Assignment 1

1.) double precision (64 bit)

Vz 1 bit | Exponent 11 bit | Mantisse 52 Bit

quadruple precision

Vz 1 bit | Exponent 15 bit | Mantisse $128-1-15$ bit $= 112$ bit

Machine Epsilon:

$$\varepsilon_{bit} = b^{-(p-1)}$$

b ... Base ≙ 2

p ... precision ≙ 113

$$\varepsilon_{128} = 2^{-(113-1)} = 2^{-112} = 1{,}9259299e{-34}$$

~~Max significant digits (decimal) = 2^{113} = 5.1922969~~

Max significant digits ~~(decimal) = $\pm(2-2^{112})\cdot 2^{16383}$~~

~~range Max sig. digits~~ $= \pm(2-2^{R1})\cdot 2^{Offset}$

~~$range_{quad} = \pm(2-2^{112})\cdot 2^{16383}$~~

significant digits ≙ $\log_{10}(2^{113}) = 34{,}01638951 \approx \underline{\underline{34}}$

Why is quadruple precision more than twice as accurate?

~~Doo~~ Precision describes the amount of bits reserved for the ~~Mantissa~~ whole floating point word. ~~Doubling the bits means to double the~~ Doubling the precision means to double the AMOUNT of bits. The decimal value of a in bits represented value gets doubled, when adding one bit (e.g.):

~~8~~ $8_{10} = 1000$

$16_{10} = 10000$

Therefore doubling the digits results in a much higher accuracy.

$\varepsilon = b^{-(p-1)}$ $\Rightarrow$ The precision is an exponent for calculating the machine epsilon which describes the error.

2.)

a) $fl(a \text{ op } b) = fl(b \text{ op } a)$

true. ✓

The rounding error ~~of~~ ~~a~~ and b does not change.

b) $fl(a+a) = fl(2 \cdot a)$

true ✓

~~The rounding error that is existing with a gets doubled, no matter what.~~

a before the operation is infinitely precise ~~it's not~~

it only is rounded after one operation.

c) $fl((a+b)+c) = fl(a+(b+c))$

false. X

a+b might produce a different rounding error than b+c. ⇒ leads to a different result.

for a & b: we only have one operation

for c: we have two operations, dependant ~~for~~ on each other.

3.)

$$S_n = \sum_{i=1}^{n} x_i$$

$$\hat{s}_1 = fl(x_1) = x_1(1+\delta_1)$$

$$\hat{s}_2 = fl(\hat{s}_1 + x_2) = (\hat{s}_1 + x_2)(1+\delta_2) =$$

$$= (x_1(1+\delta_1) + x_2)(1+\delta_2) =$$

$$= x_1(1+\delta_1)(1+\delta_2) + x_2(1+\delta_2)$$

$$\Downarrow \quad \boxed{1+\delta_i = 1 \pm \delta}$$

$$\hat{s}_2 = x_1(1\pm\delta)^2 + x_2(1\pm\delta)$$

$$\hat{s}_3 = fl(\hat{s}_2 + x_3) \cancel{(1+} = (\hat{s}_2 + x_3)(1\pm\delta) =$$

$$= \underline{x_1(1\pm\delta)^3 + x_2(1\pm\delta)^2 + x_3(1\pm\delta)}$$

Backward-Error

$$\boxed{\prod_{i=1}^{n}(1+\delta_i)^{\rho_i} = 1+\theta_n}$$

rounding errors are independant!

$$\boxed{\hat{s}_n = \sum_{i=1}^{n} x_i(1\pm\delta)^{n-i+1}}$$

Backward-Error Bound:

$$\boxed{\hat{s}_n = \sum_{i=1}^{n} x_i(1+\theta_{n-i+1})}$$

$$\left|\sum_{i=1}^{n} x_i - fl\left(\sum_{i=1}^{n} x_i\right)\right| \le \gamma_n \sum_{i=1}^{n} |x_i| \le$$

$$fl(x_1, x_2) = (x_1 + \Delta x_1) + (x_2 + \Delta x_2)$$

$$\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} \Delta x_i = \boxed{\sum_{i=1}^{n} x_i + \Delta x_i} \qquad |\Delta x_i| \le \gamma_n |x_i|$$

forward Error

$$\boxed{\sum_{i=1}^{n} x_i + \gamma_n |x_i|}$$

$$\begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 3 \\ 2 & 4 & 0 \end{pmatrix}$$

