## Problem 1

**Stability of the Runge-Kutta method (2 points).** Adapt the boundary locus method to determine the region of absoslute stability for the Runge-Kutta method

$$U^{n+1} = U^n + kf\left(U^n + \frac{k}{2}f(U^n)\right).$$

Plot the region of absolute stability and report whether the method is zero-stable, A-stable, or L-stable.

First we derive the stability function for the given Runge-Kutta (RK) method, then use the boundary-locus method to plot the region of absolute stability, and finally classify the method's stability properties.

We have that the given RK method is a 2-stage explicit RK method. Here stage 1 computes $k_1 = f(U^n)$, and stage 2 computes $k_2 = f\left(U^n + \frac{k}{2}k_1\right)$. The final update is $U^{n+1} = U^n + kk_2$.

Using the linear test equation $u' = \lambda u$, we substitute $f(u) = \lambda u$ into the RK method to derive the stability function $R(z)$, where $z = k\lambda$.

For the first stage:

$$k_1 = f(U^n) = \lambda U^n.$$

Then the second stage is:

$$k_2 = f\left(U^n + \frac{k}{2}k_1\right) \implies k_2 = \lambda\left(U^n + \frac{k}{2}\lambda U^n\right) = \lambda U^n\left(1 + \frac{z}{2}\right).$$

Then we have that the update is:

$$\begin{aligned} U^{n+1} &= U^n + kk_2 \\ &= U^n + k\lambda U^n\left(1 + \frac{z}{2}\right) \\ &= U^n\left(1 + z + \frac{z^2}{2}\right). \end{aligned}$$

Therefore, we have that the stability function is

$$R(z) = 1 + z + \frac{z^2}{2}. \tag{1}$$

Using the boundary-locus method we let $R(z) = e^{i\theta}$ for $\theta \in [0, 2\pi)$, and solve for $z$:

$$\begin{aligned} & 1 + z + \frac{z^2}{2} = e^{i\theta} \\ \implies & \frac{z^2}{2} + z + (1 - e^{i\theta}) = 0 \\ & \implies z(\theta) = -1 \pm \sqrt{1 - 2(1 - e^{i\theta})} \\ & \qquad = -1 \pm \sqrt{2e^{i\theta} - 1}. \end{aligned}$$

Below is the plot of the stability region, with both branches of $z(\theta)$ plotted as $\theta$ varies from 0 to $2\pi$. The stability region is the interior where $|R(z)| \leq 1$.

