UNSW SYDNEY

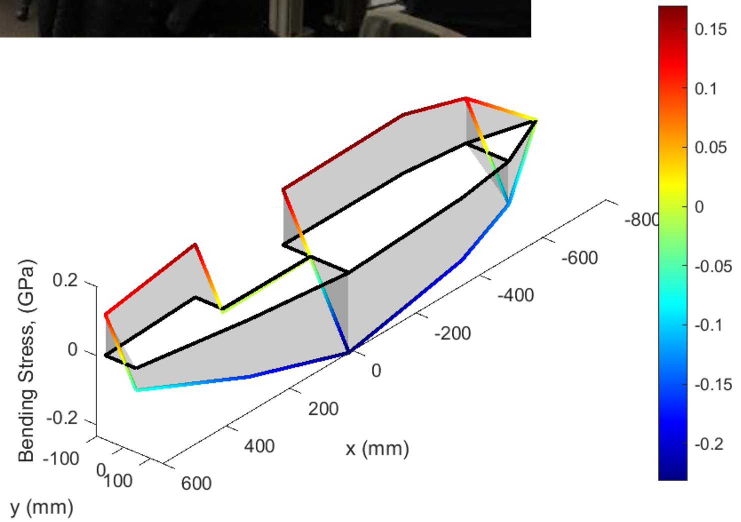
MATLAB EXPO 2021

Authentic Engineering Assessment:

From formative quizzes to high-stakes examination

Presented by

Garth Pearce
Associate Professor and Deputy Head of School (Education)
Mechanical and Manufacturing Engineering
UNSW



