AM205 HW4. PDEs, nonlinear equations, optimization. Solution

P1. Finite differences for advection

See solution code in [p1_fd.py].

(a) The central difference scheme satisfies the CFL condition for $|\nu| \leq 1$. The second-order upwind scheme satisfies the CFL condition for $0 \leq \nu \leq 2$.

(b) Assume that $u(x,t)$ is an exact solution of $u_t + cu_x = 0$. Taylor expansion of $u(x,t)$ about (x_j, t^n) gives

$$u\left(x_j, t^n + \Delta t\right) = u + u_t \Delta t + \frac{1}{2} u_{tt} \Delta t^2 + \text{h.o.t.} \tag{1}$$

$$u\left(x_j + \Delta x, t^n\right) = u + u_x \Delta x + \frac{1}{2} u_{xx} \Delta x^2 + \frac{1}{6} u_{xxx} \Delta x^3 + \text{h.o.t.} \tag{2}$$

$$u\left(x_j - \Delta x, t^n\right) = u - u_x \Delta x + \frac{1}{2} u_{xx} \Delta x^2 - \frac{1}{6} u_{xxx} \Delta x^3 + \text{h.o.t.} \tag{3}$$

$$u\left(x_j - 2\Delta x, t^n\right) = u - 2u_x \Delta x + 2u_{xx} \Delta x^2 - \frac{4}{3} u_{xxx} \Delta x^3 + \text{h.o.t.} \tag{4}$$

where (x_j, t^n) is omitted for brevity, e.g. u denotes $u(x_j, t^n)$.

For the central difference scheme, the truncation error is defined as

$$T_j^n = \frac{u(x_j, t^n + \Delta t) - u(x_j, t^n)}{\Delta t} + c \frac{u(x_j + \Delta x, t^n) - u(x_j - \Delta x, t^n)}{2\Delta x} \tag{5}$$

and takes the form

$$T_j^n = \frac{1}{2} u_{tt} \Delta t + \frac{c}{6} u_{xxx} \Delta x^2 + \text{h.o.t.} \tag{6}$$

i.e. $A_1 = 1/2$ and $A_2 = c/6$.

For the second-order upwind scheme, the truncation error is defined as

$$T_j^n = \frac{u(x_j, t^n + \Delta t) - u(x_j, t^n)}{\Delta t} + c \frac{3u(x_j, t^n) - 4u(x_j - \Delta x, t^n) + u(x_j - 2\Delta x, t^n)}{2\Delta x} \tag{7}$$

and takes the form

$$T_j^n = \frac{1}{2} u_{tt} \Delta t - \frac{c}{3} u_{xxx} \Delta x^2 + \text{h.o.t.} \tag{8}$$

i.e. $A_1 = 1/2$ and $A_2 = -c/3$.

(c) The second-order upwind scheme can be written as

$$u_j^{n+1} - u_j^n + \nu \frac{3u_j^n - 4u_{j-1}^n + u_{j-2}^n}{2} = 0 \tag{9}$$

To perform the stability analysis, we substitute the ansatz $u_j^n = \lambda^n e^{ijk\Delta x}$ into (9),

$$\lambda - 1 + \nu \left(\frac{3}{2} - 2e^{-ik\Delta x} + \frac{1}{2} e^{-2ik\Delta x} \right) = 0 \tag{10}$$

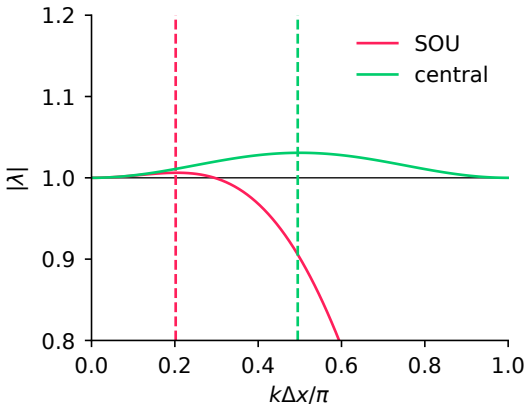
and therefore for the second-order upwind scheme

$$\lambda = 1 - \nu \left(\frac{3}{2} - 2e^{-ik\Delta x} + \frac{1}{2} e^{-2ik\Delta x} \right). \tag{11}$$

