Subject	ENGR30003: Numerical Programming for Engineers
Semester	Semester 2, 2017
Document	Assignment 2 - Report
Student Name	Sebastian Baker
Student Number	757931
Due Date	Monday, 16th October 2017, 5:00pm

Root Finding

2.1. Analytical solution

$$\tan(\theta) = 2\cot(\beta) \cdot \frac{M^2 \sin^2 - 1}{M^2(\gamma + \cos(2\beta)) + 2}$$

Find roots for $\theta = 0$:

$$\Rightarrow \quad 0 = 2\cot(\beta) \cdot \frac{M^2 \sin^2(\beta) - 1}{M^2(\gamma + \cos(2\beta)) + 2}$$

Roots can only occur if:

$$0 = \cot(\beta) \cdot (M^2 \sin^2(\beta) - 1)$$

$\therefore$ There are 2 roots:

1. One root at $0 = \cot(\beta)$

$$\Rightarrow 0 = \frac{\cos(\beta)}{\sin(\beta)}$$

$$\Rightarrow 0 = \cos(\beta)$$

$$\Rightarrow \beta = \frac{\pi}{2} = 90^\circ$$

2. Other root at $0 = M^2 \sin^2(\beta) - 1$

$$\Rightarrow \sin^2(\beta) = \frac{1}{M^2}$$

$$\Rightarrow \beta = \sin^{-1}\left(\frac{1}{M}\right)$$

$\therefore$ The two roots are:

$$\beta = 90^\circ \quad \text{and} \quad \beta = \arcsin\left(\frac{1}{M}\right)$$

