Chapter 19

Numerics in General

19.1 Introduction

1. In floating point form, rounding to $5S$,

$$84.175 \to 0.841\,75 \times 10^2 \qquad -528.685 \to -0.528\,68 \times 10^3 \tag{19.1.1}$$

$$0.000\,924\,138 \to 0.924\,14 \times 10^{-3} \qquad -362\,005 \to -0.362\,01 \times 10^6 \tag{19.1.2}$$

2. Rounding to $4S$,

$$-76.437\,125 \to -0.7644 \times 10^2 \qquad 60\,100 \to 0.6010 \times 10^5 \tag{19.1.3}$$

$$-0.000\,01 \to -1.0000 \times 10^{-5} \tag{19.1.4}$$

3. Using $5S$,

$$A = \frac{0.81534}{35.724 - 35.596} = \frac{0.81534}{0.128} = 6.3698 \tag{19.1.5}$$

$$B = \frac{0.8153}{35.72 - 35.60} = \frac{0.8153}{0.12} = 6.794 \tag{19.1.6}$$

$$C = \frac{0.815}{35.7 - 35.6} = \frac{0.815}{0.1} = 8.15 \tag{19.1.7}$$

$$D = \frac{0.82}{36 - 36} = \text{undefined} \tag{19.1.8}$$

4. Add the smallest pair of numbers together first, and then recursively repeat. This leads to the least rounding error in the final answer.

$$S = A + \epsilon + \epsilon + \cdots + \epsilon \qquad \epsilon < u \tag{19.1.9}$$

$$A + \epsilon \to A \tag{19.1.10}$$

Clearly, adding left to right will lead to $S = A$, whereas the number of small terms could be significant.

5. Rounding first before adding leads to a much greater error in the final answer than adding before rounding.

6. Direct evaluation using $3S$, and then using the nested method,

$$f(x) = x^3 - 7.5x^2 + 11.2x + 2.8 \qquad = 61.2 - 116 + 44.1 + 2.8 = -7.9 \tag{19.1.11}$$

$$g(x) = [(x - 7.5)x + 11.2]x + 2.8 \qquad = [(-3.56)x + 11.2]x + 2.8 \tag{19.1.12}$$

$$= (-14.0 + 11.2)x + 2.8 \qquad = -11.0 + 2.8 = -8.2 \tag{19.1.13}$$

Clearly the nested method is closer to the correct answer, $-8.336\,016$

7. Solving using the quadratic formula, with $6S$

$$0 = x^2 - 30x + 1 \qquad x = 15 \pm 4\sqrt{14} \tag{19.1.14}$$

$$x_1 = 15 - 4\sqrt{14} = 0.03335 \qquad x_2 = 15 + 4\sqrt{14} = 29.9666 \tag{19.1.15}$$

Solving using the product of roots formula,

$$x_1 = 29.9666 \qquad x_2 = \frac{c}{ax_1} = 0.0333705 \tag{19.1.16}$$

The second method produces a more accurate result for the smaller root.

8. Solving the quadratic equation using $4S$,

$$0 = x^2 - 40x + 2 \qquad x = \frac{40 \pm \sqrt{1600 - 8}}{2} \tag{19.1.17}$$

$$x_1 = \frac{40 - \sqrt{1592}}{2} = 0.05 \qquad x_2 = \frac{40 + \sqrt{1592}}{2} = 39.95 \tag{19.1.18}$$

Solving using the product of roots formula,

$$x_1 = 39.95 \qquad x_2 = \frac{c}{ax_1} = 0.05006 \tag{19.1.19}$$

The second method produces a more accurate result for the smaller root.

9. Repeating Problem 7,

(a) Solving using the quadratic formula, with $4S$

$$0 = x^2 - 30x + 1 \qquad x = 15 \pm 4\sqrt{14} \tag{19.1.20}$$

$$x_1 = \frac{30 - \sqrt{896}}{2} = 0.035 \qquad x_2 = 15 + 14.97 = 29.97 \tag{19.1.21}$$

Solving using the product of roots formula,

$$x_1 = 29.97 \qquad x_2 = \frac{c}{ax_1} = 0.03337 \tag{19.1.22}$$

(b) Solving using the quadratic formula, with $2S$

$$0 = x^2 - 30x + 1 \qquad x = 15 \pm 4\sqrt{14} \tag{19.1.23}$$

$$x_1 = \frac{30 - \sqrt{896}}{2} = 0 \qquad x_2 = 15 + 15 = 30 \tag{19.1.24}$$

Solving using the product of roots formula,

$$x_1 = 30 \qquad x_2 = \frac{c}{ax_1} = 0.033 \tag{19.1.25}$$

The smaller the number of significant digits, the more problematic subtraction becomes.

10. Looking at the quadratic equation,

$$(x - k)^2 = a^2 \qquad x_1, x_2 = k \pm a \tag{19.1.26}$$

As $a^2 \to 0$, $a \gg a^2$ and a slight change in a^2 produces a large change in the roots. This qualifies as instability.

11. Let the two individual values and errors be,

$$x = \widetilde{x} + \epsilon_x \qquad y = \widetilde{y} + \epsilon_y \tag{19.1.27}$$

$$|\epsilon_x| \leq \beta_x \qquad |\epsilon_y| \leq \beta_y \tag{19.1.28}$$

$$|\epsilon| = |x + y - \widetilde{x} - \widetilde{y}| \qquad = |\epsilon_x + \epsilon_y| \leq |\epsilon_x| + |\epsilon_y| \tag{19.1.29}$$

$$|\epsilon| \leq \beta_x + \beta_y \tag{19.1.30}$$

12. Let x^2 be so large that it causes overflow, and $y^2 \leq x^2$

$$f(x) = x^2 + y^2 \qquad \lambda = \left(\frac{y}{x}\right)^2 \tag{19.1.31}$$

$$\lambda \in [0, 1] \qquad f(x) = x\sqrt{1 + \lambda} \tag{19.1.32}$$

Now, the computation never involves numbers too large to cause overflow.

Other examples might use easily factorized polynomial expressions.

13. For division, with the relative erorrs in x, y being,

$$|\epsilon_{rx}| = \left|1 - \frac{\widetilde{x}}{x}\right| \qquad |\epsilon_{ry}| = \left|1 - \frac{\widetilde{y}}{y}\right| \tag{19.1.33}$$

$$|\epsilon_r| = \left|1 - \frac{\widetilde{x}}{\widetilde{y}} \cdot \frac{y}{x}\right| \qquad |\epsilon_r| = \left|1 - \frac{(x - \epsilon_x)}{(y - \epsilon_y)} \cdot \frac{y}{x}\right| \tag{19.1.34}$$

$$= \left|\frac{-x\epsilon_y + y\epsilon_x}{x(y - \epsilon_y)}\right| \qquad \approxeq \left|\frac{-x\epsilon_y + y\epsilon_x}{xy}\right| \tag{19.1.35}$$

$$= |-\epsilon_{ry} + \epsilon_{rx}| \qquad |\epsilon_r| \leq |\epsilon_{ry}| + |\epsilon_{rx}| \tag{19.1.36}$$

$$|\epsilon_r| \leq \beta_{ry} + \beta_{rx} \tag{19.1.37}$$

14. By direct computation with $6S$,

$$f(x) = \sqrt{x^2 + 4} - 2 \qquad x = 0.001 \tag{19.1.38}$$

$$f(x) = \sqrt{4 + 0.000001} - 2 = 0 \tag{19.1.39}$$

By using the product of roots shortcut,

$$0 = z^2 - 2(\sqrt{x^2 + 4})\ z + x^2 \qquad z = \sqrt{x^2 + 4} \pm 2 \tag{19.1.40}$$

$$z_2 = \frac{x^2}{z_1} \qquad z_2 = \frac{x^2}{\sqrt{x^2 + 4} + 2} \tag{19.1.41}$$

$$z_2 = \frac{0.001^2}{4} = 2.5 \times 10^{-7} \tag{19.1.42}$$

15. Computing using $6S$,

$$\ln(a) - \ln(b) = \ln(4) - \ln(3.99900) \qquad = 1.38629 - 1.38604 \tag{19.1.43}$$

$$= 0.25 \times 10^{-3} \tag{19.1.44}$$

$$\ln(a/b) = \ln(1.00025) \qquad = 0.249\,969 \times 10^{-3} \tag{19.1.45}$$

The cruder method happens to be closer to the exact value.

16. By direct computation, with $6S$,

$$1 - \cos(0.02) = 1 - 0.999800 = 2 \times 10^{-4} \tag{19.1.46}$$

By using the trigonometric identity,

$$\cos(2\theta) = \cos^2\theta - \sin^2\theta = 1 - 2\sin^2\theta \qquad 1 - \cos x = 2\sin^2(x/2) \tag{19.1.47}$$

$$2\sin^2(0.01) = 1.999\,93 \times 10^{-4} \tag{19.1.48}$$

The newer method is a closer approximation to the real value.

17. The problem $\cos(u) - \cos(v)$ when $u \approxeq v$. The straight method is problematic because it involves subtraction of close numbers.

$$\cos a - \cos b = 2\sin\left(\frac{a+b}{2}\right)\ \sin\left(\frac{a-b}{2}\right) \tag{19.1.49}$$

This is a nice computation.

18. By direct computation, using $6S$

$$\frac{1-\cos x}{\sin x} = \frac{1.3 \times 10^{-5}}{4.999\,98 \times 10^{-3}} \qquad = 2.600\,010 \times 10^{-3} \tag{19.1.50}$$

Using the formula in Problem 16,

$$\frac{1-\cos x}{\sin x} = \frac{2\sin^2(x/2)}{2\sin(x/2)\cos(x/2)} \qquad f(x) = \tan(x/2) \tag{19.1.51}$$

$$= 2.500\,00 \times 10^{-3} \tag{19.1.52}$$

The better method matches the first 6 digits exactly.

19. Using the Maclaurin series, with $6S$,

$$e^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \frac{x^5}{5!} \tag{19.1.53}$$

$$e^{-1} = 1 - 1 + \frac{1}{2} - \frac{1}{6} + \frac{1}{24} - \frac{1}{120} \tag{19.1.54}$$

$$= 0.5 - 0.166667 + 0.0416667 - 0.00833333 = 0.366666 \tag{19.1.55}$$

Using the alternative approach that avoids any subtractions.

$$e^1 = 1 + 1 + \frac{1}{2} + \frac{1}{6} + \frac{1}{24} + \frac{1}{120} \tag{19.1.56}$$

$$= 2.5 + 0.166667 + 0.0416667 + 0.00833333 = 2.71667 \tag{19.1.57}$$

$$e^{-1} = \frac{1}{2.71667} = 0.368098 \tag{19.1.58}$$

The second method is much closer to the accurate value

$$\epsilon_2 = -2.19 \times 10^{-4} \qquad\qquad \epsilon_1 = 1.213 \times 10^{-3} \tag{19.1.59}$$

20. Using the Maclaurin series, with $6S$,

$$e^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \frac{x^5}{5!} \tag{19.1.60}$$

$$e^{-10} = 1 - 10 + \frac{10^2}{2} - \frac{10^3}{6} + \frac{10^4}{24} - \frac{10^5}{120} \tag{19.1.61}$$

$$= -9 + 50 - 166.667 + 416.667 - 833.333 = -542.333 \tag{19.1.62}$$

Using the alternative approach that avoids any subtractions.

$$e^{10} = 1 + 10 + \frac{10^2}{2} + \frac{10^3}{6} + \frac{10^4}{24} + \frac{10^5}{120} \tag{19.1.63}$$

$$= 11 + 50 + 166.667 + 416.667 + 833.333 = 147.767 \tag{19.1.64}$$

$$e^{-10} = \frac{1}{147.767} = 6.767\,41 \times 10^{-3} \tag{19.1.65}$$

Since the Maclaurin series diverges for large x, the first method is bad. The second method converges to the true value.

21. Using the division algorithm to convert decimal to binary,

Dividend	Divisor	Quotient	Remainder
23	2	11	1
11	2	5	1
5	2	2	1
2	2	1	0
1	2	0	1

The algorithm ends when the quotient is 0.

22. Using the multiplication algorithm to convert decimal to binary,

Current value	Base	Product	Integer part
0.59375	2	1.1875	1
0.1875	2	0.375	0
0.375	2	0.75	0
0.75	2	1.5	1
0.5	2	1	1

The algorithm ends when the product itself is 1.

23. To check if $x = 0.1$ can be represented as a machine number,

Current value	Base	Product	Integer part
0.1	2	0.2	0
0.2	2	0.4	0
0.4	2	0.8	0
0.8	2	1.6	1
0.6	2	1.2	1
0.2	2	0.4	0
0.4	2	0.8	0

Notice that this process never ends becuase it gets stuck in an infinite loop of $0.2 \to 0.4 \to 0.8 \to 0.6 \to 0.2 \to \ldots$.

Thus, the binary representation is non-terminating.

24. All machine numbers are sums of integer powers of 2 as well as sums of fractions of the form $1/2^p$.

$$\frac{1}{2} = 0.5 = \frac{5}{10} \qquad \frac{1}{2^p} = (0.5)^p = \frac{5^p}{10^p} \tag{19.1.66}$$

Since the denominator in this rational number is always a power of 10 and the numerator is an integer, the result has terminating decimals in the base 10 system.

The converse is not true, as seen by the counterexample in Problem 23.

25. Looking at the infinite series,

$$S = \frac{3}{2} \sum_{m=1}^{\infty} \left(\frac{1}{16}\right)^m \qquad S = \frac{3}{2} \cdot \frac{1/16}{15/16} = \frac{1}{10} \tag{19.1.67}$$

Parital sums of this series using 26 terms gives

$$0.1 - 1.5 \sum_{m=1}^{\infty} \frac{1}{16^m} = 7.8883 \times 10^{-32} \tag{19.1.68}$$

This is the least number of terms for the error to be smaller than 1×10^{-30}.

26. Integrating by parts,

$$I_n = \int_0^1 e^x \ x^n \ \mathrm{d}x \qquad I_n = \Big[x^n e^x\Big]_0^1 - n \int_0^1 e^x x^{n-1} \ \mathrm{d}x \tag{19.1.69}$$

$$I_n = e - nI_{n-1} \qquad I_0 = \int_0^1 e^x \ \mathrm{d}x = e - 1 \tag{19.1.70}$$

Iteration	Value	Iteration	Value
1	1.000	6	0.3365
2	0.7183	7	0.3631
3	0.5635	8	−0.1865
4	0.4643	9	4.397
5	0.3970	10	−41.25

(a) Since the integrand is always positive in the interval [0, 1], the end result cannot be negative. The large error is a result of subtraction of close numbers, in addition to the error propagating in the recursive formula.