Similarly, for the central difference scheme

$$\lambda = 1 - i\nu \sin(k\Delta x). \tag{12}$$

- Setting $\nu = 0.25$, we get the following plot



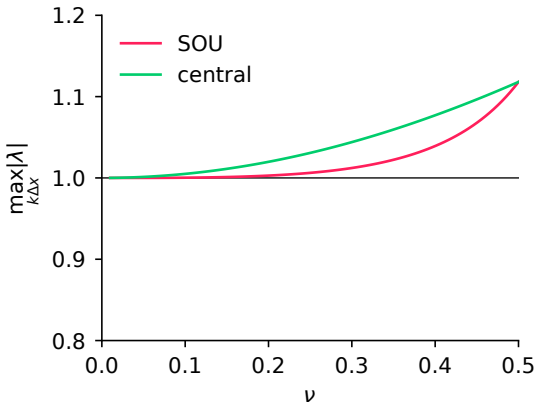
The second order upwind scheme (SOU) is stable for $k\Delta x > 0.30303\pi$, and its most unstable mode corresponds to

$$k\Delta x = 0.20202\pi, \quad |\lambda| = 1.00623. \tag{13}$$

The central difference scheme is unstable for all $k\Delta x$ (except for $k\Delta x = 0$ and $k\Delta x = \pi$), and its most unstable mode corresponds to

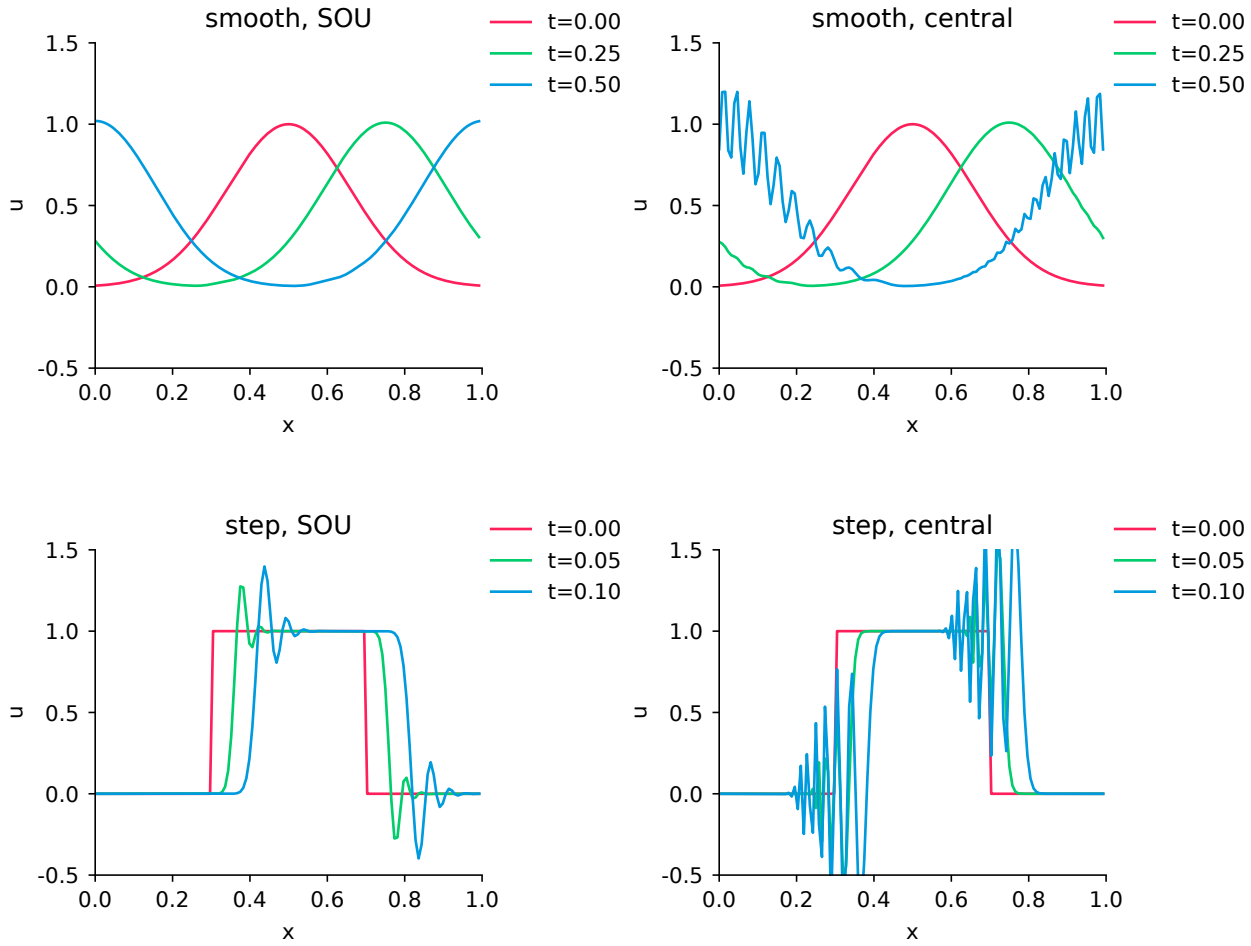
$$k\Delta x = 0.49495\pi, \quad |\lambda| = 1.03077. \tag{14}$$

