2D Advection-Diffusion Equation with Robin BC

1 Introduction

In this exercise we will consider an advection-diffusion equation in 2D and approximate solutions to it using the explicit upwind scheme to approximate the advection term, and the implicit scheme to approximate the diffusion term. We will consider only rectangular domains, and determine the order of accuracy of this method by using an example equation with a known exact solution. Finally, we will simulate the spread of oil in the ocean after an oil spill and determine the appropriate times to close beaches affected by the oil spill. We will take snapshots of the oil concentration at various times to illustrate the spread of oil over time in the water.

2 Part a

Consider a generalized advection diffusion problem in a rectangular domain $\Omega = [x_l; x_r] \times [y_b; y_t]$

$$\begin{cases} \dfrac{\partial c}{\partial t} + v \cdot \nabla c = D\Delta c + f(t,x,y), \ \forall \text{ (x,y)} \in \Omega \\ D\dfrac{\partial c}{\partial y} - v_y c = g(t,x,y) \ \forall \text{ (x,y)} \mid \text{y} = y_b \\ \quad c(t,x,y) = c_{bc}(t,x,y) \ \forall \text{ (x,y)} \mid \text{x} = x_l \text{ or x} = x_r \text{ or y}=y_t \\ c(t_{start},x,y) = c_{start}(x,y), \ \forall \text{ (x,y)} \in \Omega \end{cases} \tag{1}$$

Where the source term, Dirichlet boundary condition, and Robin boundary condition functions are given by f(t,x,y), $c_{bc}(t,x,y)$, $g(t,x,y)$. Initial conditions given by $c_{start} = c_{start}(x,y)$. D is the diffusion coefficient. We then discretize in space and time appropriately before we apply the explicit upwind scheme to the advection term, and the implicit scheme to the diffusion term. We thus obtain:

$$\tfrac{\partial c}{\partial t}(t_n,x_i,y_j) + (v_x\tfrac{\partial c}{\partial x} + v_y\tfrac{\partial c}{\partial y}) = D\tfrac{\partial^2 c}{\partial x^2}(t_n,x_i,y_j) + D\tfrac{\partial^2 c}{\partial y^2}(t_n,x_i,y_j) + f(t_n,x_i,y_j)$$

where the spacial second-derivatives are approximated by the central-difference approximation, and the time derivative and first-derivative in space is approximated and using the backward difference method:

$$\tfrac{\partial^2 c}{\partial x^2}(t_{n+1},x_i,y_j) \approx \tfrac{c^{n+1}_{i+1,j} - 2c^{n+1}_{i,j} + c^{n+1}_{i-1,j}}{\Delta x^2}$$

$$\frac{\partial^2 c}{\partial y^2}(t_{n+1}, x_i, y_j) \approx \frac{c_{i,j+1}^{n+1} - 2c_{i,j}^{n+1} + c_{i,j-1}^{n+1}}{\Delta y^2}$$

$$\frac{\partial c}{\partial t}(t_n, x_i, y_j) \approx \frac{c_{i,j}^{n+1} - c_{i,j}^{n}}{\Delta t} \tag{2}$$

$$(v_x \frac{\partial c}{\partial x} + v_y \frac{\partial c}{\partial y}) \approx v_x(\frac{c_{i+1,j}^{n} - c_{i,j}^{n}}{\Delta x}) + v_y(\frac{c_{i,j+1}^{n} - c_{i,j}^{n}}{\Delta y}) \tag{3}$$

We thus obtain the approximation scheme:

$$\frac{c_{i,j}^{n+1} - c_{i,j}^{n}}{\Delta t} + v_x(\frac{c_{i+1,j}^{n} - c_{i,j}^{n}}{\Delta x}) + v_y(\frac{c_{i,j+1}^{n} - c_{i,j}^{n}}{\Delta y}) = D\frac{c_{i+1,j}^{n+1} - 2c_{i,j}^{n+1} + c_{i-1,j}^{n+1}}{\Delta x^2} + D\frac{c_{i,j+1}^{n+1} - 2c_{i,j}^{n+1} + c_{i,j-1}^{n+1}}{\Delta y^2} + f(t_{n+1}, x_i, y_j)$$

This equation holds for all internal grid points of the domain. On the bottom boundary we apply Robin boundary conditions, since the bottom of the domain must satisfy the no-flux condition, and everywhere else (including the bottom left and right corners) of the domain, we apply the Dirichlet boundary condition function.

