From: David Winch <dmwinch@mac.com>
Date: April 27, 2007 9:05:33 AM EDT
Subject: Re: new additions/updates to our PICCUP web site

Ruth
Thank you for the file - it appears that you are further along than most of us.
Dave

On Apr 26, 2007, at 12:27 PM, Ruth Chabay wrote:

> All,
> Attached is a zip file containing a web page illustrating the computational problems currently done by students in the E&M semester of the calculus-based intro course at NC State. Click on index.html to see the collection. Included are one sample handout, WebAssign assignment for turning in the program, and student program (please don't post this publicly, since it is a solution).

From: Dan Schroeder <dschroeder@weber.edu>
Date: April 27, 2007 11:15:49 AM EDT

I've been very impressed by the M&I approach since I first saw it. However, it would not work in its present form at my institution. At the beginning of the preface to the text it states: "It is assumed that you have had an introductory physics course in high school." My course has no such prerequisite, so we necessarily spend a lot of time on the basics of mechanics and electricity and optics, leaving less time for working more advanced problems.

A couple of other differences: M&I spends almost an entire semester on E&M. Our present course allocates only half a semester to E&M. (I suppose we could change this, but I would probably argue that other topics are more important.) Also, I teach at a commuter school where group work is practical only during a scheduled lab time. I suppose we might be willing to allocate one or two lab sessions per semester to computational projects, but I can't imagine it growing beyond that. I don't see students becoming comfortable with a programming environment like VPython in just one or two lab sessions.

Sorry to be such a skeptic and naysayer. I really am open to new ideas and perhaps some of you can think of ways to work around these boundary conditions.

From: "Kenneth Jones" <kennethjone@gmail.com>
Date: April 27, 2007 12:09:30 PM EDT

We believe that very different approaches my be appropriate in different environments. This is a perfect example. Dan, I don't see you as a skeptic or naysayer. You are simply expressing that fact that a "one size fits all" approach may not make sense. We wanted this kind of diversity of experience and views when we designed the workshop. Alternative paths to the same goals are encouraged and it should not be felt that agreement on all topics is necessary or even desirable.

From: David Winch <winch@taosnet.com>
Date: April 27, 2007 12:34:11 PM EDT

Dan
I did not take your remark as skepticism but rather a statement that each institution has its own unique conditions, constraints, etc. and it is doubtful there is one correct path.

From: Ruth Chabay <rwchabay@unity.ncsu.edu>
Date: April 27, 2007 2:22:36 PM EDT

I don't think the computational activities we use are specific to M&I -- doesn't everybody calculate the electric field of a rod? There are many factors that go into a choice of an appropriate curriculum for a particular institution, and I had not intended to advocate the use of M&I -- merely to show what computational activites we use in the intro course at NCSU. Also, I want to say up front that I don't think we have yet understood how to realize the full educational potential of the computational environment and activities we do use -- after 10 years of this I still feel like a beginner.

However, in response to a couple of Dan's comments below, I should say that 20% of our engineering and science students have not taken a prior physics course, and many of the others have had very rudimentary physics courses in high school (we see no difference in the performance of these groups). Almost none of them have ever written a program before. NCSU is also in part a communter school, and the only group work time is a 2 hr weekly lab period (we have no recitations). It seems to me that flexibility is necessary in a curriculum or component of a curriculum; computational activities need to be adapted to the local situation. This is one reason there are two versions of some of our activities -- the more limited ones make fewer demands on the students. It's certainly reasonable to question the use of a programming language instead of a more familiar environment. In our context it works because (1) we teach a **very** small number of programming concepts, and use them over and over (sometimes producing inelegant but functional code), (2) instruction on 3D vectors is integrated with learning to use VPython, which provides one of the few practical ways to visualize vectors in 3D, (3) VPython allows students to do vector algebra directly (i.e. it has vector objects, which one can manipulate algebraically), and (4) VPython programs are very short, with very little setup code required. It's also nice that VPython is free and runs on Windows, Linux, and MacOS, so students can put it on their own computers.

I think the biggest bang for the buck in these E&M activities probably comes from the interactive 3D nature of the output the students produce, since the 3D nature of fields is something that is very unfamiliar to most of them. The activity students actually like the best is animating the motion of a proton in a magnetic field -- it's very simple, but elicits oohs and aahs when they get it working. Educationally, another interesting point is that because in a large course (800-1000 students) all homework is done online, programming (oddly) is the one situation in which students must work symbolically.

