Challenges to integrating computational activities

- File sharing – one student does the work; everyone else copies – enforcing individual accountability
    - Detailed commenting, 2-minute interviews
- Content coverage: How to make room for it
- Do we need to teach a programming language?
    - (standardizing? Multiple languages?)
    - (within a physics course? Adding an extra course?)
    - Can choose how much programming is needed – don't need to program from scratch
- How do you assess computational work? (coding on exams? Writing on paper on exams?)
- Faculty expertise/confidence/time
- Getting stuck debugging & wasting class time
- Student buy-in (this is what professional physicists & engineers do; student jobs)

Benefits to integrating computational activities

- Can go beyond standard textbook problems / enhanced content
- Creating models & graphing the results gives appreciation of underlying concepts
- If you can solve a problem numerically, you probably understand the problem
    - Analytical vs. numerical solutions are not an "either/or"
- Students tend to take ownership of computational solutions (open "playground" in parameter space)
- If the computation explains the physics better, that might offset some of the time that is invested in introducing computation
- Preparing students for careers and graduate school (marketable skills)
    - Computation can help the department to look more attractive in terms of where the students go, what the students do, what the students learn
- A goal of physics should be to give students the ability to see the world like a physicist. Computation can help do this.
- Doing computation in physics can help the students to do better in their math classes – making better use of the tools that they are already using (Wolfram Alpha, etc.)