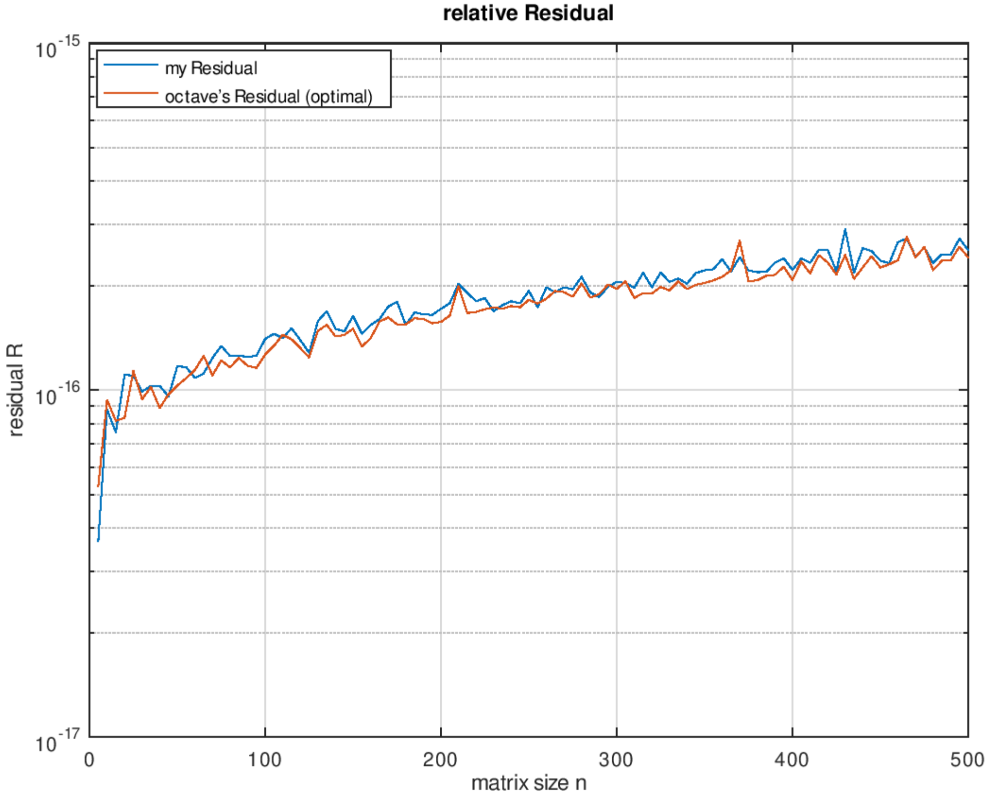

Figure 1: Error comparison of octave's built in lu function versus plu.m

Christian Lascsak
WS 2020

HW1: Numerical Algorithms

Part1