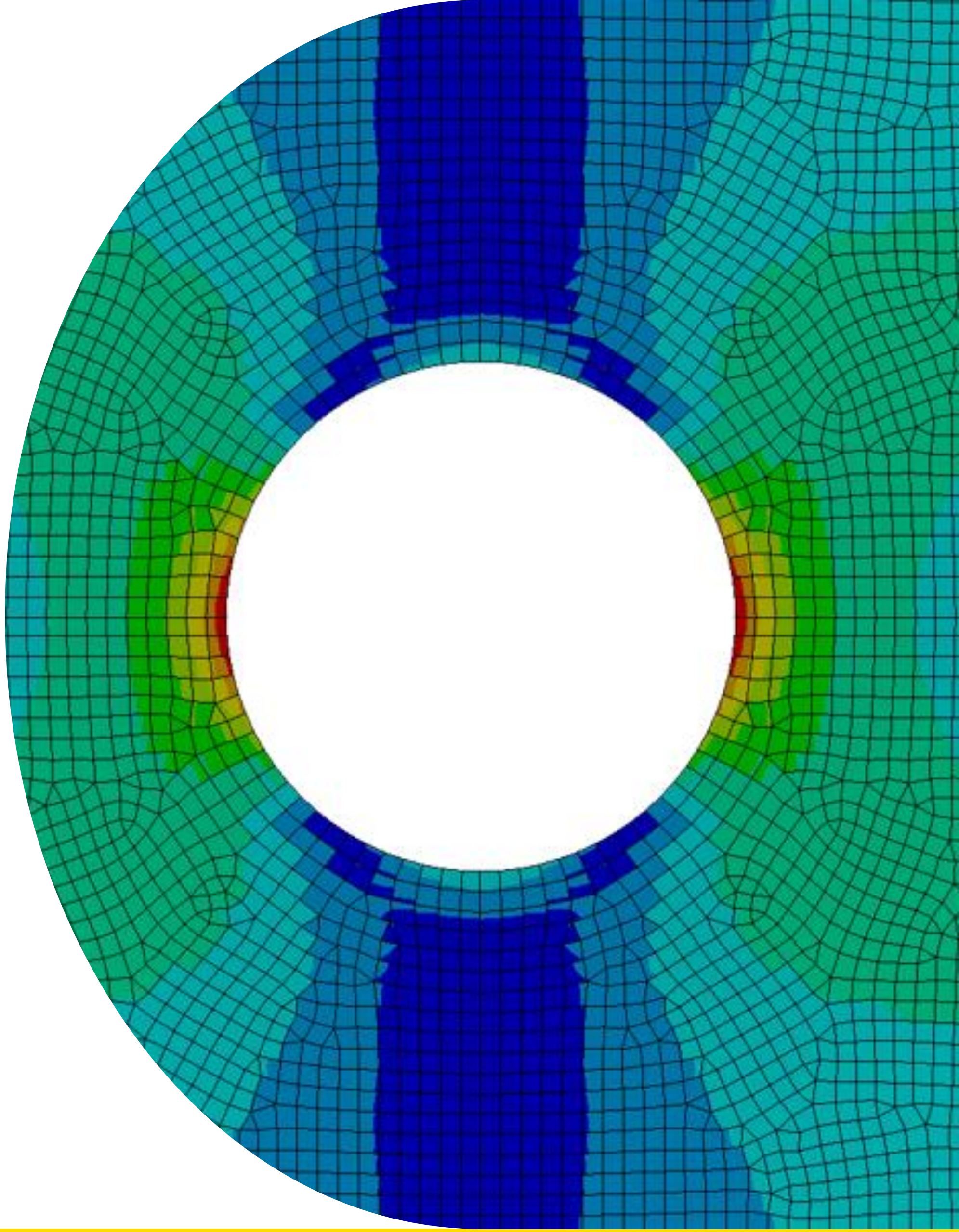
Context

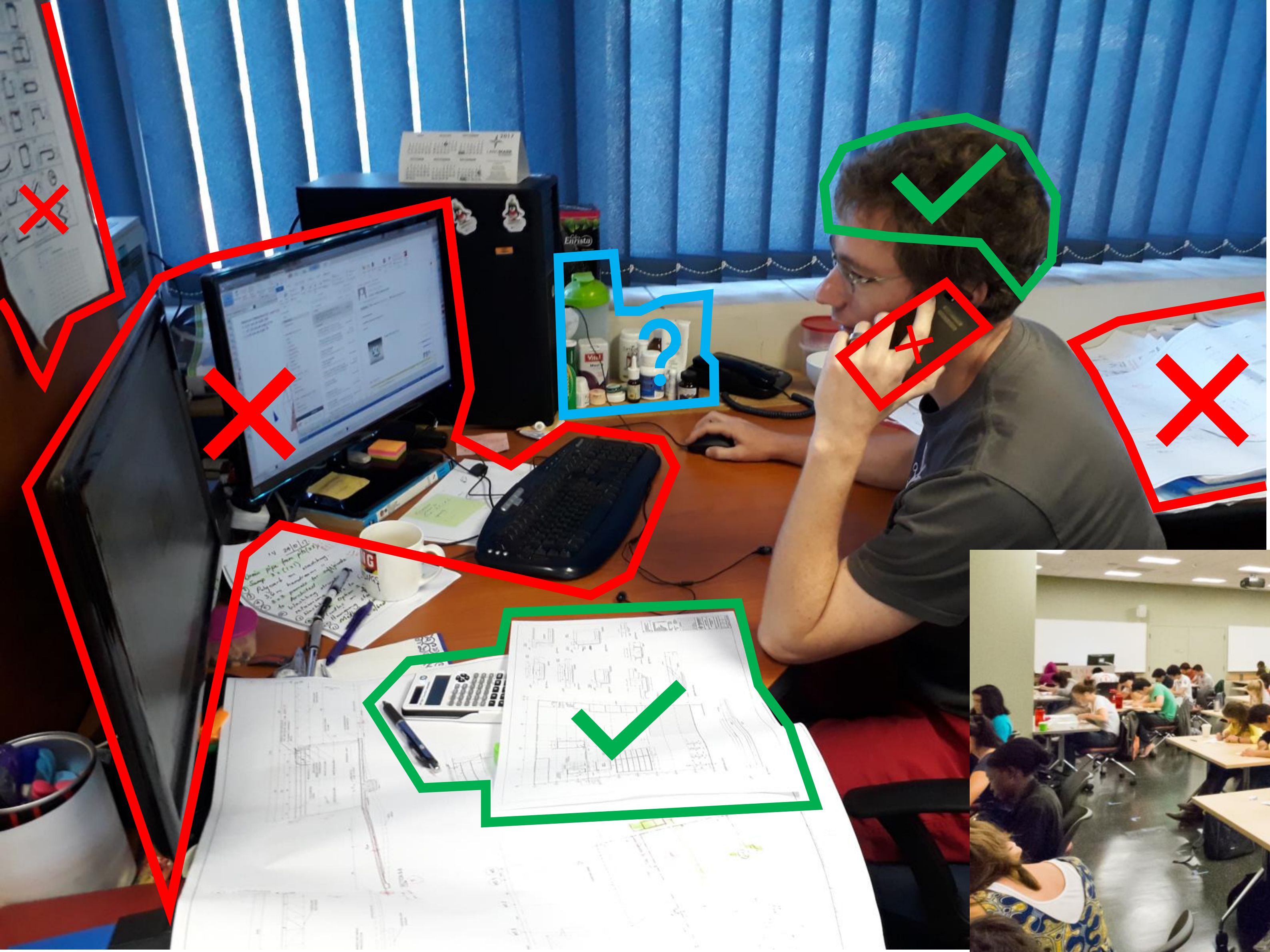
I teach 3rd yr, 4th yr and PG courses in Mechanical and Aerospace Engineering:

- Aerospace Structures (~10 yr)
- Finite Element Methods (~5 yr)

The courses skew toward engineering analysis and build on fundamental knowledge in previous years. These are not *coding* classes.

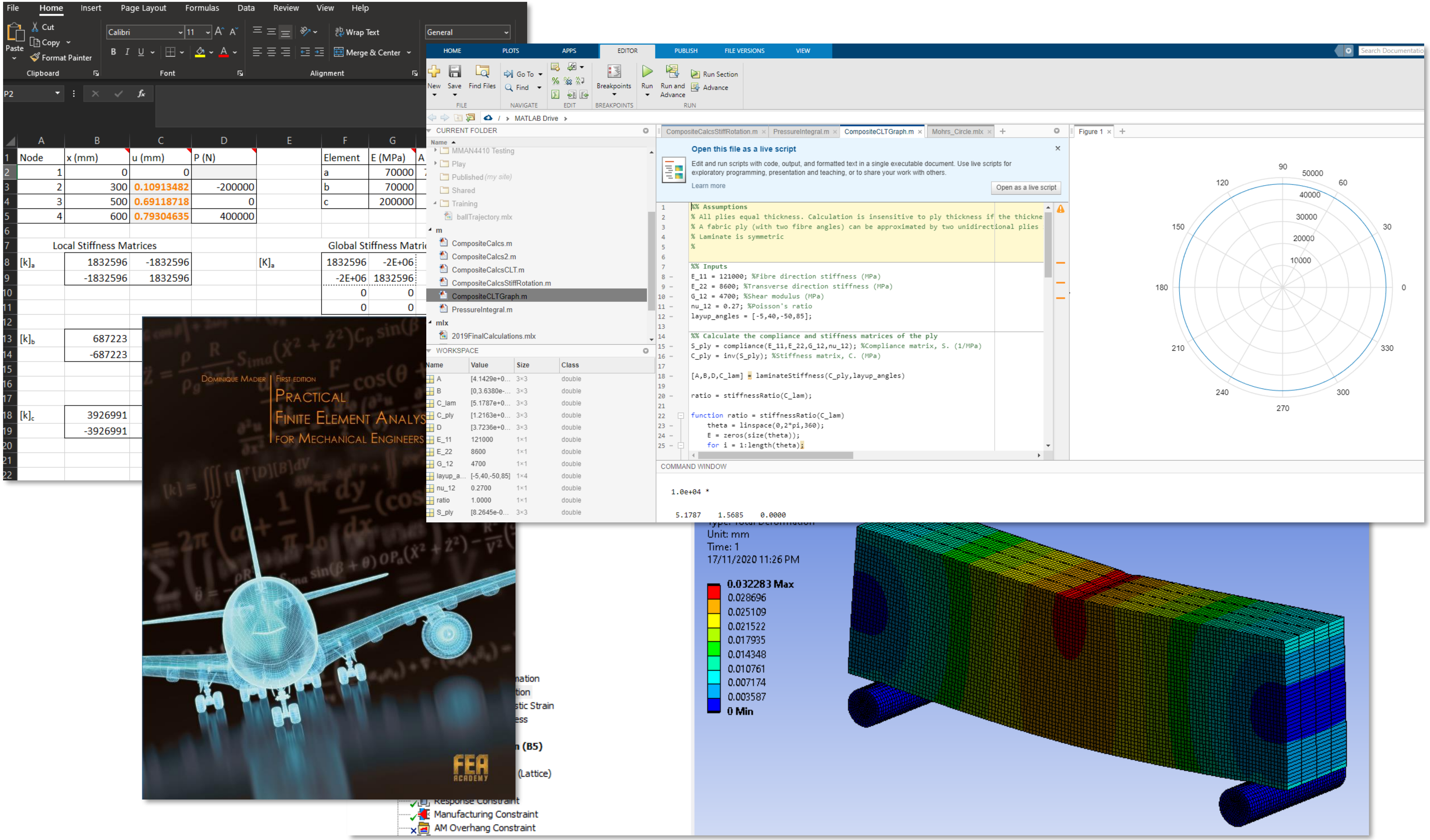
There is a diverse cohort in the classes, with between 40-60% international students. For many PG students, FEM is their first UNSW course.





