Problem 1.

Solution.

Let A be an $n \times n$ tridiagonal system. To reduce $Ax = b$ to upper triangular form,[1] the equivalent pseudocode (using MATLAB index notation) is

```
for k = 2:n
    Ab(k, :) = Ab(k, :) - (Ab(k, k) / Ab(k - 1, k)) * Ab(k, :);
end
```

By hand, this yields

$$[A \quad b] = \left[\begin{array}{cccc|c} 2 & 1 & & & 5 \\ -2 & 2 & -2 & & 4 \\ & -6 & 5 & 3 & -8 \\ & & 1 & 7 & 36 \end{array}\right] \sim \left[\begin{array}{cccc|c} 2 & 1 & & & 5 \\ & 3 & -2 & & 9 \\ & -6 & 5 & 3 & -8 \\ & & 1 & 7 & 36 \end{array}\right] \sim \left[\begin{array}{cccc|c} 2 & 1 & & & 5 \\ & 3 & -2 & & 9 \\ & & 1 & 3 & 10 \\ & & 1 & 7 & 36 \end{array}\right] \sim \left[\begin{array}{cccc|c} 2 & 1 & & & 5 \\ & 3 & -2 & & 9 \\ & & 1 & 3 & 10 \\ & & & 4 & 26 \end{array}\right].$$

This takes $(4-1)(3M+2A)$ flops. That is, $9M+6A$ flops are used to obtain the upper triangular form.

The upper triangular system can then be solved using back substitution, starting from the last row:

$$\begin{aligned} x_4 &= \frac{26}{4} = 6.5 \\ x_3 &= \frac{10 - 3x_4}{1} = -9.5 \\ x_2 &= \frac{9 - (-2)x_3}{3} \approx 9.33 \\ x_1 &= \frac{5 - 1x_2}{1} \approx 4.33. \end{aligned}$$

This takes $1M + (4-1)(2M+1A) = 7M + 3A$ flops.

[Some operations above may be elided because of the value of operands (e.g. multiplication/division by 1); discounting is correct but not required; this also applies to Problem 2]

[Other implementations of back subsitutions are valid, but complexity should not exceed $\mathcal{O}(n)$]

Problem 2.

Solution. The system is 6×6 and tridiagonal. Following the formulas in Problem 1, we expect $[(6-1)(3M+2A)] + [1M + (6-1)(2M+1A)] = 26M + 15A$ to solve this problem.

[1] LU decomposition without pivoting

Problem 3.

Solution. We begin by verify that $f(a_0) = 0.5$, and $f(b_0) = -3.485$, and we see that a_0, b_0 straddle the root.

We can pre-compute the number of iterations: $\log_2(\frac{b_0 - a_0}{2\epsilon}) = 17.6$, which rounds up to 18.

[If indexed from 1, then 19 iterations.]

Bisection yields

#	a	b	error bound	c	$f(c)$
0	0.000 000	4.000 000	2.000 000	2.000 000	0.114 389
1	2.000 000	4.000 000	1.000 000	3.000 000	−1.631 026
2	2.000 000	3.000 000	0.500 000	2.500 000	−0.739 482
3	2.000 000	2.500 000	0.250 000	2.250 000	−0.306 995
4	2.000 000	2.250 000	0.125 000	2.125 000	−0.094 809
5	2.000 000	2.125 000	0.062 500	2.062 500	0.010 176
6	2.062 500	2.125 000	0.031 250	2.093 750	−0.042 221
7	2.062 500	2.093 750	0.015 625	2.078 125	−0.015 999
8	2.062 500	2.078 125	0.007 812	2.070 312	−0.002 905
9	2.062 500	2.070 312	0.003 906	2.066 406	0.003 637
10	2.066 406	2.070 312	0.001 953	2.068 359	0.000 366
11	2.068 359	2.070 312	0.000 977	2.069 336	−0.001 269
12	2.068 359	2.069 336	0.000 488	2.068 848	−0.000 451
13	2.068 359	2.068 848	0.000 244	2.068 604	−0.000 043
14	2.068 359	2.068 604	0.000 122	2.068 481	0.000 162
15	2.068 481	2.068 604	0.000 061	2.068 542	0.000 060
16	2.068 542	2.068 604	0.000 031	2.068 573	0.000 009
17	2.068 573	2.068 604	0.000 015	2.068 588	−0.000 017
18	2.068 573	2.068 588	0.000 008	2.068 581	−0.000 004

[Students need not to include the table.]

The estimate of root is the last c, which is 2.068581.

Problem 4.

Solution. Given $f(x) = xe^x - (x^2 + 1)$, $f'(x) = (x+1)e^x - 2x$. Newton's method is given by

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

Given $x_0 = 0$, we have

$$\begin{aligned} x_1 &= 1 \\ x_2 &= 0.790\,988\,353\,434\,663 \\ x_3 &= 0.740\,771\,897\,507\,255 \\ x_4 &= 0.738\,437\,121\,204\,896. \end{aligned}$$

Problem 5.

Solution. One can convert this problem into many different forms.

