AC 2007-2127: COMPUTATION IN UNDERGRADUATE PHYSICS: WHAT NEEDS TO CHANGE AND WHAT CHANGE CAN MEAN TO COMPUTATION IN ENGINEERING COURSES.

Norman Chonacky, Yale University

Norman Chonacky is currently the editor in chief of Computing in Science and Engineering, a joint publication of the American Institute of Physics and the IEEE-Computer Society. He is a retired physicist who has spent his career split evenly between being a physics faculty member and conducting applied science and engineering research. As an academic he worked to apply computers in physics educational settings and to evaluate the impact that various methodologies had on student learning. As a research scientist he worked in optics, atmospheric physics, applied computer science, sensor engineering, instrumentation, and environmental engineering. He has been associated with, among other institutions: the Southern Connecticut State and Columbia universities; Bates, Bowdoin, and The Evergreen State colleges; the Rome Air Development Center of the U.S. Air Force; and the Pacific Northwest National Laboratory. In addition to the ASEE, he is a currently a member of the American Association of Physics Teachers, the American Physical Society, the IEEE-Computer Society, and the American Association for the Advancement of Science. He received his Ph.D. in physics from the University of Wisconsin at Madison in 1967.

© American Society for Engineering Education, 2007

Computation in Undergraduate Physics: What needs to change and what change can mean to computation in engineering courses.

Abstract

During the past year a concerted movement has begun within the undergraduate physics community to grapple with the challenges of integrating computation into its curriculum. Engineering has a stake in this process for at least two reasons: physics departments generally teach the service course in physics taught to engineering students; and the ways in which computation is viewed and used in engineering differs between the physics and engineering communities. This paper describes recent developments within the physics education community with respect to integrating computation, and attempts to outline the common challenges the physics and engineering communities face and the opportunities they have to cooperate to their mutual benefit in curriculum development efforts.

This paper starts tracing recent physics education developments using data from a national survey that was commissioned by the magazine *Computing in Science and Engineering* (*CiSE*). This publication is co-sponsored by the American Institute of Physics and the IEEE-Computer Society, hence its interest in working at the intersection between physics and engineering. The paper continues with a description of an effort by the Committee on Instructional Technology – the counterpart to CoED within the American Institute of Physics Teachers – at the summer 2006 meeting of the AAPT to take "next steps" informed by the survey results and focused on promoting discussions based on examples of current physics curricular practices. Finally, it frames an agenda for examining possible ways to change the first-year physics course, which is taught as a service to engineering students, within the context of perceived similarities and differences in the roles that computation plays between physics and the engineering disciplines.

This paper is meant to stimulate a cooperative dialogue and participation between engineering and physics educators in a process as thought and discernment move toward reformative action.

Introduction

This paper presents evidence that the use of computation in undergraduate physics nationwide falls below the need for it as judged by physics instructors themselves. This situation has a potentially greater impact upon engineering than on physics in as much as computation plays a more prominent role in engineering than it does in current physics professional practices. In addition, it is arguable that the ways in which engineers use computation are considerably different, not only in practice but also in concept, from those of physicists. It follows that engineering stands to gain from a greater presence of computation in physics curricula, but only one thoughtfully designed by a dialogue between both these communities.

The analyses and conclusions in this paper are based on the perspective of an experimental physicist who has gradually, over the years, been drawn into computation along a number of different paths. These include the physics professoriate and experimental applied research and development work conducted in a number of engineering fields – optical, bio-medical, computational, and environmental engineering. This unconventional background provides a perspective well suited to compare and contrast computation usage among these engineering disciplines and physics, as well as to view the effects of professionals' educational experiences upon their work in laboratories.

It starts with a set of considerations – really, questions - that are problematic for faculty when they try to think about if, how, and when to integrate computation into their courses. Such questions are probably somewhat different depending upon whether one teaches engineering or physics. And yet, because in many institutions students from both fields meet in the introductory physics course, it is essential to address these questions regardless of which community one belongs to. These questions are only a subset, albeit fairly representative, of important issues. However it is useful to keep such questions in mind when considering the results of a national survey of computational use in undergraduate physics courses, which form the base data for this paper. It is within the context of these questions that one may draw inferences from those survey results.

A first category of considerations is centered on "skill sets." One question that always surfaces is: Should students program applications themselves, or use packaged programs? Another is: If asked to write programs, should it be done in terms of "third generation" standardized languages such as C, Java, or Fortran; or should they be using integrated application programming environments such as Matlab or Mathematica where the programming is in "fourth generation" languages, largely proprietary command sets? A related question is: Where should the training for either of these skill sets to take place - within the courses, as needed, in a separate course and if so then taught by which department, or are the basic programming skills or the tutorial facilities of the integrated application environments sufficient so that students should be left to learn "on their own?"

Another category of considerations is institutional in character. For example: Should the department adopt a "standard" set of computational tools for computation across all courses in the curriculum, or should individual faculty be left to choose their own? Should there be interdepartmental agreement on a basic set of computational tools, or is that impractical … and/or unsound? How are costs to be treated, and who should pay the bills?

Finally, there is a category of issues that are, for want of a better word "philosophical." Should computation in the "conventional " courses, which have not usually used computation, be treated as a supplementary "add-on", for example assigned as a homework exercise, or should the materials and presentations be "re-thought" in the light of computational possibilities? A related question is whether part of the grade for a course should be determined based on computational acumen of students, or should the computations be treated more like illustrative examples that may help understanding and appreciation, but for which students are not separately assessed and evaluated? Another consideration is the degree of students' preparedness. At entry, most all have used computers. However a prevalent if not predominant use, most particularly for males, is in gaming. The primary academic computational context in high school has most likely been calculation and not simulation or modeling. There is some perception that entering students in this generation are less prepared to do analytic manipulations and thus are less prepared to do mathematical operations. Does incorporating computation, especially at the introductory level, paper-over this defect and inadequately train students to think analytically at the material?

The Survey of Physics Computational Education Attitudes and Practices

In late 2005, the magazine *Computing in Science and Engineering* commissioned a national survey to determine how much and in what ways computation was being incorporated into the courses of undergraduate physics curricula.[1] This survey was conceived in response to the results of an employment survey suggesting that computation preparation of physics majors prepared them less than adequately for the work they were expected to perform.[2]

The survey instruments assumed two forms. The first was a solicitation of free-form essays from physics faculty in general describing what each was doing individually and within the context of their department to include computation as a tool for physics in classroom and labs. The second was a formal questionnaire composed of items inspired by an ethnographic analysis of those essays. Some of these items sought to assess the degree of current computational practices and their distribution among a separate course, the labs, and the standard lecture course offerings. Other items sought to assess attitudes toward integration of computation throughout the physics curriculum. This second solicitation went out to all department chairs as well as those individuals who had responded to the first survey instrument.

