Numerical Algorithms for Visual Computing II 2010/11
Example Solutions for Assignment 8

Problem 1 (Characteristic Climax) Let us first discuss in general what characteristics mean of IVPs for hyperbolic PDEs of the type:

$$u_t + f(u)_x = 0 \tag{1}$$

The characteristics are defined as curves in the x-t-domain that fulfill the following equation:

$$\frac{\partial}{\partial t} u(x(t), t) = 0 \tag{2}$$

Essentially, the constraint of (2) defines curves along which the solution of the PDE is constant in respect to changes in the time variable t. Let us now examine equation (2) by applying the (multidimensional) chain rule:

$$\frac{\partial}{\partial t} u(x(t), t) = \frac{\partial}{\partial x} u \frac{\partial}{\partial t} x + \frac{\partial}{\partial t} u = u_t + x_t u_x \tag{3}$$

Combining this result with the equations (1) and (2), we get:

$$u_t + f(u)_x = u_t + x_t u_x$$

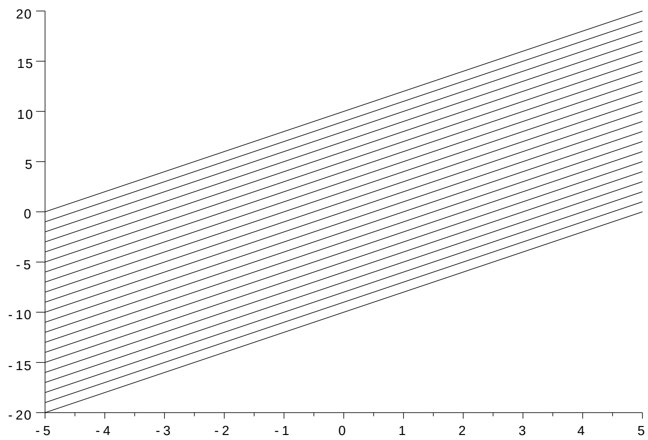
In this exercise, we consider only flux functions of the general type $f(u)_x = g(u, t) u_x$. This leads to a simple condition for the characteristics:

$$x_t(t) = g(u, t) \tag{4}$$

1. For $f(u) = 2u$ condition (4) with $g \equiv 2$ leads to $x_t \equiv 2$. Thus, the characteristics are straight lines with slope 2, i.e. they can be described by:

   $$x(t) = x_0 + 2t$$

   Naturally, there are infinitely many straight lines that differ only in x_0.



2. In the same way as in 1., we get infinitely many lines of the type $x(t) = x_0 + \frac{1}{2}t$.

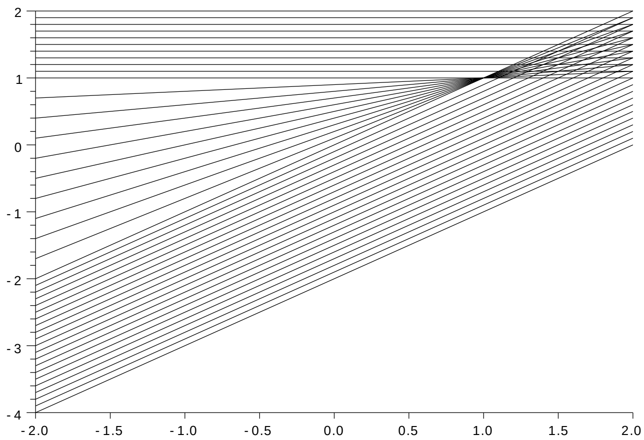
3. For the choice of $f(u) = \frac{1}{2}u^2$ we get the PDE

$$u_t + \underbrace{uu_x}_{=\left(\frac{1}{2}u^2\right)_x} = 0 \tag{5}$$

   and thus $g(u,t) = u(x,t)$ and therefore, for some fixed point x_0, as in 1. and 2., $x_t \equiv u(x_0,t)$. Since $\partial_t u(x_0,t) = 0$ by construction, we know $u(x,t)$ is constant in respect to t and by that we get $x_t \equiv u(x_0, 0)$. Again, the characteristics are straight lines:

$$x(t) = x_0 + u(x_0, 0)t \tag{6}$$

   However, this time the slopes vary which means that the lines can intersect (which in the example from this exercise they do in the point $(1,1)$). At each interesection, a shock occurs, i.e. u becomes a multi-valued function.