Skippy the Giraffe, CC BY-SA 4.0
<https://creativecommons.org/licenses/by-sa/4.0>,
via Wikimedia Commons

https://www.flickr.com/photos
/stanfordedtech/8031101633/
CC 2.0



	A	B	C	D	E	F	G
1	Node	x (mm)	u (mm)	P (N)		Element	E (MPa)
2	1	0	0			a	70000
3	2	300	0.10913482	-200000		b	70000
4	3	500	0.69118718	0		c	200000
5	4	600	0.79304635	400000			
6							
7	Local Stiffness Matrices					Global Stiffness Matri	
8	$[k]_a$	1832596	-1832596		$[K]_a$	1832596	-2E+06
9		-1832596	1832596			-2E+06	1832596
10						0	0
11						0	0
12							
13	$[k]_b$	687223					
14		-687223					
15							
16							
17							
18	$[k]_c$	3926991					
19		-3926991					

Dominique Madier | First edition

Practical Finite Element Analys

for Mechanical Engineers

FEA Academy

Open this file as a live script

Edit and run scripts with code, output, and formatted text in a single executable document. Use live scripts for exploratory programming, presentation and teaching, or to share your work with others.

Learn more

Open as a live script

```
%% Assumptions
% All plies equal thickness. Calculation is insensitive to ply thickness if the thickne
% A fabric ply (with two fibre angles) can be approximated by two unidirectional plies
% Laminate is symmetric
%

%% Inputs
E_11 = 121000; %Fibre direction stiffness (MPa)
E_22 = 8600; %Transverse direction stiffness (MPa)
G_12 = 4700; %Shear modulus (MPa)
nu_12 = 0.27; %Poisson's ratio
layup_angles = [-5,40,-50,85];

%% Calculate the compliance and stiffness matrices of the ply
S_ply = compliance(E_11,E_22,G_12,nu_12); %Compliance matrix, S. (1/MPa)
C_ply = inv(S_ply); %Stiffness matrix, C. (MPa)

[A,B,D,C_lam] = laminateStiffness(C_ply,layup_angles)

ratio = stiffnessRatio(C_lam);

function ratio = stiffnessRatio(C_lam)
    theta = linspace(0,2*pi,360);
    E = zeros(size(theta));
    for i = 1:length(theta);
```

COMMAND WINDOW

```
  1.0e+04 *

   5.1787    1.5685    0.0000
```

WORKSPACE

Name	Value	Size	Class
A	[4.1429e+0...	3×3	double
B	[0,3.6380e-...	3×3	double
C_lam	[5.1787e+0...	3×3	double
C_ply	[1.2163e+0...	3×3	double
D	[3.7236e+0...	3×3	double
E_11	121000	1×1	double
E_22	8600	1×1	double
G_12	4700	1×1	double
layup_a...	[-5,40,-50,85]	1×4	double
nu_12	0.2700	1×1	double
ratio	1.0000	1×1	double
S_ply	[8.2645e-0...	3×3	double

Unit: mm
Time: 1
17/11/2020 11:26 PM

UNSW SYDNEY

Authenticity

- Solve “real world” problems
- Align *teaching* with *practice*
- With *software*:
  - Decouple *algorithms* from *mathematics*
  - Improve *efficiency* of work and learning
  - Assess the *whole* “toolbox” of skills
  - Align student *incentives*

Daren, CC BY 2.0 <https://creativecommons.org/licenses/by/2.0>, via Wikimedia Commons

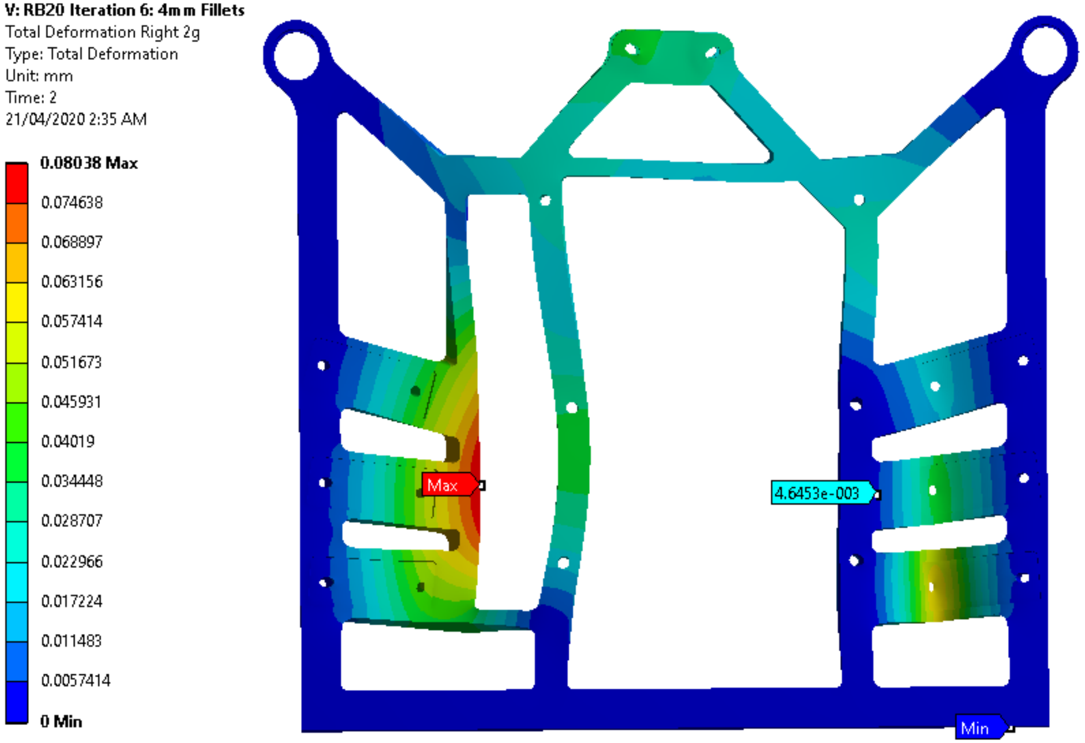
Engineers Australia Competencies

2.1: Application of established engineering methods to *complex engineering problem solving*.

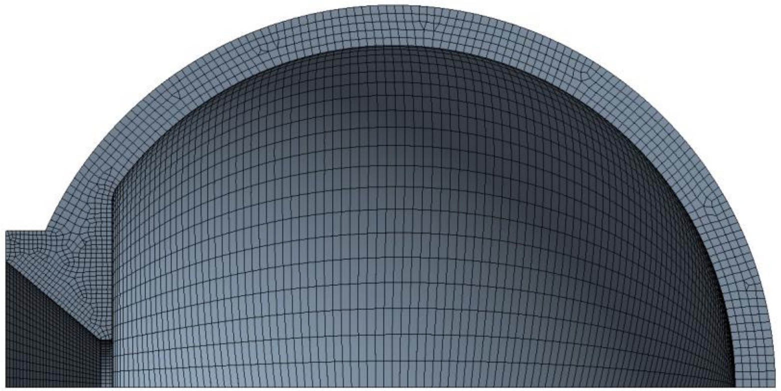
2.2: Fluent application of *engineering techniques, tools and resources*.