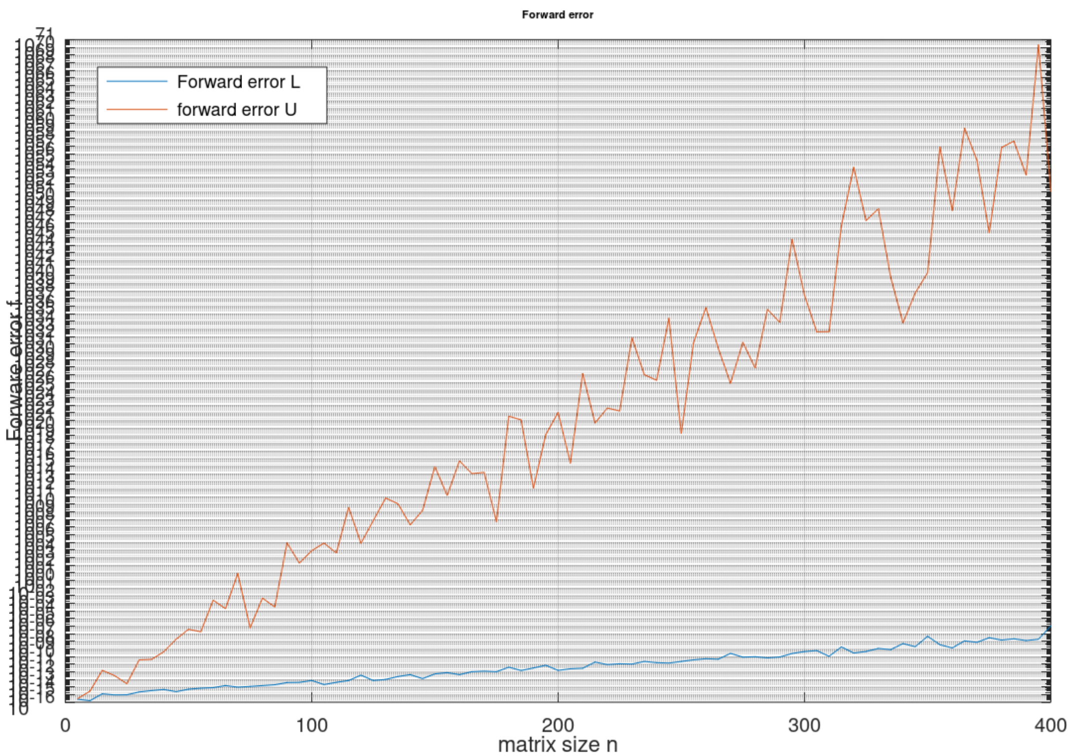

Figure 2: Forward error of solve L and solve U

Part2

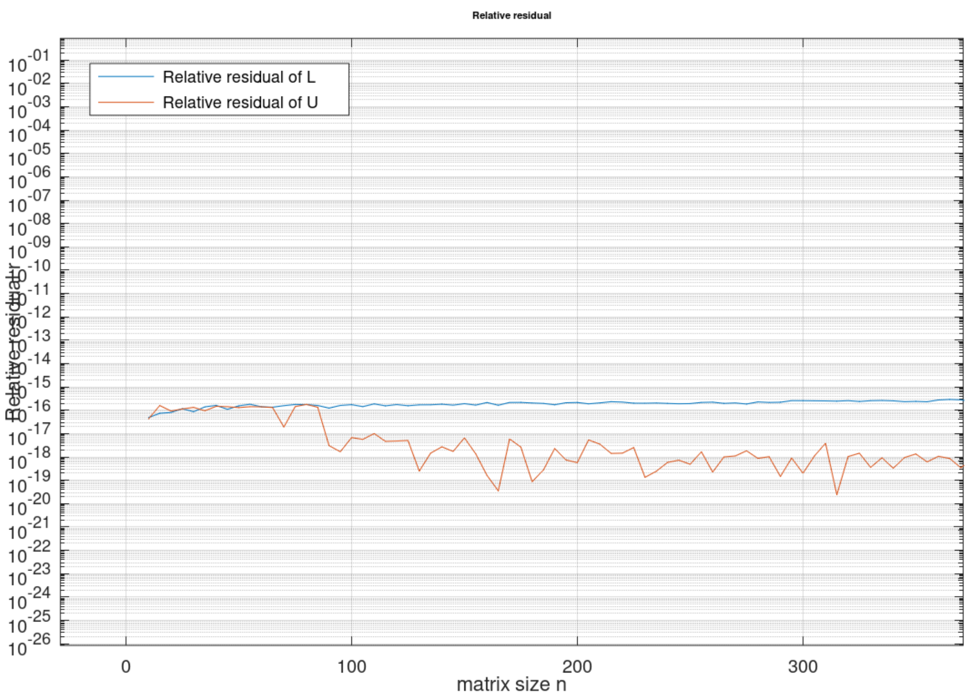

Figure 3: Residuals of solve L and solve U