Poh Leong Hien
ceeplh@nus.edu.sg

CE5377: Numerical Methods in Mechanics and Envr. Flows
CE6077: Advanced Numerical Methods in Mechanics and Envr. Flows

Outline for Part 1

- Solution of nonlinear equations
- Numerical solutions to Eigen-problems
- Solution of differential equations with Finite Difference Method
- Consistency, convergence, and stability issues

Outline for Part 2

- Applications to environmental flows
- Solution with Finite Volume Method; Comparison with Finite Difference Method
- Transport processes in environmental flows

Prerequisites

EG1109, CE2134, or equivalent courses in engineering mechanics and hydraulics

Double-Coded module

Degree by coursework (M.Sc., B.Eng) : to register under CE5377
Degree by research (M.Eng, Ph.D.) : to register under CE6077
Note: Module content is the same for all students. Assignments for CE6077 will be more "research" based, and require more fundamental discussions.

Assessment for module

- CA for Part 1 : 30% (tutorials, tests, project, **participation**)
- CA for Part 2 : 30% (tutorials, tests, project)
- Final Examination : 40%

Software required for Part 1

- Students are expected to write subroutines using MATLAB.
- MATLAB is available in the computers inside the CE Structural Lab.
- Alternative: to register for an account in NUS HPC to use MATLAB.
- Another alternative: Octave (freeware) – see help file.

References

- Bathe, K.J., Finite element procedures, Prentice-Hall, 1996.
- Lindfield, G.R., Penny, J.E.T., Numerical methods using MATLAB, Elsevier, 2012.
- Mathews, J.H., Fink, K.D., Numerical methods using MATLAB, 2004.

Poh Leong Hien
ceeplh@nus.edu.sg

Outline

- Basic equations in engineering mechanics (1 hour)
  - Governing equation, stress strain relations, strain displacement relations
  - Boundary conditions
  - Introduction to MATLAB

- Solving nonlinear equations (3 hours)

- Numerical eigenvalue problems (3 hours)
  - Inverse iteration method
  - Forward iteration method
  - Gram Schmidt Orthogonalization
  - Subspace iteration

- Mathematical models in engineering problems (3 hours)
  - Ordinary and partial differential equations in engineering
  - Boundary and initial value problems
  - Finite difference method

- Initial value problems (4 hours)

- Boundary value problems and partial differential problems (4 hours)

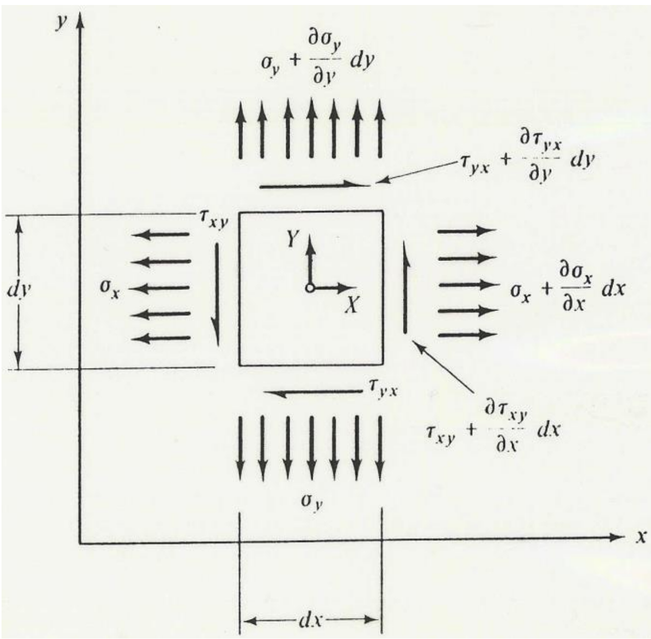
Poh Leong Hien
ceeplh@nus.edu.sg

1 Basic equations in engineering mechanics

In this section, some basic equations in engineering mechanics relevant to this course are summarized.