Responses to the first survey solicitation came from almost 140 individuals in as many institutions. The second survey solicitation, using the formal questionnaire, brought almost 190 responses, of which about 70 were from institutions represented in the first survey. Overall, about 250 of the 750 institutions (one-third) granting baccalaureate degrees in physics responded. Institutions represented ranged from community colleges (a few), through liberal arts colleges (the bulk), to major research universities and engineering schools.

The aggregated results from the questionnaire are easiest to present, and these appear below in either tabular or graphical form. The latter are based exclusively on the attitudinal items on the questionnaire -- either trend summaries or distributions. The discussions, that follow, however, are based on the descriptive essays as well as on these.

The questionnaire items themselves appear in Table 1. Note that, in the survey instrument as listed here, the attitudinal questions were placed in random order of the strength of their approval of computation, in order to avoid possible bias in the responses.

Survey of the roles of computation in your undergraduate physics curriculum

1. What is the approximate number of full time faculty in your department?
2. About what percentage of them require students to use computations in their courses as a part of the course grade?
3. In what ways has your department modified its traditional physics curriculum because of the existence of computers?
4. If your department created a separate computational physics course, please tell us here about it:
    a. prerequisites:
    b. number of credit hours:
    c. textbook used:
    d. software packages used:
    e. programming language:
    f. approximate enrollment:
    g. for how long it has been offered and now how often it is offered?
    h. is it required of all physics majors?
    i. do students with other than physics majors take this course as well?
5. Does your department require students to use computers in laboratory courses?

    If so, what software does your department use in the laboratory courses?
6. Does your department try to use numerical calculations in its non-laboratory courses?

    If so, in about what percentage of those courses are they used?
    What computer software is used in those courses?

Please rate your opinion about the following assertions on a five point scale where:
1= strongly disagree; 2= disagree; 3= neutral or no opinion; 4= agree; 5=strongly agree.

7. Computed numerical approaches to learning physics principles ought to share the stage with analytic approaches.
8. Analytic approaches to learning physics principles are necessary and sufficient for educating physics students.
9. Numerical approaches should be the major emphasis with analytic approaches reserved for statements of principles, *i.e.* as starting points for numerical calculations.
10. Numerical solutions should be used only to illustrate how the analytic solutions can be evaluated to yield actual values in specific cases.
11. Numerical modeling is too much of a departure from what physics does and should be used only, if at all, where analytic models do not work well.
12. Numerical modeling should be taught to prepare physicists for working in a world that can be modeled only numerically, except for the few special cases (*e.g.* harmonic oscillator) that we use over and over again.
13. Numerical modeling is a much-demanded skill in other sciences and engineering such that physics departments ought to commit to teaching such skills in our service courses.

Table 1: Questionnaire to survey computational physics use in undergraduate programs.

Table 2 contains the average values for quantitative items in the questionnaire. Where appropriate, these include standard deviation values. One immediately notes the very large variability in the distributions of responses for items #2 and #6. These suggest a wide divergence in the corresponding computational practices among the responding institutions. This is consistent with qualitative conclusions drawn from review of the descriptive essays and other narrative responses to the survey.

Item	No. responding	Value
2. Percentage of departmental faculty requiring computations for course grade?	167	47 ± 31%
4. Departments having a separate computational physics course:	252	53%
6. Percentage of department's non-lab courses using numerical calculations?	129	35 ± 28%

Table 2: Average results for quantitative items surveyed by the questionnaire.

The Survey Results: Distributions and Trends

The following figures describe the attitudinal results. The values represented are from a five-point Likert scale, where 1 denotes strong disagreement and 5 strong agreement with the assertion stated in each item. Note that in this analysis of attitudinal trends in responses, in contrast to the random ordering of items presented in the questionnaire as presented to the faculty, the order of presentation here corresponds to the degree to which their responses express recognition of the importance of computation in the curriculum. This reordering is clarified by the captions in each figure.

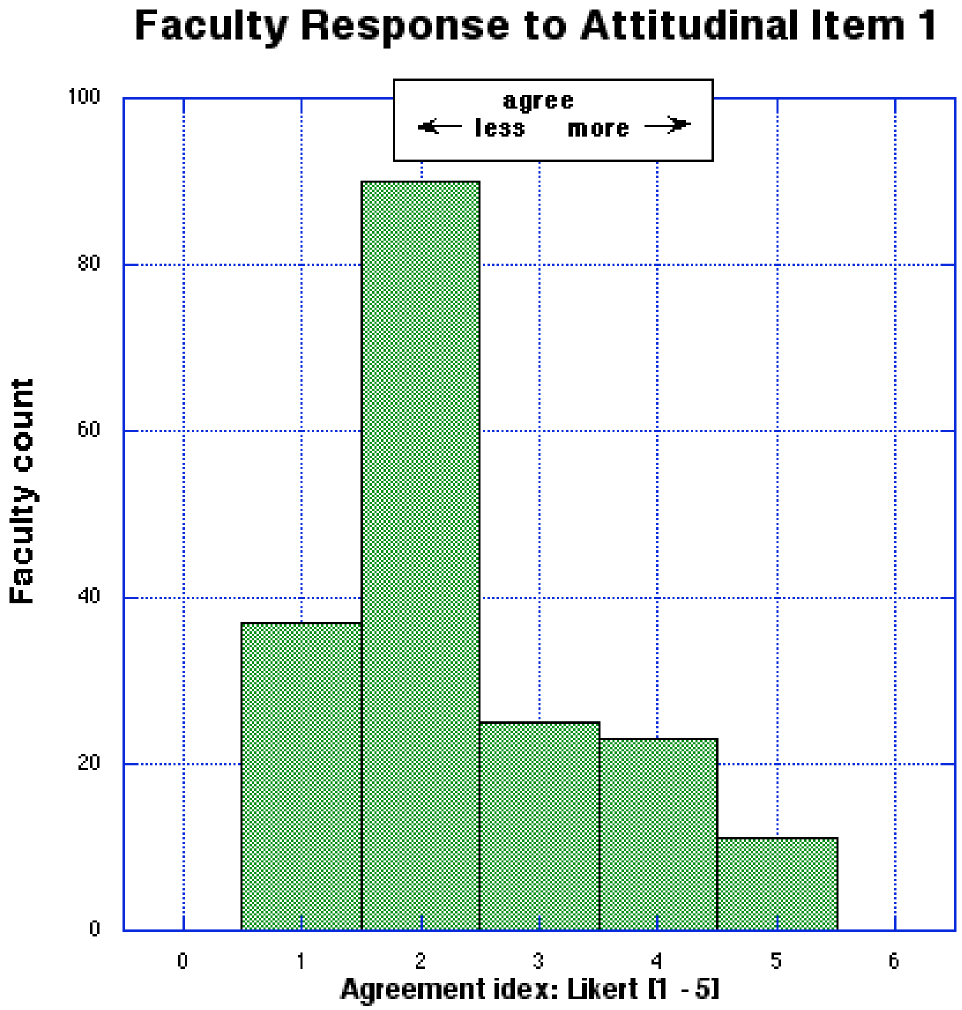

Figure 1: Item - Analytic approaches are necessary and sufficient.