2.2. Graphical solutions

Figure 1: f(beta) for M=1.5 and theta=5, 10 and 15

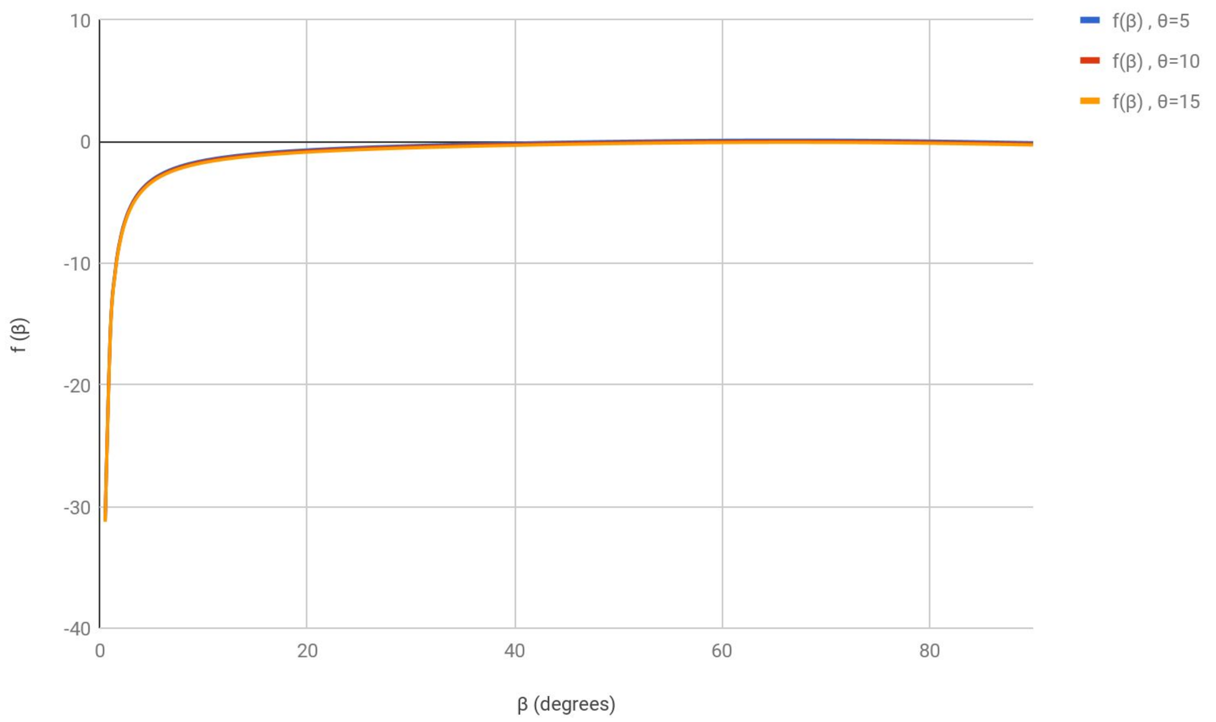


Figure 2: f(beta) for M=1.5 and theta=5, 10 and 15 over a smaller domain

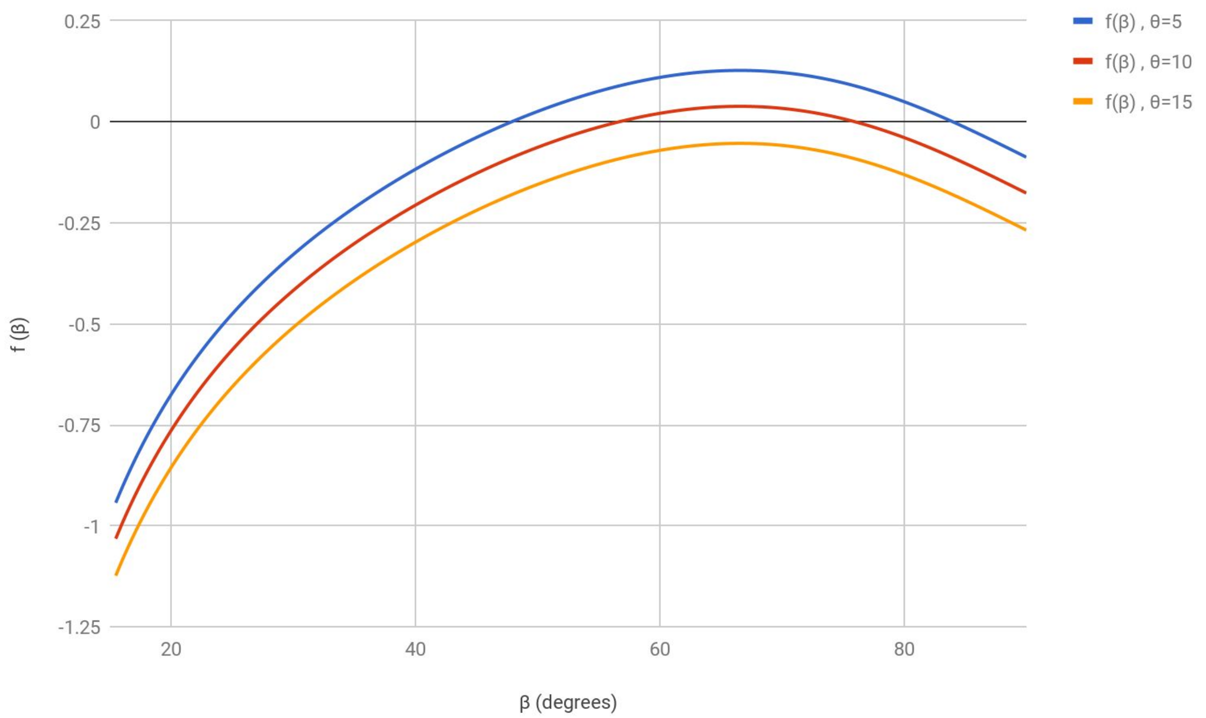
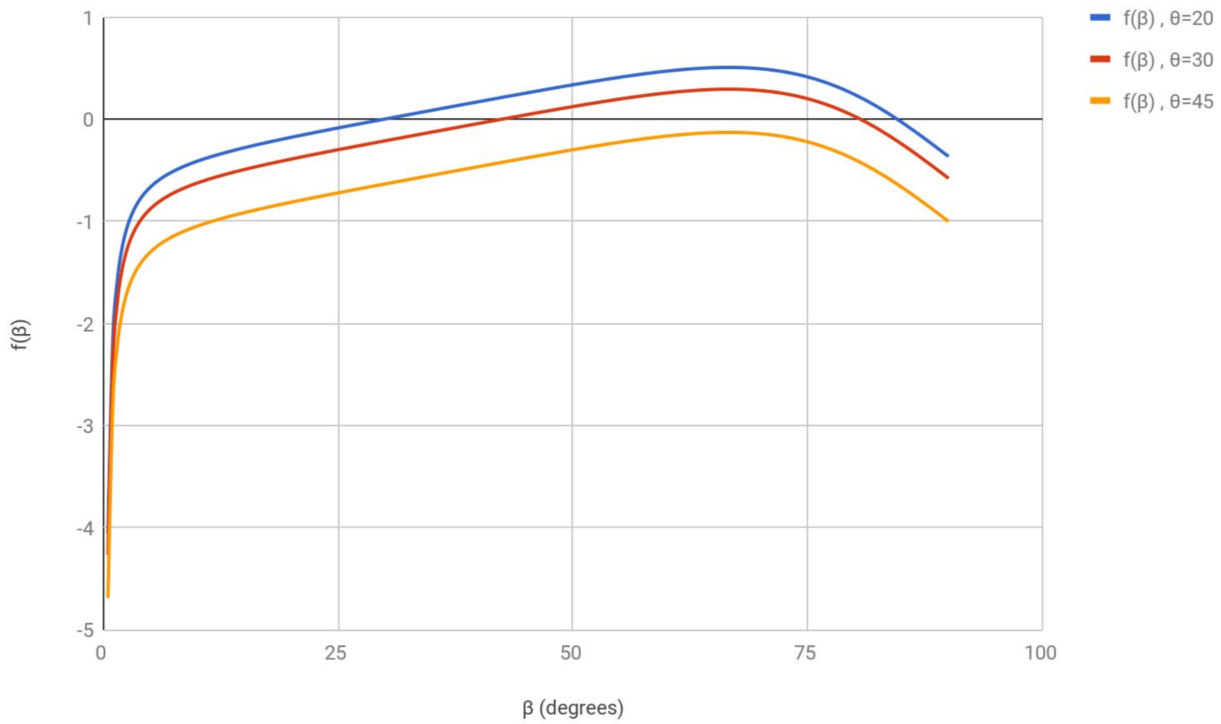


