ECE 285 – Assignment #8

Writing part

---

1 Exercise 1 (Heat equation)

Consider the one-dimensional continuous heat equation defined for $s \in \mathbb{R}$ and $t \geqslant 0$ as

$$\frac{\partial x}{\partial t}(s,t) = \alpha \frac{\partial^2 x}{\partial s^2}(s,t) \quad \text{and} \quad x(s,0) = y(s)\ .$$

1. By taking the Fourier transform (with respect to s) in both sides of the Heat equation, show that

$$\frac{\partial \mathcal{F}_s[x]}{\partial t}(u,t) = -4\pi^2 u^2 \alpha \cdot \mathcal{F}_s[x](u,t) \quad \text{and} \quad \mathcal{F}_s[x](u,0) = \mathcal{F}_s[y](u).$$

2. Deduce from the previous question that

$$\mathcal{F}_s[x](u,t) = \mathcal{F}_s[y](u) \cdot e^{(-4\pi^2 \alpha u^2)t}.$$

Hint: solution of a first order differential equation $x'(t) = ax(t)$.

3. Take the inverse Fourier transform in both sides of the previous equation, and conclude that

$$x(s,t) = (y \ * \ \mathcal{G}_{2\alpha t})(s).$$

Hint: $\mathcal{F}[\mathcal{G}_{\tau^2}] = \sqrt{2\pi\tau^2}\mathcal{G}_{1/4\pi^2\tau^2}$.

4. Consider the discretization in space and time of this problem: $x_i^k = x(i\delta_s, k\delta_t)$ and $y_i = y(i\delta_s)$, and show that for $\gamma = \frac{\alpha\delta_t}{\delta_s^2}$:

$$x_i^k \approx \frac{1}{\sqrt{4\pi\gamma k}} \sum_{u\in\mathbb{Z}} y_{i-u} e^{-\frac{u^2}{4\gamma k}} = (y * \mathcal{G}_{2\gamma k})_i.$$

Hint 1: use the change of variable $u \to u\delta_s$.
Hint 2: use Riemann's integral approximation (rectangle method).

Practical part

---

2 Exercise 2 (Anisotropic diffusion)

1. Create in `imnorm2.m`, a function

```
function a = imnorm2(v)
```

that returns the square of the ℓ_2 norm at each location of the vector field v: $a_{i,j} = \|v_{i,j,:}\|_2^2$.

2. Create in `radiffusion.m`, a function

```
function [x, alpha] = radiffusion(y, m, gamma, g)
```

that implements the explicit scheme for the regularized anisotropic diffusion given by

$$\begin{aligned} x^0 &= y \\ \alpha^k &= g(\|\nabla(\mathcal{G}_\sigma \, * \, x^k)\|_2^2) \\ v^k &= \alpha^k \nabla x^k \\ x^{k+1} &= x^k + \gamma \operatorname{div} v^k \end{aligned}$$

for $k = 1$ to m and where $g : \mathbb{R} \to \mathbb{R}$ is given in argument as a `Matlab`'s handle function, and $\sigma = 0.2$ (Gaussian limited to a support $[-1, 1]^2$).

Note that α^k is a 2d array and v^k is a 2d vector field such that

$$\alpha^k_{i,j} = g(\|(\nabla(\mathcal{G}_\sigma \, * \, x^k))_{i,j}\|_2^2) \quad \text{and} \quad v^k_{i,j,:} = \alpha^k_{i,j}(\nabla x^k)_{i,j,:}$$

NB: Don't store all iterates x^1, x^2, ... Instead overwrite the results of iteration k at iteration $k+1$.

3. Download `assignment8.zip` and extract the data:

   - `assignment8/fish.png`
   - `assignment8/immtimes.m`
   - `assignment8/mushroom.png`
   - `assignment8/imcgs.m`
   - `assignment8/imspecfunc.m`

   Create a script `test_radiffusion.m`, that loads the image `x0 = fish`, creates a noisy version of it `y` with Gaussian noise of standard deviation 10. Then, run your function on `y` for `m = 100`, `gamma = 1/8` and `g = @(u) 10 ./ (10 + u)`. Display the results, and check that your results are consistent with the following ones. Next, restart with `x0 = mushroom`.

(a) Noisy image (b) RAD (c) Conductivity

4. Create in `heatdiffusion.m`, a function

```
function x = heatdiffusion(y, m, gamma)
```

   that produces the result of the Heat equation in $O(n \log n)$ (see Exercise 1, question 4).