Because our solution at time t = t_{n+1} relies on knowing the solution at those times, it is necessary to create a system of equations and solve all $N_x N_y$ equations simultaneously. That is, we must solve the matrix equation Ax = b, where x is a column vector of the grid points at time t_{n+1}, c_s^{n+1}, where the index s is given by s = (j-1)N_x + i $\forall(i,j)$. The s^{th} row of A and b is constructed differently depending on whether the s^{th} grid point is an internal grid point, or on the boundary.

For internal grid points:

$$\begin{cases} A_{s,s} = 1 + 2D\Delta t(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}) \\ A_{s,s+1} = A_{s,s-1} = -\frac{D\Delta t}{\Delta x^2} \\ A_{s,s+N_x} = A_{s,s-N_x} = -\frac{D\Delta t}{\Delta y^2} \\ b_s = c_{i,j}^{n} + \Delta t f(t_{n+1}, x_i, y_j) - v_x \Delta t \frac{c_{i+1,j}^{n} - c_{i,j}^{n}}{\Delta x} - v_y \Delta t \frac{c_{i,j+1}^{n} - c_{i,j}^{n}}{\Delta y} \end{cases} \tag{4}$$

For bottom boundary grid points (excluding the corners of the domain), the Robin BC apply and we have:

$$\begin{cases} A_{s,s} = 1 + 2D\Delta t(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}) + \frac{2v_y \Delta t}{\Delta y} \\ A_{s,s+1} = A_{s,s-1} = -\frac{D\Delta t}{\Delta x^2} \\ A_{s,s+N_x} = -2\frac{D\Delta t}{\Delta y^2} \\ b_s = c_{i,j}^{n} + \Delta t f(t_{n+1}, x_i, y_j) - v_x \Delta t \frac{c_{i+1,j}^{n} - c_{i,j}^{n}}{\Delta x} - v_y \Delta t \frac{c_{i,j+1}^{n} - c_{i,j}^{n}}{\Delta y} - \frac{2\Delta t}{\Delta y} g(t_{n+1}, x_i, y_1) \end{cases} \tag{5}$$

Finally, for the left, right, and top boundary conditions, (including the bottom left and bottom right corners of the domain), the Dirichlet boundary conditions apply. Thus we have:

$$\begin{cases} A_{s,s} = 1 \\ b_s = c_{bc}(t_{n+1}, x_i, y_j) \end{cases} \tag{6}$$

We have so far defined values for all rows of the column vectors x and b, however A has not been defined completely. Let $A_{s,k}$ be all values (s,k) not yet defined in the matrix A. We set all of these values as $A_{s,k} = 0$ (zero everywhere else). Since A will be an $N_x N_y \times N_x N_y$ sized matrix, it will quickly grow to a very large size as grid resolution is increased. Therefore, in our MATLAB implementation of this scheme, we will have to use a sparse matrix. The sparse matrix will only save the non-zero values of the matrix in memory, while every other point in the matrix is assumed to be 0. Our matrix will contain vastly more zero than non-zero elements, and thus this will save us a considerable amount of memory and improve performance.

3 Part b

We now consider the example problem parameters:

$$\begin{cases} \Omega = [-1, 3] \times [-1.5, 1.5] \\ D = 0.7 \\ v_x = -0.8 \\ v_y = -0.4 \\ c_{exact}(t, x, y) = sin(x)cos(y)exp(-t) \end{cases} \tag{7}$$

We will use $t_{start} = 0$, $t_{final} = 1$, $(N_x, N_y) =$ (20,15),(40,30),(80,60),(160,120), $\Delta t = \Delta x/2$. with boundary conditions $c_{bc}(t, x, y)$, initial conditions $c_{start}(x, y)$, Robin boundary conditions $g(t, x, y)$, and source term f(t,x,y) calculated by plugging c_{exact} into the advection-diffusion equation. We thus obtain:

$$\begin{cases} c_{bc} = c_{exact}(t, x, y) \\ c_{start} = c_{exact}(0, x, y) = sin(x)cos(y) \\ g(t, x, y) = exp(-t)(-\lambda sin(y)sin(x) - v_y cos(y)sin(x)) \\ f(t, x, y) = exp(-t)(-cos(y)sin(x) + v_x cos(y)cos(x) - v_y sin(y)sin(x) + 2\lambda cos(y)sin(x)) \end{cases} \tag{8}$$

We will implement this in MATLAB, and then using the obtained data at t = t_{final}, calculate the error of each grid resolution as: $max(|c_{i,j} - c_{exact}(t_{final}, x_i, y_j)|)$. Then, we calculate the order of accuracy as:

$$\left| \frac{log\left(\frac{error(trial_m)}{error(trial_{m+1})}\right)}{log\left(\frac{N_x(trial_{m+1})}{N_x(trial_m)}\right)} \right|$$

Where $N_x(trial_m)$ is the grid resolution in the x dimension for the m^{th} trial. In this case, the denominator always corresponds to log(2).