Figure 3: f(beta) for M=5 and theta=20, 30 and 45



From Figures 2 and 3, the roots of f(beta) for M can be observed at:

M	theta (degrees)	Lower root (degrees)	Upper root (degrees)
1.5	1.5	48	84
1.5	10	58	76
1.5	15	N/A	N/A
5	20	30	84
5	30	42	80
5	45	N/A	N/A

The roots appear to get closer together as theta increases, and further apart as M increases. Greater theta pushes f(beta) down and greater M pushes f(beta) up,

For M=1.5, the theta_max value is between 10 degrees and 15 degrees, since two roots occur for theta=10 degrees, but there are no roots for theta=15 degrees. A graphical estimation for theta_max based would be 12 degrees.

For M=5, the theta_max value is between 30 degrees and 45 degrees, since two roots occur for theta=30 degrees, but there are no roots for theta=45 degrees. A graphical estimation for theta_max would be 40 degrees.

2.3. Newton-Raphson root finding C program

2.3.a. Choice of initial guesses

Based on the graphical solutions, a starting guess of beta_lower=20 degrees, and beta_upper=90 degrees was chosen.

2.3.c. θ − β − M diagrams for M = 1.5, 3.0, 4.0, 5.0, 7.0, 8.0

Figure 4: θ − β − M diagram for M = 1.5

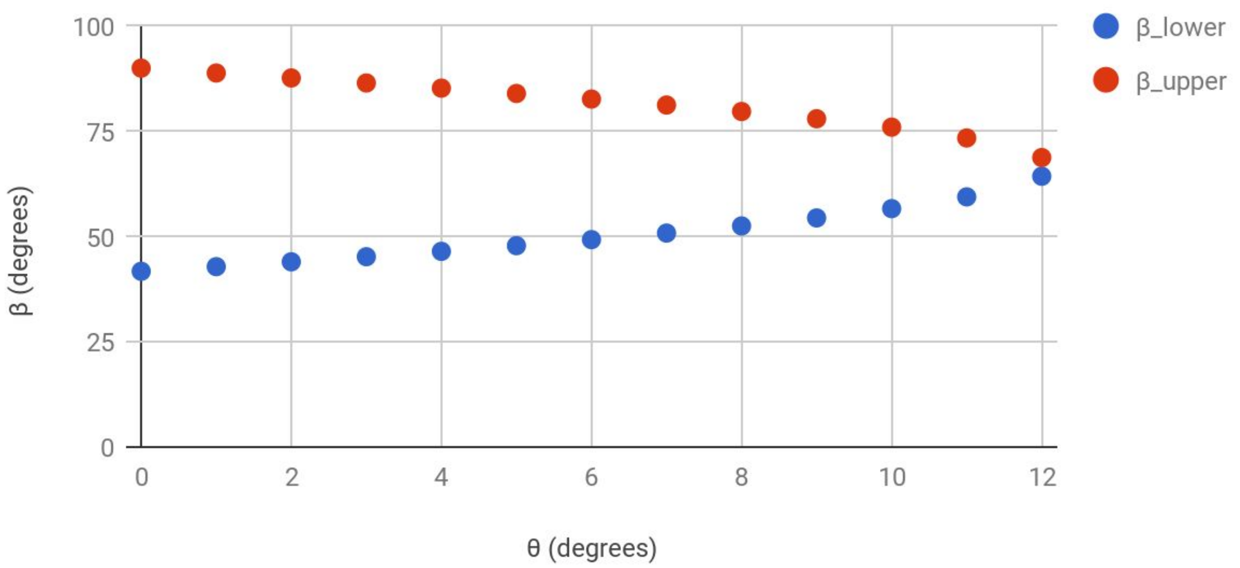


Figure 5: θ − β − M diagram for M = 3.0

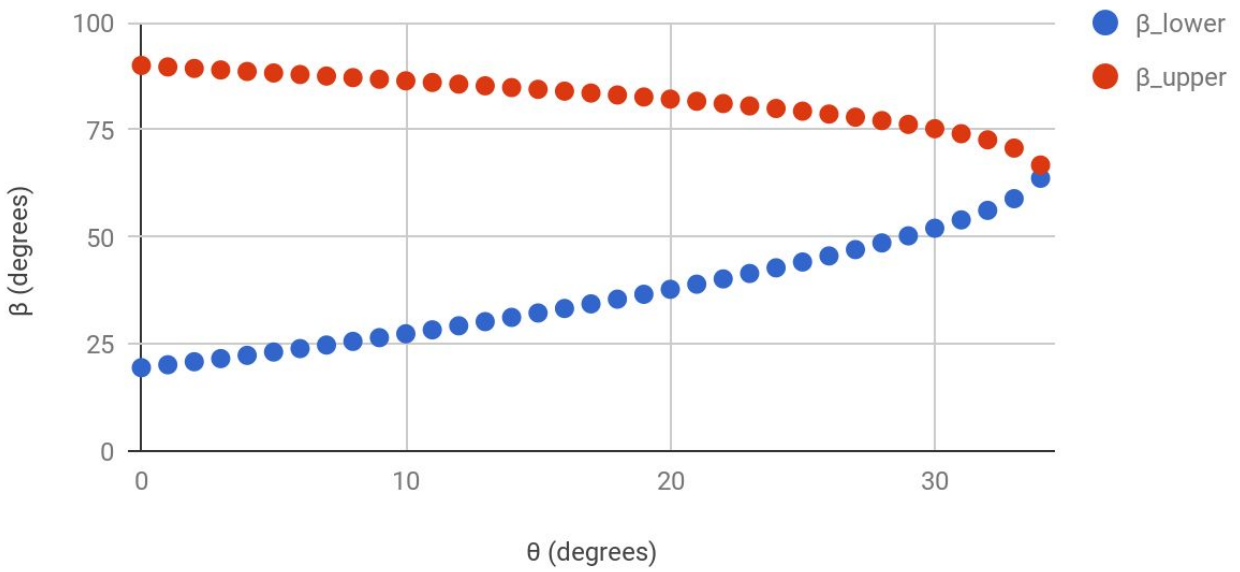


