Winfried Auzinger
Dirk Praetorius
Michele Ruggeri

Sommersemester 2014
26. März 2014

# Übungen zur Vorlesung Computermathematik

## Serie 4

**Aufgabe 4.1.** Let $f : [a, b] \to \mathbb{R}$ be a continuous function. For $N \in \mathbb{N}$ and $x_j := a + j\,(b-a)/N$ with $j = 0, \dots, N$, we define the *composite midpoint rule*

$$I_N := \frac{b-a}{N} \sum_{j=1}^{N} f\big((x_{j-1} + x_j)/2\big). \tag{1}$$

Since $I_N$ is a Riemann sum, we know that

$$\lim_{N\to\infty} I_N = \int_a^b f\,dx.$$

For $f \in C^2[a, b]$, one can even show that

$$\Big|\int_a^b f\,dx - I_N\Big| = \mathcal{O}(N^{-2}).$$

Write a MATLAB function

```
int = midpointrule(a,b,f,n)
```

which, for the sequence $N = 2^k$ and $k = 0, \dots, n$ , computes and returns the vector `int` of the corresponding values $I_N$.

**Aufgabe 4.2.** Consider the integral $I := \int_0^5 \exp(x)\,dx$. Use the composite midpoint rule from Aufgabe 4.1 to compute the sequence of approximate integrals $I_N$. Use a double logarithmic plot to show the error $E_N = |I - I_N|$ as well as the error estimator $\delta_N = |I_{2N} - I_N|$. Verify the convergence behavior $\mathcal{O}(N^{-2})$. What convergence behavior do you observe, if you use Aitken's $\Delta^2$-method? What convergence behavior do you observe if you replace the evaluation $f\big((x_{j-1} + x_j)/2\big)$ at the midpoint by the evaluation $f(x_{j-1})$?

**Aufgabe 4.3.** The following code computes a sparse matrix $A \in \mathbb{R}^{N\times N}$ (You can download the code from the CompMath webpage).

```
function A = matrix(N)

x = rand(1,N);
y = rand(1,N);
triangles = delaunay(x,y);
n = size(triangles,1);
```

```
A = sparse(N,N);
for i = 1:n
   nodes = triangles(i,:);
   B = [1 1 1 ; x(nodes) ; y(nodes)];
   grad = B \ [0 0 ; 1 0 ; 0 1];
   A(nodes,nodes) = A(nodes,nodes) + det(B)*grad*grad'/2;
end
```

Plot the computational time $t(N)$ over $N$ and visualize the growth $t(N) = \mathcal{O}(N^\alpha)$ for $N = 100\cdot 2^k$ and $k = 0, 1, 2, \dots$. What is the bottleneck of this implementation? What can be done to improve the runtime behavior? Write an improved code which leads to a better computational time. Visualize its runtime in the same plot to show that the improved code is really superior. **Hint:** You might want to have a look at `help sparse`.

**Aufgabe 4.4.** Let $m, n, N \in \mathbb{N}$. Let $I, J, a \in \mathbb{R}^N$ represent the coordinate format of a sparse matrix $A \in \mathbb{R}^{m\times n}$, i.e., for all $k = 1, \dots, N$ holds $A_{ij} = a_k$ with $i = I_k$, $j = J_k$. Write a MATLAB function

```
[II,JJ,AA] = naive2ccs(I,J,a,m,n)
```

which returns the corresponding vectors of the CCS format.

**Aufgabe 4.5.** Given the vectors of the CCS format of a sparse matrix $A \in \mathbb{R}^{m\times n}$ from the last exercise, write a MATLAB function

```
Ax = mvm(II,JJ,AA,m,n,x)
```

which computes the matrix-vector multiplication $b = Ax \in \mathbb{R}^m$ for given $x \in \mathbb{R}^n$. The complexity of the code must be $\mathcal{O}(N)$. **Hint:** You can verify your code as follows: Suppose that $A$ is a sparse matrix (e.g., the triadiagonal matrix from page 96 of the lecture notes). Then, the coordinate format of $A$ is obtained by `[I,J,a] = find(A)` in MATLAB. Use your code from Aufgabe 4.4 to compute the vectors of the CCS format and compare the outcome of your function `mvm` with the matrix-vector multiplication `A*x` in MATLAB.

**Aufgabe 4.6.** Write a function `plotPotential`, which takes a function $f : \mathbb{R}^2 \to \mathbb{R}$, a domain $[a, b]^2$ and a step size $\tau > 0$, and plots the projection of $f(x, y)$ onto the 2D plane (i.e., `view(2)`). Add a `colorbar` to the plot. For the visualization, use a tensor grid with step size $\tau$. You may assume, that the actual implementation of $f$ takes matrices $x, y \in \mathbb{R}^{M\times N}$ and returns a matrix $z \in \mathbb{R}^{M\times N}$ of the corresponding function values, i.e., $z_{jk} = f(x_{jk}, y_{jk})$. Optionally, the function `plotPotential` takes a parameter $n \in \mathbb{N}$. For given $n$, add $n$ (black or white) contour lines to the figure. To verify your code, write a MATLAB script which visualizes the potential $f(x, y) = x \cdot \exp(-x^2 - y^2)$ from the lecture notes.

**Aufgabe 4.7.** Write a MATLAB function `saveMatrix` which takes a matrix $A \in \mathbb{R}^{M\times N}$ and writes it into an ASCII file `matrix.dat` via `fprintf` (see also `help fopen`). Use `%1.16e` for `fprintf` to write the matrix coefficients! (Why does this make sense?) Optionally, the function takes a string `name` and writes the matrix to the ASCII file `name.dat`. To verify your code, write a MATLAB script which creates a random matrix $A \in \mathbb{R}^{M\times N}$ and writes it to an ASCII file `A.dat`. Load the matrix via `B = load('A.dat')` and check whether $A$ and $B$ coincide.

**Aufgabe 4.8.** Suppose you are given a C function with signature

```
double f(double x, double y);
```

Write a MEX-MATLAB function `fct` which which takes matrices $X, Y \in \mathbb{R}^{M\times N}$

```
Z = fct(X,Y)
```

and returns the matrix $Z \in \mathbb{R}^{M\times N}$ with $Z_{jk} = f(X_{jk}, Y_{jk})$. The MEX function should check whether the dimensions of $X$ and $Y$ coincide, and should throw an error if they do not. To check your implementation, implement the function $f(x, y) = x \cdot \exp(-x^2 - y^2)$ in C and reproduce some of the plots of the last lecture.