[We list a few here. Students only need to provide two, one of which converges, the other does not.]

#1: $f(x) = \arctan(x + 2x^3/3) - 0.7$, $f'(x) = \dfrac{1+2x^2}{1+(x+2x^3/3)^2}$.

Newton iterations from starting point $x_0 = 0, 1, 2$ are shown below.

0	1	2
0.7	0.583 969 922 154 570	−2.475 153 408 827 56
0.654 446 680 862 477	0.654 282 057 391 443	23.877 690 888 208 4
0.654 971 666 666 263	0.654 971 624 780 980	−63 147.457 538 798 1
0.654 971 724 522 525	0.654 971 724 522 524	$8.023\,967\,255\,959\,25 \times 10^{18}$
0.654 971 724 522 526	0.654 971 724 522 526	$-8.021\,593\,789\,161\,46 \times 10^{74}$

We see that iterations starting from $x_0 = 0, 1$ converge, whereas the iteration starting from $x_0 = 2$ diverges.

#2: $f(x) = x + 2x^3/3 - \tan(0.7)$, $f'(x) = 1 + 2x^2$.

0	1	2
0.842 288 380 463 079	0.725 207 237 932 138	1.278 772 783 014 42
0.677 596 056 795 344	0.658 346 214 941 770	0.850 120 443 065 757
0.655 329 311 226 069	0.654 979 742 246 570	0.679 423 965 394 953
0.654 971 814 661 839	0.654 971 724 567 848	0.655 389 107 326 869
0.654 971 724 522 532	0.654 971 724 522 526	0.654 971 847 325 937

#3: $f(x) = x + \sqrt[3]{\frac{3}{2}\left(x - \tan(0.7)\right)}$, $f'(x) = 1 + \frac{1}{\sqrt[3]{18}\,\sqrt[3]{x-\tan(0.7)}^{2}}$.

0	1	2
0.757 137 869 616 088	0.298 505 614 839 043	−0.378 745 467 694 460
0.671 815 830 160 573	0.702 790 496 433 078	0.822 335 503 029 322
0.655 263 126 999 129	0.657 677 973 400 205	0.739 602 817 767 492
0.654 971 805 947 435	0.654 978 816 080 601	0.665 313 385 543 077
0.654 971 724 522 532	0.654 971 724 570 694	0.655 078 589 202 672

We see that iterations starting from $x_0 = 0, 1, 2$ all converge.

Based on the results, we see that the true solution $\hat{x} \approx 0.654\,971\,724\,522\,53$. Consider #1 with $x_0 = 0$ for an example, we see that, in the first iteration $x_1 = 0.7$, there are no correct digits (or 1 if rounding is considered); in the second iteration, there are 3 correct digits; in the third, 6; in the fourth, 13; in the fifth, the digit-doubling trend is capped by machine precision.

Problem 6.

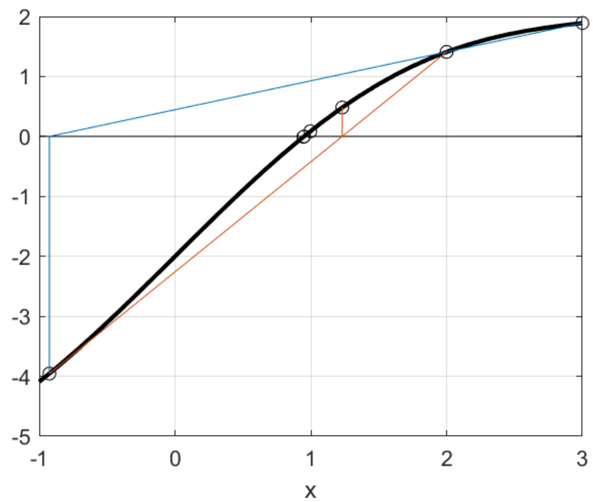
The secand rule requires two starting positions, and further iterations are given by

$$x_{k+1} = x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}.$$

In this problem, $x_0 = 3, x_1 = 2$.

#	x	$f(x)$
0	3	1.891 120 008 059 867
1	2	1.409 297 426 825 682
2	−0.924 930 216 462 200	−3.954 741 533 685 033
3	1.231 531 191 691 345	0.482 413 468 521 703
4	0.997 077 802 724 052	0.086 235 777 776 127
5	0.946 044 463 840 698	−0.006 336 141 352 894
6	0.949 537 471 784 331	0.000 068 212 679 983
7	0.949 500 267 808 440	0.000 000 052 228 887
8	0.949 500 239 300 380	−0.000 000 000 000 432

[Students need not to include the table.]



The thick black curve is the function f. The first "secant" is the blue line, which connects $(x_0, f(x_0))$ and $(x_1, f(x_1))$; it is extended and its intersection with x-axis is the new estimate x_2. The second "secant" is the orange line, which connects $(x_1, f(x_1))$ and $(x_2, f(x_2))$, whose intersection with x-axis is the new estimate x_3.