(Nx, Ny)	Error	Order k
(20, 15)	0.024041	-
(40, 30)	0.011985	1.004237
(80, 60)	0.005985	1.001925
(160, 120)	0.002992	1.000372

Table 1: Accuracy of the scheme implemented in MATLAB

The method is therefore first-order accurate in space.

4 Part c

We now apply this approximation method to the scenario of an oil spill. Parameters of this scenario are:

$$\begin{cases} \Omega = [0,12] \times [0,3] \\ D = 0.2 \\ v_x = -0.8 \\ v_y = -0.4 \\ c_{start}(t,x,y) = 0 \\ c_{bc}(t,x,y) = 0 \\ g(t,x,y) = 0 \\ f(t,x,y) = \begin{cases} \frac{1}{2}\left(1 - tanh\left(\frac{\sqrt{(x-x_s)^2+y^2} - r_s}{\epsilon}\right)\right) \text{if t} < 0.5 \\ 0 \text{ if t} \geq 0.5 \end{cases} \end{cases} \tag{9}$$

where $x_s = 10$, $r_s = 0.1$, $\epsilon = 0.1$, $t_{start} = 0$, $t_{final} = 10$, $N_x = 160$, $N_y = 40$, $\Delta t = 0.1$. We will monitor three beaches at (x,y) = (4,0), (6,0), (8,0) and plot oil concentration vs time at each beach. Implementing this in MATLAB, we obtain the following oil concentration graphs for each beach:

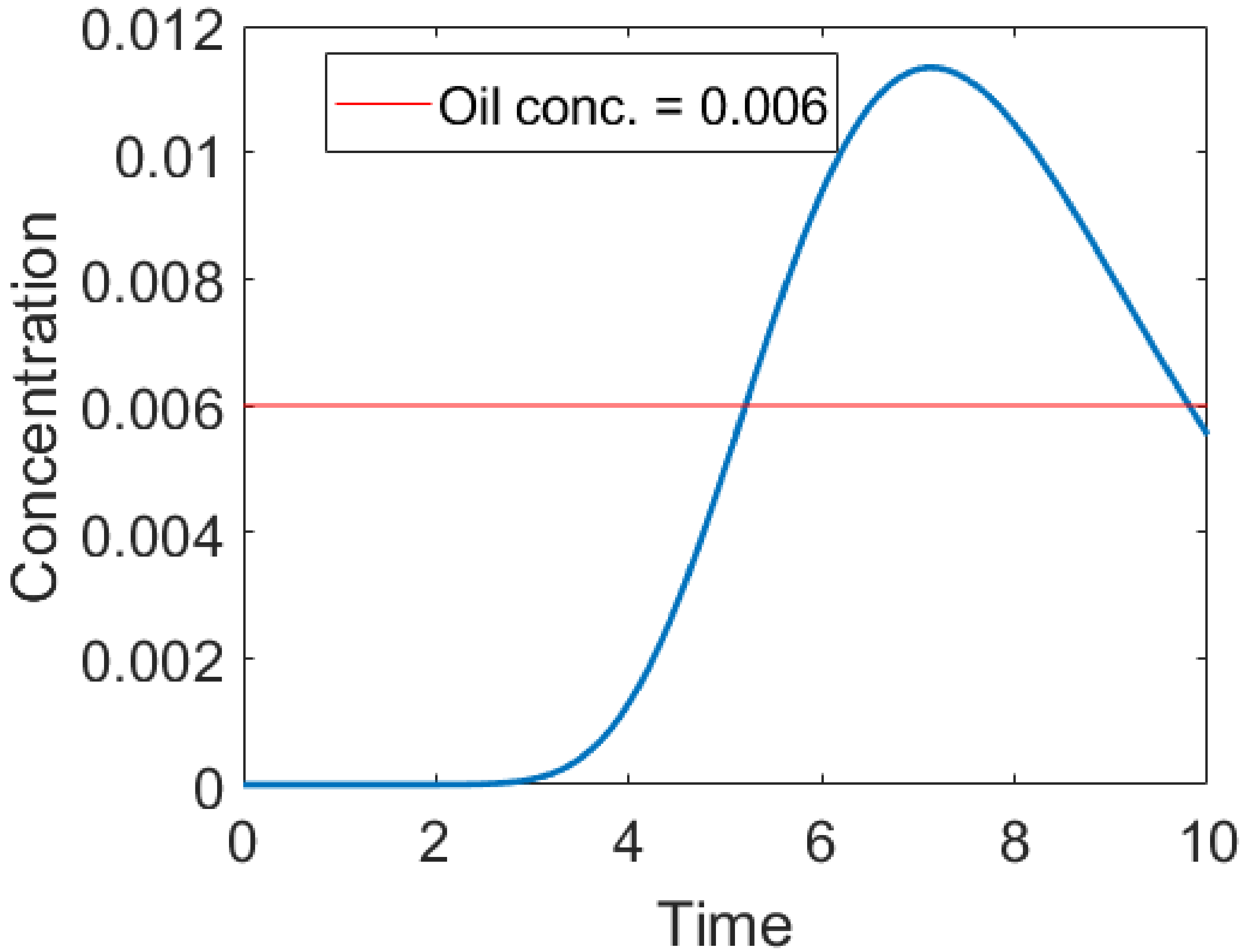

Figure 1: Oil Concentration vs Time at Beach A. This beach should close at after = 5.3 days, on May 24, 2015