(b) Using `sympy` to investigate the relationship between N and k,

$$N = p_0 + p_1 k \qquad\qquad p_0 = 0.7128, \quad p_1 = 7.740 \tag{19.1.71}$$

27. Using backward recursion, and setting $I_{15} \approxeq 0$

$$\int_0^1 e^x\ x^n\ \ dx \leq \int_0^1 e\ x^n\ \ dx \qquad\qquad |I_n| \leq \frac{e}{n+1} \tag{19.1.72}$$

$$\lim_{n\to\infty} |I_n| = 0 \tag{19.1.73}$$

n	I_n	n	I_n
15	0.000	7	0.3055
14	0.1812	6	0.3447
13	0.1812	5	0.3956
12	0.1952	4	0.4645
11	0.2103	3	0.5634
10	0.2280	2	0.7183
9	0.2490	1	1.000
8	0.2744	0	1.719

28. Looking at the harmonic series of computer numbers, let

$$\frac{1}{n} < \epsilon \qquad\qquad \forall \quad n > M \tag{19.1.74}$$

Here, ϵ is the smallest possible computer number. Underflow causes all further terms of this infinite series to be set to zero, making it a finite series.

Thus, the harmonic series of computer numbers truncates at a certain number of terms and converges to S_M.

29. For the approximation $a = 22/7$, $b = 355/113$, using $3S$

$$\epsilon_1 = \pi - a = -1.26 \times 10^{-3} \qquad \epsilon_2 = \pi - b = -2.67 \times 10^{-7} \tag{19.1.75}$$

$$\epsilon_{1r} = \left|1 - \frac{a}{\pi}\right| = 4.02 \times 10^{-4} \qquad \epsilon_{2r} = \left|1 - \frac{b}{\pi}\right| = 8.49 \times 10^{-8} \tag{19.1.76}$$

30. Computing the approximation to π using $10S$,

$$a = 16\arctan(1/5) - 4\arctan(1/239) \qquad a = 3.141592653 \tag{19.1.77}$$

This is correct to 10 significant digits.

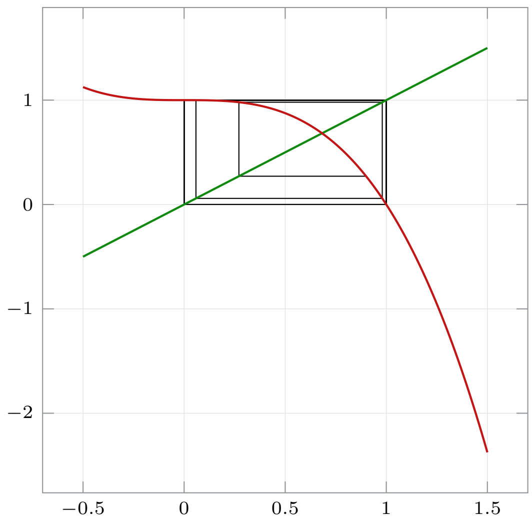
19.2 Solution of Equations by Iteration

1. In example 1, the slope of $g(x)$ is positive near the intersection of $y = g(x)$ and $y = x$, which makes the iterative process monotonic.

2. Solving the iterative relation,

$$x = 1 - x^3 \tag{19.2.1}$$

$x_0 = 0.9$

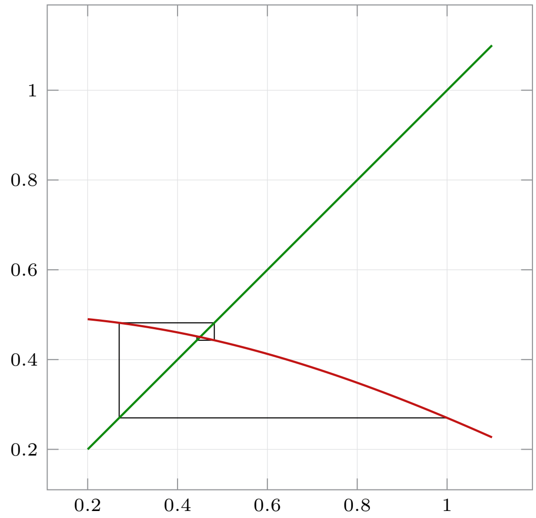


This leads to a limit cycle since $g'(0) = 0$. Initial guesses with large enough absolute value diverge to infinity.

3. Using the fixed point method,

$$g(x) = x \qquad g(x) = \frac{\cos x}{2} \tag{19.2.2}$$

$x_0 = 1$

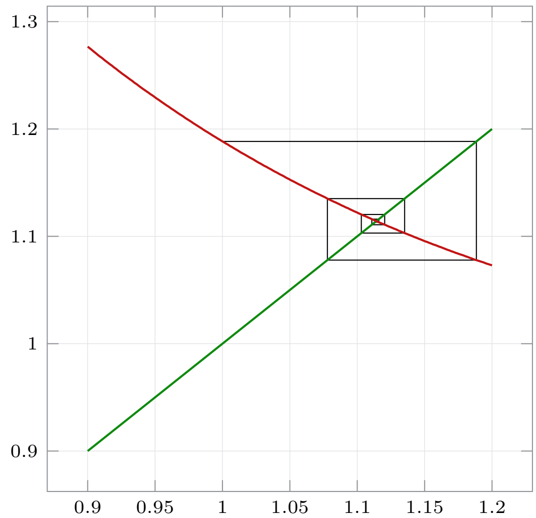


The result is $x_{10} = 0.450184$ exact to $6S$.

4. Using the fixed point method,

$$g(x) = x \qquad\qquad g(x) = \csc x \tag{19.2.3}$$

$x_0 = 1$

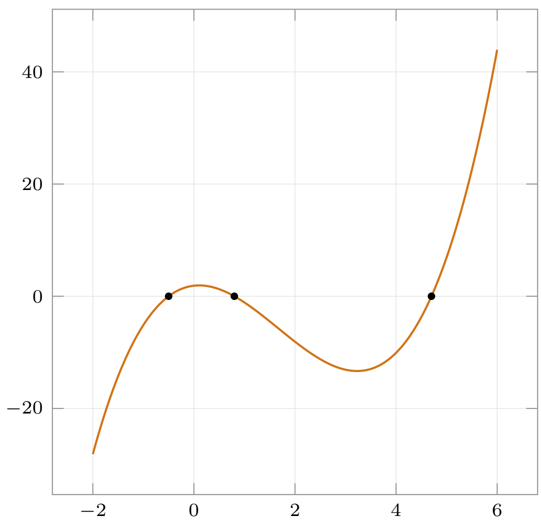


The result is $x_{10} = 1.11385$ exact to $6S$.

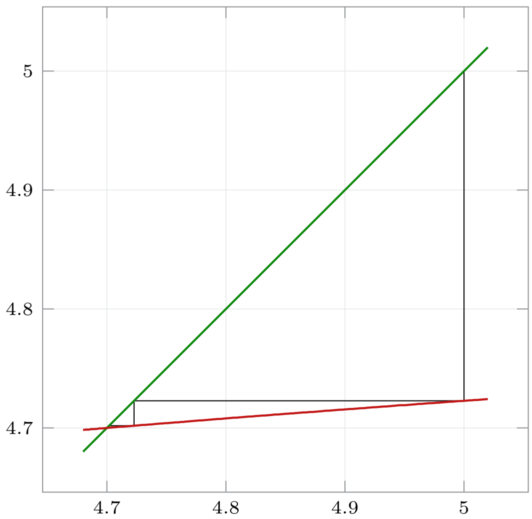
5. Showing the function with its roots and then the iterative process,

$$f(x) = x^3 - 5x^2 + 1.01x + 1.88 \qquad\qquad g(x) = x = \frac{5x^2 - 1.01x - 1.88}{x^2} \tag{19.2.4}$$

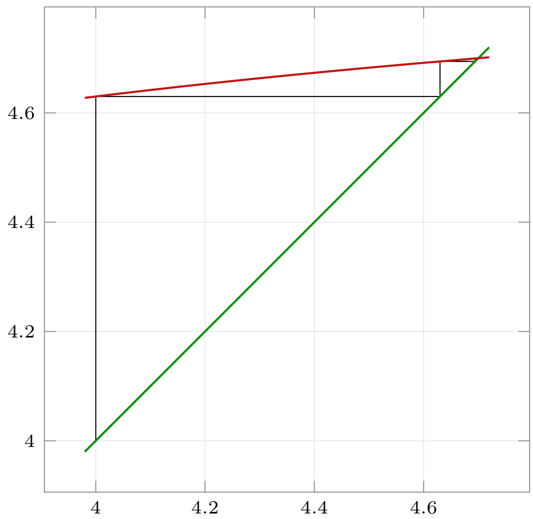
$f(x) = 0$



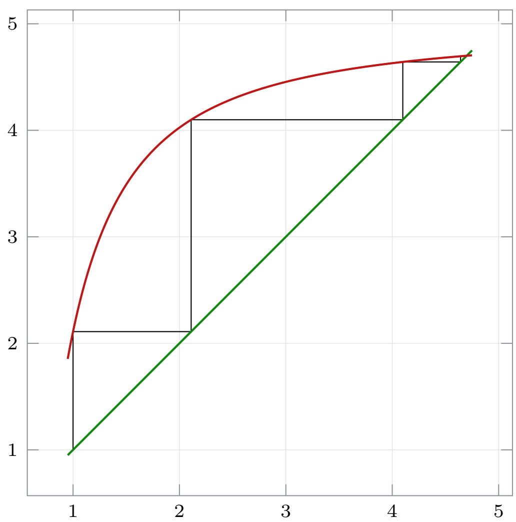
$x_0 = 5$



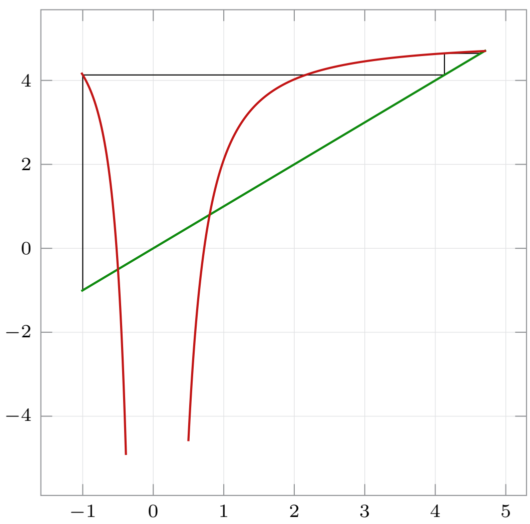
$x_0 = 4$



$x_0 = 1$



$x_0 = -1$

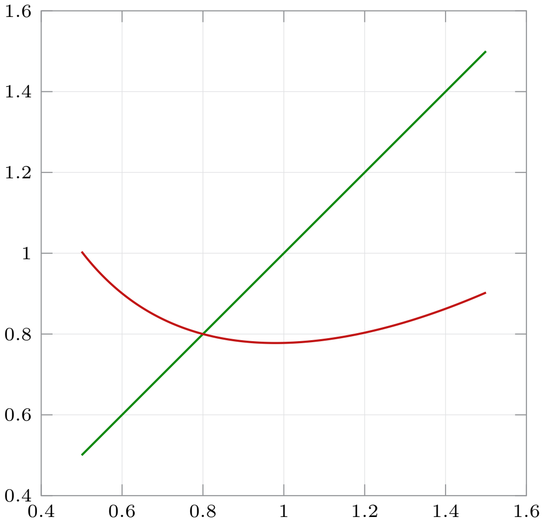


All of the guesses converge to the same root $x_3^* = 4.7$. This is because of the nature of $g'(x)$

6. Using the x^2 term to write the recursive relation,

$$g(x) = \frac{x^3 + 1.01x + 1.88}{5x} \qquad\qquad g'(x) = \frac{50x^3 - 47}{125x^2} \tag{19.2.5}$$

Convergence to $x_2^* = 1$

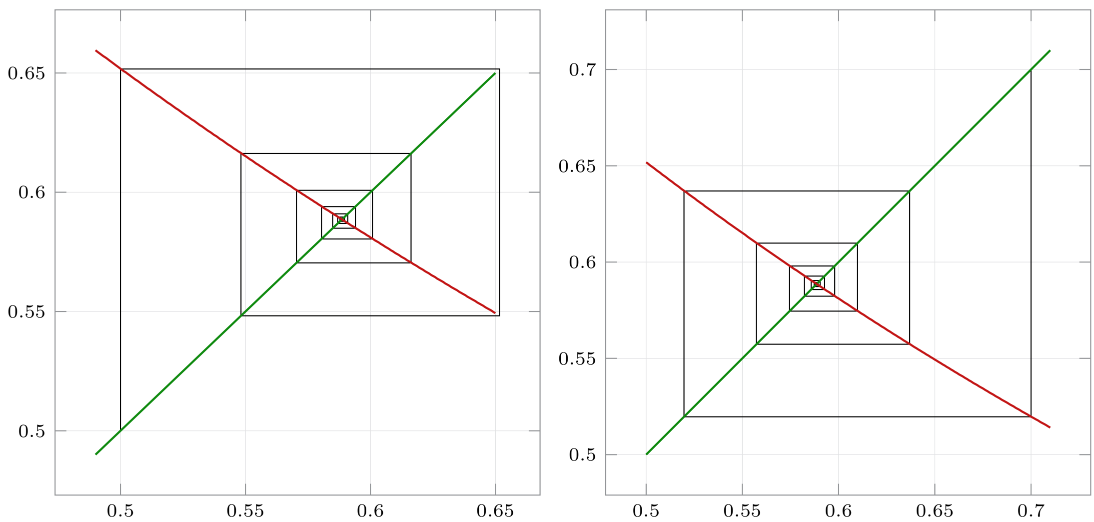


Since $|g'(x)| \leq K < 1$ for x near this solution, the convergence criterion is satisfied.

7. Using the fixed point method,

$$g(x) = \arcsin(e^{-x}) \qquad\qquad g'(x) = \frac{-e^{-x}}{\sqrt{1 - e^{-2x}}} \tag{19.2.6}$$

$x_0 = 0.5$ $\qquad$ $x_0 = 0.7$

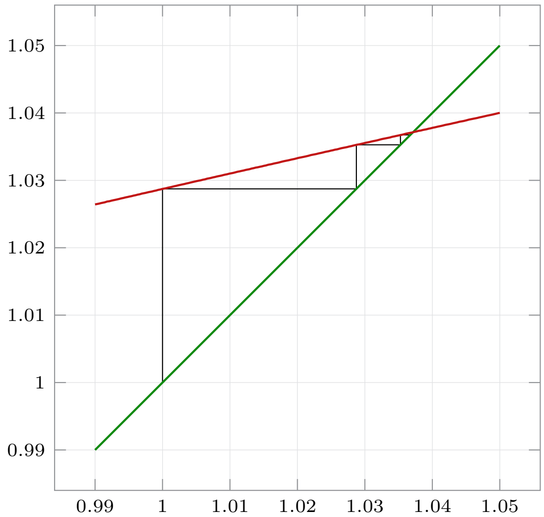


The result is $x_{30} = 0.588533$ exact to $6S$.