Looking at x column of the table above, using x_8 as the estimate for true solution we see that x_4 has 1 correct digit (after decimal point); for x_5 2; x_6 4; x_7 7. We almost have the digit doubling as in Newton's method.

Problem 7.

Solution. Given fixed-point iteration $x_{k+1} = g(x_k)$, fixed-point method is guaranteed to converge if there exists some $k \in (0, 1)$ such that

$$|g(x_2) - g(x_1)| \leq k|x_2 - x_1|$$

for all x_1, x_2. This is equivalent to $\max|g'| < 1$.

Form that is guaranteed to converge: $x = \dfrac{\arctan(2x + \frac{1}{2})}{4}$. Here,

$$g'(x) = \frac{1}{4} \times \arctan'\left(2x + \frac{1}{2}\right) \times 2.$$

The derivative of arctan is never greater than 1 (or less than -1). Hence $\max|g'| \leq \frac{1}{2} < 1$.

Form that is not guaranteed to converge: $x = \dfrac{\tan(4x - \frac{1}{2})}{2}$. Here,

$$g'(x) = \frac{1}{2}\tan'\left(4x - \frac{1}{2}\right) \times 4.$$

The derivative of tan is $\sec^2$, which is unbounded, and therefore the condition for convergence is not satisfied.

[These two forms are not the only two possible forms for use with fixed-point methods.]

Problem 8.

Solution. Fixed-point iteration $x_{k+1} = g(x_k)$.

With $g(x) = \frac{\arctan(2x+\frac{1}{2})}{4}$, iteration steps starting from $x_0 = 0, 2, 1000$ are listed below

#			
0	0	2	1000
1	0.115 911 902 250 202	0.338 031 845 230 239	0.392 574 112 951 322
2	0.157 941 624 537 384	0.216 532 888 906 576	0.227 384 934 598 999
3	0.171 087 773 325 300	0.187 696 512 614 335	0.190 565 987 719 255
4	0.174 983 497 600 905	0.179 763 138 657 429	0.180 573 110 281 140
5	0.176 118 725 736 064	0.177 499 656 671 370	0.177 732 383 289 113
6	0.176 447 896 691 087	0.176 847 318 154 673	0.176 914 524 947 884
7	0.176 543 205 251 994	0.176 658 771 055 815	0.176 678 207 238 455
8	0.176 570 789 444 505	0.176 604 229 521 598	0.176 609 852 818 266
9	0.176 578 771 889 472	0.176 588 448 355 056	0.176 590 075 490 588
10	0.176 581 081 806 272	0.176 583 881 880 776	0.176 584 352 719 000

We see that in all cases, the results have stabilized around 0.17658.

With $g(x) = \frac{\tan(4x-\frac{1}{2})}{2}$, iteration steps starting from $x_0 = 0, 2, 1000$ are listed below

#			
0	0	2	1000
1	−0.250 000 000 000 000	−3.649 855 727 610 19	0.218 187 301 101 659
2	−1.028 703 862 327 45	0.753 488 811 875 833	0.345 979 158 188 811
3	−0.984 666 902 528 119	−0.314 167 539 792 807	2.394 298 703 606 11
4	−0.761 815 294 879 434	−1.789 015 088 234 78	−0.173 195 907 374 006
5	−0.202 693 862 917 647	−0.846 131 497 188 007	−0.665 013 337 501 661
6	−0.776 043 800 336 549	−0.373 913 973 324 557	0.011 284 462 316 337 6
7	−0.231 282 537 836 976	−6.891 678 897 210 18	−0.227 415 735 365 187
8	−0.913 668 280 570 303	0.177 697 915 769 342	−0.892 737 481 809 577
9	−0.531 705 000 803 255	0.180 453 069 141 679	−0.478 921 669 772 596
10	0.554 603 619 084 111	0.190 139 687 435 473	1.141 793 337 618 11
11	−0.910 979 740 129 467	0.225 757 763 293 348	3.168 918 837 262 51
12	−0.524 663 461 917 000	0.384 016 340 394 993	−0.195 128 933 922 791
13	0.607 570 230 546 519	14.140 586 059 086 1	−0.745 141 256 151 495
14	−0.680 906 032 910 047	−0.243 161 337 749 428	−0.168 782 986 070 569
15	−0.027 930 929 134 877 4	−0.983 755 618 469 951	−0.650 320 722 026 874
16	−0.306 095 450 725 083	−0.758 096 862 215 343	0.049 925 385 256 364 8
17	−1.635 154 501 878 91	−0.195 179 297 690 436	−0.148 800 334 381 005
18	−0.381 637 215 453 758	−0.745 340 802 908 329	−0.588 557 581 207 010
19	−11.542 706 853 817 8	−0.169 192 556 577 168	0.251 968 033 438 278
20	0.453 736 176 513 745	−0.651 666 721 038 030	0.542 354 574 874 772

We see that for none of the three initial conditions do the iterations converge, as attested by the occasional spikes (#19 for $x_0 = 0$; #13 for $x_0 = 2$; #11 for $x_0 = 1000$) and the oscillatory behavior.