Computational Thinking for the Sciences

A Three Day Workshop For High School Science Teachers

Sheikh Iqbal Ahamed, Dennis Brylow, Rong Ge, Praveen Madiraju, Stephen J. Merrill, Craig A. Struble
Department of Mathematics, Statistics, and Computer Science
Marquette University
Milwaukee, WI, USA
firstname.lastname@marquette.edu

James P. Early
Department of Computer Sciences
SUNY Oswego
Oswego, NY, USA
earlyjp@cs.oswego.edu

ABSTRACT

This paper describes “Computational Thinking for the Sciences”, a 3-day summer workshop for high school science and mathematics teachers. Our workshop emphasizes the deep connections between the natural sciences, mathematics, and computer science through activities and simulation development appropriate for scientific explorations. Teachers were exposed to simulation development in VPython, a visual extension to the popular Python programming language. By broadening exposure of computational thinking to the natural sciences and developing activities specifically for science domains, we hope to attract new students into computer science and better prepare natural science students to employ the computational tools they will utilize in the future.

Categories and Subject Descriptors
K.3.2 [Computer and Info Science Education]: Computer science education

General Terms
Algorithms

Keywords
Computational thinking, K-12 outreach, high school science, high school mathematics

1. INTRODUCTION

In recent years, computer science has seen decreasing enrollments and growing gaps in participation by women and minorities. Yet, the need for trained computer scientists and training in computational thinking is growing. This is particularly true in the sciences, where the rate of data collection and generation outpaces our ability to interpret it.

In this paper, we describe “Computational Thinking for the Sciences” summer workshop, a 3-day workshop for 9-12 grade high school science and mathematics teachers sponsored by the Way Klingler College of Arts and Sciences and the Department of Mathematics, Statistics, and Computer Science at Marquette University. This workshop builds on the experiences from the “Linking Mathematics and Computer Science” workshop developed at Purdue University [6] and has the same overarching goals:

- Supporting K-12 teachers in their educational mission,
- Inspiring K-12 students in their studies,
- Increasing the pipeline of underrepresented populations into our undergraduate programs,
- Creating awareness of and interest in computer science as a discipline in the K-12 environment.

We had several reasons for expanding the workshop audience to include science teachers:

- Computational thinking is increasingly important in college and university science and mathematics training,
- Incorporating the sciences broadens the student pool exposed to computational concepts,
- Relatively few resources exist that integrate computational thinking directly into science contexts,
- Computation in the sciences demonstrates the team-oriented, interdisciplinary potential of computer science as a career.

Several initiatives already exist for disseminating knowledge about computer science to K-12 teachers and students. We briefly describe two that have gained significant attention. Computer Science Unplugged [1] is a freely available resource containing computer science activities without requiring a computer. The CS4HS [2] effort is an outreach program with workshops around the United States exposing K-12 teachers to computer science concepts. These initiatives provide outstanding resources for foundational computer science concepts and communicate these concepts to K-12 teachers. We leveraged some of these resources in our workshop.
Many initiatives, however, are largely divorced from direct applications in the sciences and have not emphasized the development of lesson plans for computation in the sciences. It was critical for our workshop to place computation in the context of addressing scientific questions. It was also important to have teachers leave with lesson plans using computational techniques that directly impact their courses in the upcoming year.

The SECANT project at Purdue University [5] has focused on embedding computational concepts directly into scientific disciplines at the college and university level. In SECANT, Python [9] and VPython [8] are used to develop simulations appropriate for different scientific disciplines. We used SECANT as a model for our workshop, but changed the structure and activities to be appropriate for high-school teachers. The emphasis, however, remained on developing and using simulations to augment scientific instruction. We saw tremendous opportunities for interactions between computer science and the natural sciences through simulations.

The remainder of this article provides workshop details. We begin with the logistics of running the workshop in Section 2, including a discussion of costs for others planning similar workshops. The individual sessions are described in Section 3, with brief overviews of activities and simulations that were developed. Evaluation of the workshop is presented in Section 4. This article concludes with a discussion of the outcomes and future work in Section 5.