8. Using the fixed point method,

$$g(x) = \frac{x + 0.12}{x^3} \qquad\qquad g'(x) = -\frac{50x + 3}{50x^2} \tag{19.2.7}$$

$x_0 = 1$

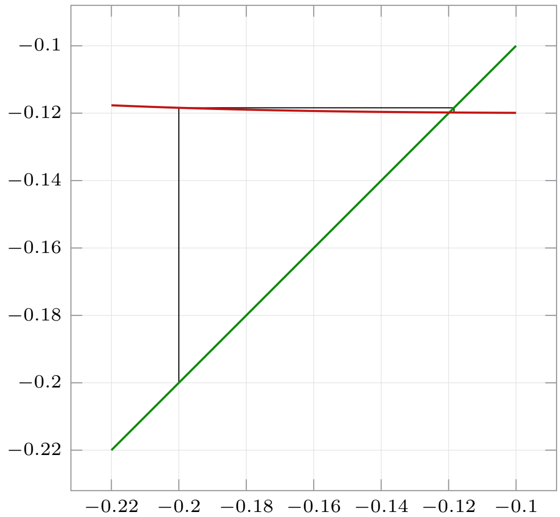


The result is $x_{30} = 1.03717$ exact to $6S$.

9. Using the fixed point method,

$$g(x) = x^4 - 0.12 \qquad\qquad g'(x) = 4x^3 \tag{19.2.8}$$

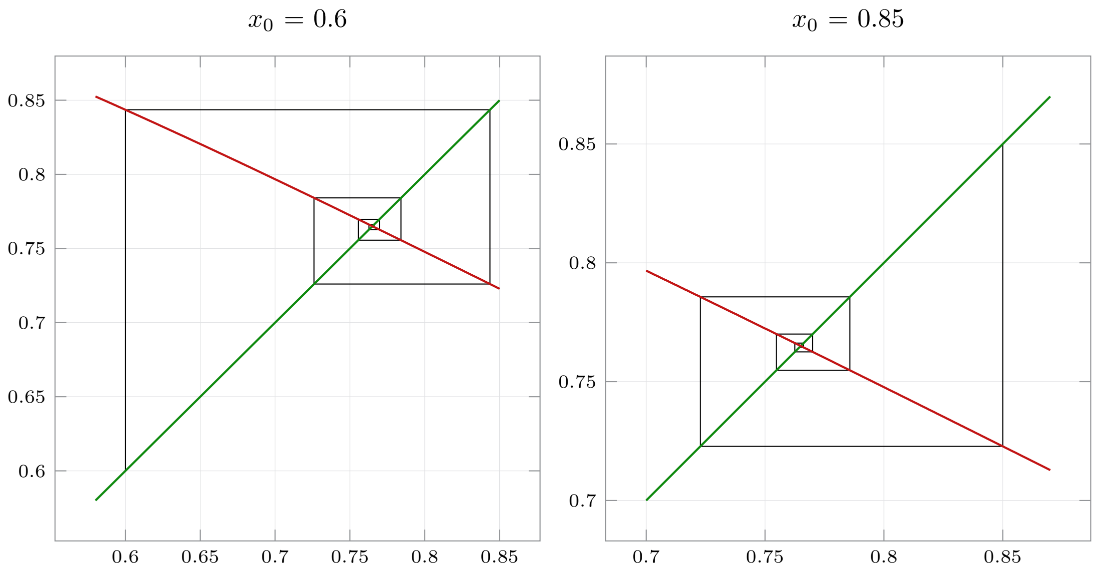
$x_0 = -0.2$



The result is $x_{20} = -0.119794$ exact to $6S$.

10. Using the fixed point method,

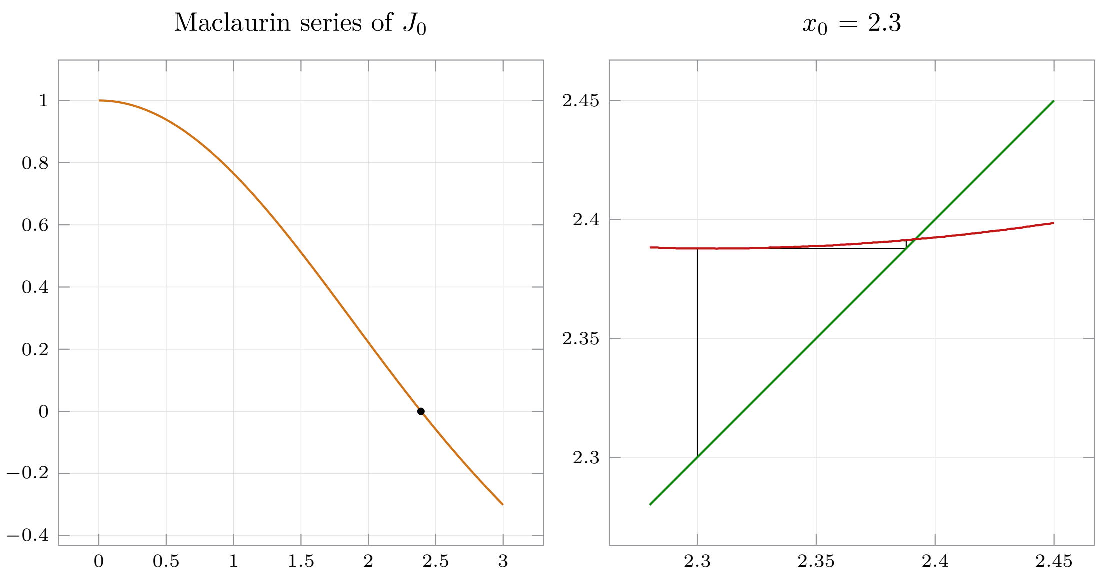
$$g(x) = \frac{1}{\cosh x} \qquad\qquad g'(x) = \arctan(\sinh x)$$ 19.2.9



The result is $x_{10} = 0.765010$ exact to $6S$.

11. Using the fixed point method,

$$g(x) = \frac{1 + x^4/64 - x^6/2304}{x/4}$$ 19.2.10



The result is $x_{10} = 2.39165$ exact to $6S$.

12. Using the fixed point method,

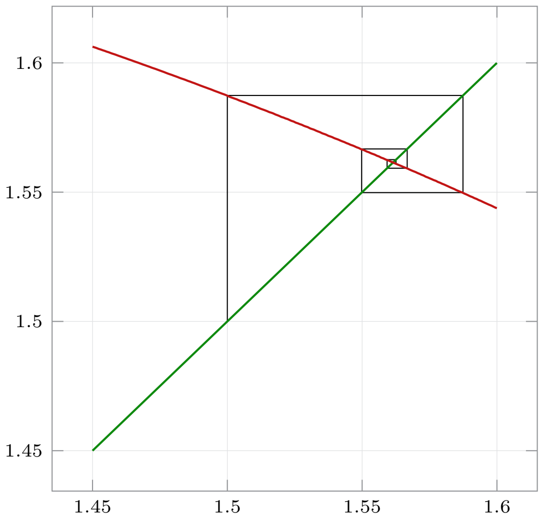
$$f(x) = x^3 + 2x^2 - 3x - 4 \qquad g_1(x) = (-3x^2 + 3x + 4)^{1/3} \tag{19.2.11}$$

$$g_2(x) = (-0.5x^3 + 1.5x + 2)^{1/2} \qquad g_3(x) = \frac{x^3 + 2x^2 - 4}{3} \tag{19.2.12}$$

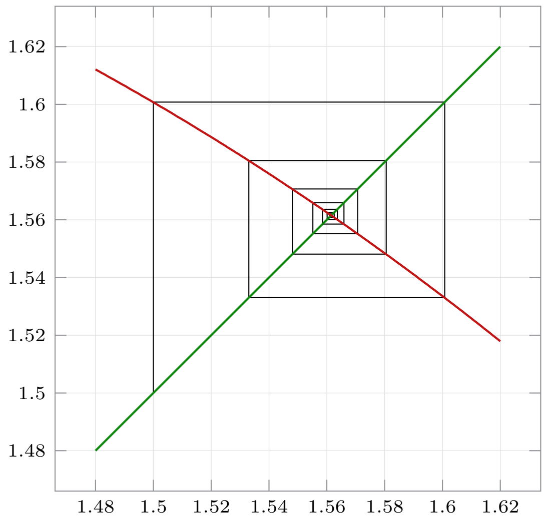
$$g_4(x) = \frac{-2x^2 + 3x + 4}{x^2} \qquad g_5(x) = \frac{-x^3 + 3x + 4}{2x} \tag{19.2.13}$$

$$g_6(x) = \frac{2x^3 + 2x^2 + 4}{3x^2 + 4x - 3} \tag{19.2.14}$$

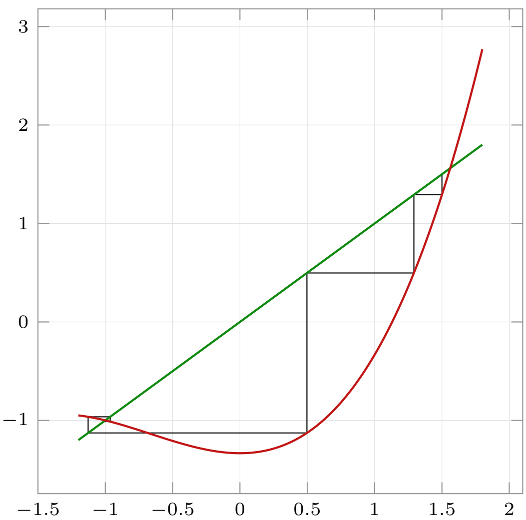
$x_1^* = 1.56155, \qquad n = 15$



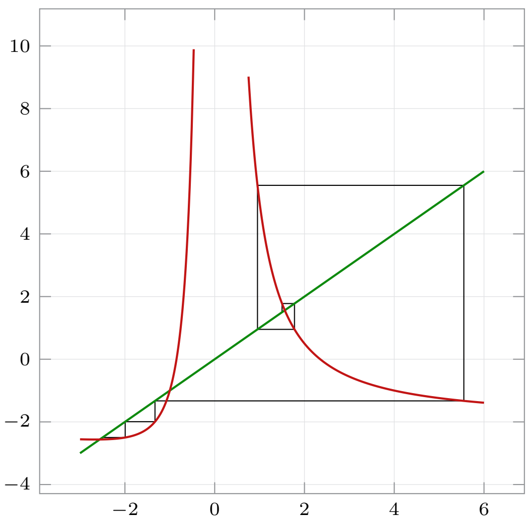
$x_2^* = 1.56155, \qquad n = 35$

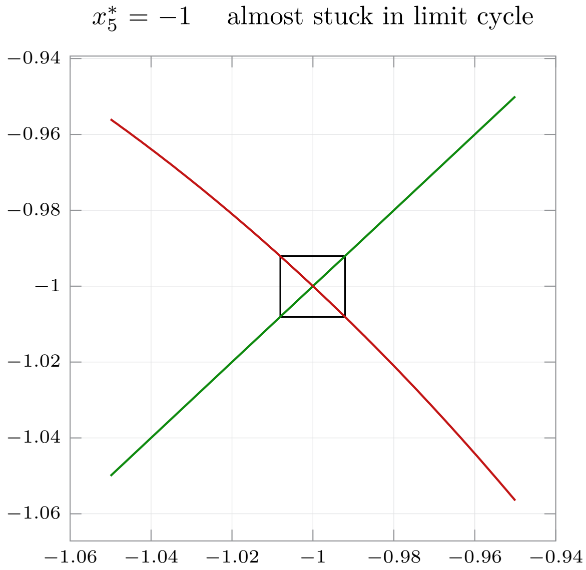


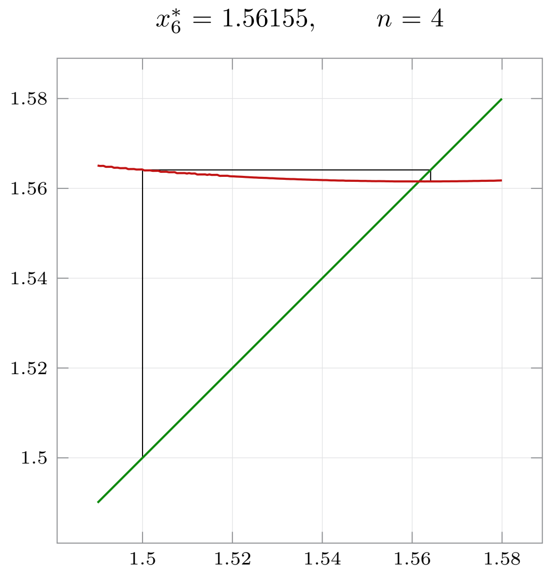
$x_3^* = -1, \qquad n = 16$



$x_4^* = -2.56155, \qquad n = 12$







13. Range of a continuous function g lies within its domain. This means that the range R is a proper subset of the domain D.

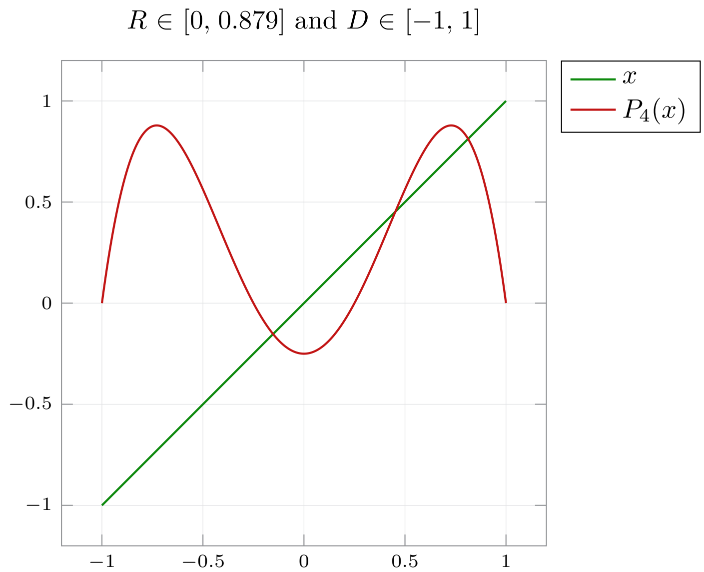
$$f(x) = g(x) - x \tag{19.2.15}$$

$$g(a) < a \qquad\qquad g(b) > b \tag{19.2.16}$$

for some a, b being the ends of the domain D

$$f(a) < 0 \qquad\qquad f(b) > 0 \tag{19.2.17}$$

By the intermediate value theorem of calculus, $f(x)$ being continuous means there exists at least one value $c \in [a, b]$ for which $f(x) = 0$.

