Numerical validation using the CADNA library practical work

F. Jézéquel, J.-L. Lamotte
LIP6 Laboratory, Sorbonne Université,
Paris, France

Contents

Chapter 1

Getting started

Download from `https://perso.lip6.fr/Fabienne.Jezequel/AFAE.html` the CADNA library: cadna_c-x.x.x.tar.gz, where x.x.x is the CADNA version.

Compile and install the library on your home directory.

```
tar -xzvf  cadna_c-x.x.x.tar.gz
cd cadna_c-x.x.x
./configure --prefix=$PWD
make
make install
```

Remark: on OSX gcc may be a wrapper for clang. In that case, use:

```
./configure  --prefix=$PWD CC=clang CXX=clang++
```

Download the exercises.

```
tar -xvzf ex_AFAE.tar.gz
```

Chapter 2

Exercises

2.1 Exercise 1

This example has been proposed by S. Rump.

$$f(x,y) = 333.75y^6 + x^2(11x^2y^2 - y^6 - 121y^4 - 2) + 5.5y^8 + \frac{x}{2y}$$

is computed with $x = 77617$, $y = 33096$. The 15 first digits of the exact result are -0.827396059946821.

2.1.1 Question 1

In the `ex_AFAE` directory compile and run the `rump.c` program:

```
make rump
./rump
```

Compare the result with the exact one.

2.1.2 Question 2

Copy the `rump.c` program into the `rumpd.c` program.
Change all the variable types from `float` to `double` in the `rumpd.c` program.
Compile and run the `rumpd.c` program:

```
make rumpd
./rumpd
```

Compare the result with the exact one.

2.1.3 Question 3

Implement the CADNA library in the `rump.c` and `rumpd.c` programs by creating two new programs called `rump_cad.cc` and `rumpd_cad.cc`.
In the `Makefile` uncomment and modify the following line:

```
#CADNAC=$(HOME)/cadna_c-x.x.x
```

Compile the `rump_cad.cc` and the `rumpd_cad.cc` programs. Then execute `rump_cad` and `rumpd_cad`.
What do you conclude?

2.2 Exercise 2

The determinant of Hilbert's matrix of size 11 is computed using Gaussian elimination. The determinant is the product of the different pivots. The 14 first digits of the exact determinant are $3.0190953344493\,10^{-65}$.

2.2.1 Question 1

Run the `hilbert.c` program. All the pivots and the determinant are printed out.
Compare the determinant value with the exact one.

2.2.2 Question 2

Implement the CADNA library in the `hilbert.c` program by creating a new program called `hilbert_cad.cc`.
Run the `hilbert_cad.cc` program.
What do you conclude?

2.3 Exercise 3

This example has been proposed by J.-M. Muller.
The `muller.c` program computes the first 30 iterations of the following sequence:

$$U_{n+1} = 111 - \frac{1130}{U_n} + \frac{3000}{U_{n-1} \times U_n}.$$

with $U_0 = 5.5$ and $U_1 = \frac{61}{11}$. The sequence limit is 6.

2.3.1 Question 1

Run the `muller.c` program. Are the results correct?

2.3.2 Question 2

Implement the CADNA library in the `muller.c` program by creating a new program called `muller_cad.cc`.
Run the program. Are the results correct?

By inserting suitable prints in the program, identify where the instabilities occur.
Are the last iterates provided by CADNA reliable? Why?

What do you conclude?

2.4 Exercise 4

This example deals with the improvement of an iterative algorithm by using CADNA functionalities.
This program computes a root of the polynomial

$$f(x) = 1.47x^3 + 1.19x^2 - 1.83x + 0.45$$

by Newton's method. The sequence is initialized with $x = 0.5$.
The iterative algorithm $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ is stopped with the criterion

$$|x_n - x_{n-1}| < 10^{-12}.$$

2.4.1 Question 1

Run the `newton.c` program.
What is the sequence limit? How many iterations are computed?

2.4.2 Question 2

Implement the CADNA library in the `newton.c` program by creating a new program called `newton_cad.cc`. Run the program. How many iterations are computed?
Look at the CADNA instability report.

Change the stopping criteria to `x==y`. How many iterations are computed?
Look at the new CADNA instability report.

What do you conclude?

2.5 Exercise 5

The following linear system is solved using Gaussian elimination with partial pivoting. The system is

$$\begin{pmatrix} 21 & 130 & 0 & 2.1 \\ 13 & 80 & 4.74\ 10^8 & 752 \\ 0 & -0.4 & 3.9816\ 10^8 & 4.2 \\ 0 & 0 & 1.7 & 9\ 10^{-9} \end{pmatrix} . x \; = \; \begin{pmatrix} 153.1 \\ 849.74 \\ 7.7816 \\ 2.6\ 10^{-8} \end{pmatrix}$$

The exact solution is $x_{sol}^t \; = \; \left(1, 1, 10^{-8}, 1\right)$.

2.5.1 Question 1

Run the `gauss.c` program. Are the four results correct?

2.5.2 Question 2

Implement the CADNA library in the `gauss.c` program by creating a new program called `gauss_cad.cc`. Run the program. Comment the results and the instabilities obtained.

By inserting suitable prints in the programs, explain the differences between the results obtained with and without CADNA.

What do you conclude?

2.6 Exercise 6

The `jacobi.c` program implements Jacobi iteration to find the solution of a linear system. The stopping criterion is based on a value ε set to 10^{-4}.

2.6.1 Question 1

Run the `jacobi.c` program.
Are the results correct? How many iterations are performed?
Try other values for ε.

2.6.2 Question 2

Implement the CADNA library in the `jacobi.c` program by creating a new program called `jacobi_cad.cc`. Run the program with ε set to 10^{-4}. How many iterations are performed?
Print the `anorm` value. Is the ε value taken into account? Why?

What do you conclude?

2.7 Exercise 7

Let us consider the logistic iteration defined by $x_{n+1} = ax_n(1 - x_n)$ with $a > 0$ and $0 < x_0 < 1$.
A mathematically equivalent sequence is: $x_{n+1} = -a(x_n - \frac{1}{2})^2 + \frac{a}{4}$.
As a remark, the logistic iteration is a chaotic sequence, not a convergent one.

2.7.1 Question 1

Run the `logistic.c` program which computes the logistic iteration.
In the program both sequence expressions are given. Compare the results obtained.

2.7.2 Question 2

Implement the CADNA library in the `logistic.c` program by creating a new program called `logistic_cad.cc`.
Run the program and describe the results obtained.
What should be the stopping criterion to avoid useless iterations? What do you obtain with this new stopping criterion?