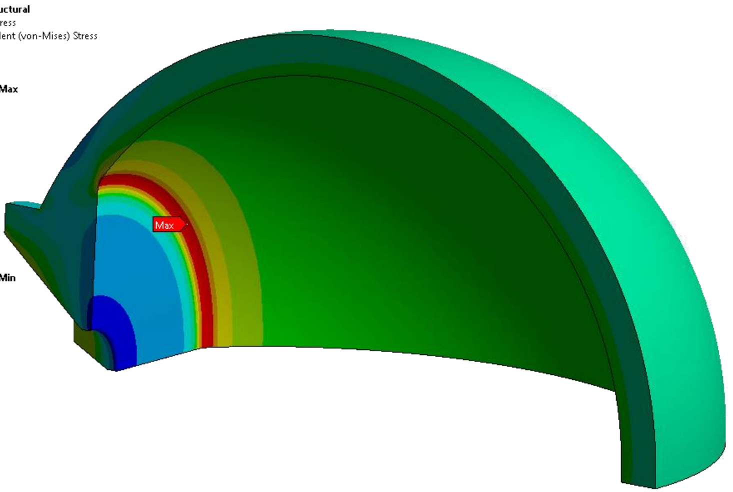
2.3: Application of systematic *engineering synthesis and design processes*.

There is a gap here. Some of our assessment encourages being good at solving *instances* of problems (i.e. traditional exams). Instead, we want students to learn methods to solve whole *classes* of problem.

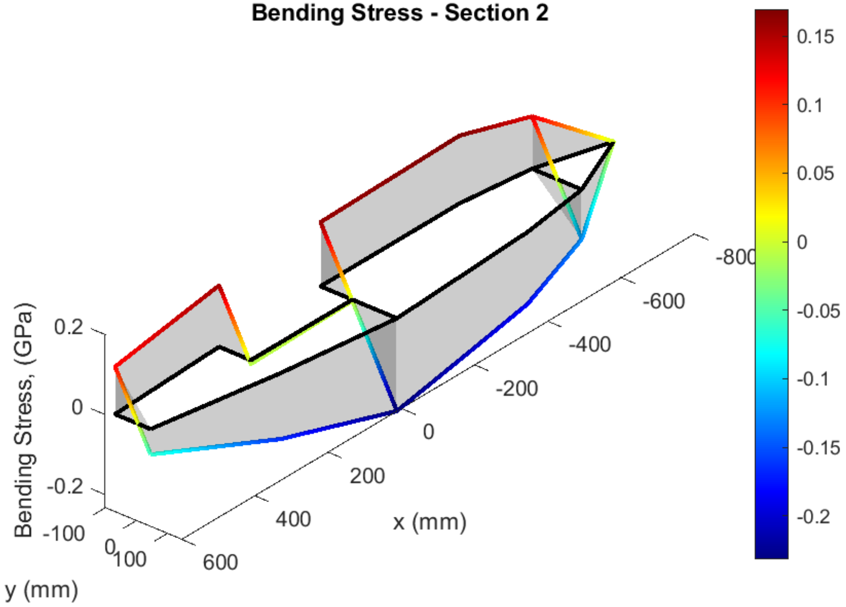


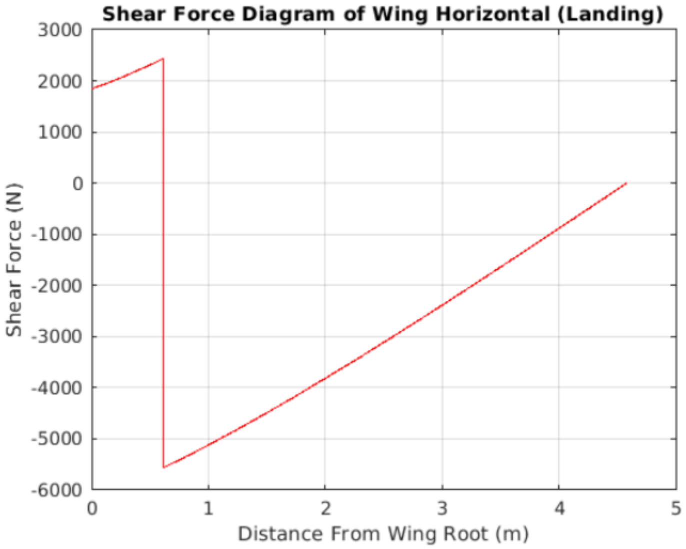
V: RB20 Iteration 6: 4mm Fillets
Total Deformation Right 2g
Type: Total Deformation
Unit: mm
Time: 2
21/04/2020 2:35 AM

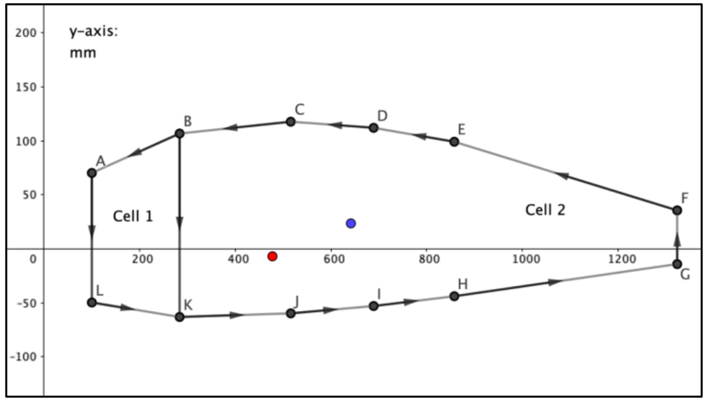


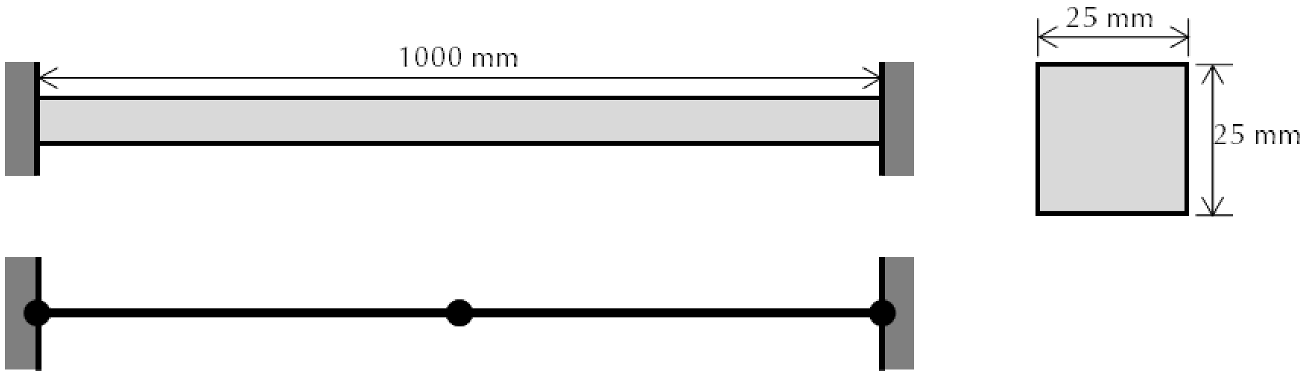


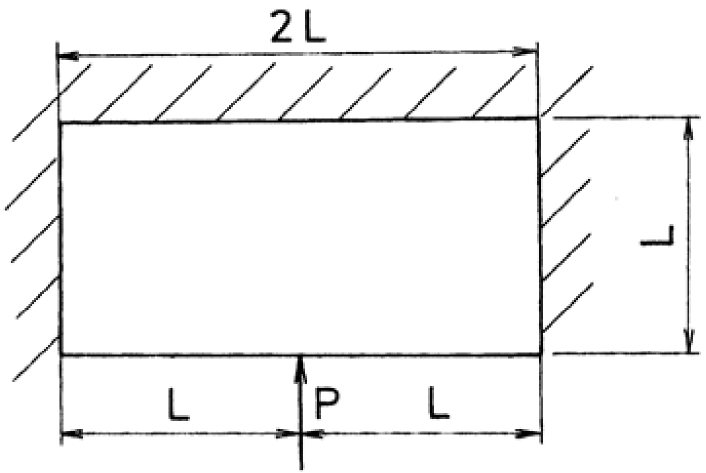
A: Static Structural
Equivalent Stress
Type: Equivalent (von-Mises) Stress
Unit: MPa
Time: 1