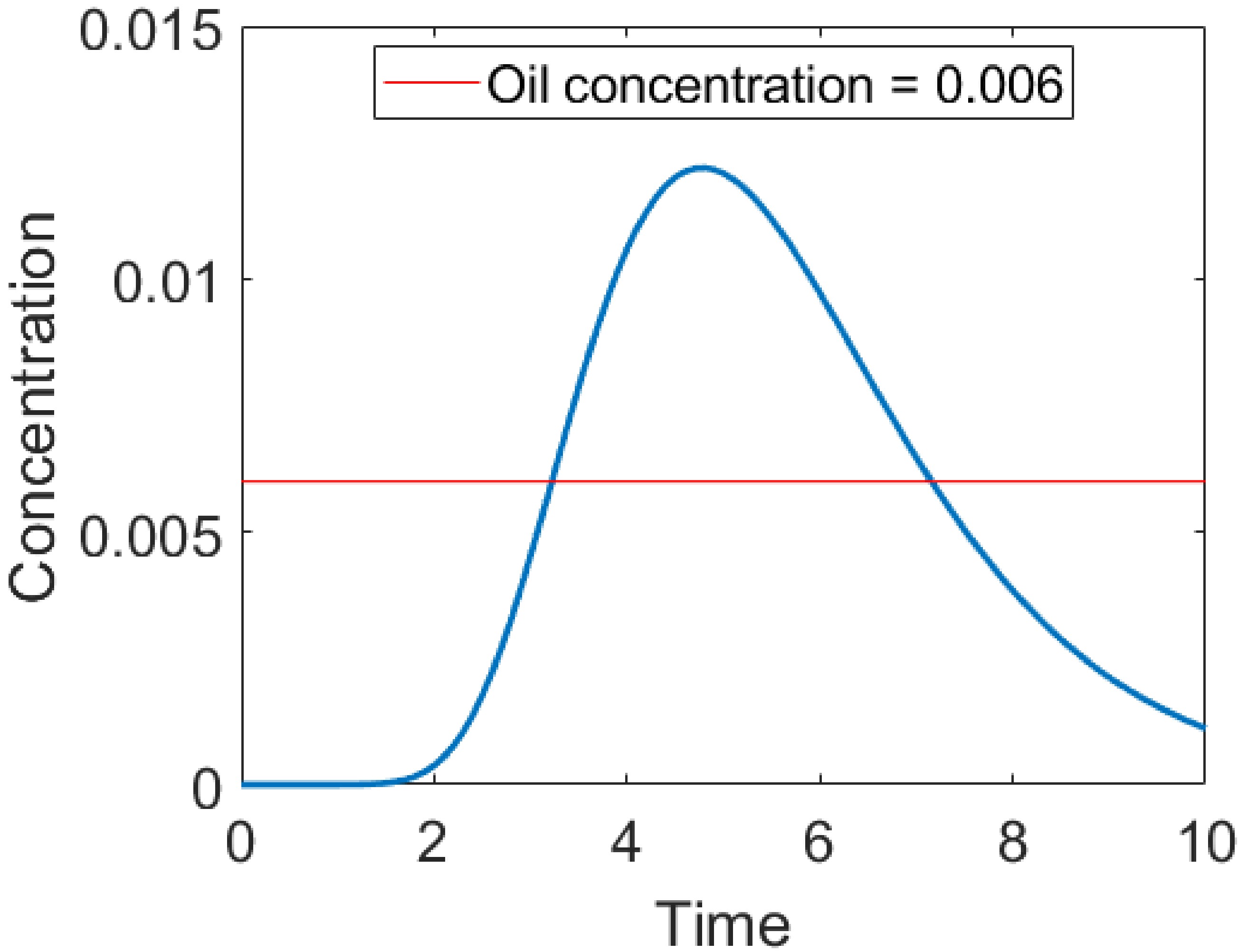

Figure 2: Oil Concentration vs Time at Beach B. This beach should close after = 3.3 days, on May 22, 2015

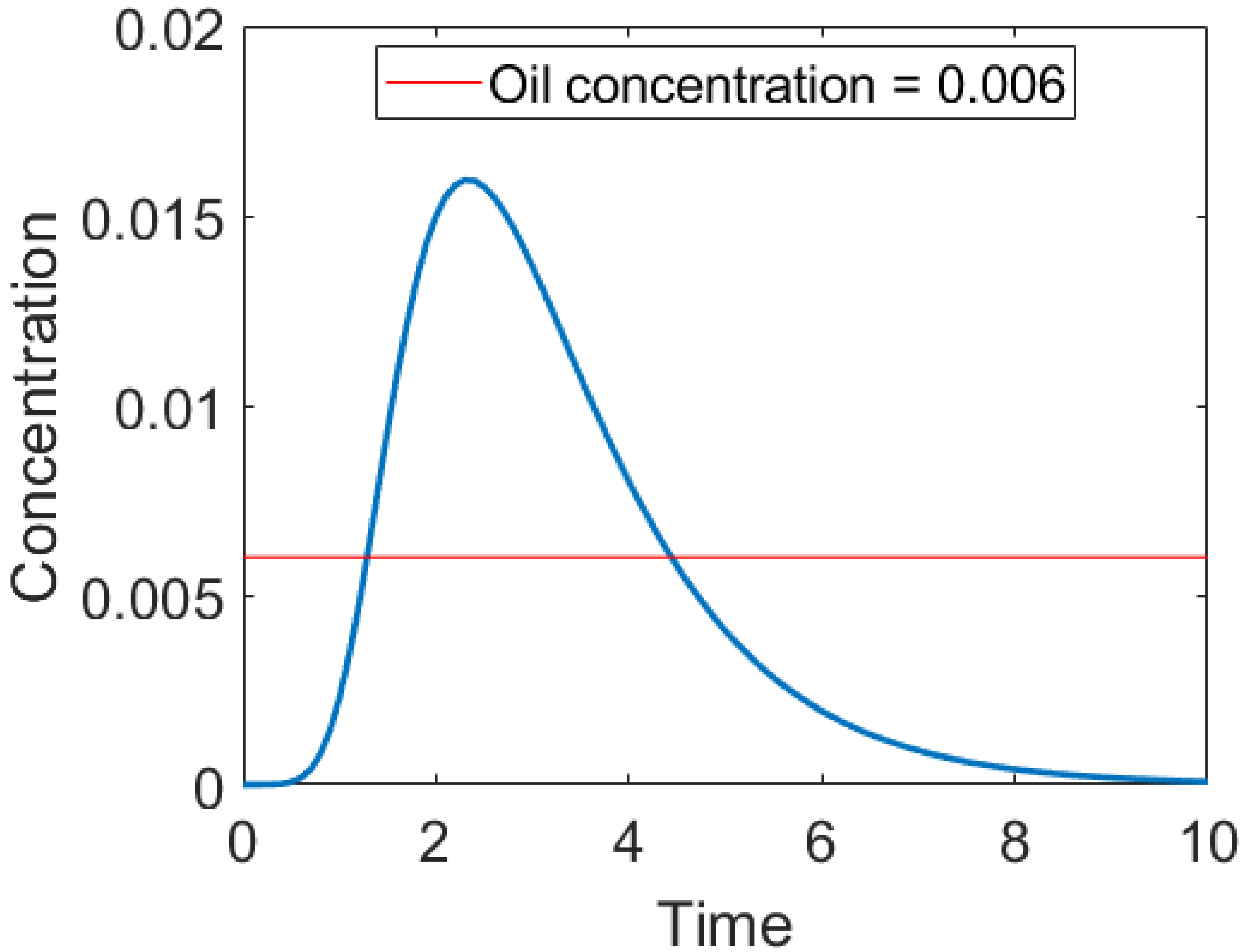

Figure 3: Oil Concentration vs Time at Beach C. This beach should close after t = 1.3 days, on May 20, 2015