These faculty would likely focus on analytic approaches and not devote time to include computation in their courses.

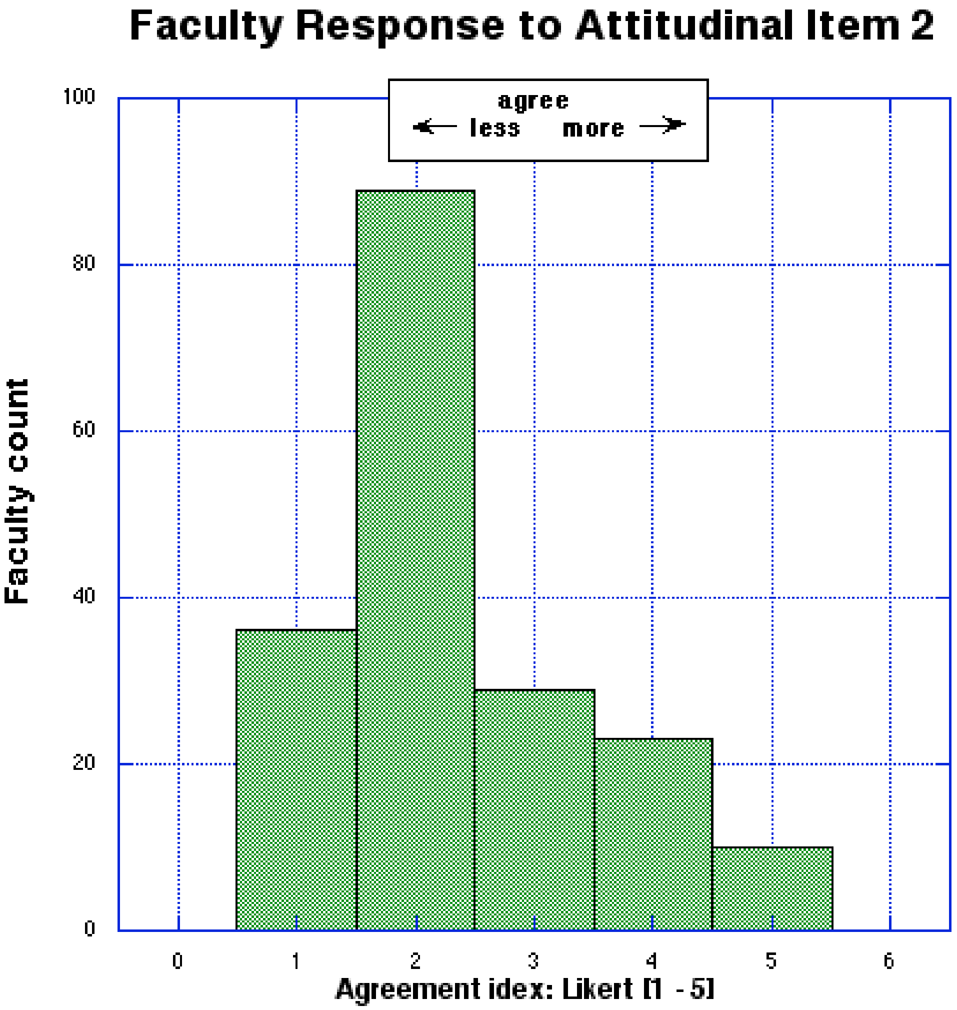

Figure 2: Item - Use numerical evaluations only to illustrate analytic solutions in specific cases.

These faculty would likely adhere to this traditional role for numerical applications in their courses, perhaps enhance by the visualization capabilities of computing.

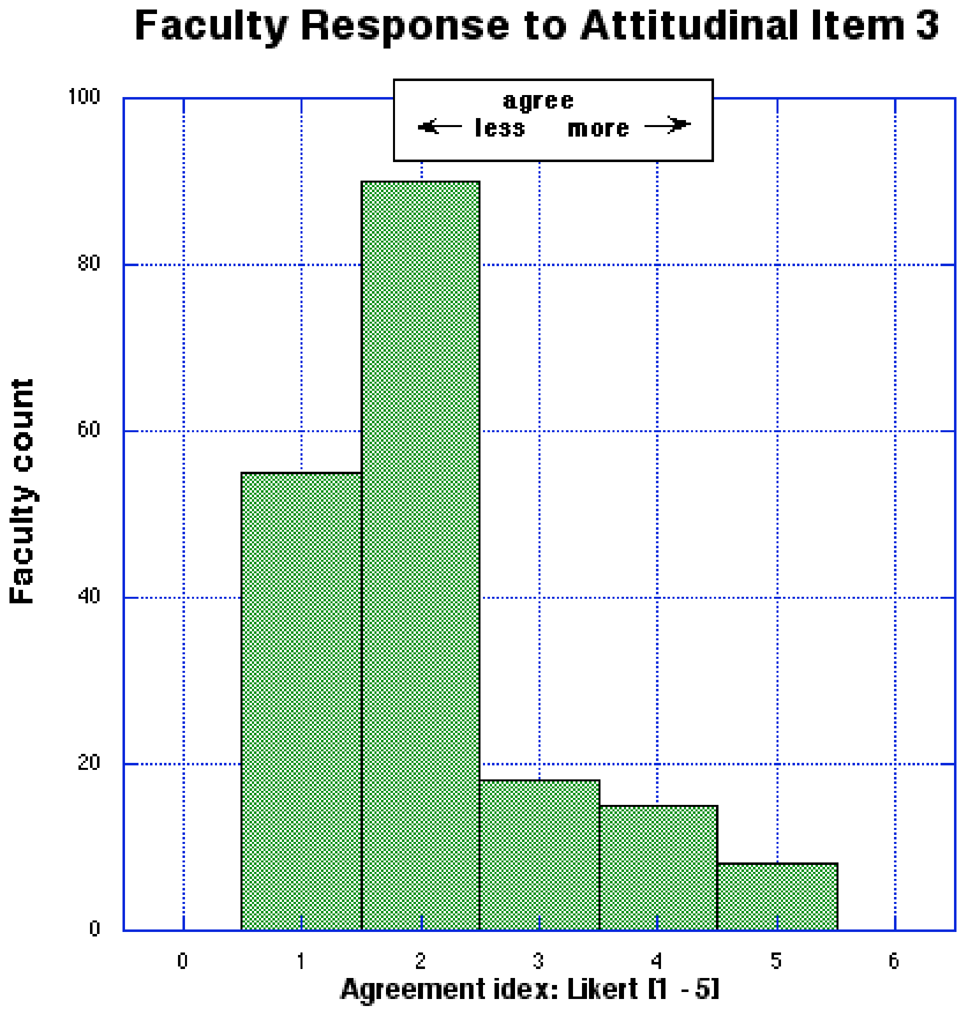

Figure 3: Item - Use numerical modeling only, if at all, where analytic models do not work well.