- The plot of $\max_{k\Delta x} |\lambda|$ as a function of ν is below



Both schemes are unconditionally unstable for all values of $\nu > 0$, i.e. there is at least one mode which is amplified.

(d)



P2. Newton fractal

See solution code in [p2_fractal.py].

(a) Denote $\Delta z = z_{k+1} - z_k = \Delta x + i\Delta y$ and $\Delta \mathbf{x} = \mathbf{x}_{k+1} - \mathbf{x}_k = (\Delta x, \Delta y)$. We will show that both update rules are equivalent to the same linear system in terms of $(\Delta x, \Delta y)$. In the following, the functions are evaluated at $\mathbf{x}_k = (x_k, y_k)$ and z_k, and the arguments are omitted for brevity, e.g. $f = f(z_k)$. The update rule from $f(z)$ can be written as

$$(\Delta x + i\Delta y) f' = -f. \tag{15}$$

Since $f = u + iv$ and $f' = u_x + iv_x$,

$$(\Delta x + i\Delta y)(u_x + iv_x) = -u - iv. \tag{16}$$

After expanding and combining like terms,

$$u_x \Delta x - v_x \Delta y + i(v_x \Delta x + u_x \Delta y) = -u - iv, \tag{17}$$

which in matrix form is

$$\begin{pmatrix} u_x & -v_x \\ v_x & u_x \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = - \begin{pmatrix} u \\ v \end{pmatrix}. \tag{18}$$

On the other hand, the update rule from $\mathbf{g}(\mathbf{x})$ can be written as

$$\mathbf{J}_\mathbf{g} \Delta \mathbf{x} = -\mathbf{g}, \tag{19}$$

which in matrix form is

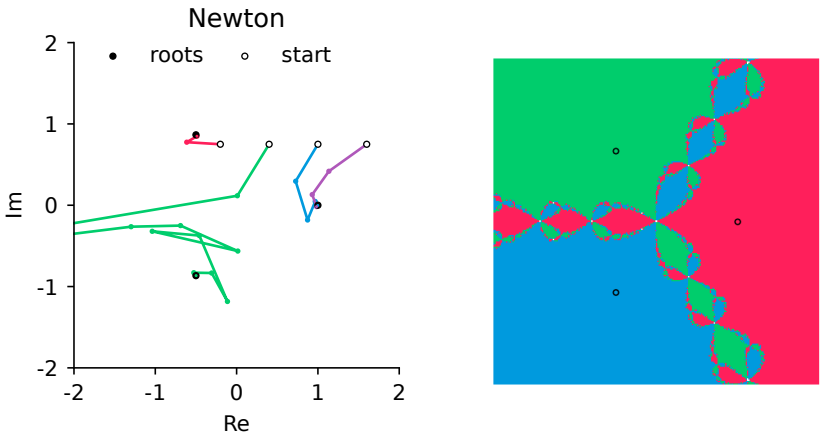
$$\begin{pmatrix} u_x & u_y \\ v_x & v_y \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = - \begin{pmatrix} u \\ v \end{pmatrix} \tag{20}$$