Finally we have the following snapshots of the oil concentration along the coast at times t = 1,4,7:

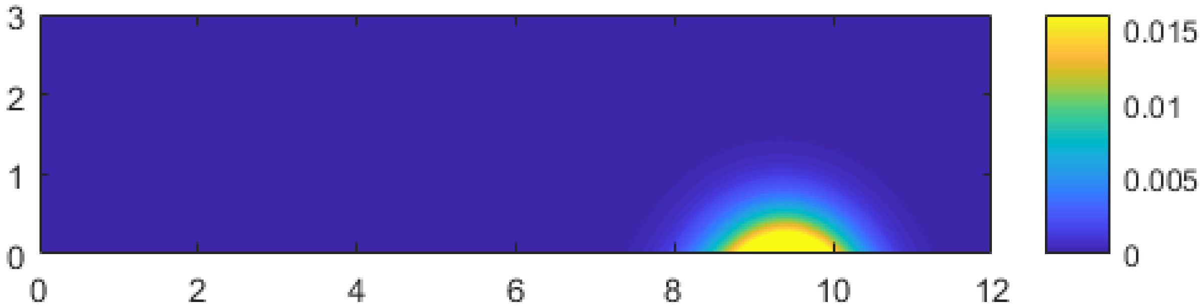

Figure 4: Oil distribution at time t = 1 day.

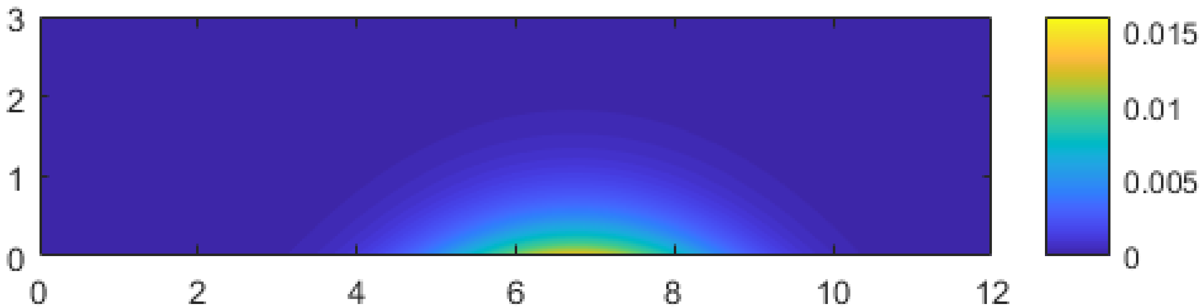

Figure 5: Oil distribution at time t = 4 days.

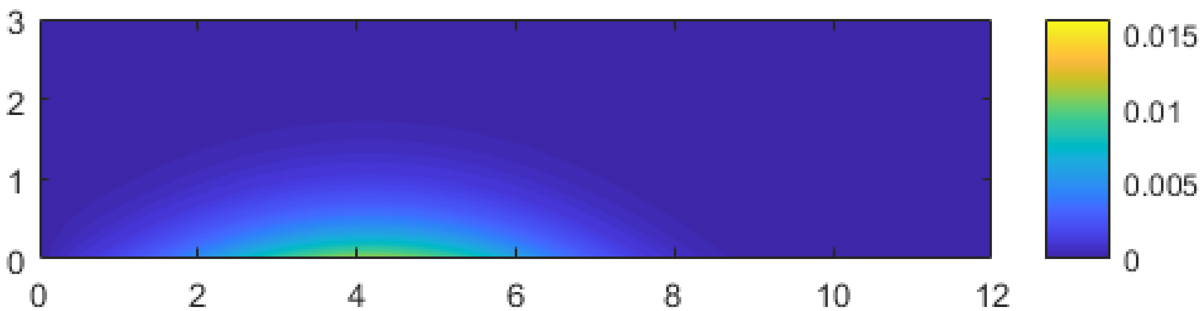

Figure 6: Oil distribution at time t = 7 days.

5 Conclusion

Combining the explicit upwind advection scheme with the implicit diffusion scheme, we obtained a scheme to approximate solutions to the advection-diffusion equation with Robin BC that is first-order accurate in space. Such a scheme can be useful in real world scenarios such as oil distribution in the ocean along the coast following an oil spill.