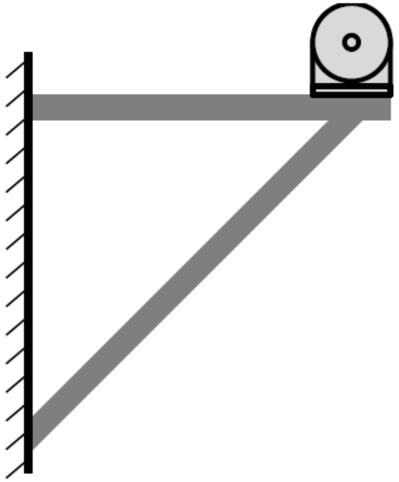


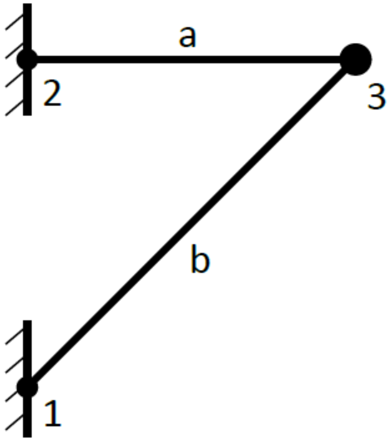


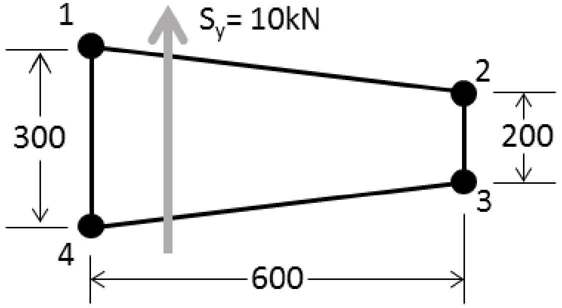


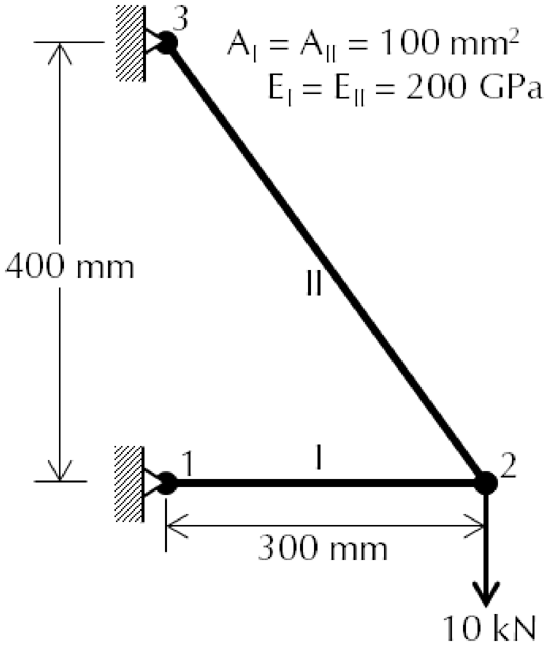

1000 mm
25 mm
25 mm

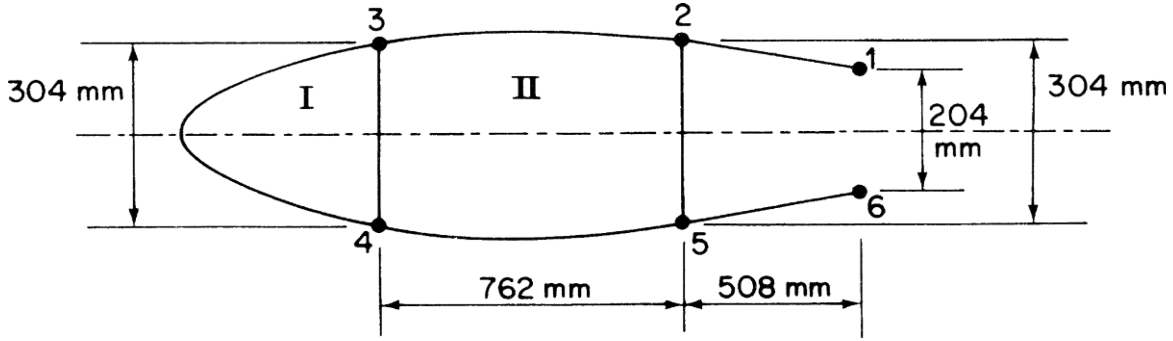

2 L
L
L
P
L




a
2
3
b
1

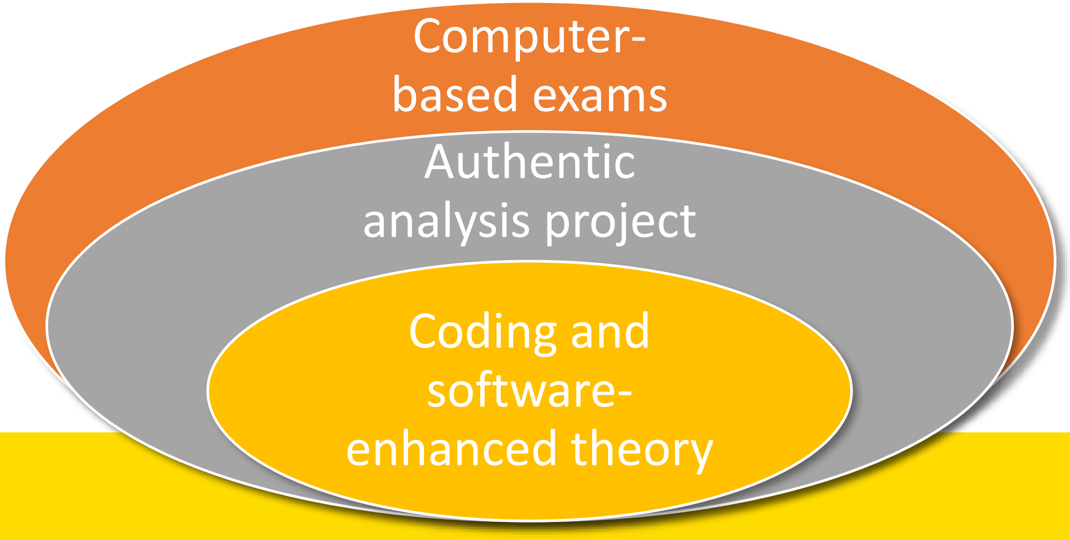

1
S_y= 10kN
2
300
200
3
4
600


3
$A_I = A_{II} = 100$ mm^2
$E_I = E_{II} = 200$ GPa
400 mm
II
1
I
2
300 mm
10 kN