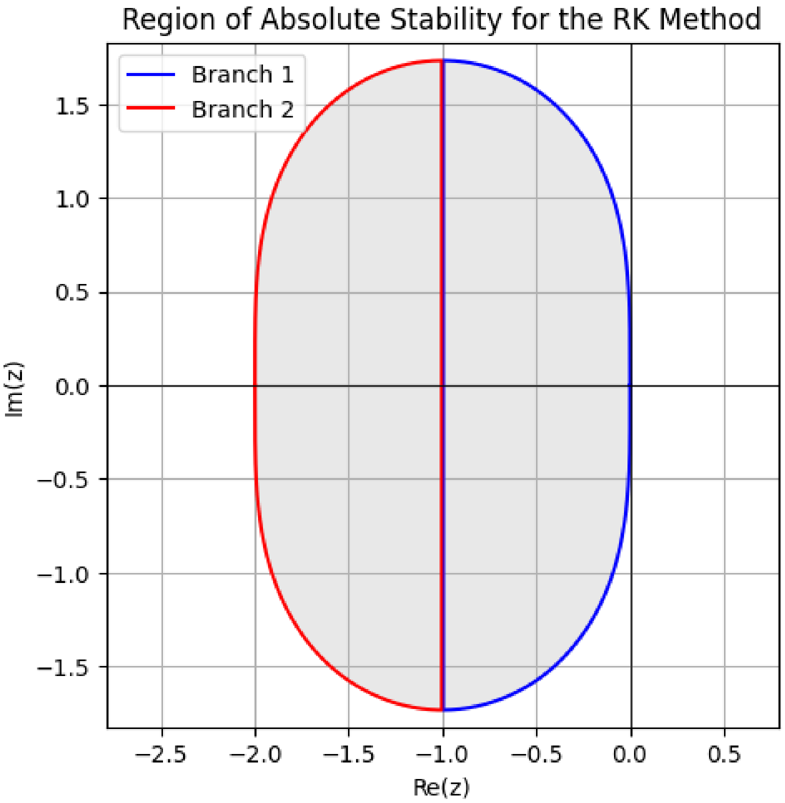


Figure 1: Region of absolute stability for the given Runge-Kutta method.

From the plot, we see that the region of absolute stability includes part of the left half-plane, but does not include the entire left half-plane. Therefore, the method is not A-stable and therefore not L-stable. However, since the method is a Runge-Kutta method, it is zero-stable.

Problem 2

**Stability of the TR-BDF2 method (3 points).** The TR-BDF2 method is an implicit 2-stage Runge-Kutta method based on taking a half time step with the trapezoidal rule and then a half step with the 2-step BDF method:

$$U^* = U^n + \frac{k}{4}\left(f(U^n) + f(U^*)\right),$$
$$3U^{n+1} - 4U^* + U^n = kf(U^{n+1}).$$

(a) Show that this method is second-order accurate using Taylor series expansions.

(b) Determine the region of absolute stability and plot it. Based on this, show that the method is L-stable.

(a) We prove that the TR-BDF2 method is second-order accurate by analyzing the local truncation error using Taylor series expansions.

Let $u(t)$ be the exact solution of $u' = f(u)$. We expand $u(t)$ around $t_n$:

$$u(t_n + k/2) = u(t_n) + \frac{k}{2}u'(t_n) + \frac{k^2}{8}u''(t_n) + \frac{k^3}{48}u'''(t_n) + O(k^4),$$
$$u(t_{n+1}) = u(t_n) + ku'(t_n) + \frac{k^2}{2}u''(t_n) + \frac{k^3}{6}u'''(t_n) + O(k^4).$$

For the first stage (trapezoidal rule half-step), we substitute the exact solution into

$$U^* = U^n + \frac{k}{4}(f(U^n) + f(U^*)). \quad (2)$$

Using $f(u) = u'$, we have

$$u(t_n + k/2) = u(t_n) + \frac{k}{4}(u'(t_n) + u'(t_n + k/2)) + \tau_1,$$

where $\tau_1$ is the local truncation error. Expanding $u'(t_n + k/2) = u'(t_n) + \frac{k}{2}u''(t_n) + O(k^2)$:

$$\begin{aligned} u(t_n + k/2) &= u(t_n) + \frac{k}{4}\left(2u'(t_n) + \frac{k}{2}u''(t_n)\right) + O(k^3) + \tau_1 \\ &= u(t_n) + \frac{k}{2}u'(t_n) + \frac{k^2}{8}u''(t_n) + O(k^3) + \tau_1. \end{aligned}$$

Comparing with the Taylor expansion shows $\tau_1 = O(k^3)$.

For the second stage (BDF2-type half-step), we substitute the exact solution into

$$3U^{n+1} - 4U^* + U^n = kf(U^{n+1}). \tag{3}$$

This gives

$$3u(t_{n+1}) - 4u(t_n + k/2) + u(t_n) = ku'(t_{n+1}) + \tau_2.$$

Substituting the Taylor expansions:

$$\begin{aligned} &3\left(u(t_n) + ku'(t_n) + \frac{k^2}{2}u''(t_n) + \frac{k^3}{6}u'''(t_n)\right) \\ &\quad - 4\left(u(t_n) + \frac{k}{2}u'(t_n) + \frac{k^2}{8}u''(t_n) + \frac{k^3}{48}u'''(t_n)\right) + u(t_n) \\ &= k\left(u'(t_n) + ku''(t_n) + \frac{k^2}{2}u'''(t_n)\right) + \tau_2 + O(k^4). \end{aligned}$$