1.1 Two dimensional elasticity equations

Consider a small differential element of size *dx* and *dy*, with components of the body force f_x and f_y. Summing the forces and divided by area (*dxdy*) in each direction gives



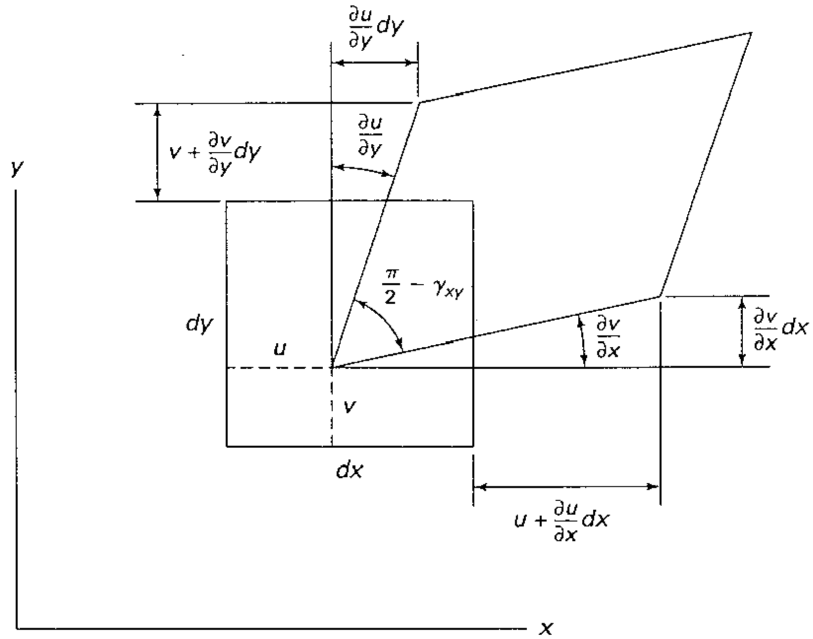
$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{yx}}{\partial y} + f_x = 0 \quad \text{and} \quad \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + f_y = 0 \tag{1.1}$$

or in tensor notation,

$$\frac{\partial \sigma_{ji}}{\partial x_j} + f_i = 0 \tag{1.2}$$

Poh Leong Hien
ceeplh@nus.edu.sg

1.2 Strain-displacement relationship



$$\varepsilon_{xx} = \frac{\partial u}{\partial x} \quad , \quad \varepsilon_{yy} = \frac{\partial v}{\partial y} \quad , \quad \gamma_{xy} = 2\varepsilon_{xy} = \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \tag{1.3}$$

or in tensor notation,

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \tag{1.4}$$

Poh Leong Hien
ceeplh@nus.edu.sg

1.3 Plane stress and plane strain

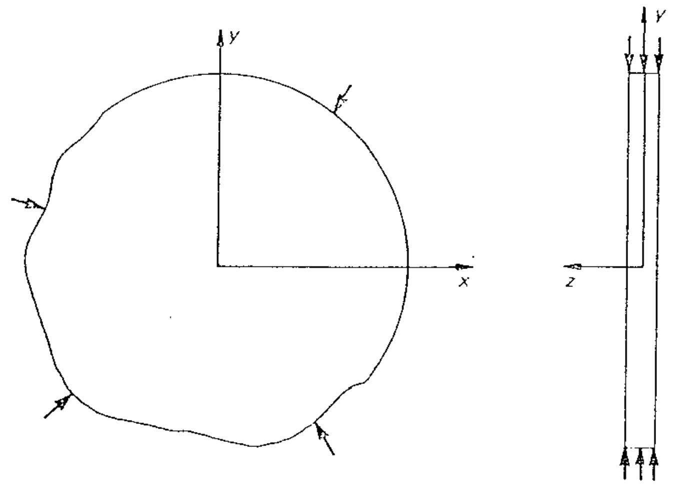
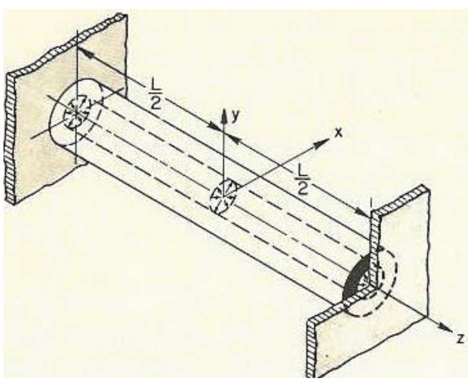


Figure 3-2 Plane stress: thin plate with in-plane loading.

