Numerical Methods for CSE
Exercise Session Week 41

Giuseppe Accaputo

October 10, 2016

Today's Plan

1. Debrief of exercises 1.8, 1.10, 1.11, and 2.2
2. Answers to some of your questions from last week
3. Installation of the `Figure` class

Initializing Parameters

- Parameters that are listed under `\param[out]` in the comment should be initialized within the function

```
/* ...
 * \param[in] B Matrix of size $n \times n$
 * \param[out] C Kronecker product of A and
    ↪ B of dim $n^2 \times n^2$
 */
void kron(..., MatrixXd & C) {
            C = MatrixXd(A.rows()*B.rows(),
              ↪ ...);
```

Initializing Parameters

- Parameters that are listed under `\param[out]` in the comment should be initialized within the function

```
/* ...
 * \param[in] B Matrix of size $n \times n$
 * \param[out] C Kronecker product of A and
    ↪ B of dim $n^2 \times n^2$
 */
void kron(..., MatrixXd & C) {
            C = MatrixXd(A.rows()*B.rows(),
              ↪ ...);
```

Initializing Variables in General

Variables can be initialized in two ways:

1. First possibility:

```
MatrixXd M = MatrixXd(n, n);
```

2. Second possibility:

```
MatrixXd M(n, n);
```

Initializing Variables in General

Watch out! The following code does not initialize the matrix `M`, but just accesses the element `M(n,n)`:

```
void fun(MatrixXd & M){
    M(n,n);
}
```

The correct way to initialize `M`:

```
void fun(MatrixXd & M){
    M = MatrixXd(n,n);
}
```

Step 1: Install All Packages

1. First of all, install the following packages:

   Ubuntu:

   ```
   sudo apt-get install git cmake libpng
       ↪ ++-dev freeglut3-dev libboost-
       ↪ all-dev
   ```

   Fedora:

   ```
   sudo dnf install zlib-devel libpng-
       ↪ devel freeglut-devel
   ```

Step 2: Install MathGL

1. Download the latest version of MathGL from `https://sourceforge.net/projects/mathgl/files/latest/download?source=typ_redirect`
2. Extract the files and `cd` into the newly extracted folder
3. Run `mkdir build && cd build`
4. Run `cmake ..`
5. Run `make -j <nrcores>` where `<nrcores>` has to be replaced with the number of available processor cores
6. Run `sudo make install`

Step 3: Install the Figure Class

1. Copy the NumCSE/MathGL folder and paste it outside of the NumCSE folder (we do not want to interfere with the git repository)
2. cd into the newfolder/MathGL/FigureClass/Install folder
3. Open the CMakeLists.txt file and comment out the following lines (35 and 36 in the file) with a #:

```
# find and include Eigen
# find_package( Eigen3 REQUIRED )
# include_directories(${
  ↪ EIGEN3_INCLUDE_DIR})
```

Step 3: Install the Figure Class

4. Run `mkdir build && cd build`
5. Run `cmake ..`
6. Run `make -j <nrcores>` where `<nrcores>` has to be replaced with the number of available processor cores
7. Run `sudo make install`

Step 4: Test the `Figure` Class

1. `cd` into the `newfolder/MathGL/FigureClass/Examples/1-`
↪ `LogScaling`
folder
2. Run `mkdir build && cd build`
3. Run `cmake ..`
4. Run `make`
5. Run `./main`
6. A file named `plot.eps` will be created

New Exercises

- Problem 2.3
- Problem 2.5 (**Core problem**)
- Problem 2.4
- Problem 2.6 (**Core problem**)