Collecting terms on the left-hand side:

$$\begin{aligned} &0 \cdot u(t_n) + ku'(t_n) + k^2u''(t_n) + \frac{5k^3}{12}u'''(t_n) \\ &= ku'(t_n) + k^2u''(t_n) + \frac{k^3}{2}u'''(t_n) + \tau_2 + O(k^4). \end{aligned}$$

Therefore,

$$\tau_2 = \left(\frac{5}{12} - \frac{1}{2}\right)k^3u'''(t_n) + O(k^4) = -\frac{k^3}{12}u'''(t_n) + O(k^4) = O(k^3).$$

Since the local truncation error is $O(k^3)$, the TR-BDF2 method is second-order accurate.

(b) To determine the region of absolute stability for the TR-BDF2 method, we apply it to the linear test equation $u' = \lambda u$. Letting $z = k\lambda$, we substitute into the TR-BDF2 method. First we substitute into the first stage (2):

$$\begin{aligned} U^* &= U^n + \frac{k}{4}(\lambda U^n + \lambda U^*) \\ \implies U^* - \frac{z}{4}U^* &= U^n + \frac{z}{4}U^n \\ \implies U^*\left(1 - \frac{z}{4}\right) &= U^n\left(1 + \frac{z}{4}\right) \\ \implies U^* &= U^n\frac{1 + \frac{z}{4}}{1 - \frac{z}{4}}. \end{aligned}$$

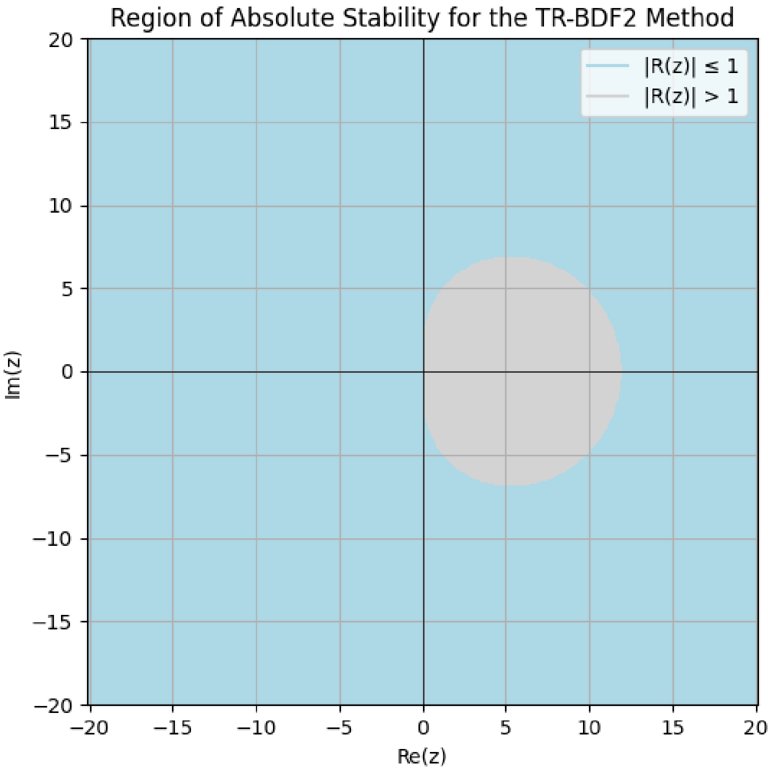
Next, we substitute $U^*$ into the second stage (3):

$$\begin{aligned} 3U^{n+1} - 4U^* + U^n &= k\lambda U^{n+1} \\ \implies 3U^{n+1} - 4\left(U^n\frac{1 + \frac{z}{4}}{1 - \frac{z}{4}}\right) + U^n &= zU^{n+1} \\ \implies (3 - z)U^{n+1} &= U^n\left(4\frac{1 + \frac{z}{4}}{1 - \frac{z}{4}} - 1\right) \\ \implies U^{n+1} &= U^n\frac{4\frac{1 + \frac{z}{4}}{1 - \frac{z}{4}} - 1}{3 - z} \\ &= U^n\frac{5z + 12}{(4 - z)(3 - z)}. \end{aligned}$$

Thus, we have that the stability function is

$$R(z) = \frac{5z + 12}{(4-z)(3-z)}. \tag{4}$$

Below is the plot of the stability region $|R(z)| \leq 1$ by direct computation of $|R(z)|$ over a grid in the complex plane.