Figure 6: θ − β − M diagram for M = 4.0

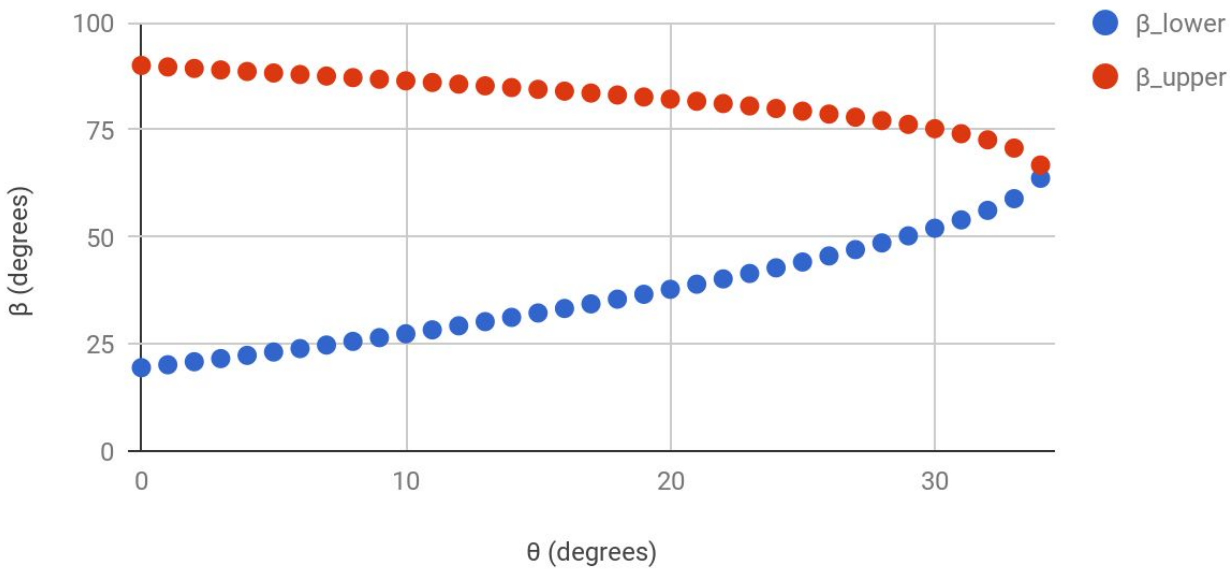


Figure 7: θ − β − M diagram for M = 5.0

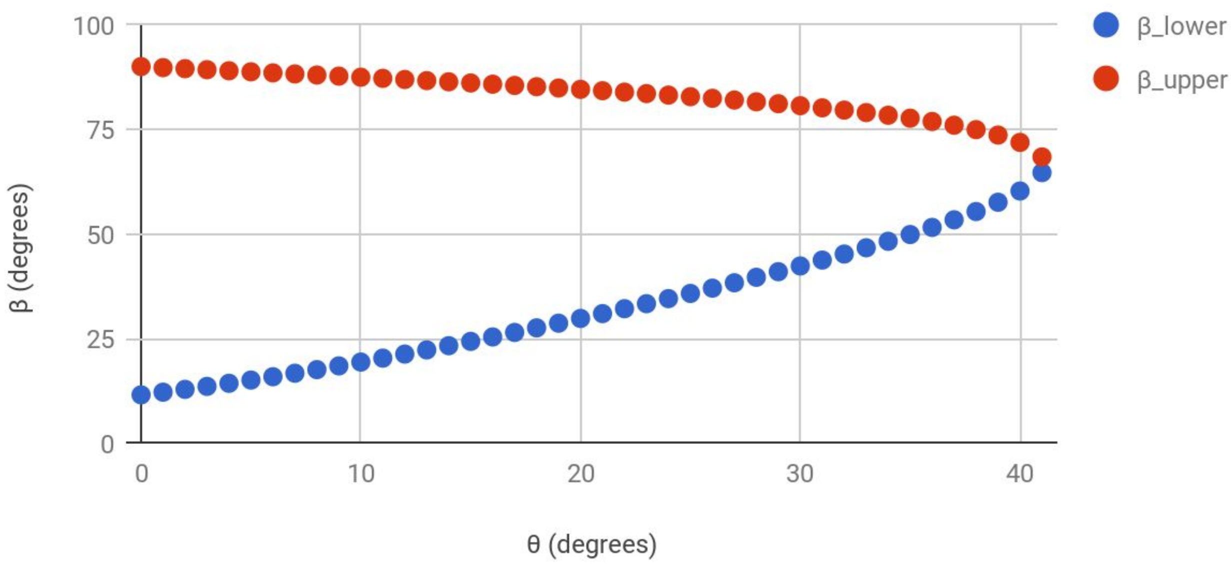


Figure 8: θ − β − M diagram for M = 7.0

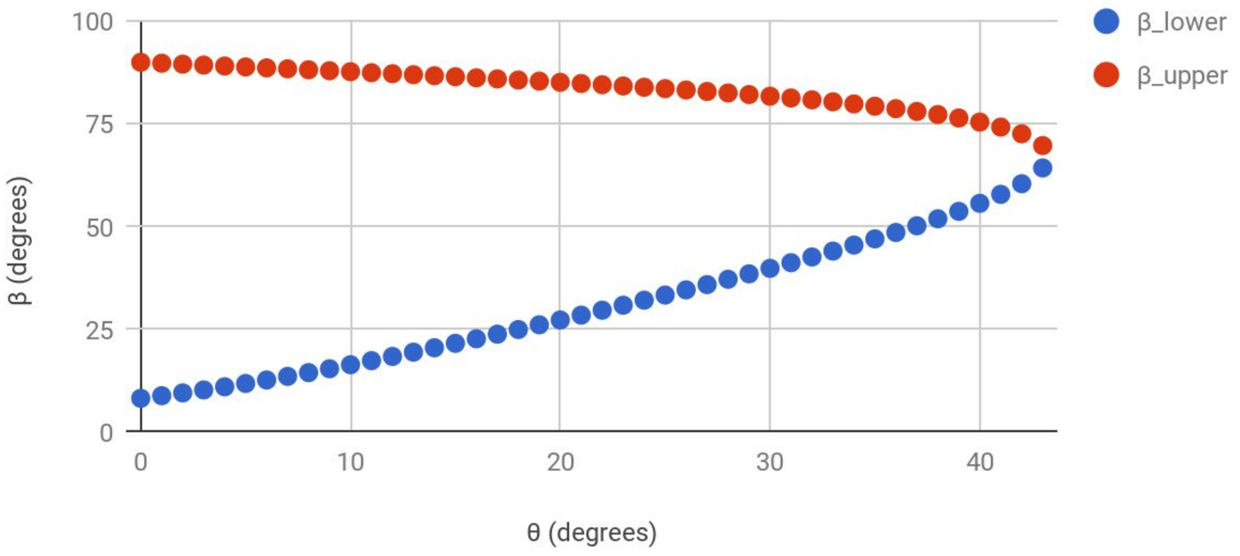
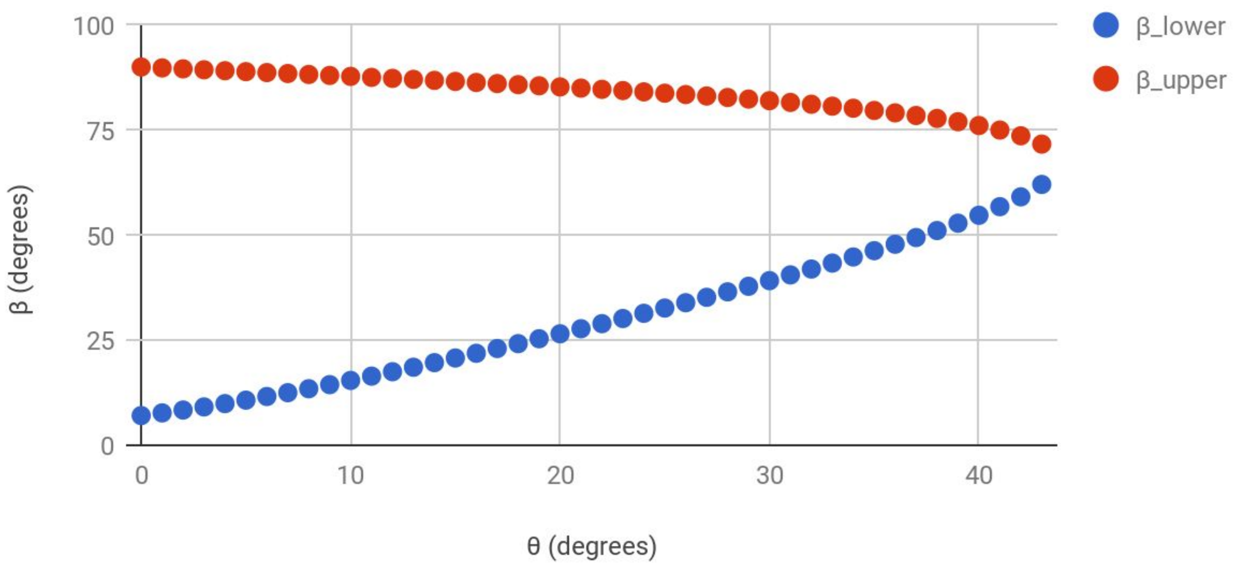


Figure 9: θ − β − M diagram for M = 8.0



Interpolation

5.* Choice of points used for Lagrange quadratic

Since a can only go through 3 points, points needed to be chosen for the Lagrange quadratic interpolations. I was decided the the three closest points to xo (with xo bounded by those points) would be used for the quadratic interpolation.

