Numerical Algorithms

Winter 2023/24

Prof. Dr. Carsten Burstedde
Hannes Brandt

Exercise Sheet 5.

Submission: **21.11.2023**

Exercise 1. (Conjugate Gradient Method)

In the conjugate gradient method presented in the lecture the residuals r_k are pairwise orthogonal with the Euclidean inner product $\langle \cdot, \cdot \rangle$ and the search directions p_k are A-conjugate, so pairwise orthogonal with the inner product $(\cdot, \cdot)_A := \langle \cdot, A \cdot \rangle$.

The preconditioned conjugate gradient method (PCG), which was introduced in the lecture, has similar properties. Choose two appropriate inner products and prove pairwise orthogonality of the residuals as well as pairwise orthogonality of the search directions of the PCG.

(6 Points)

Exercise 2. (Lanczos Iteration)

The Lanczos iteration for a symmetric matrix $A \in \mathbb{R}^{n \times n}$ and a start vector x is given by

$v_1 = \frac{x}{\|x\|_2}$
$h_{1,0} = 0, v_0 = 0$
for $k = 1, \ldots, m$ **do**
  $h_{k,k} = v_k^T A v_k$
  $u = Av_k - h_{k,k-1} v_{k-1} - h_{k,k} v_k$
  $h_{k+1,k} = \|u\|_2$
  if $h_{k+1,k} = 0$ **then**
    break
  end if
  $v_{k+1} = \frac{1}{h_{k+1,k}} u$
end for

which yields a tridiagonal matrix $H_m = V_m^T A V_m$.

a) Prove that for a unitary matrix $Q \in \mathbb{R}^{n \times n}$ the Lanczos iteration applied to $Q^T A Q$ with start vector $Q^T x$ yields the same matrix H_m as when applied to A with start vector x.

b) Let λ_1 be an eigenvalue of A with multiplicity 1 and w_1 a corresponding eigenvector. Show that the matrix $H_k \in \mathbb{R}^{k \times k}$ from the Lanczos iteration with $h_{k+1,k} = 0$ has no eigenvalue equal to λ_1, if the start vector x is orthogonal to w_1. You can use that A is unitary diagonalizable.

(2 + 4 Points)

Exercise 3. (Programming Task)

The software libraries `Triangle` (`https://www.cs.cmu.edu/~quake/triangle.html`) and `TetGen` (`https://www.wias-berlin.de/software/tetgen/1.5/index.html`) create 2D triangle and 3D tetrahedra meshes respectively based on an input domain. The resulting meshes can be output in form of a `.node` and a `.ele` file. The `.node` file begins with a line of metadata starting with

```
#nodes #dimensions
```

followed by `#nodes` many lines, each of them consisting of a node index followed by `#dim` coordinates. The `.ele` file begins with a line of metadata starting with

```
#elements #nodes_per_element
```

followed by `#elements` lines, each of them consisting of an element index followed by `#nodes_per_elements` corner indices indexing into the `.node` file. An example of such 2D and 3D output files can be found on the lecture's website.

a) Experiment with `TetGen` or `Triangle` and produce a mesh on your own.

b) Write a program that reads in a `.node` and a `.ele` file. Implement an iteration over all elements that accesses their corner coordinates, which will be needed for future exercises.

c) Find a way to visualize the triangle/tetrahedra mesh, e.g. by creating a `.vtk` file and visualizing it with an application supporting `vtk` like `ParaView`.

We recommend to implement this in `Python`, since it offers convenient packages like the `vtk`-Interface `pyVista` and `meshio` for converting between common mesh formats. However, you are free to implement your solution in a programming language of your choice.

(0 Points)