3
2
1
304 mm
I
II
304 mm
204 mm
6
4
5
762 mm
508 mm

Embedding Software in Coursework

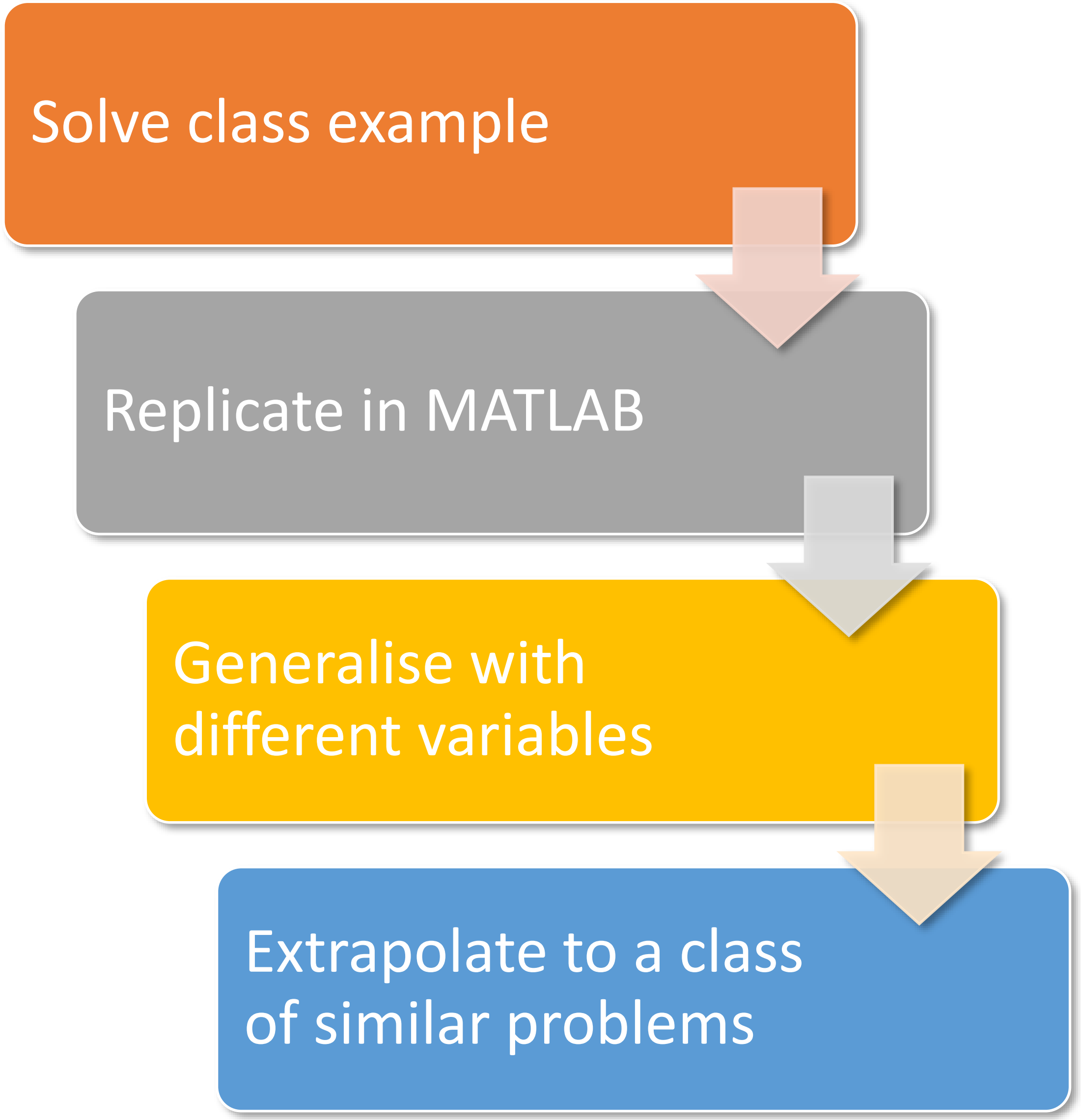
- Three step approach:
  1. *Augment* class theory with code and digital counterparts
  2. *Use* student code and software for authentic projects
  3. *Assess* skills using complex problems under software-friendly exam conditions


Computer-based exams
Authentic analysis project
Coding and software-enhanced theory

Original photo: User:Fanghong Derivative work: User:Gnomz007, CC BY-SA 3.0 <http://creativecommons.org/licenses/by-sa/3.0/>, via Wikimedia Commons

Augment Theory with Code

- Every theoretical concept has a complimentary *MATLAB Live Script*
- Live Scripts allow native embedding course theory, pictures, code, widget, graphical output, etc
- These simulations allow students to decouple the algorithm from the *implementation* of the algorithm
  - leading to better mental models and heuristics
- Seamless desktop and online code storage through MATLAB Drive is a HUGE benefit.