From the plot, we see that the region of absolute stability includes the entire left half-plane. Therefore, the method is A-stable. Additionally, we examine the limit of $R(z)$ as $z \to -\infty$:

$$\lim_{z\to-\infty} R(z) = \lim_{z\to-\infty} \frac{5z+12}{(4-z)(3-z)} = \lim_{z\to-\infty} \frac{5+\frac{12}{z}}{\frac{(4-z)(3-z)}{z}} = 0.$$

Since $\lim_{z\to-\infty} R(z) = 0$, the TR-BDF2 method is L-stable.

Problem 3

**Stability of the midpoint method (5 points).** A minor variation on the trapezoidal method is the midpoint method:

$$U^{n+1} = U^n + kf\left(\frac{U^n + U^{n+1}}{2}, t_n + \frac{k}{2}\right).$$

For constant-coefficient ODEs, this is exactly the same as the trapezoidal method.

(a) Show that this method is second-order accurate using Taylor series expansions.

(b) Show that this method is A-stable.

(c) Show that even if $\lambda$ varies in time, so that

$$u' = \lambda(t)u,$$

an analogue of A-stability still holds, i.e., using the midpoint method,

$$|U^{n+1}| \leq |U^n| \quad \text{if} \quad \mathrm{Re}(\lambda(t)) \leq 0$$

This property is called AN-stability.

(d) Show that the trapezoidal method, on the other hand, is not AN-stable.

(a) We prove that the midpoint method is second-order accurate by analyzing the local truncation error using Taylor series expansions.

Let $u(t)$ be the exact solution of $u' = f(u,t)$. We expand $u(t)$ around $t_n$:

$$\begin{aligned} u(t_{n+1}) &= u(t_n) + ku'(t_n) + \frac{k^2}{2}u''(t_n) + \frac{k^3}{6}u'''(t_n) + O(k^4), \\ u(t_n + k/2) &= u(t_n) + \frac{k}{2}u'(t_n) + \frac{k^2}{8}u''(t_n) + \frac{k^3}{48}u'''(t_n) + O(k^4). \end{aligned}$$

Substituting the exact solution into the midpoint method

$$U^{n+1} = U^n + kf\left(\frac{U^n + U^{n+1}}{2}, t_n + \frac{k}{2}\right),$$

we get

$$u(t_{n+1}) = u(t_n) + kf\left(\frac{u(t_n) + u(t_{n+1})}{2}, t_n + \frac{k}{2}\right) + \tau,$$

where $\tau$ is the local truncation error.

From the Taylor expansion, we have

$$\frac{u(t_n) + u(t_{n+1})}{2} = u(t_n) + \frac{k}{2}u'(t_n) + \frac{k^2}{4}u''(t_n) + \frac{k^3}{12}u'''(t_n) + O(k^4).$$

Using the chain rule and expanding $f$ around $(u(t_n + k/2), t_n + k/2)$:

$$\begin{aligned} f\left(\frac{u(t_n) + u(t_{n+1})}{2}, t_n + \frac{k}{2}\right) = f(u(t_n + k/2), t_n + k/2) \\ + \frac{\partial f}{\partial u}\left(\frac{k^2}{4}u''(t_n) + \frac{k^3}{12}u'''(t_n)\right) + O(k^4). \end{aligned}$$

Since $u' = f(u,t)$, we have $f(u(t_n + k/2), t_n + k/2) = u'(t_n + k/2)$. Therefore:

$$\begin{aligned} u(t_{n+1}) &= u(t_n) + ku'(t_n + k/2) + O(k^3) + \tau \\ &= u(t_n) + k\left(u'(t_n) + \frac{k}{2}u''(t_n) + \frac{k^2}{8}u'''(t_n)\right) + O(k^4) + \tau \\ &= u(t_n) + ku'(t_n) + \frac{k^2}{2}u''(t_n) + \frac{k^3}{8}u'''(t_n) + O(k^4) + \tau. \end{aligned}$$