These faculty would likely use numerical modeling as a strategy of "last resort" if they chose to address cases where analytic models don't work. It is not clear how much such material they might include in their courses.

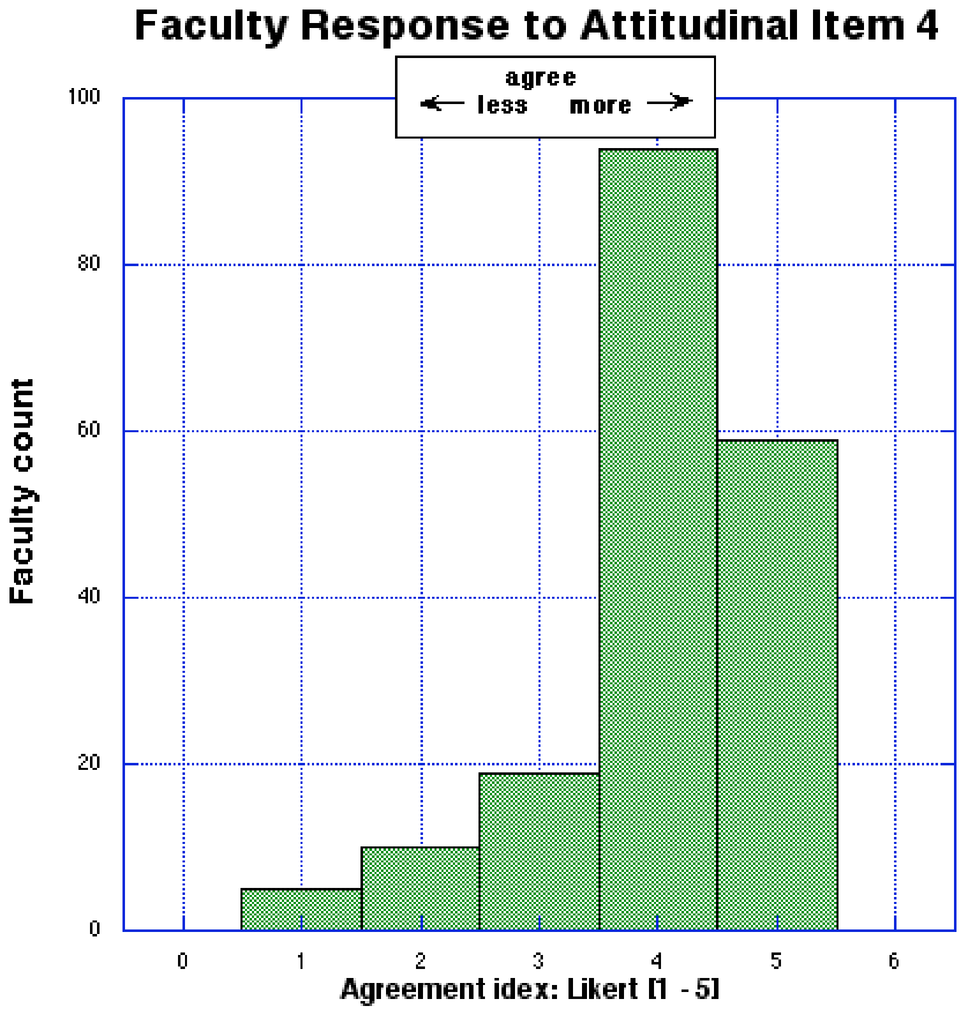

Figure 4: Item - Numerical approaches ought to share the learning stage with the analytical.

This is the "watershed point", where faculty are admitting parity in importance for including numerical computations and modeling in their physics courses.

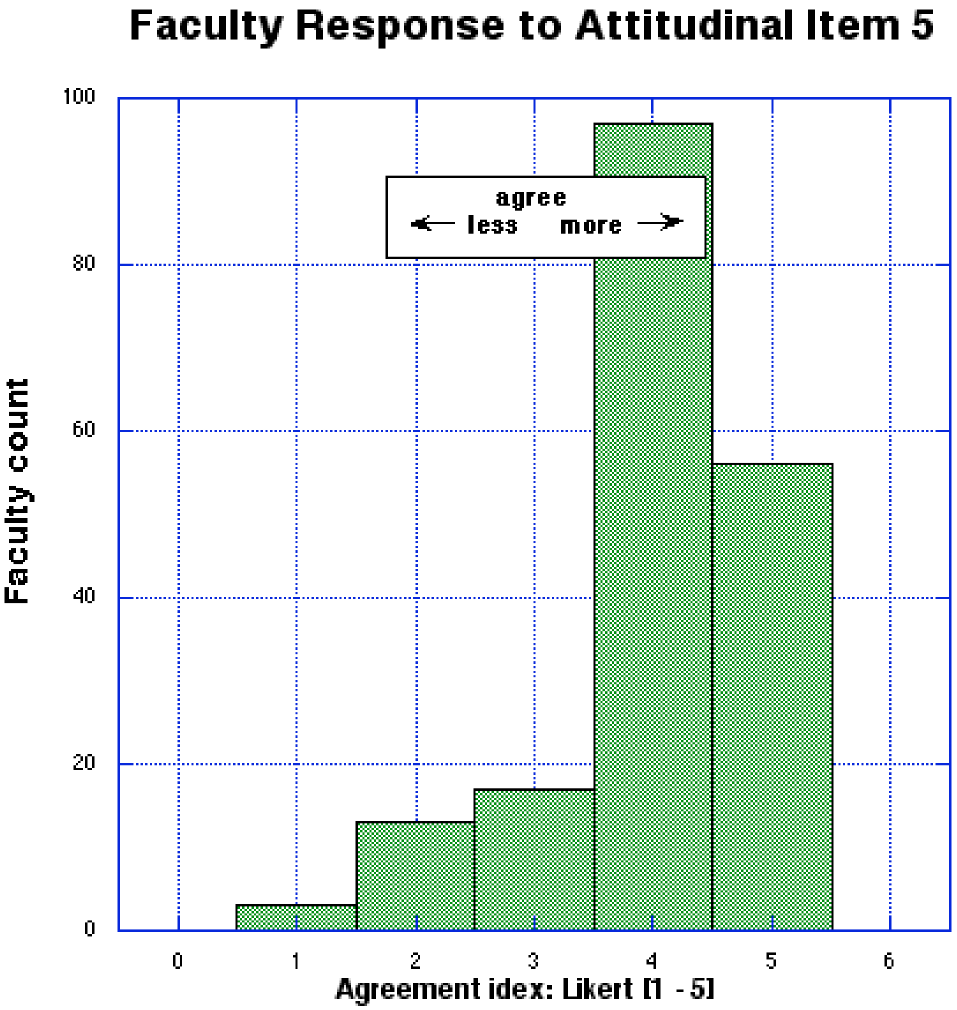

Figure 5: Item - Teach numerical modeling to prepare students to work in a world that can be modeled only numerically.