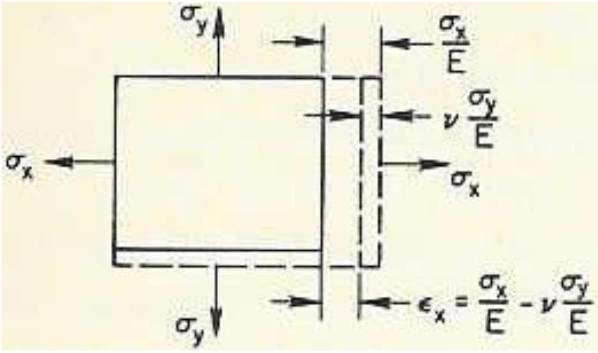
Plane stress: the stress variation across the thickness is neglected (e.g. for thin geometries). Thus $\sigma_{zz} = \tau_{xz} = \tau_{yz} = 0$.



Plane strain: the strain along the thickness is neglected (e.g. for very thick geometries). Thus $\varepsilon_{zz} = \gamma_{xz} = \gamma_{yz} = 0$. Note that the stress is non-zero, i.e. σ_{zz} = constant.

Poh Leong Hien
ceeplh@nus.edu.sg

1.4 Stress-strain relationships



For linear isotropic elasticity,

$$\begin{aligned}
\varepsilon_{xx} &= \frac{1}{E}\left[\sigma_{xx} - \nu\left(\sigma_{yy} + \sigma_{zz}\right)\right] \\
\varepsilon_{yy} &= \frac{1}{E}\left[\sigma_{yy} - \nu\left(\sigma_{xx} + \sigma_{zz}\right)\right] \\
\varepsilon_{zz} &= \frac{1}{E}\left[\sigma_{zz} - \nu\left(\sigma_{xx} + \sigma_{yy}\right)\right] \\
\varepsilon_{xy} &= \frac{1}{2G}\sigma_{xy}
\end{aligned} \tag{1.5}$$

where $G = E/2(1+\nu)$.

The stresses are given by

$$\begin{aligned}
\sigma_{xx} &= \frac{2G\nu}{1-2\nu}\left(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}\right) + 2G\varepsilon_{xx} \\
\sigma_{yy} &= \frac{2G\nu}{1-2\nu}\left(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}\right) + 2G\varepsilon_{yy} \\
\sigma_{zz} &= \frac{2G\nu}{1-2\nu}\left(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}\right) + 2G\varepsilon_{zz} \\
\sigma_{xy} &= 2G\varepsilon_{xy}
\end{aligned} \tag{1.6}$$

which can be expressed in the tensorial form as

$$\begin{aligned}
\sigma_{ij} &= \frac{2G\nu}{1-2\nu}\delta_{ij}\varepsilon_{mm} + 2G\varepsilon_{ij} \\
&= \frac{2G\nu}{1-2\nu}\left(\frac{\partial u_m}{\partial x_m}\right)\delta_{ij} + G\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)
\end{aligned} \tag{1.7}$$

Poh Leong Hien
ceeplh@nus.edu.sg

In the matrix form,

$$\begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zz} \end{Bmatrix} = \begin{bmatrix} 1/E & -\nu/E & -\nu/E & 0 & 0 & 0 \\ -\nu/E & 1/E & -\nu/E & 0 & 0 & 0 \\ -\nu/E & -\nu/E & 1/E & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/G & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/G & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/G \end{bmatrix} \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zz} \end{Bmatrix} \tag{1.8}$$

which can be rearranged to give

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zz} \end{Bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} \end{bmatrix} \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zz} \end{Bmatrix} \tag{1.9}$$

For plane strain where $\varepsilon_{zz} = 0$,

$$\begin{aligned} \varepsilon_{xx} &= \left(\frac{1-\nu^2}{E}\right)\sigma_{xx} + \left[\frac{-\nu(1+\nu)}{E}\right]\sigma_{yy} \\ \varepsilon_{yy} &= \left[\frac{-\nu(1+\nu)}{E}\right]\sigma_{xx} + \left(\frac{1-\nu^2}{E}\right)\sigma_{yy} \\ \varepsilon_{xy} &= \frac{1}{2G}\sigma_{xy} \end{aligned} \tag{1.10}$$