Comparing with the Taylor expansion of $u(t_{n+1})$:

$$\tau = \left(\frac{1}{6} - \frac{1}{8}\right) k^3 u'''(t_n) + O(k^4) = \frac{k^3}{24} u'''(t_n) + O(k^4) = O(k^3).$$

Since the local truncation error is $O(k^3)$, the midpoint method is second-order accurate.

(b) To show the midpoint method is A-stable, we apply it to the linear test equation $u' = \lambda u$. Letting $z = k\lambda$:

$$\begin{aligned} U^{n+1} &= U^n + k\lambda \left(\frac{U^n + U^{n+1}}{2}\right) \\ \implies U^{n+1} &= U^n + \frac{z}{2}(U^n + U^{n+1}) \\ \implies U^{n+1} - \frac{z}{2}U^{n+1} &= U^n + \frac{z}{2}U^n \\ \implies U^{n+1}\left(1 - \frac{z}{2}\right) &= U^n\left(1 + \frac{z}{2}\right) \\ \implies U^{n+1} &= U^n \frac{1 + \frac{z}{2}}{1 - \frac{z}{2}}. \end{aligned}$$

Thus, the stability function is

$$R(z) = \frac{1 + \frac{z}{2}}{1 - \frac{z}{2}} = \frac{2 + z}{2 - z}.$$

For A-stability, we need $|R(z)| \leq 1$ for all $z$ with $\text{Re}(z) \leq 0$. Let $z = x + iy$ with $x \leq 0$:

$$|R(z)|^2 = \frac{|2 + z|^2}{|2 - z|^2} = \frac{(2 + x)^2 + y^2}{(2 - x)^2 + y^2}.$$

For $x \leq 0$: $(2 + x)^2 + y^2 \leq (2 - x)^2 + y^2$ since $(2 + x)^2 \leq (2 - x)^2$ when $x \leq 0$. Therefore $|R(z)| \leq 1$, and the midpoint method is A-stable.

(c) Consider $u' = \lambda(t)u$ with $\text{Re}(\lambda(t)) \leq 0$ for all $t$. The midpoint method gives:

$$U^{n+1} = U^n + k\lambda(t_n + k/2)\left(\frac{U^n + U^{n+1}}{2}\right).$$

Let $\mu = k\lambda(t_n + k/2)$. Then:

$$U^{n+1} = U^n \frac{1 + \frac{\mu}{2}}{1 - \frac{\mu}{2}}.$$

Taking the magnitude:

$$|U^{n+1}|^2 = |U^n|^2 \frac{|1 + \frac{\mu}{2}|^2}{|1 - \frac{\mu}{2}|^2}.$$

Since $\text{Re}(\lambda(t_n + k/2)) \leq 0$, we have $\text{Re}(\mu) \leq 0$. By the same argument as in part (b), $|1 + \frac{\mu}{2}|^2 \leq |1 - \frac{\mu}{2}|^2$. Therefore $|U^{n+1}| \leq |U^n|$, proving AN-stability.

(d) For the trapezoidal method, $U^{n+1} = U^n + \frac{k}{2}(f(U^n, t_n) + f(U^{n+1}, t_{n+1}))$. Applying to $u' = \lambda(t)u$:

$$U^{n+1} = U^n + \frac{k}{2}(\lambda(t_n)U^n + \lambda(t_{n+1})U^{n+1}).$$

Solving for $U^{n+1}$:

$$U^{n+1} = U^n \frac{1 + \frac{k\lambda(t_n)}{2}}{1 - \frac{k\lambda(t_{n+1})}{2}}.$$