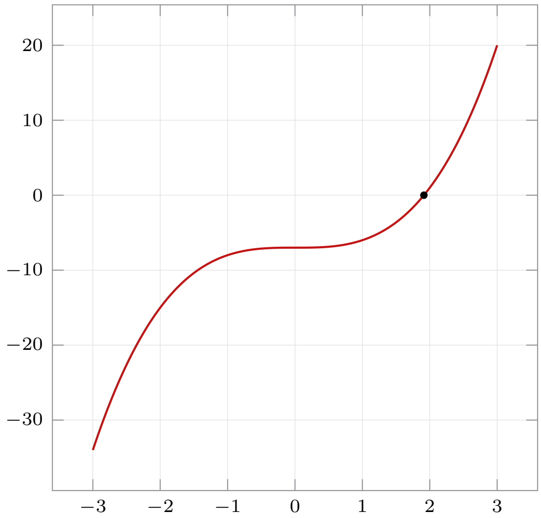


This is an example of a polynomial with roots ± 1, ± 0.5, which intersects the line $y = x$ multiple times within its domain.

14. Using the Newton Raphson method,

$$f(x) = x^3 - 7 \qquad x_{n+1} = x_n - \frac{x_n^3 - 7}{3x_n^2} \tag{19.2.18}$$

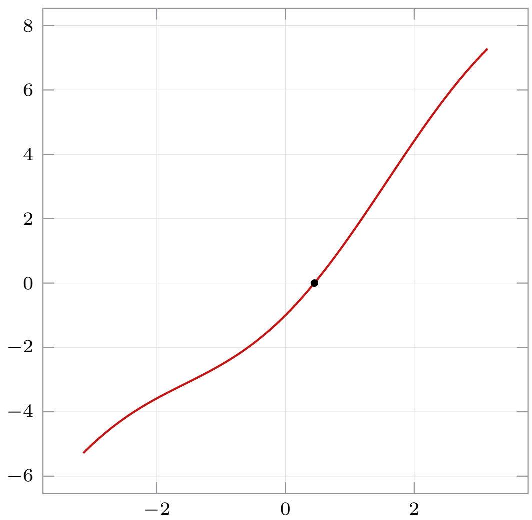
$$x_0 = 2 \qquad x^* = 1.91293, \qquad (n = 3) \tag{19.2.19}$$



15. Using the Newton Raphson method,

$$f(x) = 2x - \cos x \qquad x_{n+1} = x_n - \frac{2x_n - \cos(x_n)}{2 + \sin(x_n)} \tag{19.2.20}$$

$$x_0 = 1 \qquad x^* = 0.450184, \qquad (n = 3) \tag{19.2.21}$$

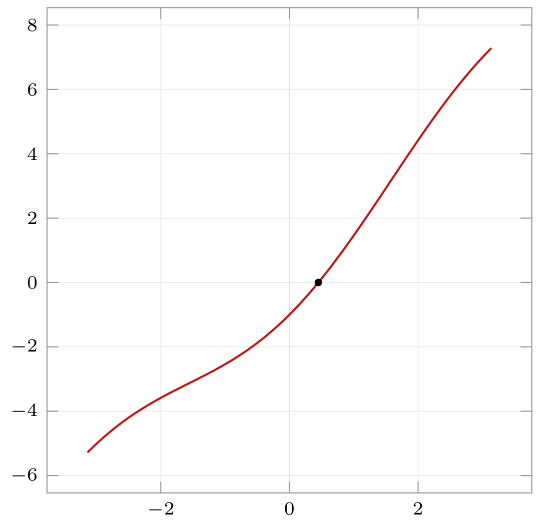


Convergence is much faster compared to the fixed point iteration method in problem 3.

16. Using the Newton Raphson method,

$$f(x) = 2x - \cos x \qquad x_{n+1} = x_n - \frac{2x_n - \cos(x_n)}{2 + \sin(x_n)} \tag{19.2.22}$$

$$x_0 = 1 \qquad x^* = 0.450184, \qquad (n = 3) \tag{19.2.23}$$

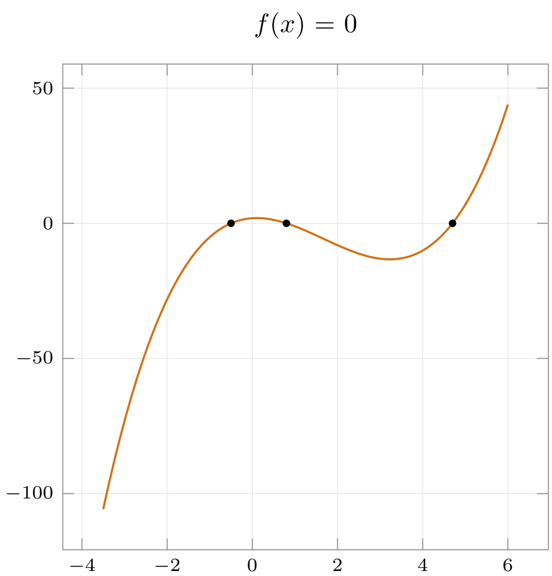


Other choices of initial guess also converge to the same root. This convergence is slightly slower for far away x_0.

17. Solving problem 5, using Newton Raphson method,

$$f(x) = x^3 - 5x^2 + 1.01x + 1.88 \tag{19.2.24}$$

$$x_{n+1} = x_n - \frac{x_n^3 - 5x_n^2 + 1.01x_n + 1.88}{3x_n^2 - 10x_n + 1.01} \tag{19.2.25}$$



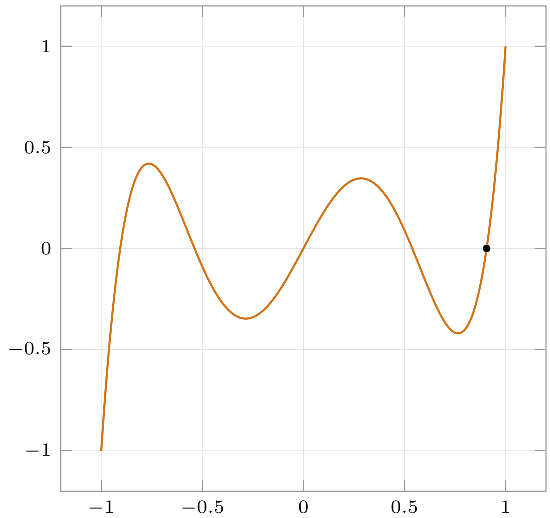
The starting points $x_0 = 5, 4, 1, -3$ converge to $4.7, 4.7, 0.8, -0.5$ respectively.

18. Using Newton Raphson method,

$$x_{n+1} = x_n - \frac{63x_n^5 - 70x_n^3 + 15x_n}{315x_n^4 - 210x_n^2 + 15} \qquad \text{19.2.26}$$

$$x_0 = 1 \qquad x^* = 0.906180, \qquad (n = 4) \qquad \text{19.2.27}$$

$P_5(x) = 0$



Factorizing the polynomial, to yield a bi-quadratic equation

$$P_5(x) = \frac{x}{8} \cdot (63x^4 - 70x^2 + 15) \qquad x^2 = 0, \frac{35 \pm 2\sqrt{70}}{63} \qquad \text{19.2.28}$$

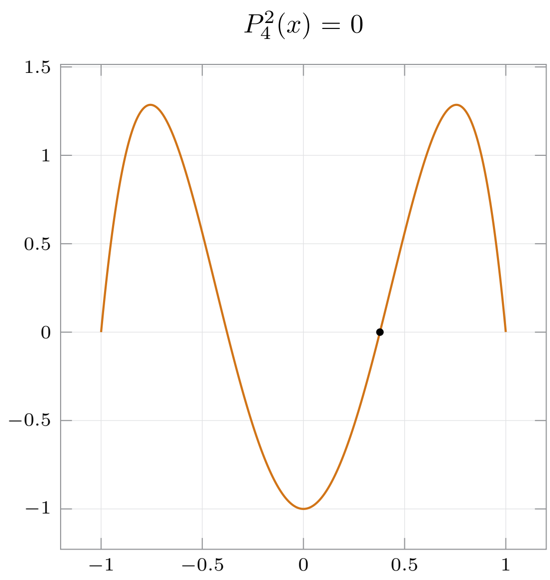
$$x = \sqrt{\frac{35 + 2\sqrt{70}}{63}} = \sqrt{0.821162} \qquad x^* = 0.906180 \qquad \text{19.2.29}$$

This matches the N.R. method to $6S$.

19. Using Newton Raphson method,

$$x_{n+1} = x_n - \frac{-7x^4 + 8x^2 - 1}{-28x^3 + 16x} \qquad \text{19.2.30}$$

$$x_0 = 0.3 \qquad x^* = 0.377964, \qquad (n = 3) \qquad \text{19.2.31}$$



Factorizing the polynomial, to yield a bi-quadratic equation

$$P_4^2(x) = \frac{15}{2} \cdot (-7x^4 + 8x^2 - 1) \qquad x^2 = \frac{1}{7}, 1 \tag{19.2.32}$$

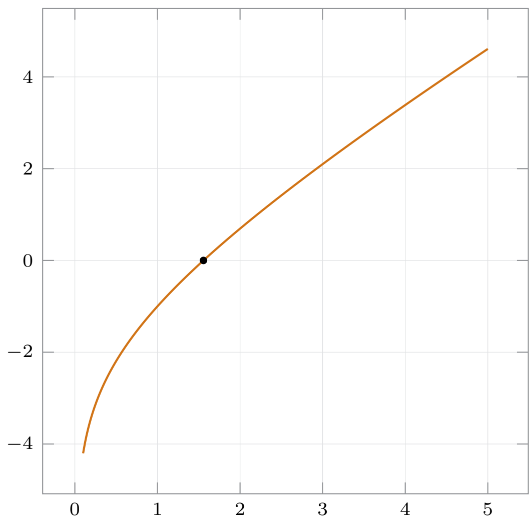
$$x = \sqrt{1/7} = \sqrt{0.142857} \qquad x^* = 0.377964 \tag{19.2.33}$$

This matches the N.R. method to $6S$.

20. Using Newton Raphson method,

$$x_{n+1} = x_n - \frac{x + \ln x - 2}{1 + 1/x} \tag{19.2.34}$$

$$x_0 = 2 \qquad x^* = 1.55715, \qquad (n = 3) \tag{19.2.35}$$



21. Using Newton Raphson method,

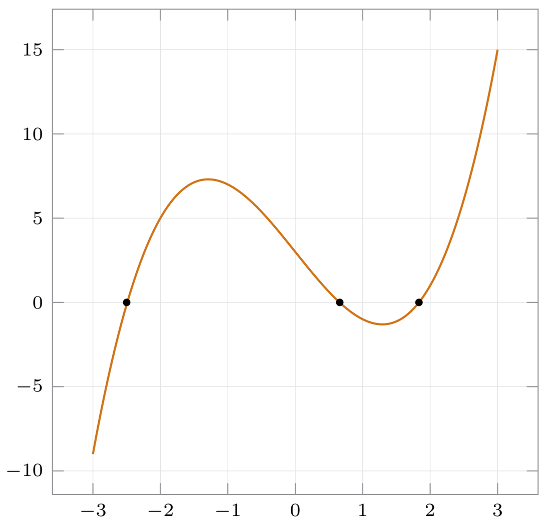
$$x_{n+1} = x_n - \frac{x^3 - 5x + 3}{3x^2 - 5} \tag{19.2.36}$$

$$x_0 = 2 \qquad x^* = 1.83424, \qquad (n = 3) \tag{19.2.37}$$

$$x_0 = 0 \qquad x^* = 0.656619, \qquad (n = 3) \tag{19.2.38}$$

$$x_0 = -2 \qquad x^* = -2.49086, \qquad (n = 4) \tag{19.2.39}$$

19.2.40



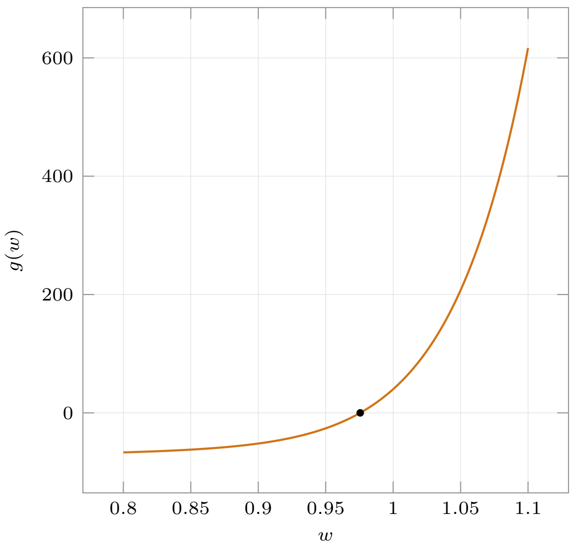
22. Using the transformation $e^{-0.01x} = w$

$$f_1 - 100(1 - w^{20}) \qquad f_2 = 40w \tag{19.2.41}$$

$$w_{n+1} = w_n - \frac{-100 + 100w^{20} + 40w}{2000w^{19} + 40} \tag{19.2.42}$$

$$w_0 = 2 \qquad w^* = 0.9756 \tag{19.2.43}$$

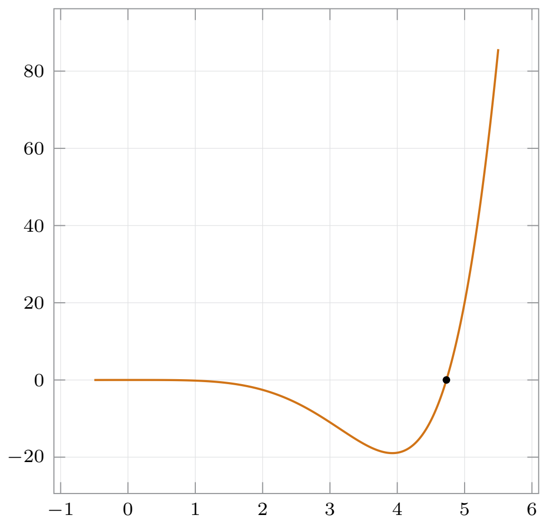
$$x^* = -100\ \ln(w^*) = 2.47 \qquad T^* = 39.02 \tag{19.2.44}$$



23. Using Newton Raphson method,

$$x_{n+1} = x_n - \frac{\cos x \cdot \cosh x - 1}{-\sin x \cosh x + \cos x \sinh x} \tag{19.2.45}$$

$$x_0 = 4.5 \qquad x^* = 4.73004, \quad (n = 4) \tag{19.2.46}$$



(a) Finding the point of intersection with the x axis,

$$\frac{y - f(b)}{x - b} = \frac{f(b) - f(a)}{b - a} \tag{19.2.47}$$

$$y = 0 \implies c - b = \frac{-f(b) \cdot (b - a)}{f(b) - f(a)} \tag{19.2.48}$$

$$c = \frac{a\ f(b) - b\ f(a)}{f(b) - f(a)} \tag{19.2.49}$$

(b) Solving using this iterative process, using $a_0 = 1$, $b_0 = 2$

$$f_1(x) = x^4 - 2 \qquad x^* = 1.18921, \quad (n = 25) \tag{19.2.50}$$

$$f_2(x) = \cos x - \sqrt{x} \qquad x^* = 0.641714, \quad (n = 8) \tag{19.2.51}$$

$$f_3(x) = x + \ln x - 2 \qquad x^* = 1.57715, \quad (n = 6) \tag{19.2.52}$$

24. Bisection method,

(a) Algorithm in `sympy`

(b) Solving using this iterative process, using

$$f_1(x) = x - \cos x \qquad a = 0, \quad b = 2 \tag{19.2.53}$$

$$x^* = 0.739085, \quad (n = 21) \tag{19.2.54}$$

The N.R. method converges to the same $6S$ result in 5 iterations starting with $x_0 = 0.5$, which is much faster.