These faculty are acknowledging a preferred status for numerical modeling in their courses, at least for the utilitarian reason of its importance in the students' future careers. They very likely are, or will in the future, insure that numerical modeling is part of how they teach physics.

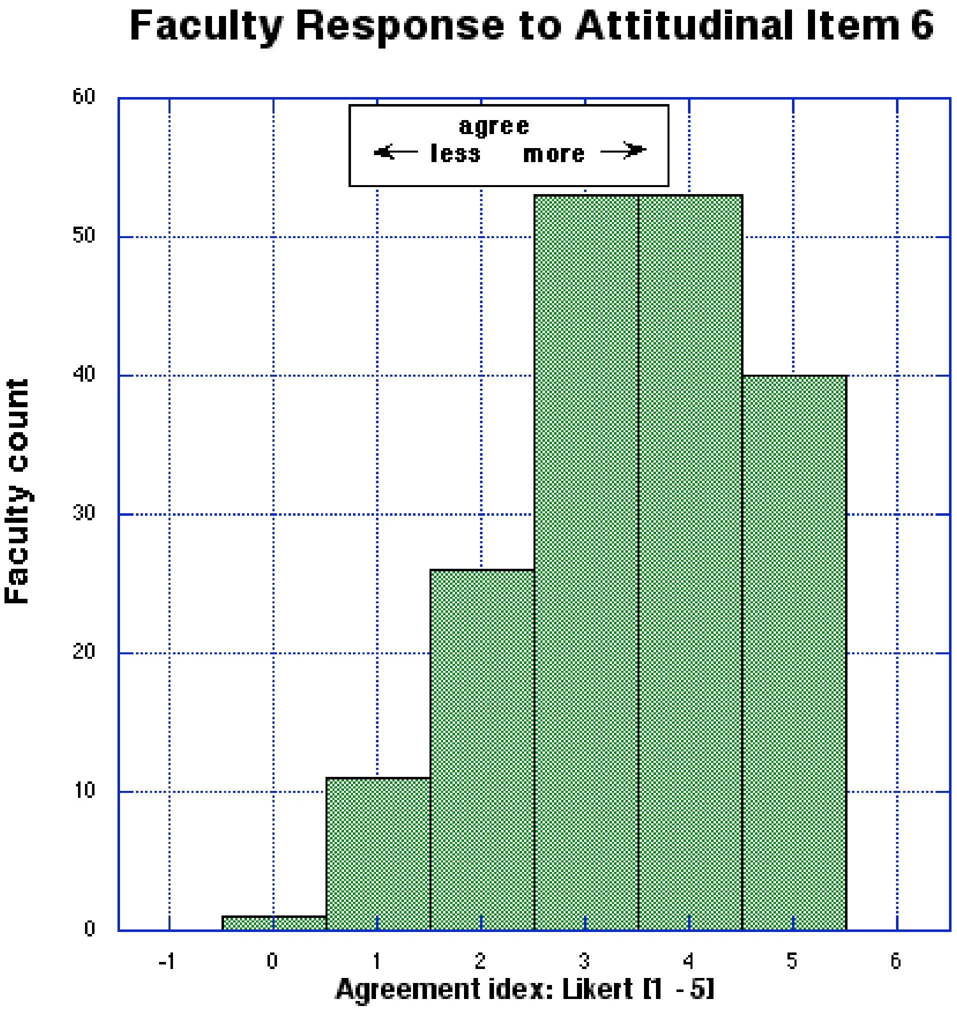

Figure 6: Item - Numerical modeling is a much-demanded skill in other sciences and engineering such that physics departments should teach it in their service courses.

These faculty are acknowledging that numerical modeling has a very important role in the futures of their students, regardless the kind of science or engineering they will be engaged in for their career work. The are probably the most responsive group in terms of their willingness to cooperate with engineering and other science departments to improve the computational content of the introductory, service physics courses.

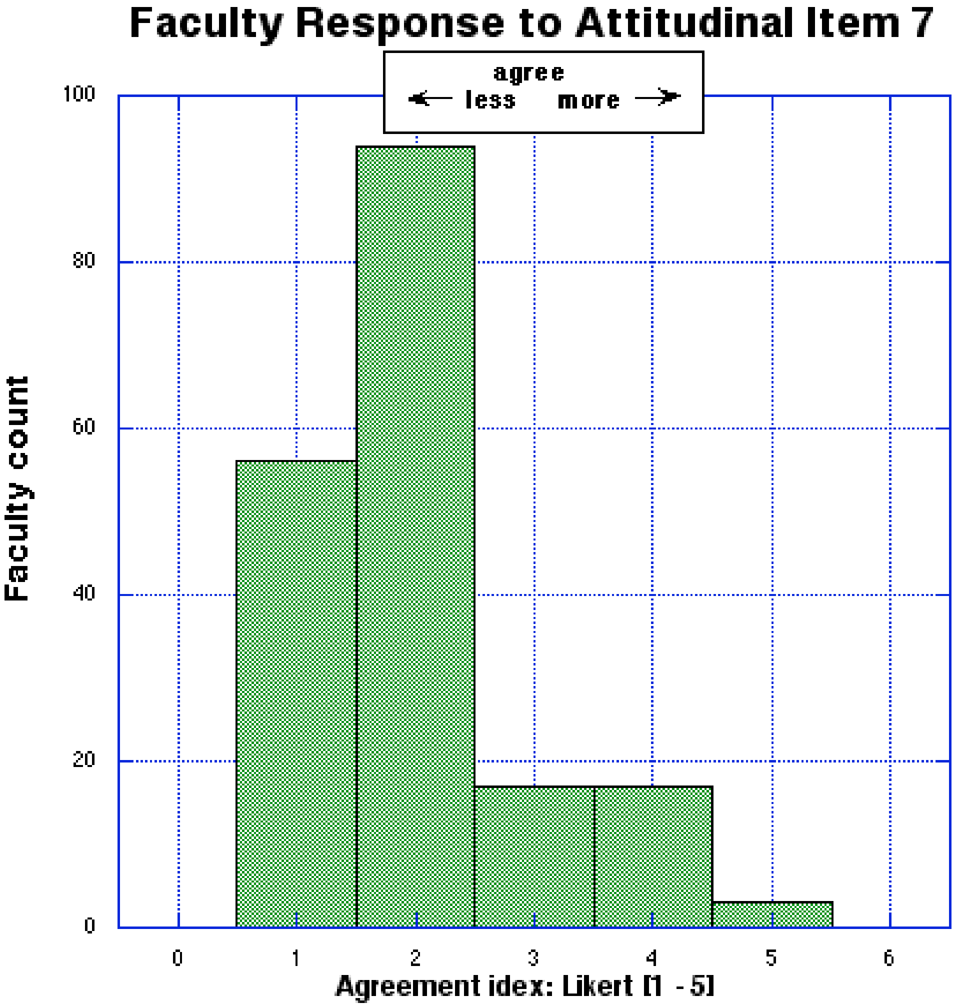

Figure 7: Item - Numerical approaches should be the major emphasis with analytic statements of principles used only as starting points for numerical models.