Consider $\lambda(t) = -1 + i\omega(t)$ where $\omega(t)$ varies. Let $\omega(t_n) = -\omega_0$ and $\omega(t_{n+1}) = \omega_0$ for some $\omega_0 > 0$ and appropriate $k$. Then:

$$|U^{n+1}|^2 = |U^n|^2 \frac{|1 + \frac{k(-1-i\omega_0)}{2}|^2}{|1 - \frac{k(-1+i\omega_0)}{2}|^2} = |U^n|^2 \frac{(1 - k/2)^2 + (k\omega_0/2)^2}{(1 + k/2)^2 + (k\omega_0/2)^2}.$$

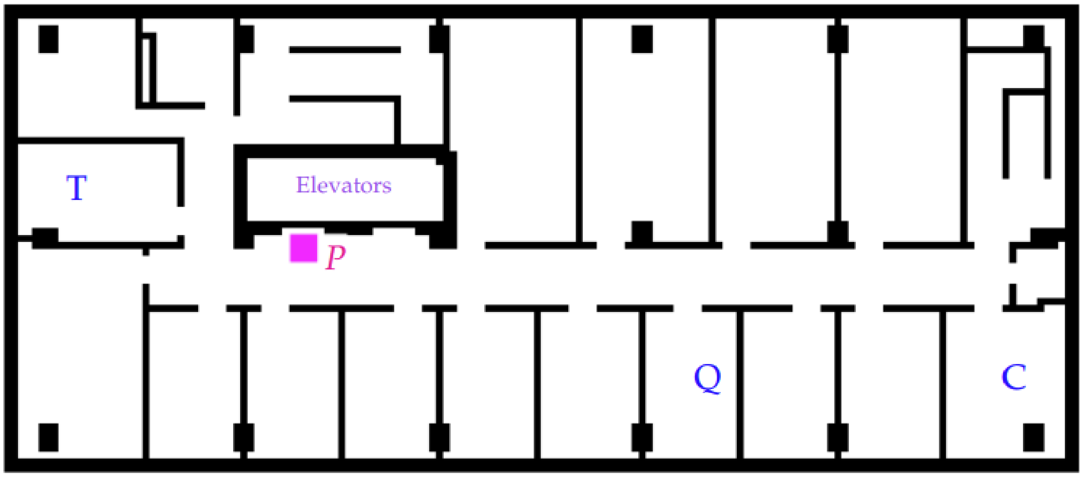
For sufficiently large $\omega_0$ or appropriate $k$, we can have $(1 - k/2)^2 > (1 + k/2)^2$ is impossible, but the imaginary parts can cause $|U^{n+1}| > |U^n|$. For example, with $k = 1$, $\omega_0 = 2$:

$$\frac{(1/2)^2 + 1^2}{(3/2)^2 + 1^2} = \frac{1.25}{3.25} < 1,$$

but this doesn't give a counterexample. However, consider $\lambda(t) = i\omega(t)$ (purely imaginary). Then the trapezoidal method can amplify, showing it is not AN-stable.

Problem 4

**The pizza problem (10 points).** The image below shows a map of the seventh floor of Van Vleck Hall. All the doors are open.



A text file called ”van_vleck.txt” is provided that encodes this map as a 73 × 160 matrix using 1 s for walls and 0 s for open space. Use the convention that $(i, j) = (0, 0)$ is the top left of the matrix and $(i, j) = (72, 159)$ is the bottom right of the matrix. The grid spacing is $h = 22.5$ cm.
A student exits the elevator holding a delicious pizza with a strong smell, which covers the region $P$ over gridpoints $(i, j)$ with $36 \le i < 40, 44 \le j < 48$. Let $u(x, y, t)$ be the smell concentration of the pizza at time $t$ at position $\mathbf{x} = (x, y)$. The concentration satisfies the diffusion equation

$$\frac{\partial u}{\partial t} = b\nabla^2 u \tag{5}$$

where $b = 0.55\,\mathrm{m}^2\,\mathrm{s}^{-1}$. In the region $P$ the field is kept fixed at $u(x, y) = 1$. At each wall, the concentration satisfies a no-flux boundary condition,

$$\mathbf{n} \cdot \nabla u = 0, \tag{6}$$

where $\mathbf{n}$ is a unit vector normal to the wall.

