From: Paul Tiesinga <tiesinga@physics.unc.edu>
Date: April 3, 2007 12:02:34 PM EDT

Group 4 could indeed be more appropriate. I also do teach an intro mechanics and EM course. We (faculty teaching intro UG courses) are planning to include more computational examples in those intro courses, but we have not started yet, so the outcome of group 1 are directly relevant to us, but I can not contribute any direct experiences.

From: Richard Gass <gass@physics.uc.edu>
Date: April 6, 2007 2:55:32 PM EDT
Subject: Here is what I sent my workgroup

All--
The PICUP Workshop is rapidly approaching and I would like to starting thinking about and discussing the contribution of workgroup four which I have agreed to convene. I don't know if you know the other members of group four but they are Tim Sullivan at Keynon, John Mallinckrodt at Cal Poly Pomona, Paul Tiesinga at UNC Chapel Hill, John Deisz at the University of Northern Iowa and myself Richard Gass at the University of Cincinnati.

Group four is to focus on the design of a prototype module or exercise for a computational physics course or for a senior project. These two are obviously somewhat different so we will have to make an earlier decision about which direction we want to go. If you have not done so you may want to look at the PICUP web page http://CompPhysEd.shodor.org .

As you know the general topic of E&M has been chosen but we need to pick a particular problem (or maybe a linked set of problems) for the prototype module or exercise. In order to efficiently use our time at Argonne I would to start an email discussion about picking a problem with the goal of either settling on a problem or at least narrowing the field to a few contenders.

So what problem or problems do you think we should look at?

From: Richard Gass <gass@physics.uc.edu>
Date: April 11, 2007 2:26:41 PM EDT
Subject: Group 4

John Mallickrod has expressed a wish to move to group one. So that leave four of us. Since I have not heard any suggestions for topics. I will throw a few out to try to get the discussion started. Some possibilities are:

Solving Maxwell's equations using the finite difference time domain method (FDTD). Applications might include near field optics

Radiation from accelerating charges

Finite difference methods for the Poisson equation.

I don't want to suggest that these are the best choices. I just want to get some discussion started. I welcome comments, criticism and suggestions.

From: Richard Gass <gass@physics.uc.edu>
Date: April 18, 2007 8:47:31 AM EDT
Subject: Prep material for workshop

Since there has been some discussion of the FDTD method I am attaching a fairly recent paper which describes how to implement the method in a spreadsheet, thus making it suitable for introductory courses. We may want to post this on the web site.

On another topic I want to echo Bob's comment's about old approximation methods gaining new currency due to the ability of computer algebra systems to handle extremely large expressions. It may not be relevant for this workshop but I think we should define computation broadly and consider not just numerical work but also symbolic computation and visualization and the advantages that are sometimes gained from moving back and forth from one to the other.

[**File: FDTD.pdf**]

From: David Cook <david.m.cook@lawrence.edu>
Date: April 18, 2007 8:55:42 AM EDT

I had always included visualization and symbolic computation along with numerical work in the term "computing", and I thank Richard for realizing that we should probably make that assumption on my part at least explicit. As some of you have perhaps heard me say, however, I have become less enthusiastic about symbolic computing over the years because of the difficulties of persuading those programs to do with complicated expressions what your intuition says you ought to be able to do. You need to guide them very explicitly step by step, and that means you need to know a name for each operation along the way. Perhaps that is good, because you then have to know what you are doing, but my early enthusiasm imagined (naively as it turns out) that these programs would be a panacea for all sorts of complicated symbolic operations. Their graphic capabilities are fairly good, but I don't think that symbolic programs are the right tools for doing serious numerical computation, even though they have that capability. I would prefer that our students become acquainted with a symbolic manipulator (MAPLE or MATHEMATICA), an array processor (MATLAB or IDL), and a standard computing language (C or FORTRAN), but one can't do it all at once. If I were forced to choose one, I am fairly certain I would lean to the array processor.