2. WORKSHOP LOGISTICS

Our target audience for the workshop was area science teachers working with grades 9–12. We also accepted several teachers who taught courses in mathematics or computer science—some in a blended load with other science courses, and some who taught purely in those fields. We recruited participants through direct mail to area schools, postings on computing- and science-related e-mail lists for secondary school teachers, and our workshop website.

The structure of our workshop is based on Purdue’s summer workshop series linking secondary school mathematics with post-secondary computer science topics [6]. The schedule of the workshop is shown in Figure 1. The content of our workshop is novel, developed to appeal to the needs of American high school science teachers. In addition to modules addressing physics, biology, chemistry, and mathematics, we also designed modules for common areas of overlap, including probability, simulations, and computational thinking. These modules are described further in Section 3.

We chose VPython [8] as a tool to enhance our discussions and demonstrations of bringing computation into the high school science classroom. The tutorials included in the workshop schedule were designed not to equip our participants to teach VPython to their students, but rather to enable them to use VPython examples in their teaching. The participants demonstrated not only that they were capable of running a wide variety of simulations and demonstrations appropriate to their courses, but also that they could make adjustments to existing demonstrations and create new content.

As has become routine practice, we introduced teachers to Computer Science Unplugged [1], usually during breaks or topic transitions, as tools for use in their classrooms.

A secondary goal of our workshop was to be cost effective so other smaller and medium-sized universities can hold similar workshops. The total cost for the workshop was $6236.59, or about $173 per participant. The budget covered catering, a stipend of $300 for each teacher and guest speaker, parking (relevant for venues in cities), and note taking materials.

We did not provide lodging for participants because we limited our recruiting effort to schools within convenient driving distance of the metropolitan area. This differs from prior work where participants traveled significant distances as in [2]. The trade-off is that evenings were not available for instructional or laboratory time.

Involved faculty donated their time for planning and carrying out the workshop. Half a dozen meetings were held for planning prior to the workshop. Individual faculty took responsibility for key areas like advertising, catering, registration and scheduling. In addition to planning meetings, each faculty member prepared for approximately two weeks to research school science standards, learn VPython, and develop materials.

3. SESSION OVERVIEWS

Our workshop consisted of nine technical sessions, a session on careers in computational sciences, and additional sessions for teachers to prepare lesson plans with assistance from the workshop organizers. The first four technical sessions emphasized foundational material in computational thinking, simulations, probability, and VPython. The remaining technical sessions illustrated computational thinking concepts in scientific domain contexts through a combination of hands-on activities and simulation development. Simulation development was often done interactively with the participants, with the session leader guiding the simulation structure. Brief descriptions of each session fol-
low. When appropriate, related Wisconsin Model Academic Standards [11] from the Wisconsin Department of Public Instruction (WDPI) are listed at the end of each description. Similar standards apply in other states.

3.1 Computational Thinking

This session introduced participants to the workshop by describing our goals and the motivation behind those goals. We highlighted current concerns in computer science and general STEM education, including: low enrollments, lack of diversity, and lack of preparedness for STEM majors. Teachers were shown the “Pathways in Computer Science & Engineering” video developed at the University of Washington [3], to convey career possibilities that apply computational thinking in the natural sciences.

This session also discussed Jeanette Wing’s article on computational thinking [10] and emphasized the concepts of abstraction, algorithms, and simulation and their application to the sciences. The session ended with participants developing algorithms for some common tasks, such as making a peanut butter and jelly sandwich and sorting by middle names. Sample puzzles were also provided to illustrate computational thinking concepts.

This session is relevant to WDPI Technology Education Standard, C12.1.

3.2 Simulations

In the sciences, simulations play an important role in understanding and exploring ideas that would otherwise be infeasible due to cost or impracticality. Participants were introduced to the wide variety of simulations that might exist, and key concepts such as deterministic and stochastic simulations. The core of a simulation is a computational model, which is frequently expressed in mathematical forms.