These faculty are assuming a perhaps (at the present time) radical view in which numerical modeling should be intimately integrated into the physics content. Presumably their objective in such integration would be to adjust the relationship between the analytical statement of physical principle and their numerical application to realistic systems. This would require that they "re-vision" the way in which they, and their students, think about physics practice.

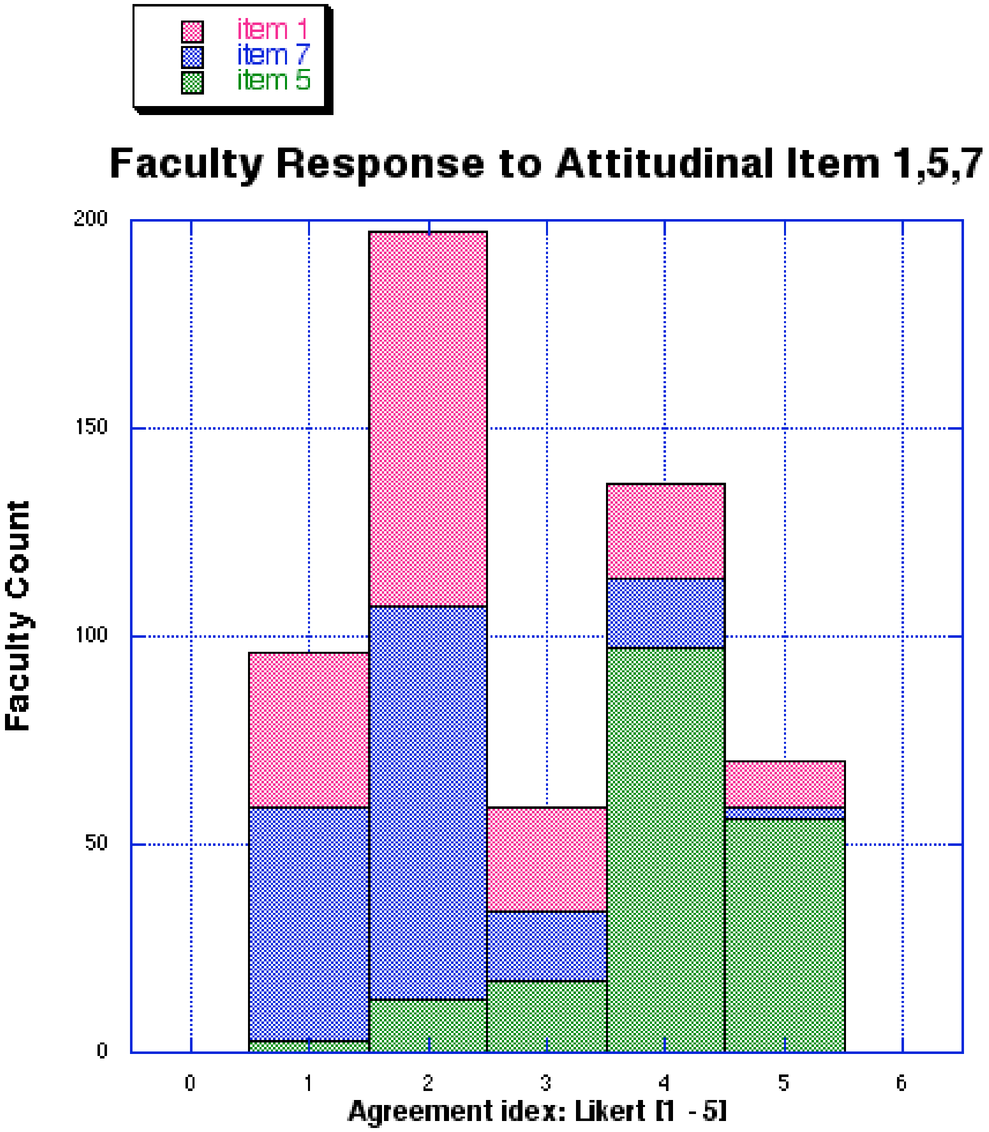

Figure 8: Comparative distributions for items #1, 5, and 7 above.

This rendering recapitulates the individual results of three previous figures, each of which represents a quite different role for computation in the curriculum. It rather clearly indicates that the current physics education community, as represented by our sample of responding institutions, is in the center with regard to two "extreme" attitudes toward computation in the undergraduate curriculum. On one extreme is business "as usual" and on the other is an ambitious rethinking of physics curricula in the light of new realities associated with evolutionary science and engineering practices and growing ease of access to computational power.