(c) Using the bisection method

$$f_1(x) = e^{-x} - \ln x \qquad x^* = 1.309800, \quad (n = 21) \tag{19.2.55}$$

$$f_2(x) = e^x + x^4 + x - 2 \qquad x^* = 0.429494, \quad (n = 21) \tag{19.2.56}$$

25. Using the secant method,

$$f(x) = e^{-x} - \tan x \qquad x_0 = 1, \quad x_1 = 0.7 \tag{19.2.57}$$

$$x^* = 0.531391 \qquad n = 5 \tag{19.2.58}$$

26. Using the secant method,

$$f(x) = x^3 - 5x + 3 \qquad x_0 = 1, \quad x_1 = 2 \tag{19.2.59}$$

$$x^* = 1.83424 \qquad n = 5 \tag{19.2.60}$$

27. Using the secant method,

$$f(x) = x - \cos x \qquad x_0 = 0.5, \quad x_1 = 1 \tag{19.2.61}$$

$$x^* = 0.739085 \qquad n = 5 \tag{19.2.62}$$

28. Using the secant method,

$$f(x) = \sin x - \cot x \qquad x_0 = 1, \quad x_1 = 0.5 \tag{19.2.63}$$

$$x^* = 0.904557 \qquad n = 6 \tag{19.2.64}$$

29. Refer notes. TBC. Methods involving more complex computations per step and more stringent assumptions on the function converge to the same result faster than crude methods.

19.3 Interpolation

1. Finding $p_1(x)$, using the given nodes,

$$L_0 = \frac{x - 9.5}{-0.5} = -2x + 19 \qquad L_1 = \frac{x - 9.0}{0.5} = 2x - 18 \tag{19.3.1}$$

$$p_1(x) = 0.1082x + 1.2234 \qquad p_1(9.3) = 2.2297 \tag{19.3.2}$$

2. Estimating the error,

$$\epsilon_1(x) = (x - x_0)(x - x_1) \cdot \frac{f''(t)}{2!} \qquad \epsilon_1(9.3) = (0.3)(-0.2) \cdot \frac{-1}{2t^2} \tag{19.3.3}$$

$$\epsilon_1(9.3) = \frac{0.03}{t^2} \qquad \epsilon_1(9.3) = [3.3241, 3.7037] \cdot 10^{-4} \tag{19.3.4}$$

3. Calculating $p_2(x)$ using the three given nodes,

$$p_2(x) = x^2 - 2.58x + 2.58 \qquad p_2(1.01) = 0.9943 \tag{19.3.5}$$

$$p_2(1.03) = 0.9835 \tag{19.3.6}$$

4. Using equation 5 to calculate the error,

$$\epsilon_2(9.2) = \frac{0.036}{t^3} \qquad \epsilon_2(9.2) \in [2.7047, 4.9383] \cdot 10^{-5} \tag{19.3.7}$$

5. Using $x_0 = 0$, $x_1 = 0.5$,

$$f(x) = e^{-x} \qquad f(0.25) \approxeq 0.8032 \tag{19.3.8}$$

$$\epsilon_1(0.25) = -0.0244 \tag{19.3.9}$$

Using $x_0 = 0.5$, $x_1 = 1$,

$$f(x) = e^{-x} \qquad f(0.75) \approxeq 0.4872 \tag{19.3.10}$$

$$\epsilon_1(0.75) = -0.0148 \tag{19.3.11}$$

Using all three nodes to find the quadratic interpolant,

$$p_2(0.25) = 0.7839 \qquad \epsilon_2(0.25) = -0.0051 \tag{19.3.12}$$

$$p_2(0.75) = 0.4678 \qquad \epsilon_2(0.75) = 0.0046 \tag{19.3.13}$$

6. Using the results of problem 2, to find $p_2(x)$

$$p_2(x) = -0.005233x^2 + 0.205x + 0.7759 \tag{19.3.14}$$

$$p_2(9.4) = 2.241 \qquad \epsilon_2(9.4) = -0.0003 \tag{19.3.15}$$

$$p_2(10) = 2.303 \qquad \epsilon_2(10) = -0.0004 \tag{19.3.16}$$

$$p_2(10.5) = 2.352 \qquad \epsilon_2(10.5) = -0.0006 \tag{19.3.17}$$

$$p_2(11.5) = 2.441 \qquad \epsilon_2(11.5) = 0.0013 \tag{19.3.18}$$

$$p_2(12) = 2.483 \qquad \epsilon_2(12) = 0.0019 \tag{19.3.19}$$

The errors are small only within the domain spanned by the nodes.

7. Using the results of problem 2, to find $p_2(x)$

$$p_2(x) = 0.1013x\,(11.49 - 3.314x) \tag{19.3.20}$$

$$p_2(-\pi/8) = -0.5089 \qquad \epsilon_2(-\pi/8) = 0.1262 \tag{19.3.21}$$

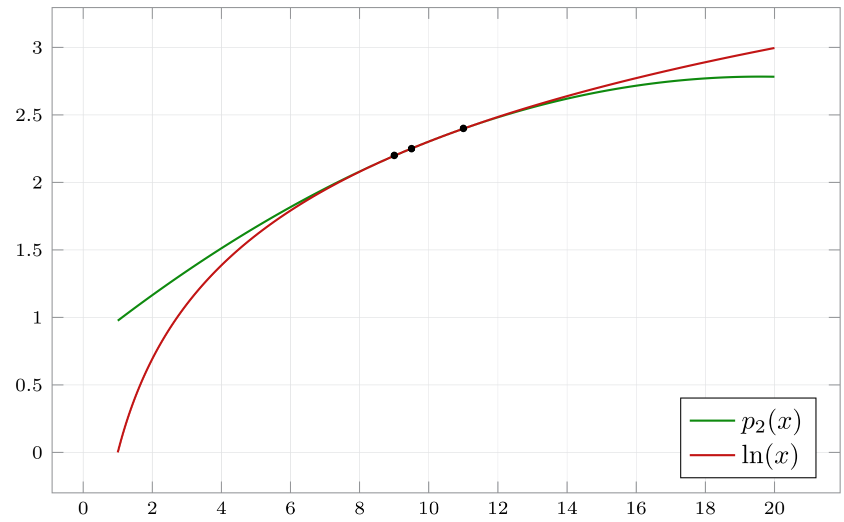
$$p_2(\pi/8) = 0.4053 \qquad \epsilon_2(\pi/8) = -0.0226 \tag{19.3.22}$$

$$p_2(3\pi/8) = 0.9053 \qquad \epsilon_2(3\pi/8) = 0.0186 \tag{19.3.23}$$

$$p_2(5\pi/8) = 0.9911 \qquad \epsilon_2(5\pi/8) = -0.0672 \tag{19.3.24}$$

The errors are small only within the domain spanned by the nodes.

8. Plotting the interpolant and the actual function, the errors are very small within the



9. For the error function,

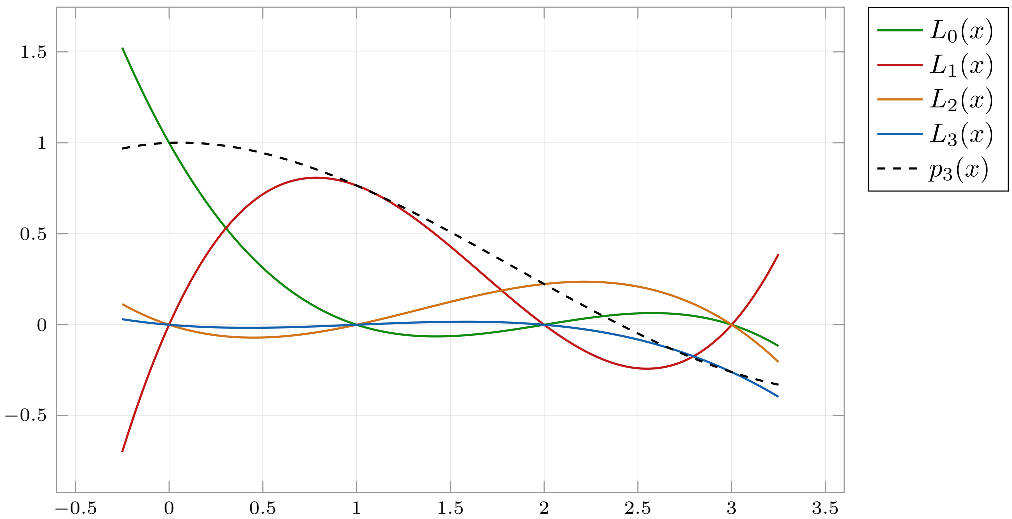
$$f(x) = \operatorname{erf}(x) \qquad p_2(x) = -0.44304x^2 + 1.30896x - 0.02322 \tag{19.3.25}$$

$$p_2(0.75) = 0.70929 \qquad \epsilon_2(0.75) = 0.00187 \tag{19.3.26}$$

10. Using the data in Problem 9,

$$\epsilon_2(x) = -0.020833(2t^2 - 1)\,\frac{e^{-t^2}}{\sqrt{\pi}} \qquad \epsilon_2(x) \in [0.0043240, 0.0096616] \tag{19.3.27}$$

11. Plotting the individual polynomials L_k,



Note that only one polynomial is nonzero at each node.

$$p_3(0.5) = 0.943654 \qquad \epsilon_3(0.5) = -0.005184 \tag{19.3.28}$$

$$p_3(1.5) = 0.510116 \qquad \epsilon_3(1.5) = 0.001712 \tag{19.3.29}$$

$$p_3(2.5) = -0.0447993 \qquad \epsilon_3(2.5) = -0.0035847 \tag{19.3.30}$$

12. Coded in `sympy`. The table of forward differences is not shown for clarity.

$$p_1(1.25) = 1.135380 \qquad \epsilon_1(1.25) = 1.107 \times 10^{-2} \tag{19.3.31}$$

$$p_2(1.25) = 1.147614 \qquad \epsilon_2(1.25) = -1.164 \times 10^{-3} \tag{19.3.32}$$

$$p_3(1.25) = 1.147004 \qquad \epsilon_3(1.25) = -5.54 \times 10^{-4} \tag{19.3.33}$$

The approximation is improving with each additional node, and thus with increasing order of p_n

13. Automating using `sympy`, and guessing that two pairs of nodes share the same value leading to a reduction in order of 2,

$$p_n(x) = 2x^2 - 4x + 2 \tag{19.3.34}$$

This is two orders lower than it could have been.

14. Coded in `sympy`. The table of forward differences is not shown for clarity.

$$p_2(1.01) = 0.9943 \qquad \epsilon_2(1.01) = 2.0 \cdot 10^{-5} \tag{19.3.35}$$

$$p_2(1.03) = 0.9835 \qquad \epsilon_2(1.03) = 5.0 \cdot 10^{-5} \tag{19.3.36}$$

$$p_2(1.05) = 0.9735 \qquad \epsilon_2(1.05) = 0 \tag{19.3.37}$$

15. Coded in `sympy`

$$p_2(x) = -0.005233\ x^2 + 0.2050\ x + 0.7759 \tag{19.3.38}$$

16. Coded in `sympy`

$$p_2(x) = -0.044304\ x^2 + 1.3090\ x - 0.02322 \tag{19.3.39}$$

$$p_2(0.75) = 0.70929 \qquad \epsilon_2(0.75) = 0.0018656 \tag{19.3.40}$$

17. Using both Newton's forward and backward difference formulas,

$$p_2(0.3) = 0.3293 \qquad \epsilon_2(0.3) = -0.0007 \tag{19.3.41}$$

18. Using Newton's backward difference formula on Example 5,

x_i	f_i	First	Second	Third
0.5	1.127626			
		0.057839		
0.6	1.185465		0.011865	
		0.069704		0.000697
0.7	1.225169		0.012562	
		0.082266		
0.8	1.337435			

$$p_3(x) = 0.116167\ x^3 + 0.38415\ x^2 + 0.050113\ x + 0.99201 \quad 19.3.42$$

$$p_3(0.56) = 1.160945 \quad 19.3.43$$

The result matches Example 5 to $7S$

19. Using the erf function whose values are given in the Appendix,

$$x_0 = \{0.2, 0.4, 0.6, 0.8, 1.0\} \qquad x^* = 0.3 \quad 19.3.44$$

$$p_1(0.3) = 0.325550 \qquad \epsilon_1(0.3) = 3.077 \times 10^{-3} \quad 19.3.45$$

$$p_2(0.3) = 0.329325 \qquad \epsilon_2(0.3) = -6.98 \times 10^{-4} \quad 19.3.46$$

$$p_3(0.3) = 0.328881 \qquad \epsilon_3(0.3) = -2.54 \times 10^{-4} \quad 19.3.47$$

$$p_4(0.3) = 0.328616 \qquad \epsilon_4(0.3) = 1.1 \times 10^{-5} \quad 19.3.48$$

Starting with the first two nodes and successively adding nodes decreases the error.

20. Interpolation and extrapolation,

(a) Lagrange method practical error estimate, uses the difference between the better and worse estimate,

$$\epsilon_1(9.2) = 2.21885 \qquad \epsilon_2(9.2) = 2.21916 \quad 19.3.49$$

$$\widetilde{\epsilon_1} = 0.00031 \qquad \epsilon_{\text{true}} = 0.00035 \quad 19.3.50$$

This is a very good approximation of the true error.

