Integrating Computation into STEM Curriculum

Calvin Berggren

Texas Lutheran University

Engaging Pedagogy Conference

17 May 2017

## Computation is increasingly important in all STEM fields, especially math, physics, and engineering

* Computation is now ubiquitous in many fields
* Computational thinking provides a different way of approaching the subject, leading to new insights and deeper understanding
* Computational methods allow one to solve a much wider range of problems
* Computational skills are valuable for a wide variety of careers in and outside your field
* Many professional societies (AAPT, NCTM, AMTE, and others) have made official statements and reports promoting the integration of computation into education in their field

## Important skills include

* Fundamental computer skills
  + operating systems
  + file systems
  + coding
  + using computational tools
* Technical computing skills
  + process data
  + visualize data
* Computational skills
  + translate models or algorithms into code
  + write/test/debug code
  + divide problem into manageable computational tasks
  + make meaning of results

## Important tools include

* General-purpose programming languages like Python and C++
* Specific packages for these languages like SciPy, Jupyter, and matplotlib (for the Python ecosystem)
* Integrated mathematical computing environments like Mathematica, Maple, and MATLAB
* Spreadsheets like Excel

## Physics at TLU is in the process of making a comprehensive update to the major to include computational elements according to best practices

* Revamp PHYS/MATH 371 Mathematical Methods for Scientists and Engineers to include use of computer algebra systems for symbolic calculations
* Revamp the dedicated computational course PHYS 390 Applied Computational Physics I to focus on computational modeling, important computational methods, and algorithms
* Revamp the dedicated computational course PHYS 391 Applied Computational Physics II as a project-based course to focus on practical tools and skills important in industry
* Supplement all existing courses where appropriate
  + Best practices indicate that development of computational skills should be integrated into the curriculum, rather than taught as isolated skills.

## Sensible opportunities for including computational elements into your courses



## Recommendations include

* Considering available tools and standardizing on a limited set sufficient for your needs
* Being strategic about where to add new material to curriculum or where to augment existing material
* Providing scaffolding to help students acclimate to computational ideas and tools, which are likely to be unfamiliar environments

## References

* *AAPT Recommendations for Computational Physics in the Undergraduate Physics Curriculum*. AAPT Undergraduate Curriculum Task Force. October 2016. <https://aapt.org/Resources/upload/AAPT_UCTF_CompPhysReport_final_B.pdf>