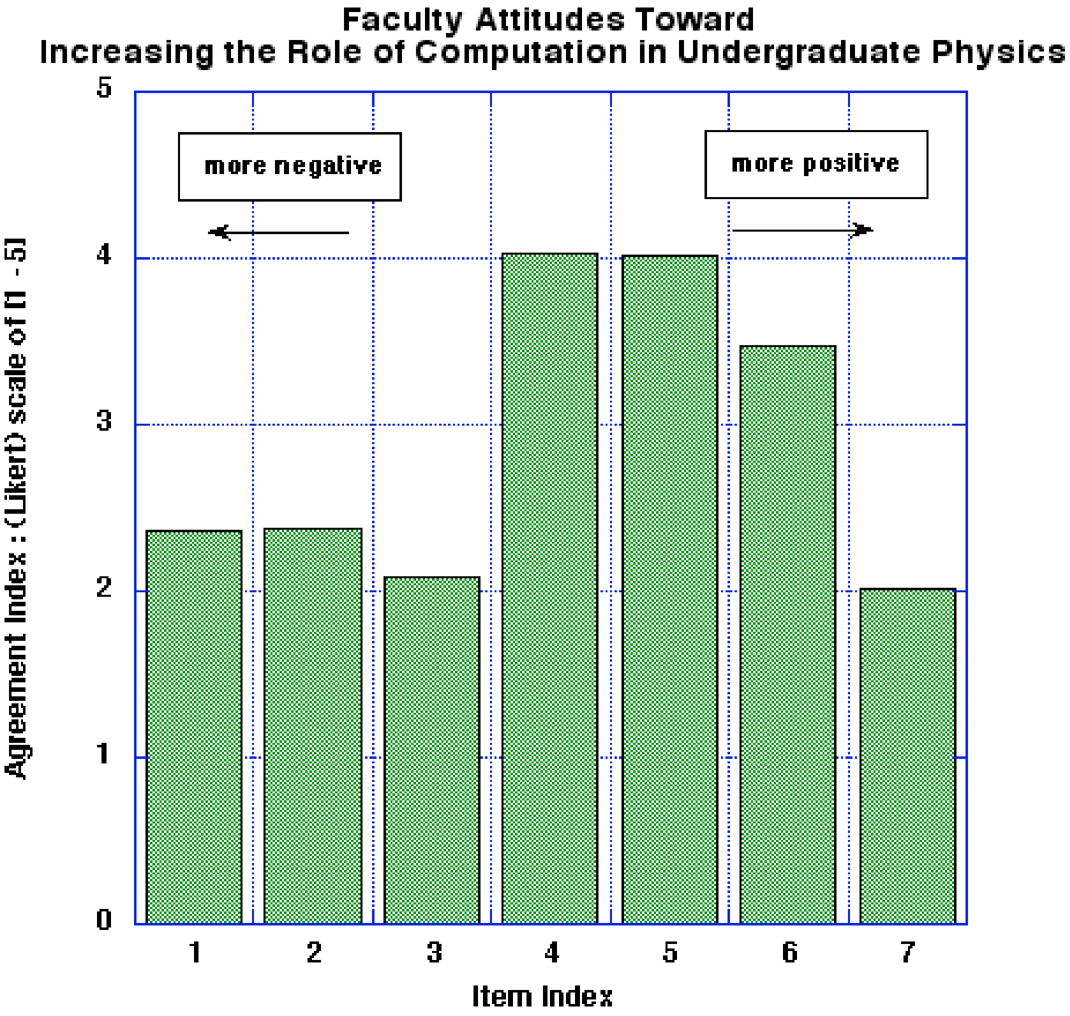

Figure 9: The trends among faculty regarding the degree of importance that they believe computation should play in physics courses.

This graph shows in greater detail the distribution of attitudes among faculty across the spectrum of possibilities offered in questionnaire items. These are indicated by the average value of their responses on the Likert scale for each attitudinal item. This reinforces the trend for faculty to have moderate to increasing supportive views for the importance of computation.

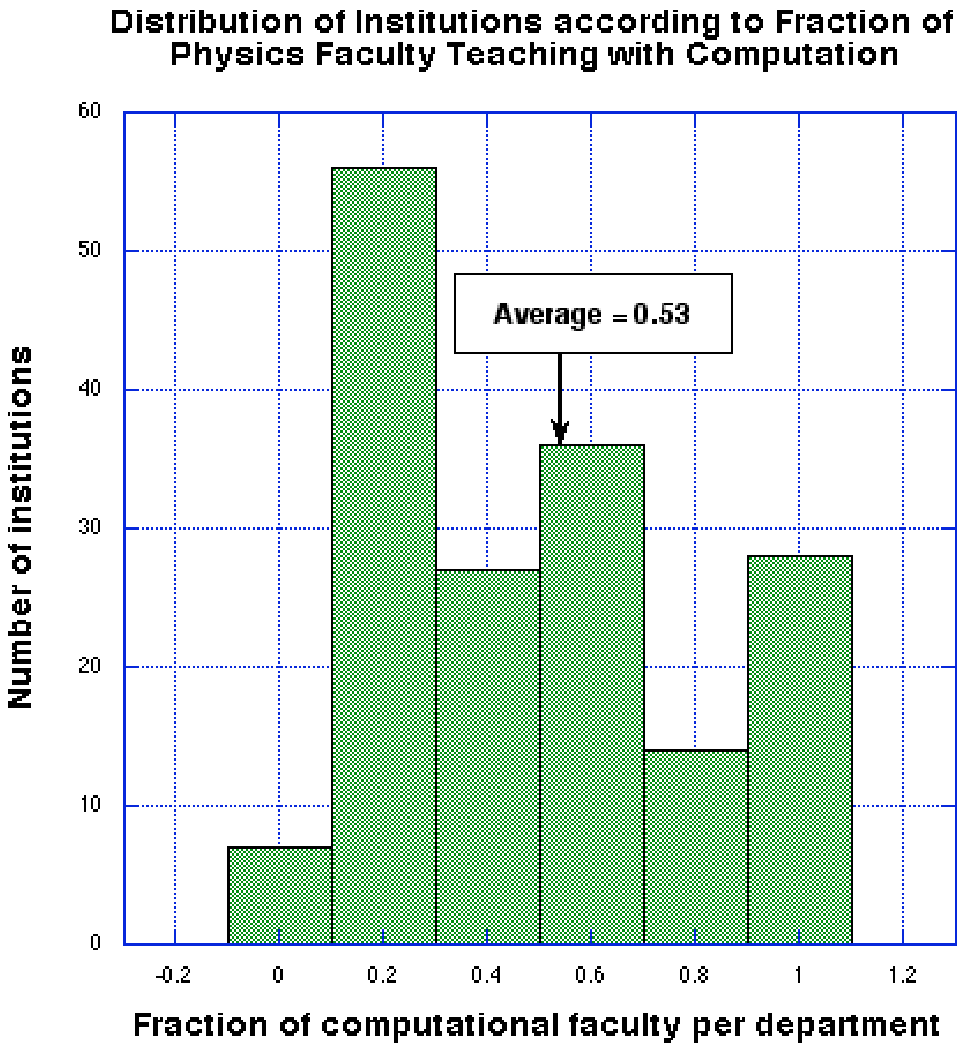

Figure 10: The distribution of responding institutions according to the percentage of physics departmental faculty using computation as part of their grading evaluation in courses.

This rendering depicts a division in institutions regarding the degree of participation that physics departmental faculty have in computational integration for their courses. The question was posed to identify only those faculty who actually reinforce their computational use by evaluating it in determining grades. It is not clear how many respondents actually noticed and observed this criterion in formulating their response. Therefore, these results probably represent the "best case" estimation of how seriously physics faculty treat computation in their courses.

Inferences and Points for Discussion

The first issue to address is the kind of engagement students should have when learning how to use computation in physics; *e.g.* coding, pre-packaged exercises; configurable simulations, *etc.*. The introduction to this paper characterized this as a question of "skill sets." But the question is deeper than that, and involves recognizing the differing ways computation is used across engineering and the various sciences other than physics. This question is crucial to physics educators because almost without exception all engineering and science students study physics as part of their undergraduate education.

In looking over the narrative responses from the survey, it is fairly clear that there is wide diversity of opinion as to whether programming is a necessary part of computational education rather than using prepared applications. There are arguments to support each opinion within this range. Rather than reviewing them in detail, consider a few things that might help rational discussion.

First, this divergence is *not* substantially aligned along questions of which programming language to use, nor even upon whether the programming should be done using third or fourth "generation" languages. (C Java, Fortran, Python, are arguably of the first category and Mathematica, MATLAB, and Maple are among the latter.) Rather the question is whether students should be required to code at all. Frequently a person's view on this is aligned with whether one has a basic research mentality or an applied research and development mentality. Physicists, especially academics, are substantially in the first community and engineers as well as many of the other sciences are in the latter. Given this division, computational integration in the introductory – or service – physics courses are going to require careful, collaborative negotiation.