5.* Interpolation plots

Figures 10 and 11 were interpolations of the set:

x	f(x)
0	0
1	11
3	17
8	17

Figure 11 used the general Lagrange interpolation through all of the points.

Figure 12 was made for testing purposes, it’s an interpolation of the set:

x	f(x)
0	0
1	11
3	17
8	17
11	11
13	17
18	17

Figure 10: Cubic spline and 2nd order Lagrange interpolation

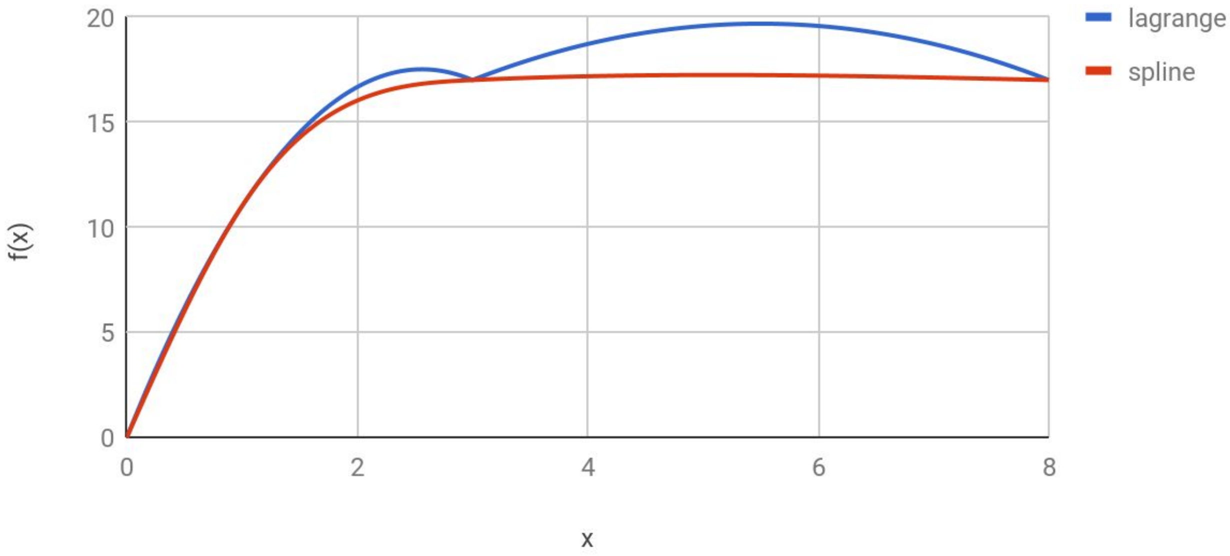


Figure 11: Cubic spline and 3rd order Lagrange interpolation

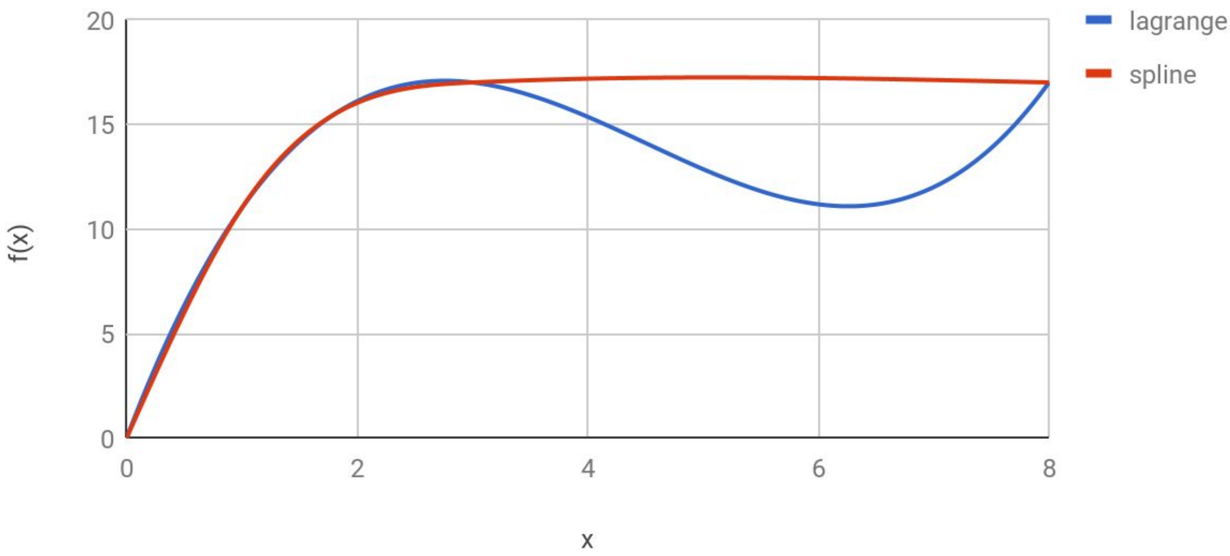
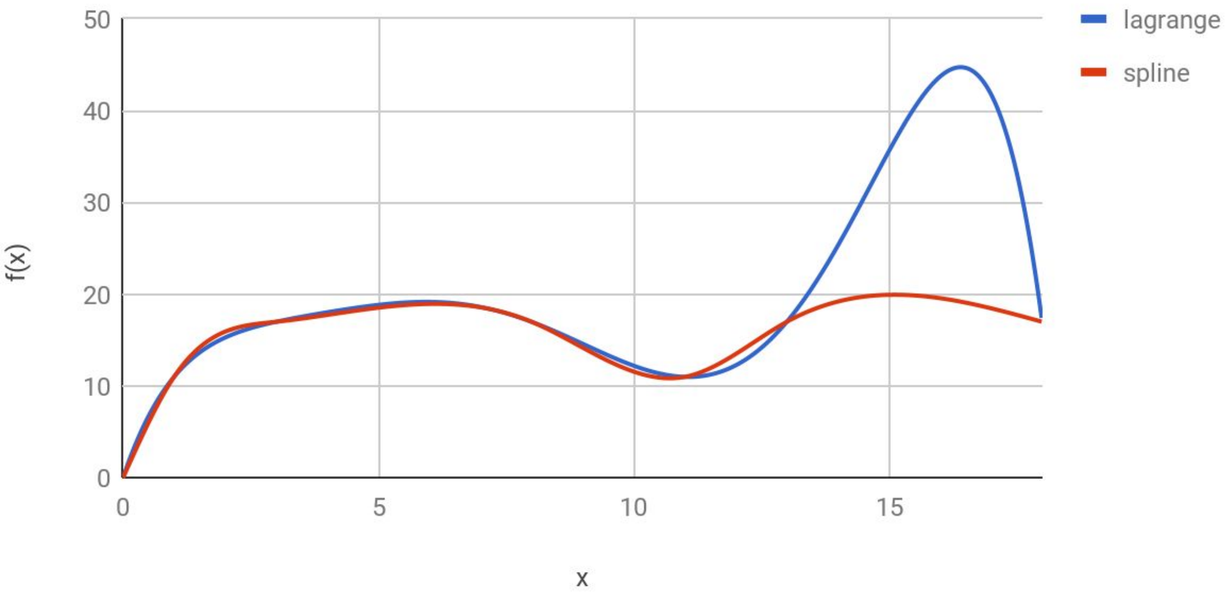


Figure 12: Cubic spline and Lagrange interpolation



5.* Comparison of interpolation techniques

Based on Figures 10 - 12, splines appear to be smoother than general Lagrange polynomials and quadratic Lagrange polynomials.

The discontinuity in derivative of f(x) at x=3 in Figure 10 is due to the choice of points for the Lagrange quadratic. At xo=4, the implementation switched from using the first 3 points to using the last 3 points. Such discontinuities may be unwanted in an interpolation, in which case a spline would be better.

Differentiation, differential equations

To evaluate the performance of the differential equation solvers, the grid-independent reference solutions were generated, as well as the solutions under varying Nx (Figure 13-14), and varying Nt (Figure 15-16). For every point in the grids, the absolute difference between the reference solutions and the variable solutions was calculated to determine the error in the solutions with varying Nx and Nt conditions.

Effect of Nx

Figure 13 shows that increasing Nx (decreasing dx) results in decreasing error, until the solution diverges, however, the implicit euler scheme could not be made to diverge.

Effect of Nt

Figure 15 shows that increasing Nt (decreasing dt) results in increasing error, until the solution diverges, however, the implicit euler scheme could not be made to diverge.

Effect of diffusion number

For both experiments, the explicit euler schemes would consistently diverge for diffusion numbers > 0.5. The implicit euler scheme did not diverge.

Effect of integration type

For all experiments, the explicit euler schemes diverged for diffusion numbers > 0.5. The implicit euler scheme did not diverge.

The fixed end explicit euler scheme had the lowest error, followed by the fixed end implicit euler and then the variable end explicit euler.

Conclusion

It appears that the implicit euler is the superior integration scheme since it does not diverge, however, there would be a performance-accuracy trade off due to the implicit scheme's use of matrices, which are slow for high Nx.

Do analyse the accuracy and precision of the various schemes properly, an analytical solution would need to be compared to euler solutions, since the grid independent reference solutions may be different from the actual solutions for the heat equation.