From: Dan Schroeder <dschroeder@weber.edu>
Date: April 27, 2007 3:01:59 PM EDT

Ruth, I would be interested to know how many of those two-hour lab sessions (in each semester) you devote to VPython programming exercises.

What do you do with transfer students who have taken the first semester elsewhere and not learned VPython?

From: Ruth Chabay <rwchabay@ncsu.edu>
Date: April 27, 2007 5:27:01 PM EDT

Hi Dan,
In E&M, out of 12-14 labs, the students do 3 to 5 computational projects per semester (the details depend on the semester). Each takes one hour of a two-hour lab session. During the second hour, students are usually doing experiments, but sometimes they are working large problems on whiteboards in groups (since this is the only small-group time we have, any problem-solving practice they get must happen here). The other lab sessions usually combine experiments and problem-solving.

Many students enter the course knowing nothing about VPython or about 3D vectors, which are typically not really used in traditional mechanics courses. Therefore, the first exercise is usually an intro to VPython and 3D vectors (a refresher for those who do know these things). The students work in groups (this semester we have instituted formal cooperative groups, based on the Minnesota scheme). Because so many students are new to VPython, we haven't been able to teach them more advanced programming constructs, such as Python lists, which could be very useful in calculating fields at many observation locations -- so this restricts the scope of the problems they can do. However, it is still possible for them to calculate fields numerically at locations other than those included in the restricted analytical solutions (e.g. locations not on the axes of a dipole; locations not on the midline of a rod); and to experiment with approximations (how many point charges should the rod be divided into?).

In mechanics, there is somewhat more computation, since it allows students to predict motion iteratively with changing forces -- for example, a spacecraft moving around an Earth and a Moon (both fixed). Some of the computational problems involve using data from their own measurements (fancarts, masses and springs). We have tried various approaches. During the current semester (which ended today), I broke the mechanics problems into small chunks, so 7 of the labs involved computing (8, counting the stat mech calculations based on your AJP paper). Programs get recycled; for example, students add energy calculations and energy graphs to previously written programs (spacecraft, mass-spring...) Again, computing takes only one hour of a two hour lab (except this last week).

To give a bit more context, I should note that the goal of the lab experiments in these courses is to connect theory to the real world. So students are usually measuring something and analyzing it using fundamental concepts and principles, instead of focusing on data collection and error analysis (which comes up only when it is really needed).

From: Dan Schroeder <dschroeder@weber.edu>
Date: April 27, 2007 8:33:20 PM EDT

Ruth,
The web page you sent lists what appear to be 8 different projects: 3D Vectors, Electric field of a charged particle, and so on, down to Charge motion in EM wave. Are you saying that in a typical semester, students would do only 3 to 5 of these projects? If they do only three, which three would these be?

You also included, as a sample, the instructions for one of the projects: Electric field of a ring. Are you saying that this project was completed by students working in groups during a single hour, that is, during half of a two-hour lab session?

Something's not adding up for me here.

One more question: How many students are in each group?

From: Ruth Chabay <rwchabay@ncsu.edu>
Date: April 27, 2007 9:44:48 PM EDT

Hi Dan,
Which programs they do depends on the details of the semester, but yes, they usually don't do them all. It's up to the person orchestrating the labs, and depends to some extent on the details of the academic calendar (there are usually no labs in weeks with a holiday.) For example, in Spring 2007 students did the field of a diplole, the field of a ring, and magnetic force (it happened that almost all of this group had used VPython in mechanics, so they didn't need an intro.) Here's the lab calendar:

http://courses.ncsu.edu/py208m/lec/003/Docs/CalendarLab.htm

In Spring 2005 they did a VPython intro, a point charge, a dipole, a rod, and magnetic force. In 2006 they did do the magnetic field of a moving charge (simpler version), and I forget what else. It varies, and it's best not to do the same thing each time.

Yes, 3 students work together for one hour on each program. Mostly they do finish during this time. There is a fair amount of scaffolding. We are working on rewriting the instructions to present ideas in bigger chunks and have more open-ended pieces. (At Carnegie Mellon we didn't give any scaffolding; at NCSU we probably went overboard in the other direction. We're aiming for some middle ground, but haven't had time to focus on this as much as we'd like.)