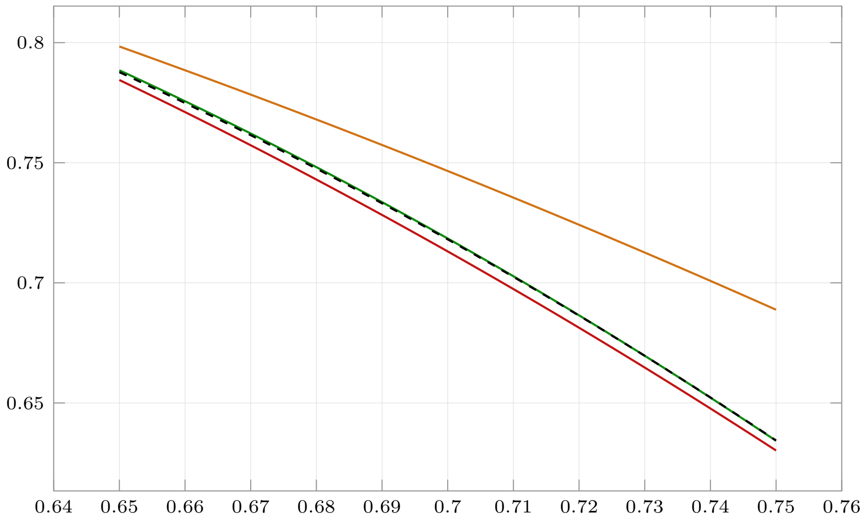
(b) Plotting the three quadratic polynomials,

$$x_0 = \{0.2, 0.4, 0.6, 0.8, 1.0\} \qquad x^* = 0.7 \tag{19.3.51}$$

$$p_{2a}(0.7) = 0.7185 \qquad \epsilon_a(0.7) = -4 \times 10^{-4} \tag{19.3.52}$$

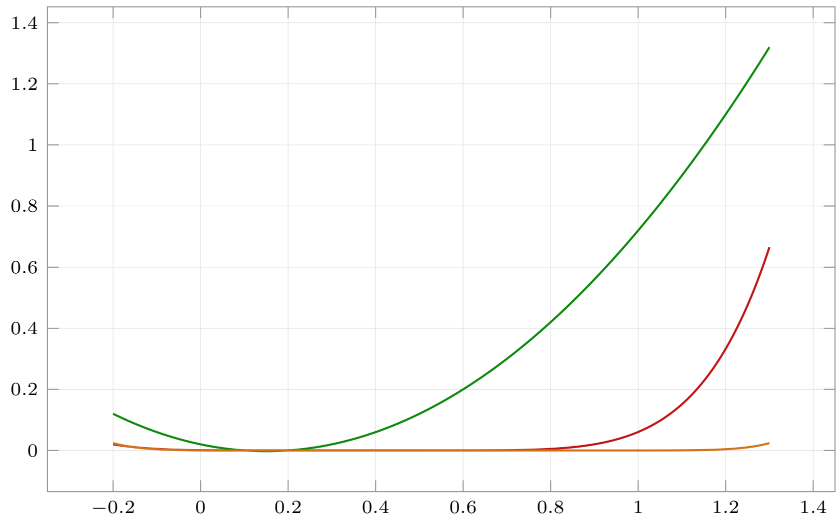
$$p_{2b}(0.7) = 0.7131 \qquad \epsilon_b(0.7) = 5 \times 10^{-3} \tag{19.3.53}$$

$$p_{2c}(0.7) = 0.7466 \qquad \epsilon_c(0.7) = -2.85 \times 10^{-2} \tag{19.3.54}$$



The error is smallest when the interpolation point is as central to the domain spanned by the nodes as possible.

(c) Plotting just 3 of the graphs for $n = 2, 6, 10$ using the points $\{0.1, 0.2, \ldots, 1\}$ as nodes,



The plynomials using a smaller number of nodes start to misbehave much quicker upon leaving the domain spanned by their respective sets of nodes.

21. Refer notes. TBC.

19.4 Spline Interpolation

1. Refer notes. TBC

2. Starting with the explicit form of the cubic polynomial q_j expanded in terms of $(x - x_j)$,

$$q_j(x_j) = f(x_j)\ c_j^2\ (x_j - x_{j+1})^2 \qquad = f(x_j)\ \frac{(x_j - x_{j+1})^2}{(x_{j+1} - x_j)^2} = f(x_j) \tag{19.4.1}$$

$$q_j(x_{j+1}) = f(x_{j+1})\ c_j^2\ (x_{j+1} - x_j)^2 \qquad = f(x_{j+1})\ \frac{(x_{j+1} - x_j)^2}{(x_{j+1} - x_j)^2} = f(x_{j+1}) \tag{19.4.2}$$

$$\tag{19.4.3}$$

Differentiating the full expression for q_j

$$q_j'(x) = f(x_j)\ c_j^2\ \Big[(x - x_{j+1})^2 \cdot 2c_j + 2(1 + 2c_jx - 2c_jx_j)(x - x_{j+1})\Big] \tag{19.4.4}$$

$$+ f(x_{j+1})\ c_j^2\ \Big[- 2c_j \cdot (x - x_j)^2 + 2(x - x_j)(1 - 2c_jx + 2c_jx_{j+1})\Big] \tag{19.4.5}$$

$$+ k_j\ c_j^2\ \Big[2(x - x_j)(x - x_{j+1}) + (x - x_{j+1})^2\Big] \tag{19.4.6}$$

$$+ k_{j+1}\ c_j^2\ \Big[2(x - x_j)(x - x_{j+1}) + (x - x_j)^2\Big] \tag{19.4.7}$$

Now, checking if this satisfies the interpolation conditions,

$$q_j'(x_j) = f(x_j)\ \Big[2c_j - 2c_j\Big] + k_j = k_j \tag{19.4.8}$$

$$q_j'(x_{j+1}) = f(x_{j+1})\ \Big[- 2c_j + 2c_j\Big] + k_{j+1} = k_{j+1} \tag{19.4.9}$$

3. Using the first derivative from Problem 1 to find $q_j''(x)$

$$q_j''(x) = f(x_j)\ c_j^2\ \Big[8c_j\ (x - x_{j+1}) + 2[1 + 2c_j\ (x - x_j)]\Big] \tag{19.4.10}$$

$$+ f(x_{j+1})\ c_j^2\ \Big[- 8c_j\ (x - x_j) + 2[1 - 2c_j(x - x_{j+1})]\Big] \tag{19.4.11}$$

$$+ k_j\ c_j^2\ \Big[6x - 2x_j - 4x_{j+1}\Big] \tag{19.4.12}$$

$$+ k_{j+1}\ c_j^2\ \Big[6x - 4x_j - 2x_{j+1}\Big] \tag{19.4.13}$$

Substituting the values of x into the above general expression,

$$q_j''(x_j) = -6f(x_j)\ c_j^2 + 6f(x_{j+1})\ c_j^2 - 4k_j\ c_j - 2k_{j+1}\ c_j \tag{19.4.14}$$

$$q_j''(x_{j+1}) = 6f(x_j)\ c_j^2 - 6f(x_{j+1})\ c_j^2 + 2k_j\ c_j + 4k_{j+1}\ c_j \tag{19.4.15}$$

4. Building upon the results of Problem 3,

$$q_j''(x_j) = 6c_{j-1}^2\ f(x_{j-1}) - 6c_{j-1}^2\ f(x_j) + 2c_{j-1}\ k_{j-1} + 4c_{j-1}\ k_j \tag{19.4.16}$$

$$q_j''(x_j) = -6f(x_j)\ c_j^2 + 6f(x_{j+1})\ c_j^2 - 4k_j\ c_j - 2k_{j+1}\ c_j \tag{19.4.17}$$

This yields a set of linear equations each with only three nonzero coefficients.

$$3\left[c_{j-1}^2\ \nabla f_j + c_j^2\ \nabla f_{j+1}\right] = [c_{j-1}]\ k_{j-1} + [2c_{j-1} + 2c_j]\ k_j + [c_j]\ k_{j+1} \tag{19.4.18}$$

Only three diagonals of the coefficient matrix can be nonzero, as seen from the above form of linear equation for $j \in [1, 2, \ldots, n-1]$

5. For the special case where, $c_j = c\ \forall\ j \in \{1, 2, \ldots, n-1\}$,

$$3c^2\left[\nabla f_j + \nabla f_{j+1}\right] = c\left[k_{j-1} + 4k_j + k_{j+1}\right] \qquad h = \frac{1}{c} \tag{19.4.19}$$

$$\frac{3}{h}\left[f_{j+1} - f_{j-1}\right] = k_{j-1} + 4k_j + k_{j+1} \tag{19.4.20}$$

6. Using the results of Problem 4,

$$q_j''(x_j) = -6f(x_j)\ c_j^2 + 6f(x_{j+1})\ c_j^2 - 4k_j\ c_j - 2k_{j+1}\ c_j \tag{19.4.21}$$

$$a_{j2} = \frac{q_j''(x_j)}{2!} = \frac{3}{h_j^2}\left[f_{j+1} - f_j\right] - \frac{1}{h_j}\left[2k_j + k_{j+1}\right] \tag{19.4.22}$$

$$q_j(x) = a_{j0} + a_{j1}\ (x - x_j) + a_{j2}\ (x - x_j)^2 + a_{j3}\ (x - x_j)^3 \tag{19.4.23}$$

$$q_j''(x) = 2a_{j2} + 6a_{j3}\ (x - x_j) \tag{19.4.24}$$

Equating the two expressions for $q_j''(x_{j+1})$

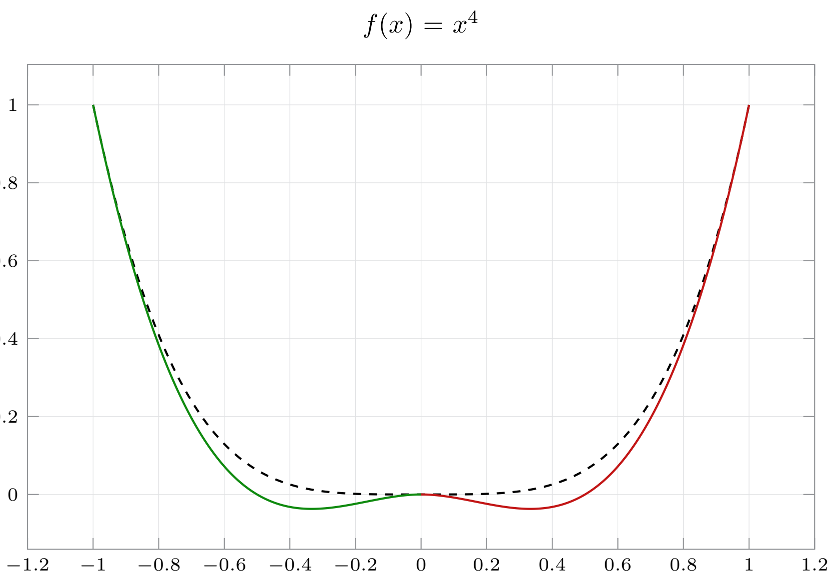
$$2a_{j2} + \frac{6a_{j3}}{c_j} = 6c_j^2\ f(x_j) - 6c_j^2\ f(x_{j+1}) + 2c_j\ k_j + 4c_j\ k_{j+1} \tag{19.4.25}$$

$$a_{j3} = c_j^3\ [f_j - f_{j+1}] + \frac{c_j^2}{3}\ [k_j + 2k_{j+1}] - c_j^3\ [f_{j+1} - f_j] \tag{19.4.26}$$

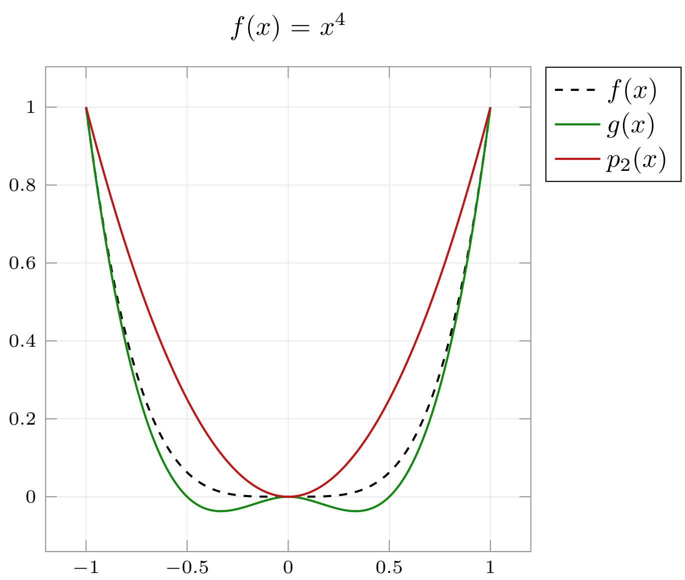
$$+ \frac{c_j^2}{3}\ [2k_j + k_{j+1}] \tag{19.4.27}$$

$$a_{j3} = \frac{2}{h_j^3}\ (f_j - f_{j+1}) + \frac{1}{h_j^2}\ (k_j + k_{j+1}) \tag{19.4.28}$$

7. Using Example 1 to test the `sympy` code, The output is a set of coefficients for the cubic equations of the form,



8. Overlaying the quadratic interpolation polynomial onto the result of Problem 7,



Using differentiation to find the greatest difference between each approximation and the true function.

$$\epsilon_1(x) = x^4 - 2x^3 + x^2 \qquad 4x^3 - 6x^2 + 2x = 0 \tag{19.4.29}$$

$$x^* = 0.5, \qquad \max|\Delta_g| = 1/16 \tag{19.4.30}$$

$$\epsilon_2(x) = x^4 - x^2 \qquad 4x^3 - 2x = 0 \tag{19.4.31}$$

$$x^* = \frac{1}{\sqrt{2}}, \qquad \max|\Delta_p| = 1/4 \tag{19.4.32}$$

The spline interpolant has a much smaller maximum error over the interval.

