Numerical Algorithms for Visual Computing III: Optimisation

Michael Breuß and Kai Uwe Hagenburg

Released: 19.05.2011
Assigned to: Tutorial at 26.05.2009

Assignment 3 – Pratical Assignments

Exercise No. 1 – Slow starters (2+2=4 points)

1. Compute the directional derivative of $f(x, y, z) = 2x^2 + 3y^2 + z^2$ at point $P = (2, 1, 3)^\top$ in the direction of the vector $v = (1, 0, -2)^\top$.
2. Compute the directional derivative of $f(x, y) = \exp(x)\cos(y)$ at point $P = (2, \pi)^\top$ in the direction of the vector $v = (2, 3)^\top$.

Exercise No. 2 – Germany's Next Top Stencil (1+1+8=10 points)

Consider the following simple energy functional:

$$E(u) \quad = \quad \int_1^{11} \frac{1}{2}(u'(x))^2 \,\mathrm{d}x$$

This exercise should give you a feeling how to avoid bad discretisation choices. Given are an initial signal $f_i = 0$ with $i \in \{1, \ldots, 11\}$ and Dirichlet boundary conditions $u(1) = 10$ and $u(11) = 20$.

(a) Compute the Euler-Lagrange equation of the given energy functional.

(b) What kind of function would you expect as a solution that satisfies this boundary conditions?

(c) Discretise the PDE from (a) by means of the following derivation rules:

- Use <u>only</u> forward differences for <u>all</u> occuring derivatives.
- Use <u>only</u> backward differences for <u>all</u> occuring derivatives.
- Use <u>only</u> central differences for <u>all</u> occuring derivatives.

Construct an iterative scheme for each discretisation, implement this in scilab and perform some experiments, i.e. iterate until convergence. What problems occur when implementing boundary conditions? Discuss your results! We will provide a simple template with the basic structure of the program for your implementations.

Exercise No. 3 – Tortellini and Ambrosia Revisited (8 points)

On the last assignment sheet we have introduced the 1-D variant of Ambrosio-Tortorelli by

$$E_{TA}(u, v) = \int_a^b \beta\,(u - f)^2 + v^2\,(u')^2 + \alpha\left(\gamma\,(v')^2 + \frac{(1 - v)^2}{4\gamma}\right)\,dx$$

Implement this method in Scilab with the discretisation from assignment 2 together with Neumann boundary conditions. Test your program at hand of a given noisy signal with varying parameters α, β, γ. What can you say about the time step size? What is a good initialisation for u and v? For your implementation use the given template for iterative schemes that we will provide on the webpage as well as an initial signal.

Exercise No. 4 – Rationalising with gold (8 points)

In this exercise, we want to devise an algorithm for finding minima that does not rely on derivatives.
Let f be a function $f : [a, b] \to \mathbb{R}$ with a minimum ξ.

(a) Given an interval $[a, b]$. Choose two points x_1, x_2 in that interval that fulfill the properties

$$\begin{aligned} x_1 &= b - \tau(b - a) \\ x_2 &= a + \tau(b - a), \end{aligned}$$

i.e., that it also holds $x_1 - a = b - x_2 = (1 - \tau)(b - a)$. Compute the optimal value for τ.

(b) Assume that the initial interval $[a, b] = [a^0, b^0]$ with the inner points x_1^0, x_2^0. Devise an iterative algorithm that rapidly decreases the size of the interval with the property $b^i - a^i = \tau(b^{i-1} - a^{i-1})$. Implement this function and test it again on the function $f(x) = x^4 + 3x^3 - 3x^2 - 7x + 6$.