calculations with the IMD code. The system studied here was crack propogation in silicon. Again here examples of script files and input to that groups preferred visualization tool, atomeye, are given. Script files for the Technion Parallel computer, TAMNUN are also presented, and images are shown on [1].

The final project here concerns the temperature dependence of conductivity in graphene using MATHEMATICA and MATLAB. This project is a detailed numerical realization of a study by Hwang and Das Sarma [14], with downloadable, well documented, MATHEMATICA and MATLAB codes.

5. Biophysics projects

The projects in this section all apply techniques of statistical physics to biophysics models. They all also happen to involve a slightly more advanced level of programming than some of the others and provide optimized code with excellent instructions for installation and compilation of the codes.

The first is an significant extension of the concept from the animated simulated annealing PGPLOT code of Adler and Silverman [15] to a model for confined water molecules. In both cases the idea is to view the sample while carrying out annealing in order to monitor the progress to the required state. This project was made within cygwin [16] and used Fortran90, pushing the envelope of previous class experience. One snapshot is given in Figure 7.

Another interesting project is the simulation of a dilute polymer solution with the dissipative particles dynamics method. Here MATLAB with the parallel toolbox and GPUs was used and a visualization in MATLAB has been prepared for youtube. Both this and the previous project simulate and visualize concurrently.

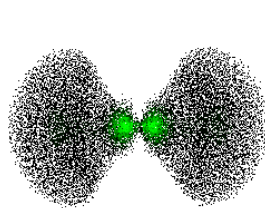
The last project with an image featured herein describes intermittent reaction paths of tfs (transcription factors). This project uses C# for the code; although an executable is also given on the website and AViz for the visualization, and gives instructions for both on Ubuntu, with monodevelop recommended for the former.

6. Optics and relativity

The three projects in this section are closely connected to specific research groups in the Physics Department, and more details may be published by those groups in the future. Two use MATLAB, all are at a high level. The first concerns solution of the Schroedinger equation and the second light propagation on glial cells.

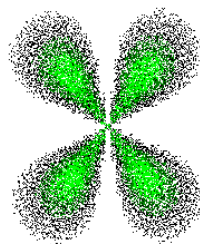
The third and final project to be described is in relativity, and concerns the numerical solution of the CGHS equations [17]. Although the equations date from 1991, and are considered to be a “toy model”, a non-pertubative numerical solution has been evasive. Unlike the “toy model” of statistical physics, viz. the Ising model, which has been studied ad nauseum, little intensive numerical effort has been made here. However, since the equations can be turned into difference equations and solved with a second order predictor corrector, it is amenable to a full numerical solution, if the intial conditions are chosen with care and sufficient computer power can be found. This is a classic example of applying very basic numerical methods to a problem outside the range of previous applications in the class programme, and achieving success. It was fun to present at CCP2013 where all teachers of basic numerical analysis enjoyed hearing from I. M. Khalatnikov about the early days of numerical methods for differential equations.

The C# equation solving code was originally designed for windows, but can be ported to the monodevelop in Ubuntu that was installed for the DNA project described above. The data plotting is done in MATLAB. Both parts have a nice interactive interfaces.



orbital 3p m=1

Figure 1. Image of the $n = 3, l = 1, m = 1$ orbital.



orbital 4f m=2

Figure 2. Image of the $n = 4, l = 3, m = 2$ orbital.

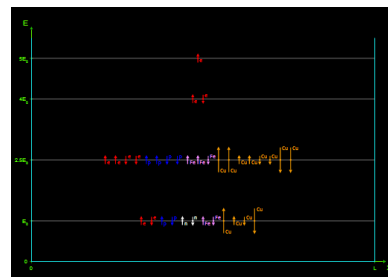


Figure 3. Energy levels in an infinite well.

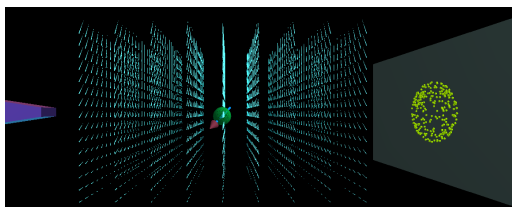


Figure 4. Stern-Gerlach experiment with classical spins.

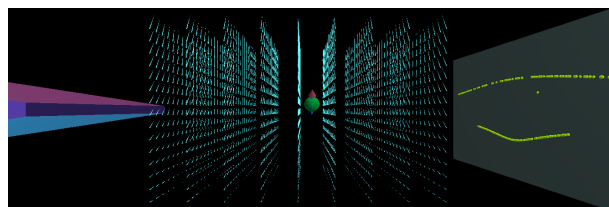


Figure 5. Stern-Gerlach experiment with $s = \pm(1/2)$ spins.

