1. Let $f : [-1, 1] \times [-1, 1] \to [0, 1]$ and suppose the image is corrupted in the domain $D \subseteq [-1, 1] \times [-1, 1]$. Consider inpainting in $D$ by total variation diminishing flow

$$
f_t = \frac{1}{\sqrt{\| \nabla f \|^2 + \epsilon^2}} \left[ \Delta f - \frac{\nabla f^T \nabla^2 f \nabla f}{\| \nabla f \|^2 + \epsilon^2} \right]
$$

according to the following procedure:

- Replace values of $f$ in $D$ by 0.5;
- Under first order von Neumann boundary conditions at the boundary of the image, and $\epsilon = \delta = 0.1$ and time stepsize $k = \delta^3/4$, use central differencing on all spatial derivatives and Euler’s method on the resulting ODE system to get $f_{ij}^{(n+1)}$, for all $i, j$, from $f_{ij}^{(n)}$, for all $i, j$.
- Stop when iterations satisfy

$$
||f^{(n)} - f^{(n-1)}||_\infty < 10^{-6}.
$$

Apply the procedure to the following images, with the following corrupted domains $D$. Submit your final image and write down the number of iterations and the final infinity norm error.

(a) 500 $\times$ 500 image with intensities 0.75, for $x \leq -1/3$; 0.25 for $-1/3 < x \leq 1/3$; 0.5 otherwise, and where $D$ includes all pixels $(i, j)$ satisfying $j$ or $j+1$ or $j+2$ divisible by 10.

(b) 500 $\times$ 500 image with intensities 0.25, for $x \leq 0$; 0.75 otherwise, and $D = \{(x, y)\mid -1/3 \leq x \leq 1/3, -1/3 \leq y \leq 1/3\}$.

2. Use the same procedure as in the previous problem, but with the following stopping condition, on the following images $f : \mathbb{R} \to [0, 1]$, with the following corrupted domains $D$. Submit your final image and write down the final infinity norm error.

(a) dunes.bmp, for 1000 iterations, where $D$ includes all pixels $(i, j)$ satisfying $j$ divisible by 2.

(b) trains.bmp, for 5000 iterations, where $D$ includes all pixels $(i, j)$ satisfying $i$ or $i+1$ or $i+2$ divisible by 7, or $j$ divisible by 4.

(c) snake.bmp, for 18900 iterations, where $D$ includes all pixels $(i, j)$ satisfying $i \leq 50$ and $j \leq 50$.

3. Show the approximation for $f_{xy}$ at pixel $(i, j)$ using

$$
(f_x)_{ij} \approx \frac{f_{i+1,j} - f_{i-1,j}}{2h},
$$
for all \( i, j \), and
\[
((f_x)_y)_{ij} \approx \frac{(f_x)_{i,j+1} - (f_x)_{i,j-1}}{2h},
\]
is the same as that using
\[
(f_y)_{ij} \approx \frac{f_{i,j+1} - f_{i,j-1}}{2h},
\]
for all \( i, j \), and
\[
((f_y)_x)_{ij} \approx \frac{(f_y)_{i+1,j} - (f_y)_{i-1,j}}{2h}.
\]

4. Use Taylor series to prove
\[
\left| (f_{xy})_{i,j} - \frac{f_{i+1,j+1} - f_{i-1,j+1} - f_{i+1,j-1} + f_{i-1,j-1}}{4h^2} \right| = \mathcal{O}(h^2).
\]

5. Denoise the following images with the following number of heat flow steps applied to the whole image. Also write down the intensity of pixel \((17,17)\) of your final result.

(a) “plant.bmp” using 400 steps.
(b) “modelnoisy.bmp” using 20 steps.
(c) “snakenoisy.bmp” using 3 steps.
(d) “dunesnoisy.bmp” using 50 steps.
(e) “trainsnoisy.bmp” using 30 steps.

6. Denoise the following images with the following number of total variation diminishing flow steps \((\epsilon = h = 0.1)\) applied to the whole image. Also write down the intensity of pixel \((17,17)\) of your final result.

(a) “plant.bmp” using 400 steps.
(b) “modelnoisy.bmp” using 100 steps.
(c) “snakenoisy.bmp” using 50 steps.
(d) “dunesnoisy.bmp” using 100 steps.
(e) “trainsnoisy.bmp” using 200 steps.