Problem 2 (Parabolic Recall)

1. We begin our analysis by considering the discretised diffusion equation

$$u_t = \nabla \cdot (D(u)\nabla u) \tag{7}$$

by taking into account the four pixel boundary segments $l_{ij,1}, \dots, l_{ij,4}$ as in (12.3):

$$\frac{\mathrm{d}}{\mathrm{d}t}\overline{u}_{ij}(t) = \frac{1}{|\sigma_i|}\sum_{k=1}^{4}\int_{l_{ij,k}} (D(u)\nabla u)\cdot \vec{n}\mathrm{d}s. \tag{8}$$

As a note, in the following we denote $D \equiv D(u)$.

$$\underbrace{\int_t^{t+\delta t}\left[\frac{\mathrm{d}}{\mathrm{d}t}\overline{u}_{ij}(t)\right]\mathrm{d}t}_{\overline{u}_{ij}(t+\delta t)-\overline{u}_{ij}(t)} = \frac{1}{|\sigma_i|}\int_t^{t+\delta t}\left\{\sum_{k=1}^{4}\int_{l_{ij,k}}(D\nabla u)\vec{n}\mathrm{d}s\right\}\mathrm{d}t. \tag{9}$$

In order to give a shorter notation for the time discretisation, we set in the following $\bar{u}_{ij}(t+\delta t) =: u_{ij}^{n+1}$ and $\bar{u}_{ij}(t) =: u_{ij}^n$. Let us rewrite this now as an explicit scheme, i.e.

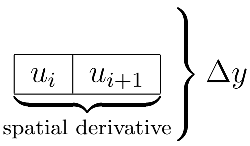
$$u_{ij}^{n+1} = u_{ij}^n + \frac{1}{|\sigma_i|}\int_t^{t+\delta t}\left\{\sum_{k=1}^{4}\int_{l_{ij,k}}(D\nabla u)\vec{n}\mathrm{d}s\right\}\mathrm{d}t \tag{10}$$

Now we have to do the spatial discretisation, as in (16.11) we can rewrite this as

$$\approx \quad u_{ij}^n + \frac{\delta t}{|\sigma_i|}\left\{\sum_{k=1}^{4}\int_{l_{ij,k}}(D\nabla u)\big|_t \vec{n}\mathrm{d}s\right\} \tag{11}$$

$$= \quad u_i^n + \frac{\delta t}{\Delta x \Delta y}\left\{\sum_{k=1}^{4}\int_{l_{ij,k}}(D\nabla u)\big|_t \vec{n}\mathrm{d}s\right\} \tag{12}$$

Now, we have to approximate the boundary integrals as in (16.13), however, we need to be careful here. In the lecture we have used $h = \Delta x = \Delta y$. In most applications, this suffices, however in order to be precise, we have to consider the following. Each direction needs proper approximation, so if we would want to approximate with respect to the direction $l_{ij,1}$, i.e. the pixels right neighbour, then we have to consider this setup:



This means, that we consider for the spatial derivative the standard approach, however, as the pixel width in y-direction, we have to consider that the spatial differentiation occurs along the entire border between both pixels u_i and u_{i+1}. As the size of this border is Δy, we have to include this in the approximation, i.e.

$$\int_{l_{ijk}}(D\nabla u)\cdot\vec{n}\mathrm{d}s \quad \approx \quad \begin{cases} \Delta y \cdot [(D\nabla u)\cdot\vec{n}]\big|_{m_{ij,k},t} & \text{if } k = 1,3 \\ \Delta x \cdot [(D\nabla u)\cdot\vec{n}]\big|_{m_{ij,k},t} & \text{if } k = 2,4 \end{cases}$$

From this we can plug this in into (12):

$$\begin{aligned} = u_{ij}^n + \frac{\delta t}{\Delta x \Delta y}\Big(& \quad \Delta y[D\nabla u\cdot\vec{n}]\big|_{m_{ij,1},t} + \Delta x[D\nabla u\cdot\vec{n}]\big|_{m_{ij,2},t} \\ & + \quad \Delta y[D\nabla u\cdot\vec{n}]\big|_{m_{ij,3},t} + \Delta x[D\nabla u\cdot\vec{n}]\big|_{m_{ij,4},t}\Big) \end{aligned} \tag{13}$$

As being given in the instructions, we set as the diffusion tensor (with the function $g(s^2) = \frac{1}{1+\frac{s^2}{\lambda^2}})$

$$
\begin{aligned}
D = g(|\nabla u|^2)I &= \begin{pmatrix} g(|\nabla u|^2) & 0 \\ 0 & g(|\nabla u|^2) \end{pmatrix} \\
&= \begin{pmatrix} \dfrac{1}{1+\dfrac{u_x^2+u_y^2}{\lambda^2}} & 0 \\ 0 & \dfrac{1}{1+\dfrac{u_x^2+u_y^2}{\lambda^2}} \end{pmatrix}
\end{aligned}
$$