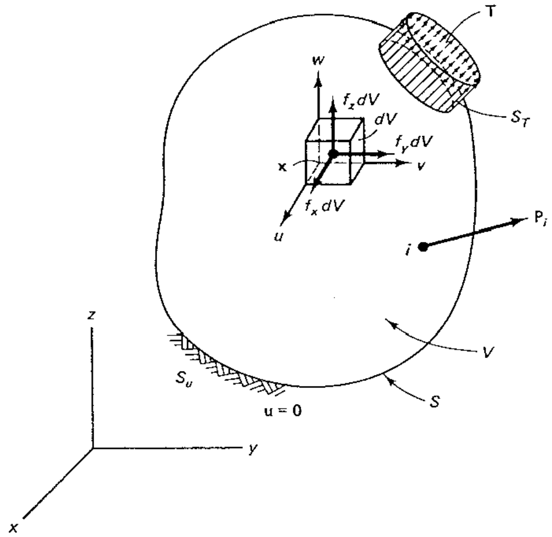
For plane stress where $\sigma_{zz} = 0$,

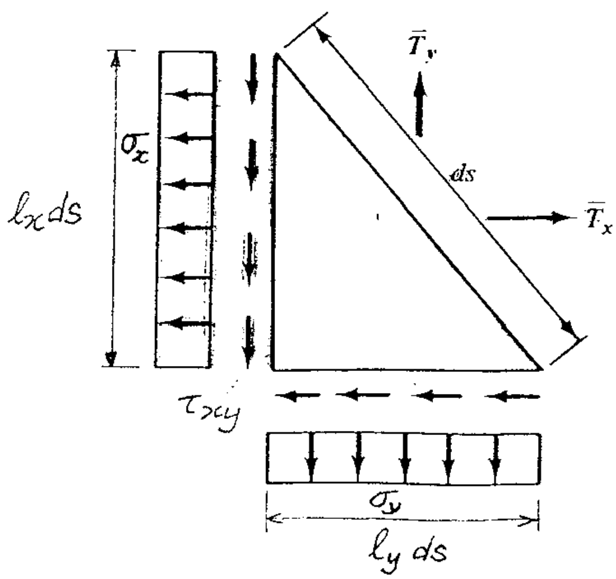
$$\begin{aligned} \varepsilon_{xx} &= \left(\frac{1}{E}\right)\sigma_{xx} + \left[\frac{-\nu}{E}\right]\sigma_{yy} \\ \varepsilon_{yy} &= \left[\frac{-\nu}{E}\right]\sigma_{xx} + \left(\frac{1}{E}\right)\sigma_{yy} \\ \varepsilon_{xy} &= \frac{1}{2G}\sigma_{xy} \end{aligned} \tag{1.11}$$

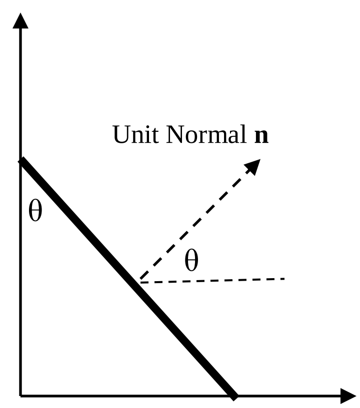
Note that the same set of equations for plane strain in (1.10) can be used for plane stress by substituting $E(1+2\nu)/(1+\nu)^2$ for E and $\nu/(1+\nu)$ for ν to get (1.11).