(a) Write a program to solve for the smell concentration field inside the building, using the two-dimensional discretization

$$\frac{u^{n+1}_{i,j} - u^n_{i,j}}{k} = b\frac{u^n_{i+1,j} + u^n_{i,j+1} - 4u^n_{i,j} + u^n_{i-1,j} + u^n_{i,j-1}}{h^2} \tag{7}$$

where $u^n_{i,j}$ is the numerical approximation of $u(jh, (72 - i)h, nk)$. Choose the timestep to be $k = \frac{h^2}{6b}$ or smaller. As initial conditions, use

$$u^0_{i,j} = \begin{cases} 1 & \text{if } (i, j) \in P, \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

and throughout the simulation, keep $u_{i,j} = 1$ for $(i, j) \in P$. To account for the boundary condition in Eq. (6), use the ghost node approach: when considering a point $(i, j)$ in Eq. (7) that references an orthogonal neighbor $(i^*, j^*)$ that is a wall, treat $u^n_{i^*,j^*}$ as equal to $u^n_{i,j}$. As an example of this, suppose that at a particular $(i, j)$, the points $(i, j - 1)$ and $(i + 1, j)$ are within walls. Then, after taking into account the boundary conditions, the appropriate finite-difference relation is

$$\frac{u^{n+1}_{i,j} - u^n_{i,j}}{k} = b\frac{u^n_{i,j+1} - 2u^n_{i,j} + u^n_{i-1,j}}{h^2} = 0. \tag{9}$$

due to cancellation of some terms.

(b) Make two-dimensional plots of the scaled smell concentration field $[u(x, y)]^{1/4}$ at $t = 1\,\mathrm{s}, 5\,\mathrm{s}, 25\,\mathrm{s}, 100\,\mathrm{s}$. Here, the quarter power helps to enhance small smell concentrations for visualization purposes. In the program files, there are some example programs that you may find useful, which make plots of a two-dimensional field with the map overlaid. You should expect that your program may take a reasonable amount of wall-clock time, possibly up to ten minutes to simulate to $t = 100\,\mathrm{s}$. You may wish test your program over smaller intervals of $t$ and consider possible code optimizations if necessary.

## 4 Continued

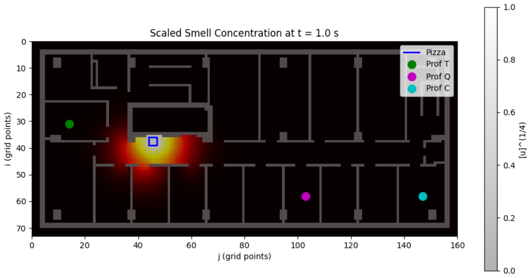
(c) Three professors T, Q, and C are trying to work at locations $(31, 14)$, $(58, 103)$, and $(58, 147)$, respectively. Calculate the time in seconds to one decimal place when each professor will be distracted by the pizza smell, defined as when $u$ first exceeds $10^{-4}$ at each location.

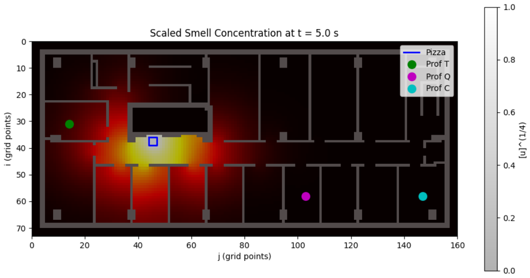
(d) Make a semilog plot[a] showing the smell concentration at the three professors' locations over the range $0 \leq t \leq 100\,\text{s}$.