5. Complete `test_radiffusion.m`, to compare the `heatdiffusion` with `raddiffusion` for $\gamma = 1/8$. Test with `g = @(u) 10 ./ (10 + u)` and `g = @(u) 1`. Conclude.

6. Bonus: implement the implicit scheme as seen in class using `imcgs`.

3 Exercise 3 (Truly anisotropic diffusion)

We are going to implement the explicit scheme for the truly anisotropic diffusion given by

$$\begin{aligned} x^0 &= y \\ M^k &= (\nabla \mathcal{G}_\sigma * x^k)(\nabla \mathcal{G}_\sigma * x^k)^T \\ M^k_{\text{conv}} &= \mathcal{G}_\rho * M^k \\ T^k &= h[M^k_{\text{conv}}] \\ v^k &= T^k \times \nabla x^k \\ x^{k+1} &= x^k + \gamma \operatorname{div} v^k \end{aligned}$$

for $\rho = 0.5$ (Gaussian limited to a support $[-1,1]^2$), where M^k is a matrix field of 2×2 tensors (size: $n_1 \times n_2 \times 2 \times 2$), $(M^k_{\text{conv}})_{:,:,p,q} = \mathcal{G}_\rho * M^k_{:,:,p,q}$, $h[A]$ is a matrix spectral function that applies g to the eigenvalues of A, and $T^k \times \nabla x^k$ is a vector field obtained as the punctual matrix vector product $(T^k \times \nabla x^k)_{i,j,:,:} = T^k_{i,j,:,:} \times (\nabla x^k)_{i,j,:}$.

1. Copy paste `radiffusion.m` into `tadiffusion.m`, and change the function signature to

```
function x = tadiffusion(y, m, gamma, g)
```

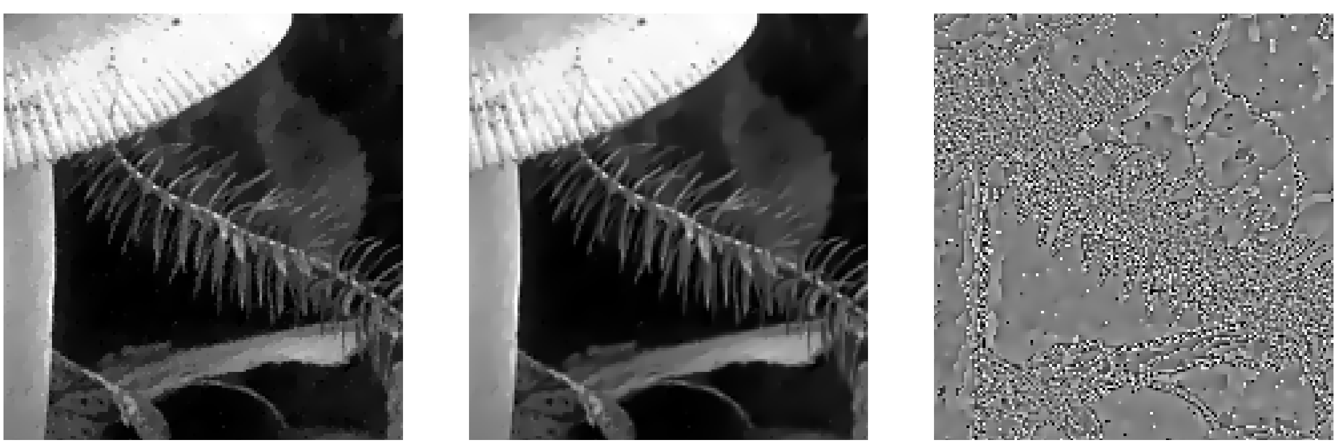
2. Inside the loop for k, write a double loop for p and q creating the matrix field M_{conv}:

```
M_conv = zeros(n1, n2, 2, 2);
for p = 1:2
    for q = 1:2
```

3. Use `imspecfunc` to create T the matrix obtained by applying g to the eigenvalues of M_{conv}.

4. Using the function `immtimes`, complete the function `tadiffusion`.

5. Create a script `test_tadiffusion.m`, that loads the image `x0 = mushroom`, creates a noisy version of it `y` with Gaussian noise of standard deviation 10. Then, run your function on `y` for `m = 100`, `gamma = 1/8` and `g = @(u) 10 ./ (10 + u)`. Display the results, compare PSNRs and execution times, and check that they are consistent with the following ones. Next, restart with `x0 = fish`.



(d) RAD (2.2s / 30.8dB) (e) TAD (6s / 31.4dB) (f) Difference

6. Play with the parameters γ, σ, ρ and g and the noise level.