We used Microsoft Excel 2007 to illustrate the development and application of simulations. We chose Excel since it is a commonly available tool and to demonstrate that simulations can be done without the need for traditional programming. One of the simulations investigated Newton’s Law of Cooling, which allowed participants to explore different cooling rates and ambient temperatures, to plot those explorations, and draw conclusions. Other simulation concepts covered including generating measurement noise, model fitting (using linear models), and extrapolation.

This activity is relevant to WDPI Mathematics Performance Standards A12.3, E12.1 and E12.2.

3.3 Probability

While probability is not strictly a high school science topic, it underlies ideas explored in physics, biology and chemistry, and is at the core of many computational simulations. In the workshop, we introduced probability with a simple version of the Monty Hall game, based on the television game show “Let’s Make a Deal.” This was used to illustrate interactive and non-interactive simulations and randomness.

A second round of workshop material on probability demonstrated simulated percolation, in which a fluid propagates through a grid of spaces randomly blocked or unblocked according to some probability, ρ. While an abstract implementation of this simulation is not difficult, this conceptual framework is applicable to scientific simulations in hydrogeology, semiconductor structure, and forest fire propagation.

Taking this another step, we constructed a VPython simulation that runs many random percolations with successive values of the blocking probability, ρ, to demonstrate a macro property “tipping point” of the problem that could not be predicted via closed-form equations.

This activity is relevant to WDPI Mathematics Performance Standards, E12.1, E12.2 and E12.5.

3.4 Python and VPython

The goal of this session was to provide teachers with tools they can use to inspire students to explore problems using computational thinking. Several lecture and lab exercises focused on VPython [8]. We chose VPython because its syntax is easy to learn, and users can produce complex animations with just a few lines of code.

None of the participants had experience with VPython (or Python), and only a few had any programming experience. So, our challenge was to present the material such that the participants could grasp key elements quickly. We prepared small lectures with corresponding lab exercises highlighting each of the following topics: i) objects, attributes, and methods; ii) conditional and iterative statements; iii) and graphing capabilities.

In Lab 1, participants created instances of sphere objects and manipulated various attributes (e.g., color, position in the scene). Through this exercise, they observed how changes to attributes cause the VPython interpreter to change the object rendering. Lab 2 introduced conditional and iterative statements with exercises built on ideas from Lab 1. For example, participants animated the Earth’s rotation about the Sun using two sphere objects. Producing a trace of the Earth’s rotation (using an arc object) was a bonus challenge.

In the final lab, teachers were given exercises highlighting graphing capabilities in VPython. They were shown the curve, gvbars, and gvhistogram objects, used to plot continuous curves, curves formed by vertical bars, and histograms respectively. They then produced a variety of plots in 2D using the range object to generate data.

3.5 Mathematics

The mathematics session of the workshop illustrated concepts behind cryptographic protocols and public key encryption using CS Unplugged [1]. The teachers formed groups with 2-4 teachers in each group. The idea was to have 1-2 teachers in each group play as “Amy”, another teacher play as “Bill”, and one other teacher play as intruder. Bill gives his public key in the form of a map to Amy. Amy encrypts the message and sends the encrypted message (map) to Bill. The intruder is also given the encrypted message, but cannot decrypt the message without the private key (map) which only Bill has. The interactive and simple examples from CS Unplugged conveys the important points about encryption.

This activity is relevant to WDPI Mathematics Performance Standard B.8.6.

3.6 Physics

We used two representative problems that both directly address existing high school standards and showcase the usefulness of computational simulations in the classroom.

The Inclined Plane is a ubiquitous physics problem. A rectangular block sits on a flat surface inclined at a particular angle. Gravitational force acts to accelerate the block
down the inclined plane, and frictional force acts to slow the down-
ward acceleration. In the simplest, static phrasing of the
problem, concepts explored include Newton’s Second Law
\(F = ma\), gravity, and trigonometry for calculating the
perpendicular and parallel components of gravity.

![Image of inclined plane simulation](image)

**Figure 2: Friction on an inclined plane.**

The VPython simulation constructed during the workshop
allows teachers to visually demonstrate the problem. Stu-
dents can also “play” to develop a greater intuition for the
forces at work by trying various angles of incline, and check-
ing simulation results against their calculations. Visible vec-
tors are included to show the forces and velocities at work.