Now, we have to employ a discretisation of the derivatives. For this, we will use the central differences, i.e.

$$
g(|\nabla u|^2)\Big|_t \approx \frac{1}{1+\dfrac{\left(\dfrac{u^n_{i+1,j}-u^n_{i-1,j}}{2\Delta x}\right)^2+\left(\dfrac{u^n_{i,j+1}-u^n_{i,j-1}}{2\Delta y}\right)^2}{\lambda^2}}
$$

In the following, we abbreviate this to $D^n_{ij} := g(|\nabla u|^2)\big|_{i,j,t}$. The discretisation of the tensor are then given as in (12.15)

$$
\begin{aligned}
D\Big|_{m_{i,j,1},t} &=: D^n_{i+\frac{1}{2},j} = \frac{D^n_{ij}+D^n_{i+1,j}}{2} \\
D\Big|_{m_{i,j,2},t} &=: D^n_{i,j+\frac{1}{2}} = \frac{D^n_{ij}+D^n_{i,j+1}}{2} \\
D\Big|_{m_{i,j,3},t} &=: D^n_{i-\frac{1}{2},j} = \frac{D^n_{ij}+D^n_{i-1,j}}{2} \\
D\Big|_{m_{i,j,4},t} &=: D^n_{i,j-\frac{1}{2}} = \frac{D^n_{ij}+D^n_{i,j-1}}{2}
\end{aligned}
$$

We can plug this now in into our initial equation (together with the scalar products with the respective norms) and get

$$
\begin{aligned}
u^{n+1}_{ij} = u^n_{ij} + \frac{\delta t}{\Delta x \Delta y}\Big(&\Delta y \frac{D^n_{ij}+D^n_{i+1,j}}{2}\frac{u^n_{i+1,j}-u^n_{i,j}}{\Delta x} + \Delta x \frac{D^n_{ij}+D^n_{i,j+1}}{2}\frac{u^n_{i,j+1}-u^n_{i,j}}{\Delta y} \\
- &\Delta y \frac{D^n_{ij}+D^n_{i-1,j}}{2}\frac{u^n_{ij}-u^n_{i-1,j}}{\Delta x} - \Delta x \frac{D^n_{ij}+D^n_{i,j-1}}{2}\frac{u^n_{i,j}-u^n_{i,j-1}}{\Delta y}\Big)
\end{aligned}
$$

This leads to our sought iterative method

$$
\begin{aligned}
u^{n+1}_{ij} = u^n_{ij} &+ \frac{\delta t}{\Delta x^2}\left(\frac{D^n_{ij}+D^n_{i+1,j}}{2}(u^n_{i+1,j}-u^n_{i,j}) - \frac{D^n_{ij}+D^n_{i-1,j}}{2}(u_{ij}-u_{i-1,j})\right) \\
&+ \frac{\delta t}{\Delta y^2}\left(\frac{D^n_{ij}+D^n_{i,j+1}}{2}(u^n_{i,j-1}-u^n_{i,j}) - \frac{D^n_{ij}+D^n_{i,j-1}}{2}(u^n_{i,j}-u^n_{i,j-1})\right),
\end{aligned}
$$

which concludes our calculations.

2. Let us consider a general nonlinear diffusion equation

$$u_t = \nabla \cdot (D(u)\nabla u), \tag{14}$$

where $D \equiv D(u)$ is a nonlinear diffusion tensor, with only diagonal entries. Let us employ the Theorem of Gauß at (14) over a pixel σ_i and divide by $|\sigma_i|$ (as in (11.3):

$$\frac{1}{|\sigma_i|}\int_{\sigma_i} \frac{\mathrm{d}}{\mathrm{d}t} u(\vec{x},t)\mathrm{d}\vec{x} = \frac{1}{|\sigma_i|}\int_{\partial\sigma_i} \nabla \cdot [D(u)\nabla u]\mathrm{d}\vec{s}$$

Assuming smoothness of u, we can pull the temporal derivative in front of the left hand side integral, i.e.

$$\frac{\mathrm{d}}{\mathrm{d}t} \underbrace{\left[\frac{1}{|\sigma_i|}\int_{\sigma_i} u(\vec{x},t)\mathrm{d}\vec{x}\right]}_{\substack{\text{Average grey value}\\ \text{over pixel } \sigma_i}} = \frac{1}{|\sigma_i|}\int_{\partial\sigma_i} \nabla \cdot [D(u)\nabla u]\mathrm{d}\vec{s}$$

$$\Leftrightarrow \frac{\mathrm{d}}{\mathrm{d}t}\overline{u}_i(t) = \frac{1}{|\sigma_i|}\int_{\partial\sigma_i} [D(u) \cdot \nabla u] \cdot \vec{n}\mathrm{d}s. \tag{15}$$