From the Cauchy-Riemann equations, the matrices in (18) and (20) coincide

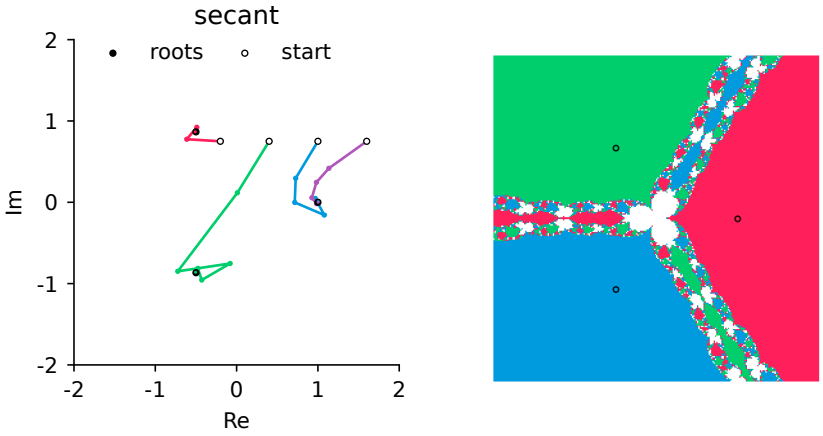
$$\begin{pmatrix} u_x & u_y \\ v_x & v_y \end{pmatrix} = \begin{pmatrix} u_x & -v_x \\ v_x & u_x \end{pmatrix}, \tag{21}$$

which shows that both linear systems are identical.

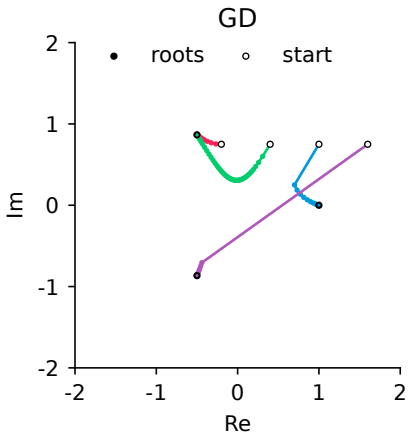
(b) Newton In this case, Newton's method converges from any starting point, with minor exceptions that would disappear for a larger threshold A. The updates in the trajectory starting from $0.4 + 0.75i$ appear to be larger in magnitude than with other methods.

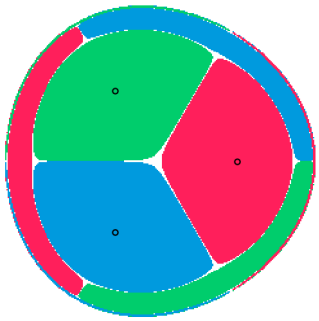


(c) Secant The secant method diverges for certain sets of starting points shown in white. The convergence behavior for three out of four starting points is qualitatively similar to that of Newton's method.



(d) Gradient descent (GD) Gradient descent tends to diverge for points that are sufficiently far from zero. This is consistent with the definition of $f(z) = z^3 - 1$ since its derivative $f'(z) = 3z^2$ determines the size of the first step and takes larger values away from zero. If it converges, the method needs to take many small steps to approach a root.