The deluxe formulation of the inclined plane introduces
more advanced concepts, like static and kinetic friction, nor-
mal force, frictional coefficients, and summation of forces.
The full featured simulation includes a slide control that
changes the angle of incline in real time, allowing users to
slide the block back and forth on the plane.

The inclined plane simulation is directly relevant to WDPI
Physical Science Performance Standards D.12.7 and D.12.8.

### 3.7 Biology

Mendelian genetics is often taught with a combination of
mathematics and actual in vivo experiments with kits such as
Fast Plants [4]. Students compare the behavior of breed-
ing with the expected theoretical results, but frequently the
small plant sample does not match theoretical expectations.

A genetics simulation was developed in VPython, which simulated mating of a population consisting of dominant
and recessive alleles (e.g., green vs. yellow peas). Simulation
parameters include population size, number of generations,
and selection probability of dominant alleles. Teachers and
students can explore small populations, where Mendelian
theory breaks down, and larger populations where it holds.

### 3.8 Chemistry

The motivating application presented for computation in
chemistry was virtual screening in drug design, where small
molecules are computationally docked or placed into med-
ically important protein structures. Conceptually, docking
is similar to putting puzzle pieces together. Docking is ac-
complished by minimizing the energy that occurs after bond
formation between the atoms in the molecule and those in
the protein. A key component of computational docking
models is the Lennard Jones potential formula to model van
der Waals forces, \(V(r) = 4\epsilon \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^{6} \), where \(\epsilon\) is
the minimum energy potential, \(\sigma\) is the distance where the en-
ergy potential is 0, and \(r\) is the distance between atoms.

Teachers and the session leader developed an interactive
simulation to calculate the van der Waals forces between
two hydrogen atoms. In the simulation, dragging one of
the two hydrogen atoms plots the current energy potential with
a persistent trail to show the shape of the energy potential
function. Using this, one can discover the ideal distance
between two hydrogen atoms, which is where the energy
potential reaches its minimum.

Protein docking and the van der Waals simulation are di-
rectly related to WDPI Physical Science Performance Stan-

### 3.9 Careers in Computational Sciences

The goal of the careers session was to bring awareness
to high school educators regarding the possible careers for a
Computer Science/Computational major. The broad ranges
of topics covered in the session are: (i) career roles/titles, (ii)
career outlook, (iii) salary outlook, and (iv) women/under
represented minorities. Additional videos from the Univer-
sity of Washington [3] were used to inform the teachers that
computer science is not just about programming and sit-
ing in front of a computer. Several discussions with addi-
tional resources, such as information from the Bureau of
Labor Statistics, dispelled the myths about the impact of
outsourcing and the future of technology jobs. The teach-
ers communicated that this information was very useful and
that many negative stereotypes and perceptions of technol-
ogy careers are still propagated in their schools.

### 3.10 Lesson Plans

At the conclusion of the workshop, each participant pre-
sented lesson plans illustrating how they plan to incorpo-
rate computational thinking in their classroom. Most par-
ticipants used VPython simulations as part of their lesson
plan. Examples of simulations included conservation of mo-
mentum, the genetics simulation described in Section 3.7, an
isotope abundance estimator, cell volume vs. surface area,
and a derivation of the formula for the volume of a sphere
using increasing numbers of pyramids. One pair of partici-
ants used Excel to develop an analytical spreadsheet for a
seed germination laboratory, which explored model fitting,
interpolation, and extrapolation concepts.

### 4. EVALUATION

The effectiveness of our workshop was evaluated using a 14
statement anonymous survey given before and at the con-
clusion of the workshop (shown in Table 1). Participants
were asked to rate how strongly they believed a statement
5. CONCLUSIONS AND FUTURE WORK

This workshop is the initial step toward our ultimate goal for increasing high school students’ interest in the careers in STEM, and computer science in particular. We chose to introduce computational thinking to high school teachers to change long-term perceptions about computer science and its relationship with other disciplines. We are considering new workshops for high school students. Direct interaction with students expands the opportunities for activities, which will not have to conform to state curricular standards. The challenge, however, is having the necessary experience to make the topics interesting and content levels appropriate for high school students. We hope that partnerships with teachers attending the workshop can guide development of materials for the students.

