Just-in-Time Teaching of Numerical Methods

Dhavide Aruliah

Faculty of Science
University of Ontario
Institute of Technology

Western Conference on Science Education
Jul. 6, 2011
One minute of grumbling

Axiom of Universities (implicit)

Research >> Teaching
One minute of grumbling

Axiom of Universities (implicit)

Research $\gg$ Teaching

Conjecture

Teaching $\geq$ Research
1. Challenges of teaching Numerical Methods for Engineers

2. The Just-in-Time Teaching approach

3. Details of my implementation of JiTT

4. Reflections and Conclusions
1. Challenges of teaching Numerical Methods for Engineers

2. The Just-in-Time Teaching approach

3. Details of my implementation of JiTT

4. Reflections and Conclusions
### Central question

How can I structure the course *Numerical Methods for Engineers* effectively?
Technical details

<table>
<thead>
<tr>
<th></th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>17:00–18:30</td>
<td></td>
<td></td>
<td>17:00–18:30</td>
<td></td>
</tr>
<tr>
<td></td>
<td>135 students</td>
<td></td>
<td></td>
<td>135 students</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20:30–22:00</td>
<td>19:00–20:30</td>
<td>65 students</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>65 students</td>
<td>65 students</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Technical details

- two 80 minute classes per week, one 50 minute tutorial
- two evening sections [65+135 students]
- three teaching assistants
Technical details

- two 80 minute classes per week, one 50 minute tutorial
- two evening sections [65+135 students]
- three teaching assistants
- inhomogeneous range of engineering students: automotive, electrical, manufacturing, mechanical, nuclear
Technical details

- two 80 minute classes per week, one 50 minute tutorial
- two evening sections [65+135 students]
- three teaching assistants
- inhomogeneous range of engineering students: automotive, electrical, manufacturing, mechanical, nuclear
- engineering students generally take 6 courses/term
- extensive laundry list of "mandatory" topics
Larger challenges

- students generally do not read textbooks
Larger challenges

- students generally do not read textbooks
  - reading skills not strong enough on average
  - reading not perceived as related to success
Larger challenges

- students generally do not read textbooks
  - reading skills not strong enough on average
  - reading not perceived as related to success
  - questionable coherence of numerics textbooks
Larger challenges

- students generally do not read textbooks
  - reading skills not strong enough on average
  - reading not perceived as related to success
  - questionable coherence of numerics textbooks
- disparate modes of thought required
  - mathematical analysis vs. computer programming
Larger challenges

- students generally do not read textbooks
  - reading skills not strong enough on average
  - reading not perceived as related to success
  - questionable coherence of numerics textbooks
- disparate modes of thought required
  - mathematical analysis vs. computer programming
  - prerequisites: Linear Algebra, Calculus
    ⇒ no programming prerequisite!
  - some students “know” C++; others are novices
Larger challenges

- students generally do not read textbooks
  - reading skills not strong enough on average
  - reading not perceived as related to success
  - questionable coherence of numerics textbooks
- disparate modes of thought required
  - mathematical analysis vs. computer programming
  - prerequisites: Linear Algebra, Calculus
    ⇒ no programming prerequisite!
  - some students “know” C++; others are novices
- student effort proportional to grades assigned
<table>
<thead>
<tr>
<th>Challenges</th>
<th>The JiTT approach</th>
<th>Implementation</th>
<th>Reflections and Conclusions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Challenges of teaching Numerical Methods for Engineers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>The Just-in-Time Teaching approach</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Details of my implementation of JiTT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Reflections and Conclusions</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
My traditional course-planning strategy

1. identify learning objectives
2. select textbook
3. plan schedule of lectures
4. construct grading scheme
5. construct lectures/assignments/exams
## Course topics

1. *Modelling, computing, error analysis*
2. *Roots and optimisation*
3. *Linear systems*
4. *Curve fitting*
5. *Integration and differentiation*
6. *Ordinary differential equations*

[Begin with one–two weeks teaching MATLAB]
**Just-in-Time Teaching (JiTT)**

- web-based warm-up exercises
- interactive classroom
- “Just-in-Time”: instructor reads student responses to warm-up immediately before class
- feedback shapes class discussion
Common elements of JiTT