Poh Leong Hien
ceeplh@nus.edu.sg

1.5 Traction boundary conditions







The prescribed surface tractions at the boundary of the body (for static case), T_x and T_y (force per unit length per unit thickness), is related to the stress by

$$T_x(ds \times 1) = \sigma_{xx}(ds \times \cos\theta \times 1) + \tau_{xy}(ds \times \sin\theta \times 1) \tag{1.12}$$

The unit normal to surface is given by

$$\mathbf{n} = \begin{bmatrix} n_x \\ n_y \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \tag{1.13}$$

From (1.12) and (1.13),

$$T_x = \sigma_{xx}\ n_x + \tau_{xy}\ n_y \tag{1.14}$$

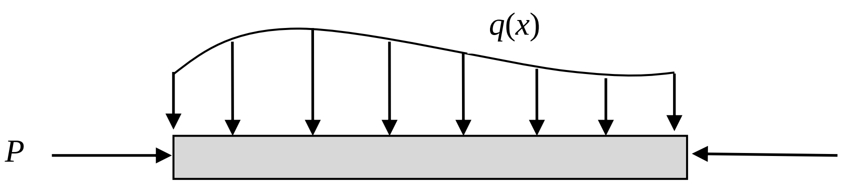
Repeat the same process to get

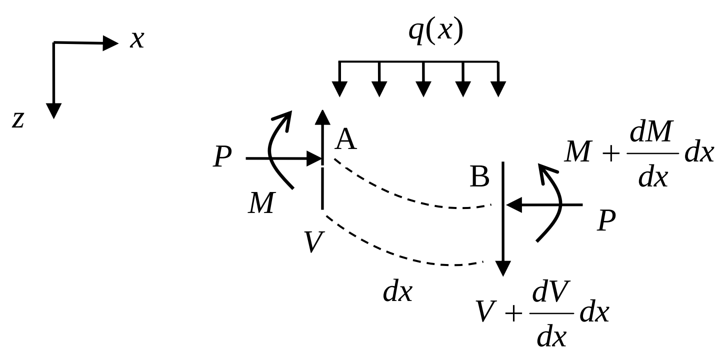
$$T_y = \sigma_{yy}\ n_y + \tau_{yx}\ n_x \tag{1.15}$$

In tensorial form,

$$T_i = \sigma_{ij}\ n_j \tag{1.16}$$

1.6 Beam under axial and transverse load





From equilibrium,

$$\sum F_z = 0 \Rightarrow V + \frac{dV}{dx}dx - V + q(x)\ dx = 0 \Rightarrow \frac{dV}{dx} + q(x) = 0$$

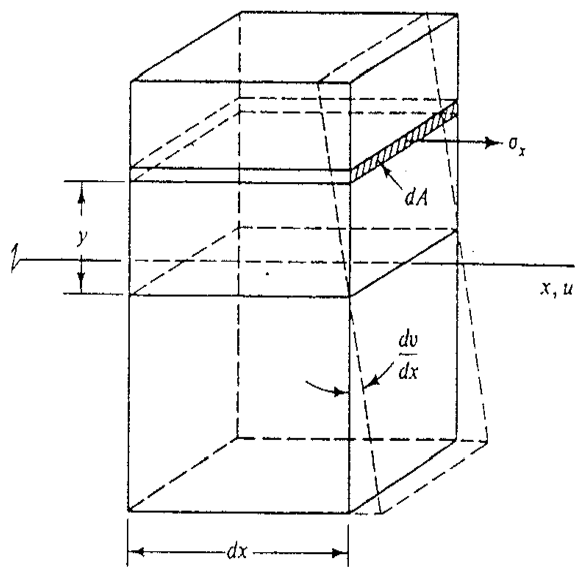
$$\sum M_B = 0 \Rightarrow M + \frac{dM}{dx}dx - M - Vdx - Pdw + qdx\frac{dx}{2} = 0 \tag{1.17}$$

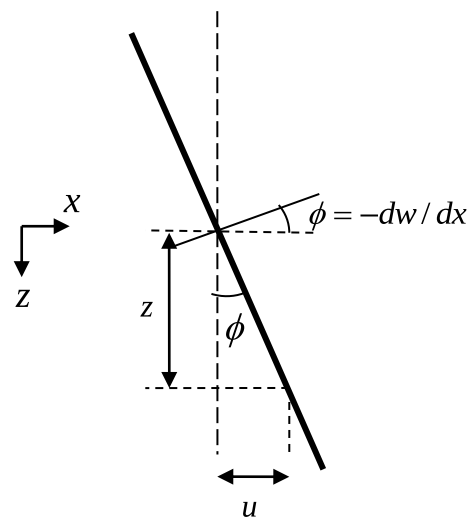
$$\Rightarrow V = \frac{dM}{dx} - P\frac{dw}{dx} \quad \text{(ignoring higher order term)}$$