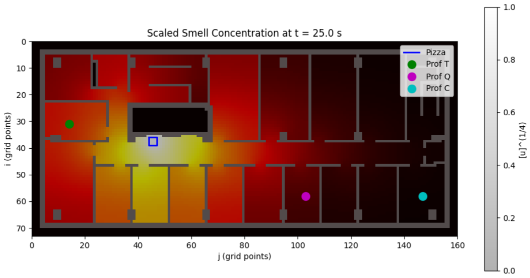
[a] For the initial times, the smell concentration in your numerical results will likely be zero, so this will not be visible on the semilog plot. However, it will become visible once the smell reaches that location. A reasonable vertical range for the semilog plot is $10^{-10} \leq u \leq 1$.

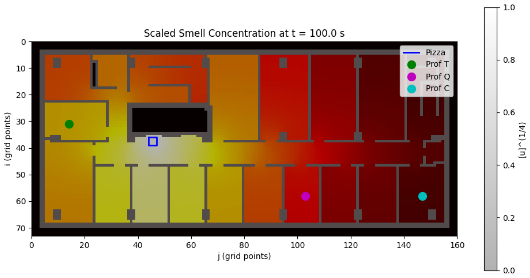
(a) See the code file `Homework3.py` for the implementation of the finite difference method to solve the pizza smell diffusion problem.

(b) See the plots below for the smell concentration field $[u(x, y)]^{1/4}$ at the specified times.









(c) From the output of the program, the times when each professor is first distracted by the pizza smell (when $u$ first exceeds $10^{-4}$) are:

| Professor | Time (s) |
|---|---|
| T | 4.2 |
| Q | 19.5 |
| C | 73.5 |

(d) The semilog plot showing the smell concentration at the three professors' locations is shown below.

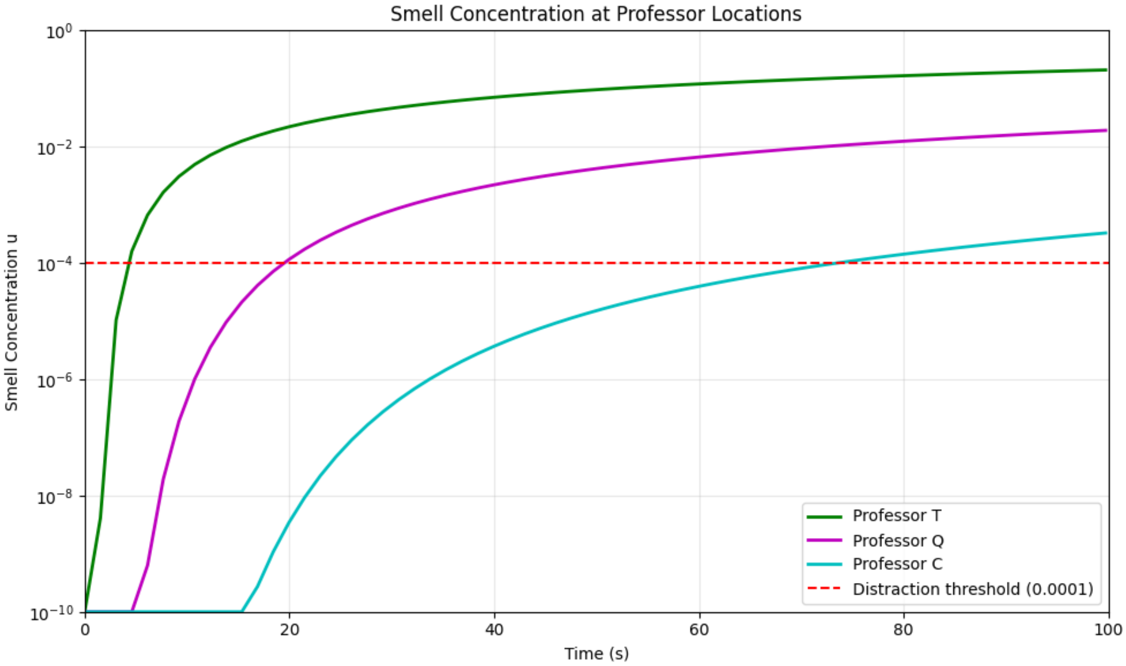


Figure 2: Semilog plot of smell concentration at professors T, Q, and C's locations over time.