- “Warm-ups”
- “Puzzles”
- “Good fors” (enrichment material)
<table>
<thead>
<tr>
<th>Challenges</th>
<th>The JiTT approach</th>
<th>Implementation</th>
<th>Reflections and Conclusions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Challenges of teaching Numerical Methods for Engineers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>The Just-in-Time Teaching approach</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Details of my implementation of JiTT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Reflections and Conclusions</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
UOIT Infrastructure

- all students have standard IBM ThinkPad laptops
- lecture hall seats have power, ethernet ports
- science instructors have tablets for lectures
- WebCT & Mapleta available for course management
Reading assignments

- released Monday night & Thursday night on WEBCT
- readings assigned from textbook (roughly 20 pages)
- recommended textbook problems (taken up in tutorial)
- due 3 hours prior to next lecture
- binary scoring based on submission
In-class assignments

- sometimes on paper, sometimes electronic
- done in groups of three of four
- usually discussion of solution prior to end of class
- binary scoring, largely based on submission
Questions asked every reading assignment

- How long did it take you to complete the reading?
- How thoroughly did you reading?
  (a) I didn’t read it at all.
  (b) I quickly skimmed over parts of the reading.
  (c) I skimmed over most of it but went over a small part in great detail.
  (d) I skimmed over a small part but went over most of it in great detail.
  (e) Very thorough; I went over each sentence of each paragraph meticulously.
- What part of the reading was most difficult?
- Do you have any specific concerns about the topics covered?
The following MATLAB transcript is used to find the three smallest positive solutions of the nonlinear equation

\[ \mu x = \cot(x) \quad \text{for} \quad \mu = 1. \]
The following MATLAB transcript is used to find the three smallest positive solutions of the nonlinear equation

\[ \mu x = \cot(x) \quad \text{for} \quad \mu = 1. \]

```matlab
>> mu = 1; f = @(x) mu*x - cot(x);
>> x1 = fzero( f, [0.75,1.25] ); % Zero in [0.75,1.25]
>> x2 = fzero( f, [2.75,3.25] ); % Zero in [2.75,3.25]
>> x3 = fzero( f, [3.25,3.75] ); % Zero in [3.25,3.75]
```

The first three zeros of \( f \) are:

0.8603335890
3.1415926536
3.4256184595

What is wrong with these results?
The following MATLAB transcript is used to find the three smallest positive solutions of the nonlinear equation

\[ \mu x = \cot(x) \quad \text{for} \quad \mu = 1. \]

\[
\begin{align*}
\text{>> } & \mu = 1; \quad f = @(x) \mu x - \cot(x); \\
\text{>> } & x1 = \text{fzero}(f, [0.75,1.25]) \quad \% \text{ Zero in } [0.75,1.25] \\
\text{>> } & x2 = \text{fzero}(f, [2.75,3.25]) \quad \% \text{ Zero in } [2.75,3.25] \\
\text{>> } & x3 = \text{fzero}(f, [3.25,3.75]) \quad \% \text{ Zero in } [3.25,3.75] \\
\text{>> } & \% \text{ Display results} \\
\text{>> } & \text{fprintf(} \text{The first three zeros of } f \text{ are:} \text{\n}); \\
\text{>> } & \text{fprintf(} \%12.10f\text{\n,} [x1;x2;x3] \text{)}
\end{align*}
\]

What is wrong with these results?
The following MATLAB transcript is used to find the three smallest positive solutions of the nonlinear equation

\[ \mu x = \cot(x) \quad \text{for} \quad \mu = 1. \]

```matlab
>> mu = 1; f = @(x) mu*x - cot(x);
>> x1 = fzero( f, [0.75,1.25] ); % Zero in [0.75,1.25]
>> x2 = fzero( f, [2.75,3.25] ); % Zero in [2.75,3.25]
>> x3 = fzero( f, [3.25,3.75] ); % Zero in [3.25,3.75]
>> % Display results
>> fprintf('The first three zeros of f are:
');
>> fprintf('%.12f
', [x1;x2;x3])
The first three zeros of f are:
0.8603335890
3.1415926536
3.4256184595
```