For $\nabla u \equiv \vec{0}$ along $\partial\sigma_i$ according to von Neumann boundary conditions we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\overline{u}_i(t) = \frac{1}{|\sigma_i|}\int_{\partial\sigma_i} [D(u) \cdot \vec{0}] \cdot \vec{n}\mathrm{d}s = 0, \tag{16}$$

i.e., analogously to the procedure in §11, we see that the average grey value is conserved for von Neumann boundary conditions. In order to have a proper numerical implementation, this von Neumann boundary needs some further considerations. Consider a 1-D signal $u = (u_1, \ldots, u_N)^\top$. Now consider, we would have to introduce ghost boundary pixels u_0 and u_{N+1} in order to employ von Neumann conditions for the derivatives on the border pixels. For this, you may just mirror pixels u_1 and u_N, so that you have $u_0 = u_1$ and $u_{N+1} = u_N$. This suffices for the ensurance of the average grey value.

Problem 3 (Discrete Theorem of Gauß)

The conservation form of a hyperbolic PDE-discretisation is defined in (16.2) in the script:

$$\begin{aligned} u_{n+1} &= u_j^n - \frac{\Delta t}{\Delta x}(g_{j+\frac{1}{2}} - g_{j-\frac{1}{2}}) \\ g_{j+\frac{1}{2}} &\equiv g(u_j^n, u_{j+1}^n) \\ g_{j-\frac{1}{2}} &\equiv g(u_{j-1}^n, u_j^n) \end{aligned}$$

A discretisation that can be written in this form is grey-value conservative, i.e. the average grey value is preserved. In order to write the Lax-Wendroff-discretisation in the conservation form we have to find a fitting numerical flux function $g(\cdot, \cdot)$.

With some simple operations we can rewrite the Lax-Wendroff scheme in a form that gives us an idea how g might look like:

$$U_j^{n+1} = U_j^n - \tfrac{\Delta t}{\Delta x}\left[\underbrace{\frac{1}{2}(f(U_{j+1}^n) - f(U_{j-1}^n))}_{:=G_1} \underbrace{-\frac{\Delta t}{2\Delta x}(A_{j+\frac{1}{2}}(f(U_{j+1}^n) - f(U_j^n)) - A_{j-\frac{1}{2}}(f(U_j^n) - f(U_{j-1}^n)))}_{:=G_2}\right]$$

In order to achieve the conservation form, we have to find g sucht that

$$G_1 + G_2 = g(u_j^n, u_{j+1}^n) - g(u_{j-1}^n, u_j^n)$$

holds. For G_2 we can easily find a function $g^{(2)}$ that fulfills this equation. The factors $A_{j+\frac{1}{2}}$ and $A_{j-\frac{1}{2}}$ already suggest how to split the term:

$$G_2 = \underbrace{-\frac{\Delta t}{2\Delta x}A_{j+\frac{1}{2}}(f(U_{j+1}^n) - f(U_j^n))}_{=g_{j+\frac{1}{2}}^{(2)}} + \underbrace{\frac{\Delta t}{2\Delta x}A_{j-\frac{1}{2}}(f(U_j^n) - f(U_{j-1}^n))}_{=g_{j-\frac{1}{2}}^{(2)}}$$

The resulting function $g^{(2)}$ is of the form:

$$g^{(2)}(U, V) = -\frac{\Delta t}{2\Delta x}\underbrace{f'\left(\frac{1}{2}(U - V)\right)}_{=A_{j\pm\frac{1}{2}}}(f(U) - f(V))$$

For G_1 we first have to rewrite the term with a little trick in order to achieve a similar situation as in G_2:

$$G_1 = \frac{1}{2}(f(U_{j+1}^n) - f(U_{j-1}^n)) = \underbrace{\frac{1}{2}(f(U_{j+1}^n) - f(U_j^n)}_{=g^{(1)}_{j+\frac{1}{2}}} + \underbrace{\frac{1}{2}(f(U_j^n) - f(U_{j-1}^n))}_{=g^{(1)}_{j-\frac{1}{2}}}$$

Combining the results of the examination of G_1 and G_2 yields $g = g^{(1)} + g^{(2)}$:

$$g(U,V) = \frac{1}{2}(f(U) - f(V) - \frac{\Delta t}{2\Delta x} f'\left(\frac{1}{2}(U-V)\right)(f(U) - f(V))$$