Overall, the workshop met our designed goals for high school teachers. We measured improvements in defining computational thinking, understanding its role in the natural sciences, and in developing and using simulations in their classes. Work remains on making the material easier to incorporate into the high school curriculum, so that teachers have the confidence needed to include computational thinking concepts in their lessons. We will be sending a follow-up survey to the teachers after a full semester to see if the workshop had longer-term effects.

Python and VPython were very successful in this workshop. All teachers were engaged in the lab assignments and successfully completed the exercises, to different degrees of complexity. Given that most of the teachers had limited programming experience, it was encouraging to see that the skills to develop simple simulations including randomness, conditional logic, functions, and graphics could be taught in a very short period of time. We hope to hear that the teachers pass along Python and VPython to their students.

6. ACKNOWLEDGEMENTS

The authors thank the Helen Way Klingler College of Arts and Sciences at Marquette University for providing funding. We also thank Mindy Hart from the Purdue University Computer Science Department for her work on the Linking Mathematics and Computer Science workshop, which formed the foundation of our workshop. Finally, the authors deeply appreciate the participation of the high school teachers, who made the workshop enjoyable and an incredible success.

7. REFERENCES


Table 1: Statements used in our evaluation survey.

<table>
<thead>
<tr>
<th>Q</th>
<th>Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I can define computational thinking.</td>
</tr>
<tr>
<td>2</td>
<td>Computational thinking is important in the courses I teach.</td>
</tr>
<tr>
<td>3</td>
<td>Computational thinking is important in my field.</td>
</tr>
<tr>
<td>4</td>
<td>Computational thinking is too complex to use at the level of my classes.</td>
</tr>
<tr>
<td>5</td>
<td>I incorporate computational thinking in my classroom.</td>
</tr>
<tr>
<td>6</td>
<td>I am familiar with some computational tools.</td>
</tr>
<tr>
<td>7</td>
<td>I use computational tools in my classes.</td>
</tr>
<tr>
<td>8</td>
<td>I routinely look for computational tools to enhance my teaching.</td>
</tr>
<tr>
<td>9</td>
<td>I routinely update my lesson plan to incorporate computational tools.</td>
</tr>
<tr>
<td>10</td>
<td>I am familiar with general software applications such as MS Word and Excel.</td>
</tr>
<tr>
<td>11</td>
<td>I know a high level programming language such as C#, Java, C/C++, etc.</td>
</tr>
<tr>
<td>12</td>
<td>I believe my students can benefit from seeing or using computational tools in their science and math courses.</td>
</tr>
<tr>
<td>13</td>
<td>I am familiar with Python OR VPython.</td>
</tr>
<tr>
<td>14</td>
<td>I believe that incorporating computational thinking activities into my teaching could increase student interest in science and technology.</td>
</tr>
</tbody>
</table>

on a 5 point Likert scale: strongly disagree (SD), disagree (D), neutral (N), agree (A), and strongly agree (SA).

All twelve (12) participants completed the before-workshop survey, but only eleven (11) completed the after-workshop survey. Mean results are presented as (Q# before/after).

We saw substantial improvements in participants’ understanding of computational thinking (Q1 2.92/4.91), and importance to their fields (Q2 3.67/4.73, Q3 3.83/4.91). There is no change, however, in the perception of how complex it is to teach computational thinking (Q4 2.08/2.00). We also saw improvements in teaching computational thinking in their classrooms (Q5 2.58/3.91) and use of computational tools (Q6 3.33/4.08, Q7 3.08/3.72). We saw slight gains in perceptions of the role of computational tools and computational thinking in their classes (Q12 4.42/4.64, Q14 4.33/4.91). Interestingly, there is essentially no change in participants perception of knowing a high level language (Q11 2.25/2.00), yet all participants responded with A or SA to being familiar with VPython (Q13 1.08/4.27).