P3. Himmelblau's function

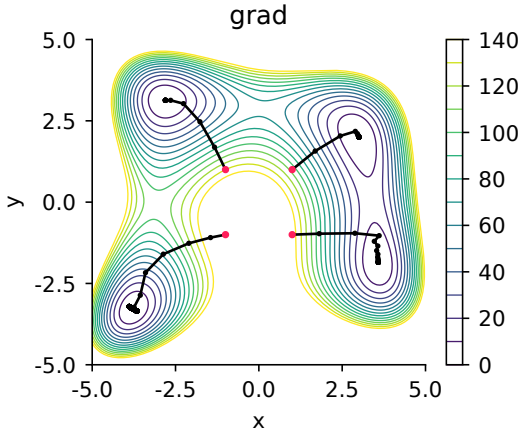
See solution code in [p3_opt.py]. The gradient of the function $f(x, y)$ is

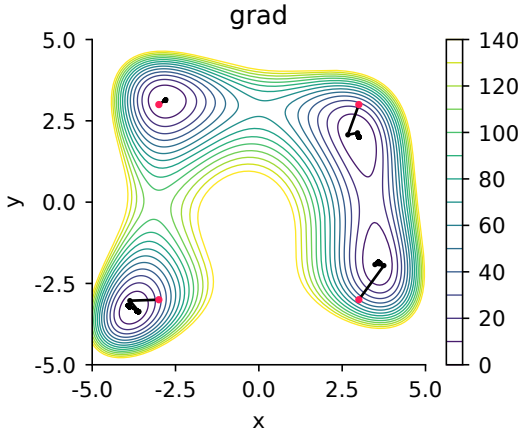
$$\nabla f(x,y) = \begin{pmatrix} 4x(x^2+y-11) + 2(x+y^2-7) \\ 2(x^2+y-11) + 4y(x+y^2-7) \end{pmatrix}, \tag{22}$$

and its Hessian is

$$H_f(x,y) = \begin{pmatrix} 2+8x^2+4(x^2+y-11) & 4x+4y \\ 4x+4y & 2+8y^2+4(x+y^2-7) \end{pmatrix}. \tag{23}$$

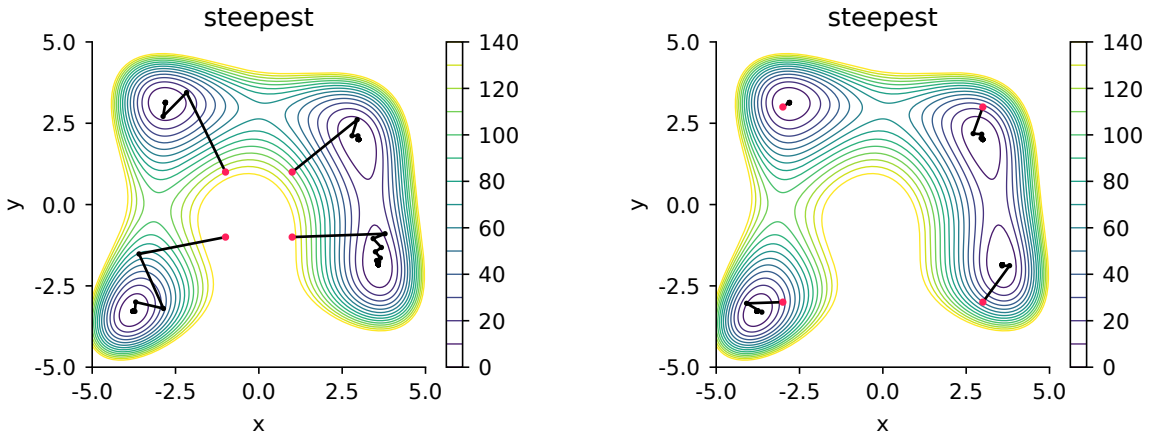
(a) Gradient descent with constant step factor The method converges to the global minima from six starting points and takes between 10 and 33 iterations. With two starting points, $(-1, -1)$ and $(-3, -3)$, the method fails to converge after 1000 iterations, although it approaches the minimum within 0.11.