Solve class example

Replicate in MATLAB

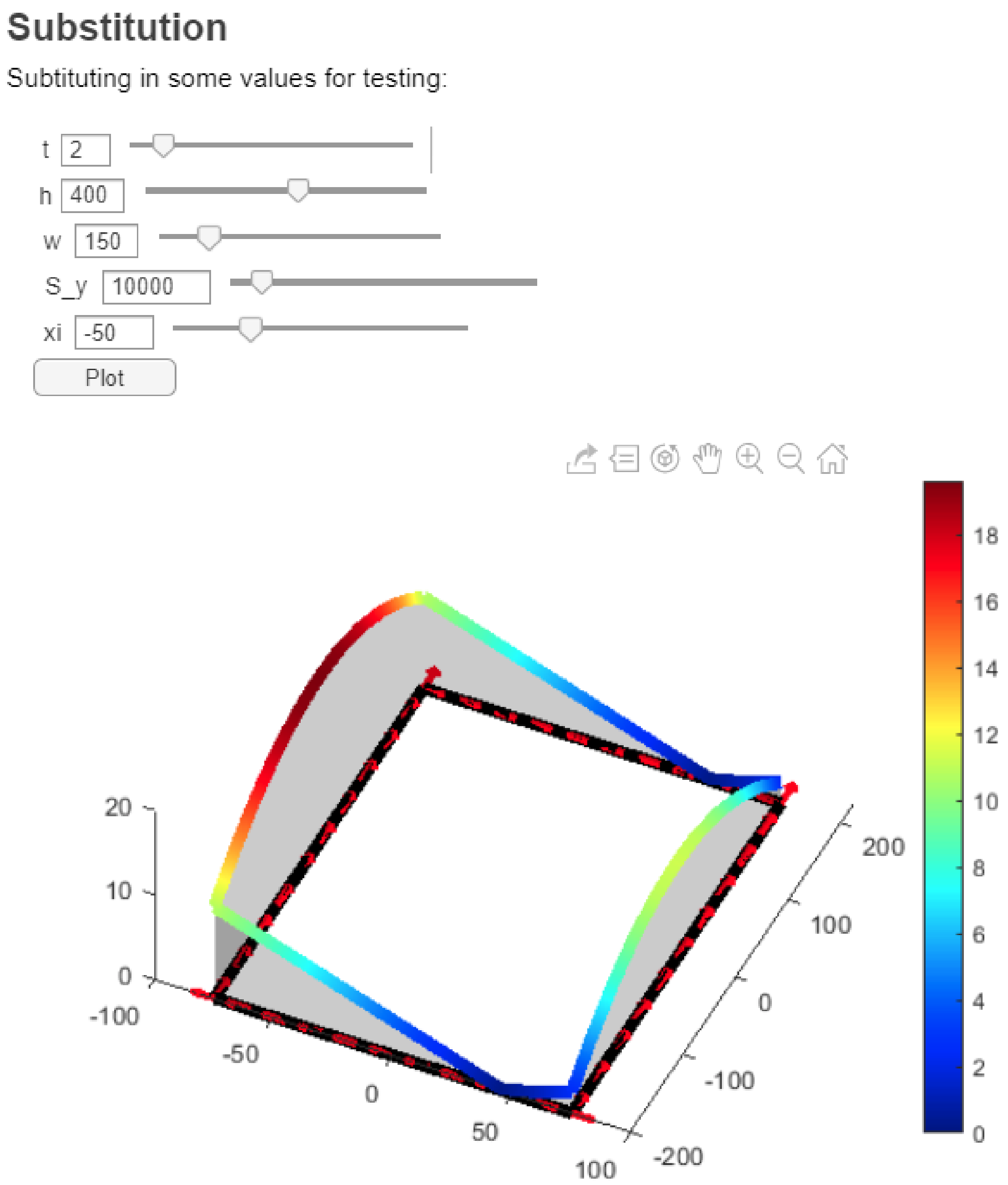
Generalise with different variables

Extrapolate to a class of similar problems

Simulation

- Widgets and live scripts offer opportunities to make simple system simulations (without Simulink)

- Some evidence suggests that exposing students to guided simulations *before* theoretical analysis enhances the learning experience



Substitution

Subtituting in some values for testing:

t 2

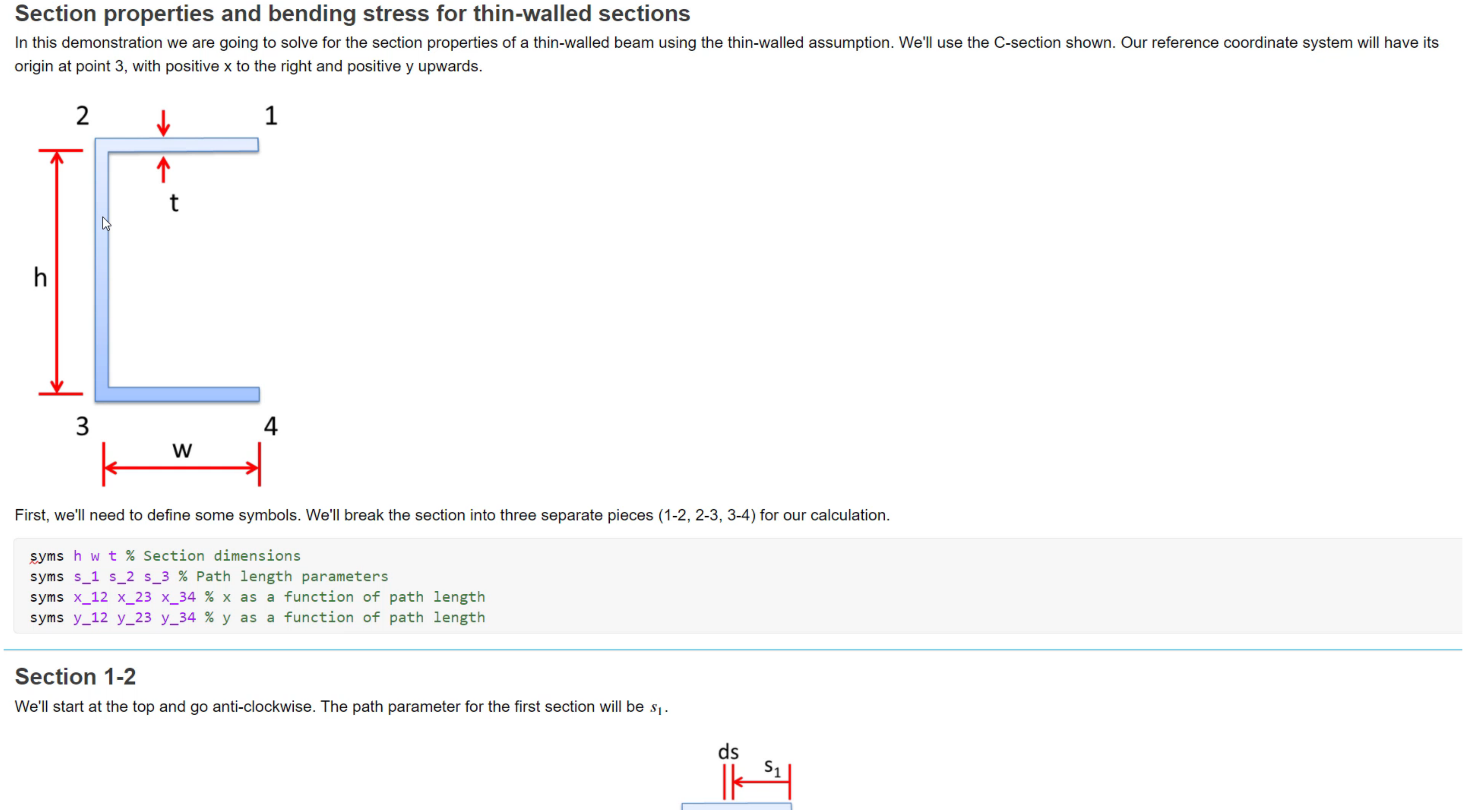
h 400

w 150

S_y 10000

xi -50

Live Script Example



Section properties and bending stress for thin-walled sections

In this demonstration we are going to solve for the section properties of a thin-walled beam using the thin-walled assumption. We'll use the C-section shown. Our reference coordinate system will have its origin at point 3, with positive x to the right and positive y upwards.

