Numerical Algorithms

Winter Semester 2014/2015
Lecturer: Prof. Dr. Beuchler
Assistent: Katharina Hofer

universitätbonn

Excercise sheet 1.

Closing date **Theory: 14.10.2014, Programming: 21.10.2014**.

All exercises shall be solved in a group of two persons. For admittance to the final exam you have to score 50% or more of the overall score (for both, the theoretical and the programming exercises, separately). All theoretical exercises can be written by hand on a sheet. For the programming exercises please prepare a pdf where you explain the code, put the corresponding code parts there and where you show your results. Furthermore you shall send me a running version of your program. There is a code with a lineare algebra package and some additional functionality. This programm will be extended during exercises and for all 1d programming exercises there will be a sample solution. This gives you the possibility to use code which you mabe didn't implement the week before. If you want to use the code please send a mail to hofer@ins.uni-bonn.de and I will send you the code. If you have question to the prepared code don't hesitate to ask. Furthermore I will explain the parts you will need in the first tutorial.
If you wish you can also use your own code. For the exercise you need at least a working lineare algebra package.

1. **Theoretical exercise. [5 points.]** The integrated Legendre-polynomials are given by

$$\hat{L}_n(x) = \int_{-1}^{x} L_{n-1}(s)\,\mathrm{d}s \qquad n \geq 1, \tag{1}$$

($L_n(x)$ denotes the n-th Legendre-polynomial), the scaled integrated Legendre-polynomials by

$$\hat{K}_n(x) = (-1)^n \gamma_n \int_{-1}^{x} L_{n-1}(s)\,\mathrm{d}s \qquad n \geq 2$$

with $\gamma_i = \sqrt{\frac{(2i-3)(2i-1)(2i+1)}{4}}$ and

$$\hat{K}_0(x) = \frac{1-x}{2} \qquad \hat{K}_1(x) = \frac{1+x}{2}.$$

(a) Show the relation

$$\hat{L}_n(x) = \frac{1}{2n-1}\left(L_n(x) - L_{n-2}(x)\right) \qquad \forall n \geq 2. \tag{2}$$

(b) Show

$$\hat{L}_n(\pm 1) = 0 \qquad \forall n \geq 2.$$

(c) Show the Orthogonality:

$$\int_{-1}^{1} \hat{L}_n(x)\hat{L}_m(x)\,\mathrm{d}x = 0 \qquad |n-m| \notin \{0, 2\}.$$

(d) Show the relation

$$\hat{L}_n(x) = \frac{x^2 - 1}{2n - 2} P_{n-2}^{(1,1)}(x) \qquad n \geq 2$$

where $P_n^{(1,1)}$ denotes the n-th Jacobi-polynomial.

(e) Calculate and scetch/plot the first five integrated Legendre-Polynomials $\hat{L}_n(x)$ and the first five scaled integrated Legendre-Polynomials $\hat{K}_n(x)$ on the interval $[-1, 1]$. (HINT: For this item you can use Mathematica/Maple or Matlab.)

2. **Theoretical exercise. [5 points.]** The Gaussian quadrature rule for

$$I(f) = \int_{-1}^{1} f(x)\,\mathrm{d}x$$

is given by

$$\sum_{i=1}^{n} w_i f(x_i) = Q_n(f)$$

where the weights w_i and the points x_i are chosen such that

$$I(f) = Q_n(f) \qquad \forall f \in P_{2n-1}.$$

Explain how the weights w_i and the points x_i are computed.

3. **C++/C. [5 points.]** Legendre-polynomials:

(a) Implement a routine for the evaluation of $L_n(x)$ for given values n and x. You can either use the class LegendreBasis1D in the files basis.hpp, basis.cpp (Attention, they need the linalg-Package) or you write your own code (please consider that you will need your basis functions for the 1d and 2d code, so make sure that this will be possible!). For a given polynomial degree p and a given point x the Legendre-polynomials up to polynomial degree p shall be return in the Vector basis.

```
void LegendreBasis1D::get_basisvalues(Vector & basis, int p, double x);
```

(HINT: Use the recurrence relation of the Legendre polynomials!)

(b) Test your implementation: Calculate the values of the Legendre-polynomials up to $n = 7$ at the points $x_0 = -1, x_1 = -0.3, x_2 = 0.2, x_3 = 0.9$.

4. **C++/C. [5 points.]** Scaled integrated Legendre-polynomials:

(a) Implement a routine for the evaluation of $\hat{K}_n(x)$ for given n and x. (call analogue to the Legendre-Polynomials, if you use the files basis you can use the class IntLegendreBasis1D). (HINT: Use the relation (2), derive a similar relation between the Legendre and the polynomials

$$\tilde{K}_n(x) = (-1)^n \hat{K}_n(x)$$

and correct the sign at the end of the evaluation. You can use the global functions intleg and gamma in glob.hpp. Another possibility would be to use a recurrence relation for $\hat{K}_n(x)$.)

(b) Test your implementation: Calculate the values of the scaled integrated Legendre-polynomials up to $n = 7$ at the points $x_0 = -1, x_1 = -0.3, x_2 = 0.2, x_3 = 0.9$.

(c) Implement a routine for the evaluation of the first derivative of the scaled integrated Legendre-polynomials for a given n and x. The call in IntLegendre-Basis1D is

```
void IntLegendreBasis1D::get_diffbasisvalues(Vector & basis, int p, double x);
```

(d) Test your implementation: Calculate the values of the first derivatives of the scaled integrated Legendre-polynomials at the points $x_0 = -1$ and $x_1 = 0.4$ (HINT: You can use (1)).

5. **C++/C. [5 points.]** Gauss-Legendre Integration (Attention: you will need your implementation of the Legendre-Polynomials!)

   (a) Implement the Gauss-Legendre Integration routine given in the lecture. You can use the files integration.hpp, integration.cpp, then you need to implement

   ```
   void Int1D::calculateWeightsIntpoints(int n);
   ```

   (HINT: The weights and points of the integration formula can be efficiently computed by solving an eigenvalue problem, see lecture “Einführung in die Grundlagen der Numerik ”! Use the Lapack-Blas Library and use the function dsteqr_, documentation can be found at http://www.netlib.org/lapack/double/dsteqr.f If you use the prepared code, everything you need to use the function dsteqr_ was already done.)

   (b) Test your routine with $f_1(x) = 100x \cdot \sin(\pi x)$ and $f_2(x) = \frac{1}{2}x^2 + x^{12}$. Calculate

   $$\int_{-1}^{1} f_i(x)dx \qquad i = 1, 2$$

   for $n = 2, 5, 8, 11$ where n denotes the number of integration points. Which n do you need to calculate f_2 exactly?