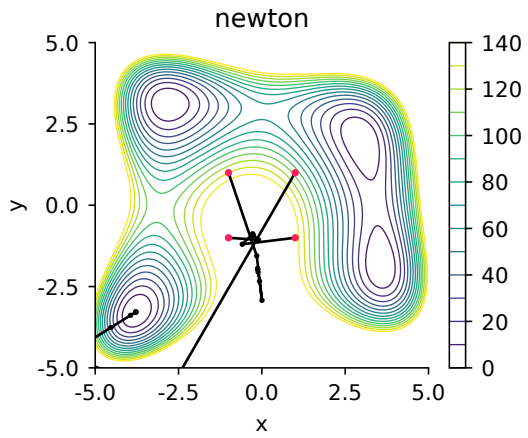
$\mathbf{x}_0$	k	$\mathbf{x}_k$	f	∇f
-1,-1	1000	-3.6634,-3.3476	1.1546	14.674,-9.123
-1,1	13	-2.8051,3.1313	2.6848e-15	-5.4813e-08,-6.5538e-07
1,-1	33	3.5844,-1.8481	3.4908e-13	3.266e-07,4.5075e-06
1,1	31	3,2	5.2529e-13	-1.9891e-06,4.8021e-06
-3,-3	1000	-3.6632,-3.3476	1.1573	14.691,-9.1333
-3,3	10	-2.8051,3.1313	9.3641e-15	-1.0237e-07,-1.224e-06
3,-3	29	3.5844,-1.8481	1.0094e-13	4.5669e-06,1.6886e-08
3,3	29	3,2	6.6652e-13	-2.2406e-06,5.4092e-06

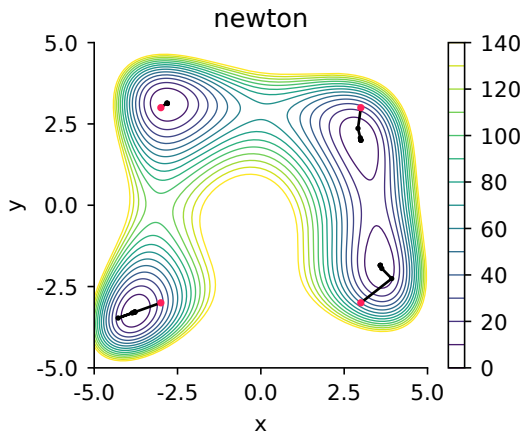
(b) Steepest descent The direction of steepest descent is $s = -\nabla f$. To perform the line search, we apply `scipy.optimize.line_search()` to the function $g(\alpha) = f(x_k + \alpha s)$. The method converges to the global minima from all eight starting points and takes between 8 and 29 iterations.



$\mathbf{x}_0$	k	$\mathbf{x}_k$	f	∇f
-1,-1	14	-3.7793,-3.2832	1.8856e-15	-4.1119e-07,5.3412e-07
-1,1	9	-2.8051,3.1313	4.3699e-15	6.8785e-07,3.5589e-07
1,-1	29	3.5844,-1.8481	2.3498e-13	1.9338e-06,3.6686e-06
1,1	12	3,2	5.3967e-17	-8.8949e-08,-2.9431e-08
-3,-3	15	-3.7793,-3.2832	1.2345e-14	1.5875e-06,1.0949e-07
-3,3	7	-2.8051,3.1313	7.7761e-15	8.0205e-07,6.9002e-07
3,-3	23	3.5844,-1.8481	1.4626e-13	2.0287e-06,-2.5692e-06
3,3	11	3,2	2.802e-14	2.0364e-06,5.4309e-07

(c) Newton's method The method converges from all eight starting points and takes between 4 and 8 iterations. However, in the cases $(-1,-1)$, $(-1,1)$, and $(1,-1)$ it converges to a maximum instead of a minimum. This is expected since Newton's method does not distinguish maxima and minima.





$\mathbf{x}_0$	k	$\mathbf{x}_k$	f	∇f
-1,-1	5	-0.27084,-0.92304	181.62	1.0658e-14,-7.4607e-14
-1,1	8	-0.12796,-1.9537	178.34	0,0
1,-1	6	-0.27084,-0.92304	181.62	1.7764e-15,-3.5527e-15
1,1	8	-3.7793,-3.2832	7.8886e-31	-1.7764e-15,1.1664e-14
-3,-3	6	-3.7793,-3.2832	7.8886e-31	1.7764e-15,-1.1664e-14
-3,3	4	-2.8051,3.1313	7.8886e-31	1.7764e-15,1.1125e-14
3,-3	6	3.5844,-1.8481	0	0,0
3,3	6	3,2	0	0,0