Mathematical Modeling in Engineering

RICK SALAY

UNIVERSITY OF TORONTO, CANADA

Question/Claim:

Why does math in software engineering make work harder when in physical engineering it makes work easier?

Work Context

A physical engineer can't do their work without math

- PE needs math because the world is complex

A software engineer can do their work without math

- SE is reluctant to use math
  - Or even semiformal models like UML
- Agile: Just code!
- (Exception: Safety critical systems)

Main use of math

Physical engineering:

- Math is used to get useful approximations
- Makes PE work doable

Software engineering:

- Math is used to get precise (specifications, refinement, etc.)
- Makes SE work more tedious

Lesson: approximation seems more fruitful than precision

- Probably leads to higher adoption by software engineers…

Question: Can we use approximation in SE?

Answer: we already are, somewhat:

- Time/space complexity analysis of programs
- Mockups and RAD (not mathematical)
  - approximation to assess UI, functionality and user interaction
- Queueing model for server load analysis
- Abstract Interpretation (over-approximation)
  - But only to support verification
- UML models (semi-mathematical)

Question: Why isn't UML used that much?

Cost Context

Physical Engineering:

- Cost of creating throwaway physical systems too high compared to cost of creating models (physical and mathematical)
- Maybe 3D printing will change this!

Software Engineering:

- Cost of creating throwaway code is low compared to cost of creating models

Lesson: Need really cheap approximate methods

- Is this realistic?

Environmental Context

- Physical engineers work in the real world
  - Amenable to approximation
    - Real world is constrained by fixed physical laws
    - Real world is continuous
- Software engineers create their own worlds
  - Constrained only by limits of computability
    - Or maybe socially constructed worlds
  - Fewer opportunities for approximation?

Conclusion

Physical engineering science is more successful in "technology transfer" than software engineering science

- We need to find out why!

How can math make a software engineer's work easier?

- Maybe "cheap approximate methods" should be the focus of research