One can argue that these two mentalities can be served simultaneously. One resolution may come in having students code some models in the introductory course where coding helps them understand the physical principles behind the models, using a fourth generation language to do so. A good example of such a course is that laid out by the introductory texts of Chabay and Sherwood.[3,4] The more complex programming, where attention to algorithms and program verification are desired, could be left to the intermediate and/or upper level courses, which is where many physics departments place it now.[5]

To some degree a different approach can address the same problem. This involves the use of interactive simulations in the introductory course and reserves coding, in any one of its many forms, to later courses or a special computational course. One example that follows this approach involves packaged simulations at the introductory level and open source software modules for more advanced work.[6],[7] In other cases, a commercial application "package" is used to create preprogrammed exercises for the introductory course and subsequent advanced courses use the same package as a programming environment for coding.

What is problematic in this approach is the manifest difference between the physics and engineering communities in their preferred package. Some physics departments insist on object-oriented, general purpose programming environments such as Java or Python. Consider for the moment only three "major" commercial programming packages – MATLAB, Mathematica, and Maple – all of which can be applied more or less readily to the same tasks. The survey sample of physics departments reveals Mathematica as most widely used of these three.[8] There is anecdotal evidence that a similar survey of engineering departments would reveal that MATLAB is predominantly used. Reasons have much to do with the subsequent use of these programming and application tools in the world of professional work after graduation.[9,10]

Next are questions that are institutional in character. The introduction to this paper suggests that among these questions is: Is a *standard* computing environment, across all courses in a department and possibly across all departments, possible, necessary, or even desirable? Another related and very crucial, but seldom discussed question is: How are faculty going to be able to dedicate sufficient time to the tasks required to make a difference in the computational character of their courses? The answers are going to involve institutional policies of rewards, support, and leadership. This not the appropriate place to discuss these questions. However, this author's experiences in a variety of institutions suggest that the current conditions for development of reformative curricula are, except for a very few academic institutions, not conducive to resolution of these questions. As evidence, consider the survey result suggesting that in many institutions computational initiatives and indeed implementation depend upon a single individual. If this person leaves or dies then the whole computational program is in jeopardy.

Finally, there are questions of implementation that vary enormously according to type of institution. The introduction offers a few of these. Of special significance to the engineering community are those regarding academic load and those regarding the quality of students' preparation. These are linked, and their (mutual) solution is both institutionally situational and educationally philosophical. Where education is less valued than research, course loads and expectations about productivity can make computational "reform" a non-starter. Where value added to students competence is more valued that standardized "coverage," the choice of reformative program will be quite different.

The introductory physics sequence in many institutions already is pressing the limits of reasonable academic load. Many institutions are alarmed at the mathematical capabilities, or lack thereof, in their students. So how can anyone suggest "adding" anything more to an already overloaded curriculum? Since this question will take many forms, each of which is specific to a different category of institution, any reformative rationale or suggested approach would be purely speculative. Nonetheless, here are some comments drawn mostly from the survey results.

Many institutions – just over one-half in the survey sample – offer a separate course dedicated to computational physics. Interestingly, at least among the majority of those with such an offering, this does not mean that they have abandoned the piecemeal integration of computation on a course-by-course basis as an additional path. So there is evidence of some willingness among physics departments to include computation within the existing introductory course structure. It is not clear however that there is anywhere near consensus on whether this inclusion should be relegated to the margins – *e.g.* homework problems – or whether it should be woven into the fabric of the courses – *i.e.* highly integrated.

A final note on this topic has to do with "preparation" of incoming students to deal with the different kind of mathematical perspective on physics that is provided by numerical methods. One can note the success of the mathematical community with inventing a new approach to calculus that involves discrete numerical calculations and includes practical contexts from the get-go. This approach has been particularly successful with non-major students. One may speculate that a similar discovery may eventuate in the physics community once an analogous restructuring is accomplished in service physics courses. The other comment concerns the value inherent in explicitly including numerical approaches in the introductory course and relates to an assertion that students' understanding of numerical techniques and their applicability to physical systems will emerge eventually and inevitably. Countering this are the numerous personal testimonies of practicing physicists trained a generation ago admitting that they never understood the full implications of Newton's second law of motion until encountering its expression via finite difference methods much later in life.

Summary and Projections

What more can one say about the overall results of the survey and what hopes are there for rapid progress with the task of getting physics curricula to better serve the needs of all, but especially engineering students?

First, it is not clear that physicists realize how deeply the use of computational applications has penetrated into engineering practice. Even now, a great deal of physics research does not depend upon numerical modeling, for example, to the degree that does engineering development. Moreover, the majority of engineers will never write computational code. One can argue that they should, but they simply don't have the time or inclination. For example, there is evidence of a surprisingly widespread use of spreadsheets for working out design problems in the applied science and engineering communities. The reasons: low cognitive overhead and sufficient flexibility. In short, it is the right tool for the job.

Second, the survey results provide an enlightening look into a cross section of computational practices in undergraduate physics programs. The indicators are that there is enormous variability among institutions regarding existing practices. However, there is also evidence for a limited number of paradigms that have emerged and from which one may take heart that something "works."

Finally, it is clear that there needs to be a extensive and continuing conversation between the physics and engineering communities in the process of evolving the physics curriculum to make it more computationally enriched. Such a conversation would be greatly aided if there were a survey of engineering computational practice in the undergraduate curriculum that was analogous to the one that *CiSE* recently commissioned for physics, which has been reported here.

[1] Fuller, R. G. (2006). "Numerical Computations in US Undergraduate Physics Courses." Computing in Science and Engineering 8(5): 16-21.

2. Ivie, R. and K. Stowe (2002). Early Careers of Physics Bachelors. College Park MD, American Institute of Physics.

3. Chabay, R. W. and B. A. Sherwood (2007). Matter and Interactions I: Modern Mechanics. Somerset, NJ, J.W. Wiley.

4. Chabay, R. W. and B. A. Sherwood (2007). Matter and Interactions II: Electric and Magnetic Interactions. Somerset, NJ, J.W. Wiley.

5. Taylor, J. R. and B. A. K. III (2006). "Using Computational Methods to Reinvigorate an Undergraduate Physics Curriculum." Computing in Science and Engineering 8(5): 38-43.

6. Belloni, M. and W. Christian (2003). "Physlets for Quantum Mechanics." Computing in Science and Engineering 5(1): 90-96.

7. Belloni, M., W. Christian, et al. (2007). "Open Source Physics Curricular Material for Quantum Mechanics: Dynamics and Measurement of Quantum Two-state Superpositions ".

8. Fuller, R., D. Winch, et al. (2006). Responses to the CiSE survey of roles for computation in undergraduate physics courses, 2005-2006, IEEE-Computer Society. Database available for educational research – contact *CiSE* publication office: <cise@computer.org> for conditions and to request.

9. Chonacky, N. and D. Winch (2005). "3Ms for Instruction: Reviews of Maple, Mathematica, and Matlab: Part I." Computing in Science and Engineering 7(3): 7-13.

10. Chonacky, N. and D. Winch (2005). "3Ms for Instruction: Reviews of Maple, Mathematica, and Matlab: Part II." Computing in Science and Engineering 7(4): 14-23.