Poh Leong Hien
ceeplh@nus.edu.sg

From (1.17a) and (1.17b),

$$\frac{d^2M}{dx^2} - p\frac{d^2w}{dx^2} + q = 0 \tag{1.18}$$





From the figures, we have

$$\tan\phi = u/z = -dw/dx \quad \Rightarrow \quad u(x) = -z\frac{dw(x)}{dx} \tag{1.19}$$

such that

$$\varepsilon(x) = -z\frac{d^2w(x)}{dx^2} \tag{1.20}$$

(*check: negative curvature produce positive strain for positive z*)

For thin beams ($\sigma_z = 0$)

$$\sigma_x = E\varepsilon_x \tag{1.21}$$

The bending moment is thus

$$M = \int_A \sigma_x z\, dydz = E\int_A \varepsilon_x z\, dydz = -E\int_A z^2 \frac{d^2w(x)}{dx^2}\, dydz = -EI\frac{d^2w(x)}{dx^2} \tag{1.22}$$

where (1.20) and (1.21) are utilized.

Poh Leong Hien
ceeplh@nus.edu.sg

Substitute (1.22) into (1.17b) to get the expression for shear

$$V = -\frac{d}{dx}\left(EI\,\frac{d^2 w}{dx^2} \right) - P\,\frac{dw}{dx} \tag{1.23}$$

The governing equation is obtained by substituting (1.22) into (1.18)

$$\frac{d^2}{dx^2}\left(EI\,\frac{d^2 w}{dx^2} \right) + P\,\frac{d^2 w}{dx^2} = q \tag{1.24}$$

For a beam of constant *EI*, the equation for beam becomes

$$EI\,\frac{d^4 w}{dx^4} + P\,\frac{d^2 w}{dx^2} = q \tag{1.25}$$

The boundary conditions (for constant *EI*) are:

- Fixed end

$$w = 0 \quad , \quad \theta = \frac{dw}{dx} = 0$$

- Free end

$$V = 0 \;\Rightarrow\; \frac{d^3 w}{dx^3} = 0 \quad , \quad M = 0 \;\Rightarrow\; \frac{d^2 w}{dx^2} = 0$$

- Simply supported

$$w = 0 \quad , \quad M = 0 \;\Rightarrow\; \frac{d^2 w}{dx^2} = 0$$

Poh Leong Hien
ceeplh@nus.edu.sg

Some useful MATLAB commands and functions:

Case sensitive	‘A’ is not the same as ‘a’
%	Comment
A^2	A*A (applicable to square matrix only)
A**.**^2	Square each element of A (dot means element-level)
cos(A)	Take cosine of each element
cos(A**.**/2)	Divide each element by 2 and take cosine
C = [A B]	Augment arrays A and B and assign it to C
A = zeros(2,3)	Create 2x3 matrix of all zeros
A = ones(2,3)	Create 2x3 matrix of all ones
A = eye(3)	Create 3x3 identity matrix
b=1:2:9	b = [1 3 5 7 9] (1:9 generates 1 to 9 with default increment of 1)
b= diag(A)	Create vector b which is extracted from the main diagonal of matrix A
A = diag(b)	Create diagonal matrix A (square) with vector b on its main diagonal
A’	Transpose (conjugate transpose if A is complex)
[m,n] = size(A)	m= no. of rows in A, n= no. of columns in A
m = length(b)	m = Length (size) of vector b
det(A)	Determinant (avoid using this an indicator of singularity)
inv(A)	Inverse of matrix A (**avoid** using this to solve equations)
X = A\B	Matrix “left division”. Reads like X = A^{-1}*B, but actually solves for X in AX=B by Gauss elimination (and its variation where appropriate).
X = A**.**/B	X is a matrix where X(i,j) = A(i,j)/B(i,j). (A and B must be of the same size unless one of them is a scalar.)
t=0:0.2:2*pi; plot(2*cos(*t*), 3*sin(*t*))	Plot an ellipse defined by $(2\cos(t), 3\sin(t))$.