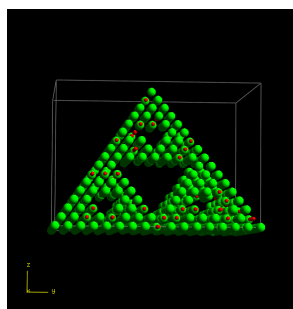


Figure 6. Sierpinski square pyramid (green balls) with 67 sites removed and indicated as small red balls.

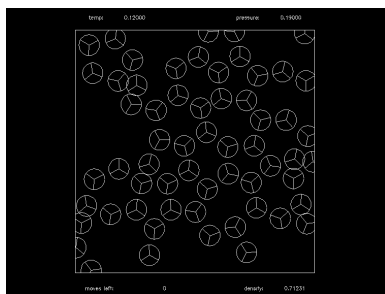


Figure 7. A “good enough” two dimensional ice model system with temperature of 0.12 and a pressure of 0.19.

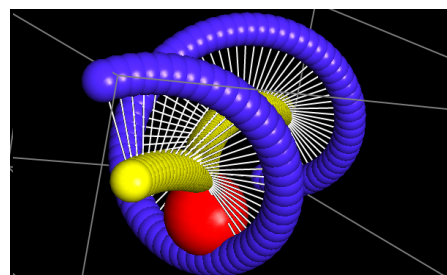


Figure 8. Double stranded DNA with diffusing protein (red ball).

7. Conclusions

The codes described above cover many areas of Computational Physics and we expect they will be useful both in Physics Education for future students, in Computational Physics Education and in Computational aspects of Physics research. Future extensions will no doubt be made to some of these, and already several new undergraduate physics projects are in progress, some of which extend these directions. Relevant updates will be placed on [1] as appropriate.

As well as usable codes and websites which should enable others to run and extend the codes, many participants learnt new tools, and some of those who already came with good computer skills learnt something about documentation and transfer to other operating systems. There was a lot of cooperation within the class. A word on pitfalls: while the end result in all cases was excellent, some flexibility in topics and goals was needed in a few cases, and everyone worked very hard throughout and after the semester.

Acknowledgments

We thank those class students whose projects are on the webpage but not mentioned explicitly above (due to separate publication or confidentiality) for also being part of the experience, and also acknowledge other departmental graduate students for helping class members in many ways. We thank J. Halupovich and D. Pomerancblum for help with OpenGL, and R. Kraizelman and Y. Katz for support and installation on classroom computers.

References

- [1] <http://phycomp.technion.ac.il/~comphy/projects2012.html>
- [2] J. Adler, “Educating the next generation of Computational Physicists” Physics Procedia, to appear (Athens, 2013).
- [3] D. Mazvovsky, G. Halioua and J. Adler, “The role of projects in (Computational) Physics Education”, Physics Procedia, **34**, p 1-5, (2012).
- [4] D. Peled, A. Silverman and J. Adler “3D visualization of atomistic simulations on every desktop’, IOP Conference Series, **454**, p. 012076, (2013), CCP2012.
- [5] J. Adler and O. Cohen, “Solid Hydrogen - New twists on an old problem” in “Recent Developments in Computer Simulation Studies in Condensed Matter Physics, X”, edited by D. P. Landau, S. P. Lewis and B. Schuttler, Physics Procedia **6**, p 2-6, (2010).
- [6] J. Adler, N. Ierushalmi, A. Silverman and R. Kalish, IOP Conference Series, to appear (Joao Pessoa, 2013).
- [7] <http://phycomp.technion.ac.il/~aviz>
- [8] J. Adler, A. Hashibon, N. Schreiber, A. Sorkin, S. Sorkin, and G. Wagner, “Visualization of MD and MC simulations for atomistic modeling”, Computer Physics Communications, **147** pp 665-9, (2002).

- [9] J. Adler, "Visualization in Atomistic and Spin Simulations", Computers in Science and Engineering, **5** pp 61-65, (2003).
- [10] J. Adler, J. Fox, R. Kalish, T. Mutat, A. Sorkin and E. Warszawski, "The essential role of visualization for modeling nanotubes and nanodiamond", Computer Physics Communications **177** pp 19-20, (2007).
- [11] J. Adler, Y. Koenka and A. Silverman, "Adventures in carbon visualization with AViz", Physics Procedia, **15** pp 7-16, (2011).
- [12] A. J. G. Hey and P. Walters, "The quantum universe", Cambridge University Press, (1987).
- [13] <http://lammps.sandia.gov>
- [14] E. H. Hwang and S. Das Sarma, "Screening induced temperature-dependent transport in two-dimensional graphene", Phys Rev. B **79** p.1654044, (2009).
- [15] A. Silverman and J. Adler, "Animated simulated annealing", Computers in Physics, **6**, pp. 277-281, (1992).
- [16] <http://www.cygwin.com>
- [17] C. Callan, S. Giddings, J. Harvey, A. Strominger, "Evanescent black holes", Phys. Rev. D **45** pp. 1005-9, (1992).