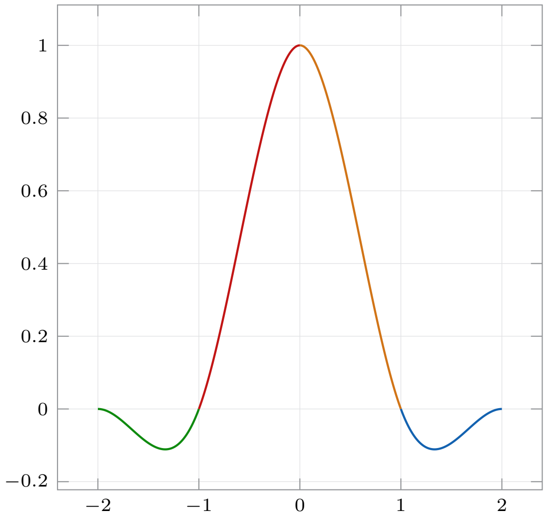
9. Using the edge splines,

$$q_5(x) = 1.5 - 1.13\ (x-5) - 1.39\ (x-5)^2 + 0.58\ (x-5)^3 \tag{19.4.33}$$

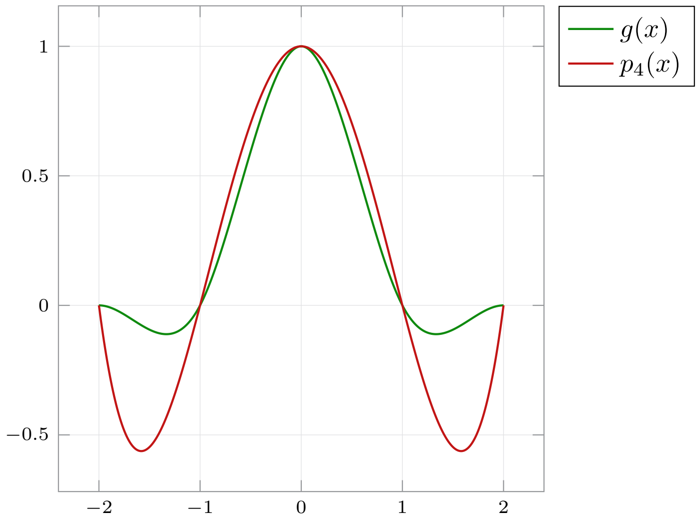
$$q_5''(5.8) = -1.39\ (2) + 0.58\ (6)(5.8-5) = \frac{1}{250} \tag{19.4.34}$$

This result is not zero due to roundoff error. Since the function is even, the second derivative is the same value at the other extreme.

10. Using the `sympy` code,

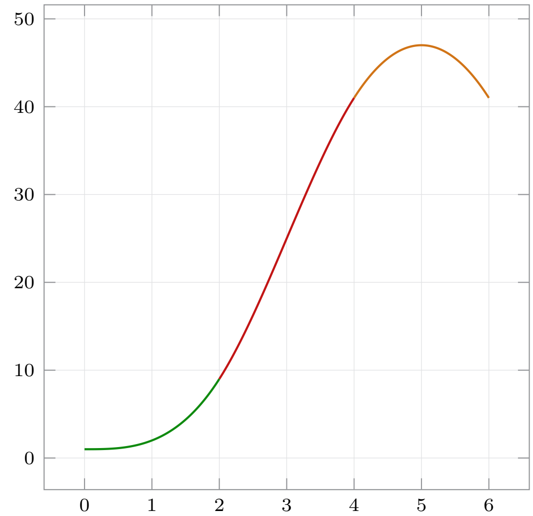


11. Overlaying the fourth degree polynomial on top of the cubic spline obtained in Problem 10,

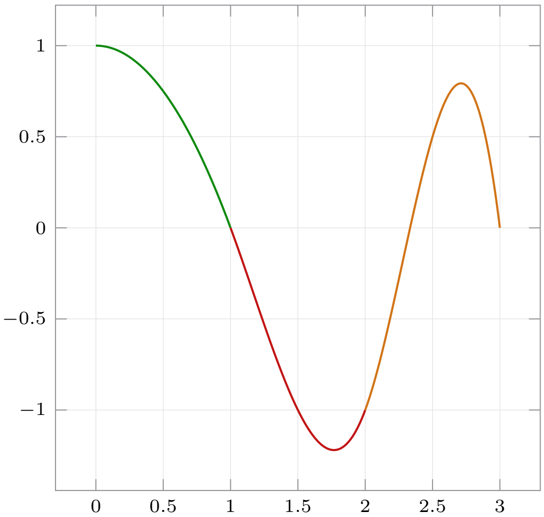


The fourth degree polynomial is a much worse approximation than the cubic spline.

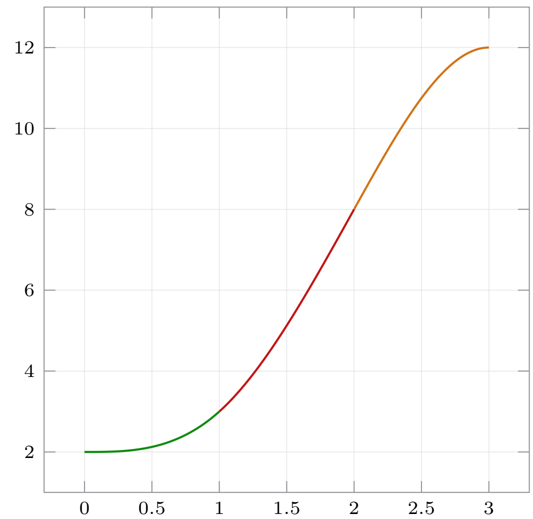
12. Using the `sympy` code,



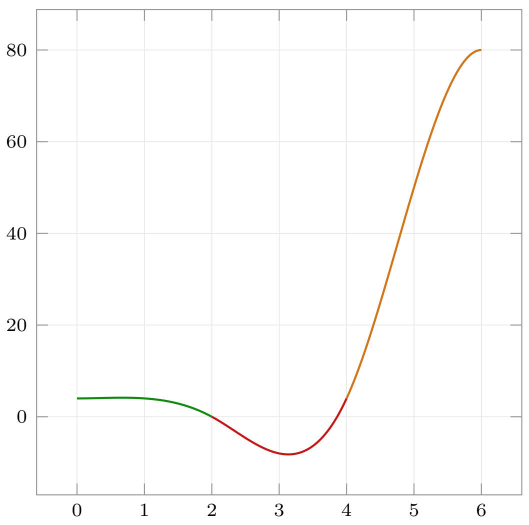
13. Using the `sympy` code,



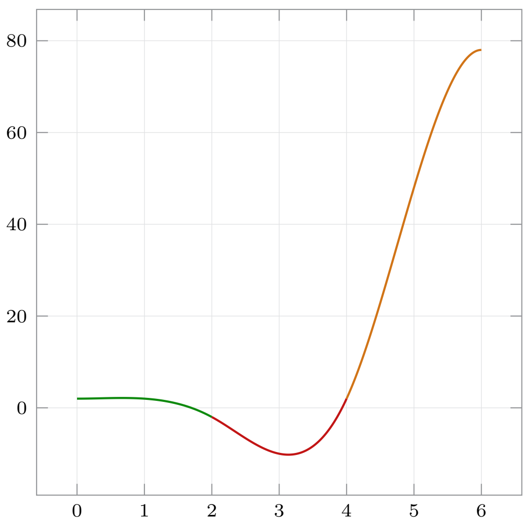
14. Using the `sympy` code,



15. Using the `sympy` code,



16. Using the `sympy` code, or simply subtracting 2 from each of the cubic splines in Problem 15,



17. The spline $g(x)$ has continuous derivatives upto order 3.

$$q_j'''(x_j) = q_{j+1}'''(x_j) \implies 6a_{j,3}\ x_j = 6a_{j+1,3}\ x_j \tag{19.4.35}$$

$$a_{j,3} = a_{j+1,3} \tag{19.4.36}$$

$$q_j''(x_j) = q_{j+1}''(x_j) \implies 2a_{j,2}\ x_j = 2a_{j+1,2}\ x_j \tag{19.4.37}$$

$$a_{j,2} = a_{j+1,2} \tag{19.4.38}$$

$$q_j'(x_j) = q_{j+1}'(x_j) \implies a_{j,1}\ x_j = a_{j+1,1}\ x_j \tag{19.4.39}$$

Including the fact that the at the nodes, q_j and q_{j+1} have to match to ensure continuity, the coefficients of adjacent cubic polynomials are equal.

18. TBC

19. The curvature is given by,

$$k = \frac{y''}{(1+y'^2)^{3/2}} = 0 \qquad \Longrightarrow \qquad y'' = 0 \tag{19.4.40}$$

In physical systems, this corresponds to a free end of the beam which makes the spline a straight line outside the domain of interpolation.

20. Bezier curves

(a) Let the general equation be,

$$\mathbf{r}(t) = \mathbf{r}_0 + \mathbf{v}_0\ t + (a_1\mathbf{r}_0 + a_2\mathbf{r_1} + b_1\mathbf{v}_0 + b_2\mathbf{v}_1)\ t^2 \tag{19.4.41}$$

$$+ (c_1\mathbf{r}_0 + c_2\mathbf{r_1} + d_1\mathbf{v}_0 + d_2\mathbf{v}_1)\ t^3 \tag{19.4.42}$$

This already satisfies $\mathbf{r}(0) = \mathbf{r}_0$ and $\mathbf{r}'(0) = \mathbf{v}_0$, the two starting boundary conditions. For the other two boundary conditions,

$$\mathbf{r}(1) = (1 + a_1 + c_1)\ \mathbf{r}_0 + (1 + b_1 + d_1)\ \mathbf{v}_0 + (a_2 + c_2)\ \mathbf{r}_1 + (b_2 + d_2)\ \mathbf{v}_1 \tag{19.4.43}$$

$$\mathbf{r}'(1) = (2a_1 + 3c_1)\ \mathbf{r}_0 + (1 + 2b_1 + 3d_1)\ \mathbf{v}_0 + (2a_2 + 3c_2)\ \mathbf{r}_1 + (2b_2 + 3d_2)\ \mathbf{v}_1 \tag{19.4.44}$$

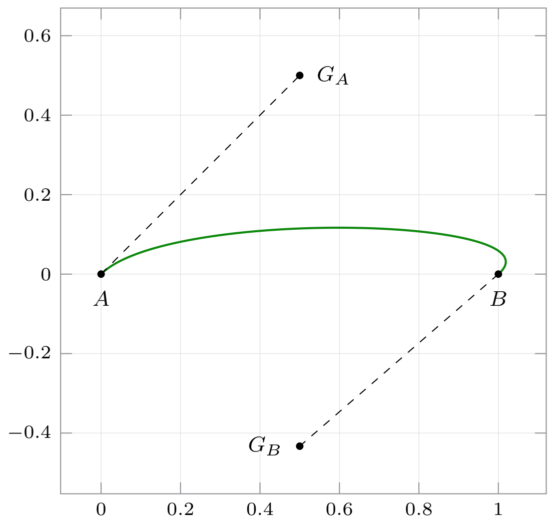
This is a set of 8 equations in 8 variables, that has the solution,

$$\mathbf{r}(t) = \mathbf{r}_0 + \mathbf{v}_0\ t + (-3\mathbf{r}_0 + 3\mathbf{r}_1 - 2\mathbf{v}_0 - \mathbf{v}_1)\ t^2 + (2\mathbf{r}_0 - 2\mathbf{r}_1 + \mathbf{v}_0 + \mathbf{v}_1)\ t^3 \tag{19.4.45}$$

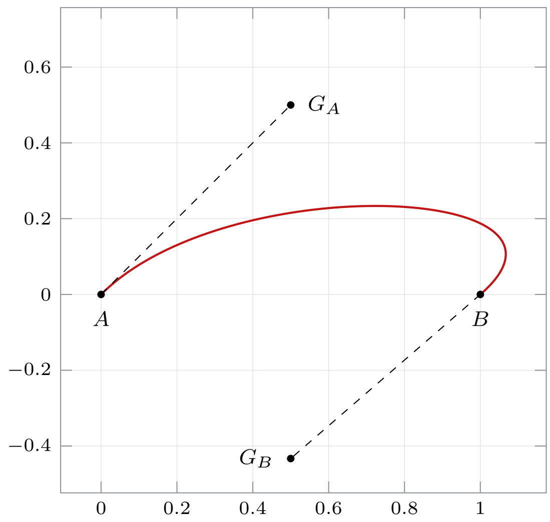
In components this decomposes simply into

$$\mathbf{r} = x\ \hat{\mathbf{i}} + y\ \hat{\mathbf{j}} \qquad \mathbf{v} = x'\ \hat{\mathbf{i}} + y'\ \hat{\mathbf{j}} \tag{19.4.46}$$

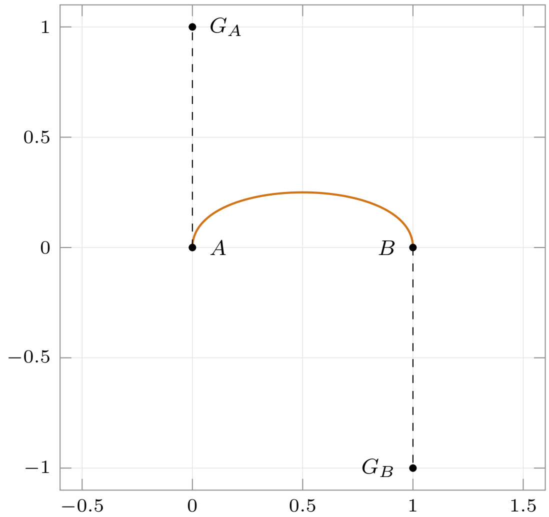
(b) Plotting the Bezier curve with the given guidepoints,

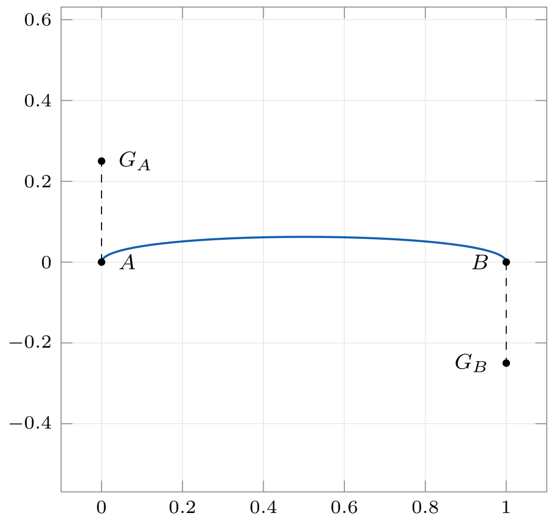


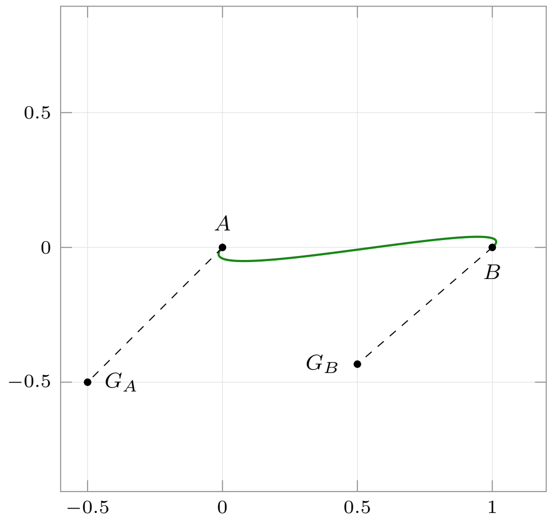
(c) The curve conforms to the guidepoints for a larger window in t, as the magnitude of the velocity vectors increases, keeping their direction unchanged.