First, we'll need to define some symbols. We'll break the section into three separate pieces (1-2, 2-3, 3-4) for our calculation.

```
syms h w t % Section dimensions
syms s_1 s_2 s_3 % Path length parameters
syms x_12 x_23 x_34 % x as a function of path length
syms y_12 y_23 y_34 % y as a function of path length
```

Section 1-2

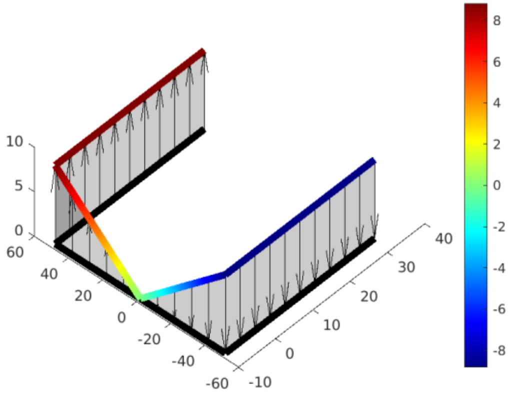
We'll start at the top and go anti-clockwise. The path parameter for the first section will be s_1.

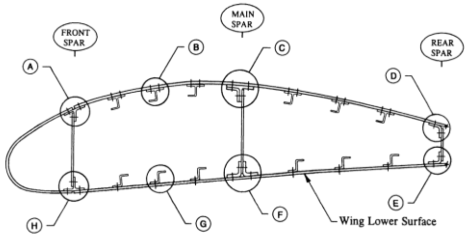
Use Code in Authentic Projects

Example

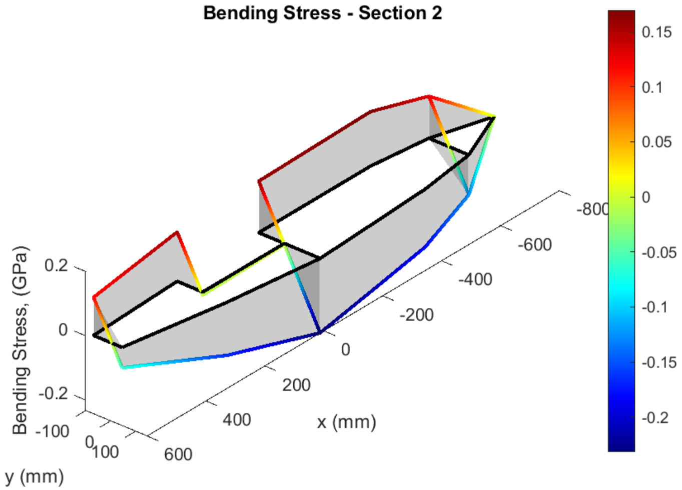
Students work in groups to analyse stress in the wing of a plane (that they eventually get to fly!)

Code built up through classes is progressively applied to solve the different challenges





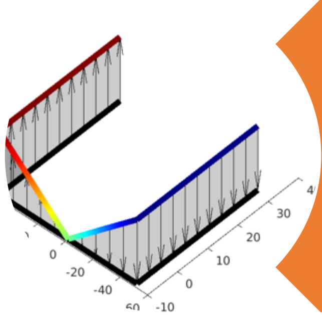
Students synthesise and adapt the code they have been given and build their own “toolbox”



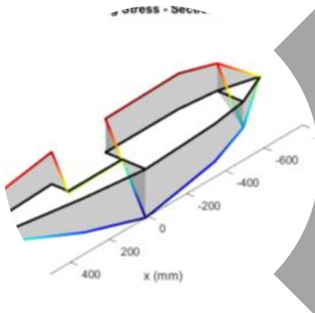
Software-Friendly Exams

- High-stakes examination in open-web, open-software computer lab
- Students have access to all the tools in their software arsenal

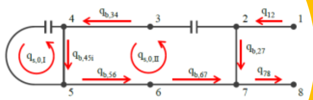
Pre-COVID invigilated exam



Learn theoretical concepts using code



Synthesise and adapt code to solve deeper problems



Apply code to solve problems in exam setting

Aligned Student Incentives

MATLAB Changes Exams

- Questions can be more authentic and address deeper concepts
- Fewer limits on question scope (e.g. asking students to solve systems of linear equations or complex integrals)
- More efficient solves mean more questions can be asked in the same period.
  - greater coverage of course topics
  - encourages students to study broadly

Concern

Learning concepts with code undermines theoretical understanding.

Code is easy to share and copy.

Students will just share code with a full worked solution.

Response

Separating concepts and algorithms from mathematical implementation creates *deeper intuition*.

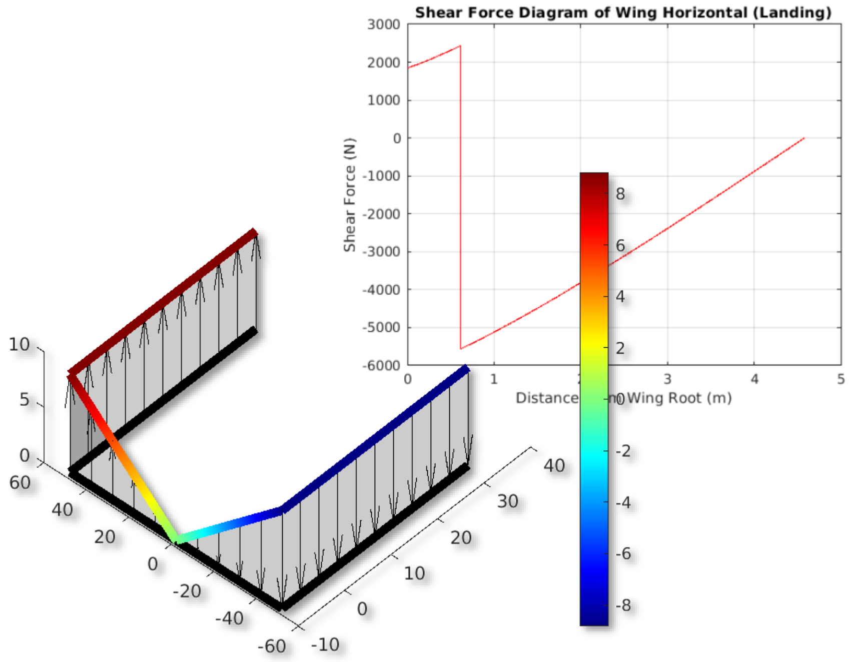
Copying is a feature, not a bug.

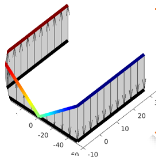
Leverage sharing to help everyone.

Ask high level questions to ensure students understand and can use the software.

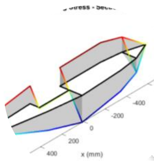
Summary

- Software skills are a necessary requirement of authentic learning and assessment for engineers
- Coding can augment theoretical concepts with rapid simulation and visualisation
- Integration of software into classes, projects and exams is critical to align student incentives and motivation

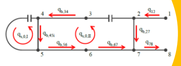




Learn theoretical concepts using code



Synthesise and adapt code to solve deeper problems



Apply code to solve problems in exam setting

Student Feedback

The way the course strongly emphasized the use of MATLAB and other calculation softwares felt very relevant for employment in the future.

… the use of Matlab, it really shows how Matlab is usefully in real life

Garth's MATLAB code proved to be the aspect of the course I found the most useful.