Figure 13: Average absolute error vs x step

Figure 14: Average absolute error with vs diffusion number (varying Nx)

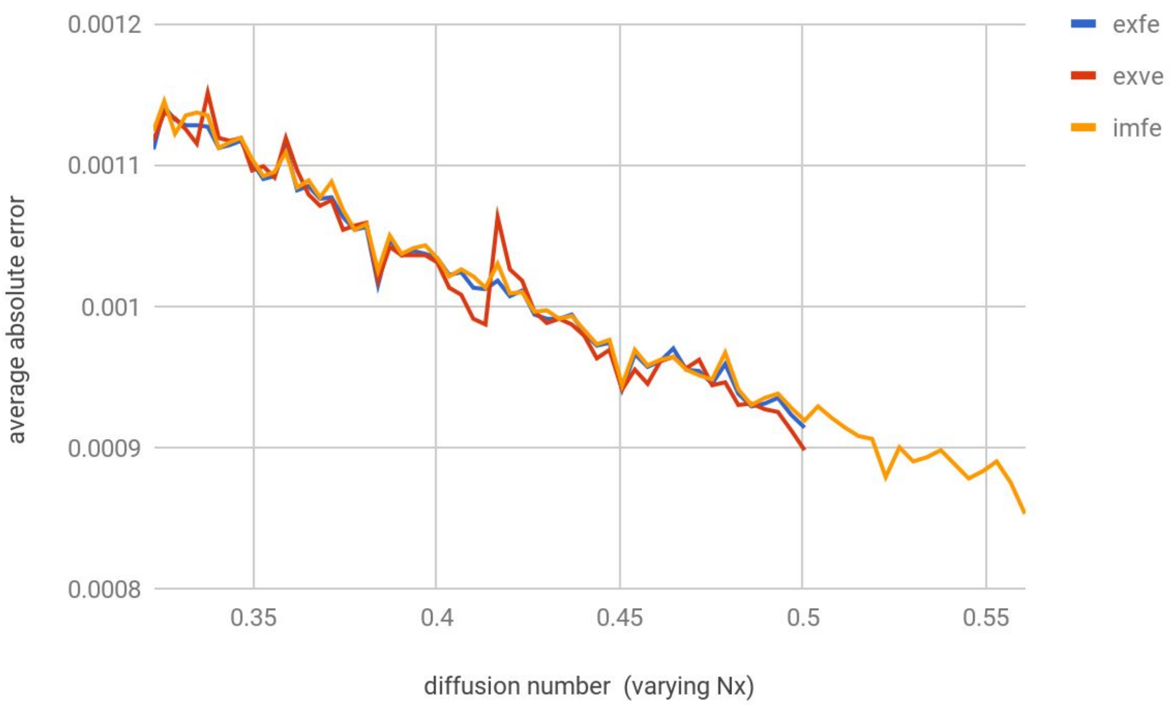


Figure 15: Average absolute error vs t step

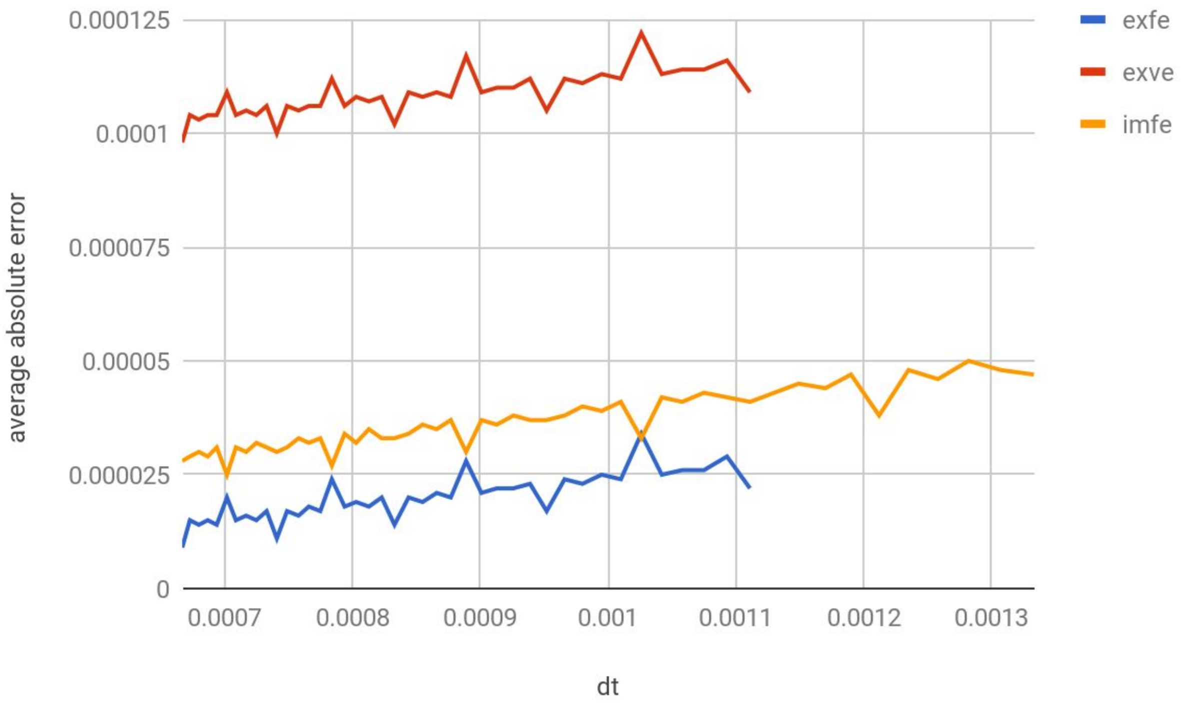


Figure 16: Average absolute error with vs diffusion number (varying Nt)