(d) Other changes TBC.







19.5 Numeric Integration and Differentiation

1. Coded in `sympy`

$$J_r = 0.747131 \qquad \epsilon_r = -3.07 \times 10^{-4} \tag{19.5.1}$$

Since the function is monotonically decreasing in the domain $[0, 1]$, J_r is larger than J_t

2. For a single subinterval, whose midpoint is $c = x_0 + h/2 = (x_0 + x_1)/2$,
Using the Taylor expansion of $(x - c_0)$ about c,

$$f(x) = f(c) + (x - c)\ f'(c) + (x - c)^2\ \frac{f''\{t(x)\}}{2!} \tag{19.5.2}$$

$$\int_{x_0}^{x_0+h} [f(x) - f(c)]\ \mathrm{d}x = \int_{c-h/2}^{c+h/2} (x - c)\ f'(c)\ \mathrm{d}x \tag{19.5.3}$$

$$+ \int_{c-h/2}^{c+h/2} (x - c)^2\ \frac{f''\{t(x)\}}{2!}\ \mathrm{d}x \tag{19.5.4}$$

Here, $t^* \in [x_0, x_0 + h]$ in accordance with the Mean value theorem for integration.

$$\epsilon_{M,j} = 0 + \left[\frac{(x - c)^3}{6}\ f''(t_j^*)\right]_{c-h/2}^{c+h/2} \qquad \epsilon_{M,j} = \frac{h^3}{24}\ f''(t_j) \tag{19.5.5}$$

$$\epsilon_M = \frac{(b - a)}{24}\ h^2 f''(t^*) \tag{19.5.6}$$

Here, t^* is some suitable value in $[a, b]$, whereas t_j is a value between x_j, x_{j+1}.
Applying this error formula to Problem 1,

$$h = 0.1 \qquad (b - a) = 1 \tag{19.5.7}$$

$$\epsilon = \frac{0.001}{24}\ f''(t) \qquad f''(t) = e^{-x^2}[-2 + 4x^2] \tag{19.5.8}$$

$$f'''(t) = e^{-x^2}[12x - 8x^3] \tag{19.5.9}$$

Since $f'''(t) > 0$ in the entire interval of integration, the second derivative is monotonically increasing

$$\epsilon_{\min} = -0.0008333 \qquad \epsilon_{\max} = 0.0003066 \tag{19.5.10}$$

$$J_M = [0.7462977, 0.7474376] \tag{19.5.11}$$

3. Tabulating the results and the corresponding errors,

h	Value	Error
1	0.5	-0.166667
0.5	0.375	-0.04167
0.25	0.34375	-0.010412
0.1	0.335	-0.001667

The relation $\epsilon \propto h^2$ seems to hold.

4. Using the practical error estimate, for $f(x) = x^4$

h	J_h	$J_{h/2}$	Practical Error	True Error
1	0.5	0.28125	-0.0729167	-0.3
0.5	0.28125	0.220703	-0.020182	-0.08125
0.25	0.220703	0.205200	-0.00516767	-0.020703

5. Using the practical error estimate, for $f(x) = \sin(\pi x/2)$

h	J_h	$J_{h/2}$	Practical Error	True Error
1	0.5	0.603553	0.0345178	0.033066
0.5	0.603553	0.628417	0.008288	0.008202
0.25	0.628417	0.634573	0.002052	0.002047

6. Let the individual roundoff errors be δ_j

$$\epsilon = \frac{h}{2} \ |\delta_0 + 2\delta_1 + \cdots + 2\delta_{n-1} + \delta_n| \qquad |\delta_j| \le u \tag{19.5.12}$$

$$\epsilon \le \frac{(b-a)}{2n} \ (2nu) \qquad \epsilon \le (b-a)u \tag{19.5.13}$$

Since the error bound is independent of the number of subintervals n, the algorithm is numerically stable.

7. Using Simpson's rule coded in `numpy`

$$J = \int_1^2 \frac{1}{x} \ \mathrm{d}x \qquad 2m = 4 \tag{19.5.14}$$

$$J_S = 0.693254 \qquad \epsilon_S = -0.000107 \tag{19.5.15}$$

8. Using Simpson's rule coded in `numpy`

$$J = \int_1^2 \frac{1}{x} \ \mathrm{d}x \qquad 2m = 10 \tag{19.5.16}$$

$$J_S = 0.693150 \qquad \epsilon_S = -0.000003 \tag{19.5.17}$$

9. Using Simpson's rule coded in `numpy`

$$J = \int_0^{0.4} xe^{-x^2} \, \mathrm{d}x \qquad 2m = 4 \tag{19.5.18}$$

$$J_S = 0.0739303 \qquad \epsilon_S = -2.2494 \times 10^{-6} \tag{19.5.19}$$

10. Using Simpson's rule coded in `numpy`

$$J = \int_0^{0.4} xe^{-x^2} \, \mathrm{d}x \qquad 2m = 10 \tag{19.5.20}$$

$$J_S = 0.0739282 \qquad \epsilon_S = -5.6817 \times 10^{-8} \tag{19.5.21}$$

11. Using Simpson's rule coded in `numpy`

$$J = \int_0^{1} \frac{1}{1+x^2} \, \mathrm{d}x \qquad 2m = 4 \tag{19.5.22}$$

$$J_S = 0.785392 \qquad \epsilon_S = 6.0065 \times 10^{-6} \tag{19.5.23}$$

12. Using Simpson's rule coded in `numpy`

$$J = \int_0^{1} \frac{1}{1+x^2} \, \mathrm{d}x \qquad 2m = 10 \tag{19.5.24}$$

$$J_S = 0.785398 \qquad \epsilon_S = 9.9126 \times 10^{-9} \tag{19.5.25}$$

13. For the practical error estimate,

$$J = \int_0^{1} \frac{1}{1+x^2} \, \mathrm{d}x \qquad 2m = 8 \tag{19.5.26}$$

$$J_8 = 0.785398125 \qquad J_4 = 0.785392157 \tag{19.5.27}$$

$$\epsilon_8 \approxeq \frac{J_8 - J_4}{15} = 3.9786 \times 10^{-7} \qquad \epsilon_8^* = 3.778 \times 10^{-8} \tag{19.5.28}$$

14. Using Simpson's rule coded in `numpy`

$$J = \int_0^2 e^{-x} \ \mathrm{d}x \tag{19.5.29}$$

$$J_2 = 0.868951 \qquad \epsilon_2 = 4.2862 \times 10^{-3} \tag{19.5.30}$$

$$J_4 = 0.864956 \qquad \epsilon_4 = -2.9152 \times 10^{-4} \tag{19.5.31}$$

$$\epsilon_4^* \approxeq \frac{J_4 - J_2}{15} = -2.6633 \times 10^{-4} \tag{19.5.32}$$

15. Checking the number of subintervals needed,

(a) Using the trapezoidal rule,

$$A = \int_1^2 \frac{1}{x} \ \mathrm{d}x = \ln 2 \qquad \epsilon_T = -\frac{(2-1)^3}{12n^2} \frac{2}{t^3} \tag{19.5.33}$$

$$\epsilon_T = \frac{1}{6n^2} < 0.5 \times 10^{-5} \qquad \implies \qquad n \geq 183 \tag{19.5.34}$$

(b) Using the simpson rule,

$$A = \int_1^2 \frac{1}{x} \ \mathrm{d}x = \ln 2 \qquad \epsilon_S = -\frac{(2-1)^5}{180n^4} \frac{24}{t^5} \tag{19.5.35}$$

$$\epsilon_S = \frac{2}{15n^4} < 0.5 \times 10^{-5} \qquad \implies \qquad n \geq 13 \tag{19.5.36}$$

However, since n has to be even, the answer is $n = 14$

16. Using the practical error estimate, for $f(x) = \sin(x)/x$

h	J_h	$J_{h/2}$	Practical Error	True Error
0.2	0.9450787	0.9458321	0.0644415	0.0010043
0.1	0.9458321	0.9460203	0.0002511	0.000250998

17. Using the practical error estimate, for $f(x) = \sin(x)/x$

$2m$	J_h	$J_{h/2}$	Practical Error	True Error
2	0.94614588	0.94608693	-0.015244	-6.28 $\times 10^{-5}$
4	0.94608693	0.94608331	-3.929 $\times 10^{-6}$	-2.86 $\times 10^{-6}$

18. Using the tabulated data in Problem 17,

$$J_4 \approxeq 0.94608693 - 3.929 * 10^{-6} \qquad J_4 \approxeq 0.946083001 \tag{19.5.37}$$

$$\epsilon_4^* = 6.9 \times 10^{-8} \tag{19.5.38}$$

This is 2 orders of magnitude a smaller error than the J_4 calculation on its own.

19. Using the practical error estimate, for $f(x) = \sin(x)/x$

$2m$	J_h	True Error
10	0.94608316883	-9.847 $\times$ 10^{-8}

20. Using the practical error estimate, for $f(x) = \sin(x^2)$

$2m$	J_h	True Error
10	0.54594097	2.213 14 $\times$ 10^{-5}

21. Using the practical error estimate, for $f(x) = \cos(x^2)$

$2m$	J_h	True Error
10	0.97745853	2.085 92 $\times$ 10^{-5}

22. Transforming from $x \to t$,

$$x = \frac{a(1-t) + b(1+t)}{2} = \frac{\pi}{4}\,(1+t) \qquad J = \int_0^{\pi/2} \cos(x)\ \mathrm{d}x \tag{19.5.39}$$

$$J^* = \pi/4 \int_{-1}^{1} \cos\left[\frac{\pi(t+1)}{4}\right]\ \mathrm{d}t \tag{19.5.40}$$

n	Approximation	True Error
5	$1 + \delta$	-3.956×10^{-11}

23. Transforming from $x \to t$,

$$x = \frac{a(1-t) + b(1+t)}{2} = \frac{1}{2}\,(1+t) \qquad J = \int_0^{1} xe^{-x}\ \mathrm{d}x \tag{19.5.41}$$

$$J^* = \frac{1}{2} \int_{-1}^{1} \cos\left[\frac{\pi(t+1)}{4}\right]\ \mathrm{d}t \tag{19.5.42}$$

n	Approximation	True Error
5	0.26424111766	-2.2874×10^{-12}

24. Transforming from $x \to t$,

$$x = \frac{a(1-t)+b(1+t)}{2} = \frac{5}{8}\,(1+t) \qquad J = \int_0^{1.25} \sin(x^2)\;\mathrm{d}x \tag{19.5.43}$$

$$J^* = \frac{5}{8}\int_{-1}^{1} \sin\left[\frac{5(t+1)}{8}\right]^2\;\mathrm{d}t \tag{19.5.44}$$

n	Approximation	True Error
5	0.545962669	-3.8273×10^{-7}

25. Transforming from $x \to t$,

$$x = \frac{a(1-t)+b(1+t)}{2} = \frac{1}{2}\,(1+t) \qquad J = \int_0^{1} e^{-x^2}\;\mathrm{d}x \tag{19.5.45}$$

$$J^* = \frac{1}{2}\int_{-1}^{1} \exp\left[-\frac{(1+t)^2}{4}\right]\;\mathrm{d}t \tag{19.5.46}$$

n	Approximation	True Error
5	0.74682412676	$6.046\,17 \times 10^{-9}$

26. TBC

27. Using the three formulas,

$$f(x) = x^4 \qquad f'(x) = 4x^3 \tag{19.5.47}$$

$$f'(0.4) = \frac{-3(0.4)^4 + 4(0.6)^4 - 1(0.8)^4}{2(0.2)} = 0.08 \qquad \epsilon_a = 0.176 \tag{19.5.48}$$

$$f'(0.4) = \frac{-(0.2)^4 + (0.6)^4}{2(0.2)} = 0.32 \qquad \epsilon_b = -0.064 \tag{19.5.49}$$

$$f'(0.4) = \frac{1(0)^4 - 4(0.2)^4 + 3(0.4)^4}{2(0.2)} = 0.176 \qquad \epsilon_c = 0.08 \tag{19.5.50}$$

$$f'(0.4) = \frac{1(0)^4 - 8(0.2)^4 + 8(0.6)^4 - (0.8)^4}{12(0.2)} = 0.256 \qquad \epsilon_d = 0 \tag{19.5.51}$$

The central difference formula using nearest neighbours is more accurate than the forward and backward three point formulas.

Since the last formula used a fourth degree Lagrange polynomial as interpolant, it was able to replicate the function $f(x)$ exactly.

28. Using the three formulas,

$$f(x) = x^4 \qquad\qquad f'(x) = 4x^3 \tag{19.5.52}$$

$$f'(0.4) = \frac{-2(0.2)^4 - 3(0.4)^4 + 6(0.6)^4 - (0.8)^4}{6(0.2)} = 0.24 \qquad\qquad \epsilon^* = 0.016 \tag{19.5.53}$$

This is worse than the three point centered difference formula, but better than the three point backward and forward formulas.

Since this interpolant is only a third degree polynomial (four nodes in the formula), it cannot exactly match a fourth degree polynomial.

29. Using the point $x = 0.4$ and the next 4 forward points,

n	Approximation	True Error
2	0.52	-0.264
3	0.08	0.176
4	0.304	-0.048
5	0.256	0

Once again, incorporating 4 forward points, enables an exact computation of the derivative, since this polynomial can be of degree 4.

30. Differentiating Equation 14 from Section 19.3 on both sides,

$$f'(x) = \Delta f_0 + \frac{r + (r-1)}{2!}\,\Delta^2 f_0 + \frac{(r-1)(r-2) + r(r-2) + r(r-1)}{3!}\,\Delta^3 f_0 + \ldots \tag{19.5.54}$$

$$f'(x_0) = \left[\Delta f_0 - \frac{1}{2!}\,\Delta^2 f_0 + \cdots + \frac{(-1)^{n-1}\,(n-1)!}{n!}\,\Delta^n f_0\right] \cdot \frac{\mathrm{d}r}{\mathrm{d}x} \tag{19.5.55}$$

$$f'(x_0) = \frac{1}{h} \cdot \left[\Delta f_0 - \frac{1}{2}\Delta^2 f_0 + \cdots + \frac{(-1)^{n-1}}{n}\,\Delta^n f_0\right] \tag{19.5.56}$$

This is because $x = x_0 \implies r = 0$ and all but the first term in the sum of each numerator reduce to zero.