What is wrong with these results?
The following MATLAB transcript is used to find the three smallest positive solutions of the nonlinear equation

\[ \mu x = \cot(x) \quad \text{for} \quad \mu = 1. \]

```matlab
>> mu = 1; f = @(x) mu*x - cot(x);
>> x1 = fzero( f, [0.75,1.25] ); % Zero in [0.75,1.25]
>> x2 = fzero( f, [2.75,3.25] ); % Zero in [2.75,3.25]
>> x3 = fzero( f, [3.25,3.75] ); % Zero in [3.25,3.75]
>> % Display results
>> fprintf(The first three zeros of f are:
);
>> fprintf(%12.10f
,[x1;x2;x3])
The first three zeros of f are:
0.8603335890
3.1415926536
3.4256184595
```

What is wrong with these results?
The error in the above argument is that it mistakenly states that the function has 3 roots, when it actually only has 2 real roots. What the argument is doing is confusing a vertical asymptote that occurs between the two real roots as a third root. Considering that a vertical asymptote is not a root the results proved to be incorrect. The roots are located at $x=0.8603335890$ and $x=3.4256184595$. 
The program does not work properly because of the discontinuity in the function. The function has two roots and one asymptote. The values of the roots are 0.8603335890 and 3.4256184595; whereas the value of the asymptote is equal to PI which is 3.1415926536.
Sample student responses
very good response; third zero correctly identified

To verify the correctness of the results one could graph the function and determine if there is a change in signs where the zeros are located. The answer could be verified by subbing the answers into the initial equation. They are incorrect because the asymptote in cot causes a rapid sign change without actually causing a zero, and since fzero looks for a sign change it assumes there is a zero in between. The real zeros occur at 0.8603335890, 3.4256184595 and 6.4373
A not-so-successful in-class assignment

Reproduce this figure (based on Anscombe, 1973)
A not-so-successful in-class assignment
Reproduce Anscombe’s plots

- prior reading assignment: find the original paper
- Anscombe’s original data provided on WEBCT in class
- students instructions:
  - use the file `anscombe_plots.m` as a template
  - use `linregr2.m` (from the textbook) as a guide to help you produce the plots of the regression lines
  - save final file as `anscombe_plots.m` and upload to WEBCT
1. Challenges of teaching Numerical Methods for Engineers

2. The Just-in-Time Teaching approach

3. Details of my implementation of JiTT

4. Reflections and Conclusions
Lessons learned

- reading assignments: generally useful, well received
- in-class assignments: must be linked to assessment
Lessons learned

- reading assignments: generally useful, well received
- in-class assignments: must be linked to assessment
- 22 reading assignments out of 25 lectures
- 18 in-class assignments out of 25 lectures
Lessons learned

- reading assignments: generally useful, well received
- in-class assignments: must be linked to assessment
- 22 reading assignments out of 25 lectures
- 18 in-class assignments out of 25 lectures
- Generally $\approx 150$ submissions (most legitimate!)
- around 75% participation from engineering is good!
- students who bother to attend put in reasonable effort
### Principal challenges

- constructing suitable deep reading questions
- constructing right questions to do in-class
- managing marking workload
Principal lesson learned

- Make students responsible for reading *explicitly*.
  - "reading assignments" → put into syllabus!
  - structure class-time around actual reading
  - impose suitable incentives into grading scheme
Future plans

- will definitely keep using reading assignments
- will work harder at not lecturing
- will avoid obsessing about “covering material in class”
- will structure class around completing activities
- for larger classes, prepare questions sooner
The way to save time, make every moment count, and integrate grading, learning, and motivation is to plan your grading from the moment you begin planning the course. To do otherwise—to regard grading as an afterthought—is to create wasted time, dead-end efforts, and post-hoc rationalizations as students question their grades.

Effective Grading: A Tool for Learning and Assessment in College
Barbara E. Walvoord, Virginia Johnson Anderson (2nd ed., 2009)