Assignment-2 Bridging course

Linus Falk

14 september 2022

1 Workout 4.2

For computing the inner products $s_n = x^T y$ of vectors $x, y \in F^n$ for an even n assume $m = \frac{n}{2}, s_1 = x(1:m)^T y(1:m), s_2 = x(m+1:n)^T y(m+1:n)$ and $s_n = s_1 + s_2$ prove that forward error bound

$$\|\hat{s}_n - s_n\| \leq \gamma_{\frac{n}{2}+1} \|x\|^T \|y\| \tag{1}$$

which shows the error bound is almost halved by accumulating the inner product in two pieces. Generalize the idea and by breaking the inner product into k pieces and find the optimal k.

We start with the first multiplication in the inner product and then continue building it with more terms of products.

$$\hat{s_1} = x_1 y_1 = \text{fl}(x_1 y_1) = x_1 y_1 + (1 + \delta_1) \tag{2}$$

$$\hat{s_2} = \text{fl}(s_1 + x_2 y_2) = (s_1 + x_2 y_2 (1 + \delta_2))(1 + \delta_3) =$$

$$(x_1 y_1 + (1 + \delta_1) + x_2 y_2 (1 + \delta_2))(1 + \delta_3)$$

$$x_1 y_1 + (1 + \delta_1)(1 + \delta_3) + x_2 y_2 (1 + \delta_2)(1 + \delta_3) \tag{3}$$

We can from this pattern see that if we split the matrix in half and calculate each part we can nearly half the number of error bound. If we then generalize this idea to split the inner product into k pieces instead of two as in (1) we get:

$$\|\hat{s}_n - s_n\| \leq \gamma_{\frac{n}{k}+k+1} \|x\|^T \|y\| \tag{4}$$

The extra addition of k is because of the extra summations of parts. By examining this expression and look for the minimum we can derive what the optimal k would be: $\frac{n}{k} + k + 1$. We derive the derivative of this expression with respect to k and set i to zero to locate the optimal k.

$$\frac{d}{dk}\left(\frac{n}{k + k + 1}\right) = 1 - \frac{n}{k^2} = 0 \tag{5}$$

We can now see that $k = \sqrt{n}$ is the optimal k.

2 Workout 6.3

Consider the algebraic equation

$$x^n + ax + 1 = 0, a > 0, n \geq 2 \tag{6}$$

show that the equation has exactly one positive root $\xi(a)$. Show that ξ is well-conditioned with respect to perturbations in a

$$(\text{cond}\xi)(a) = \frac{|\xi'(a)a|}{|\xi(a)|} \tag{7}$$

Since can't put ξ in explicit form and we use implicit differentiation to compute $\xi'(a)$

$$n\xi' \left[\xi(a)\right]^{n-1} + \xi(a) + a\xi'(a) = 0 \tag{8}$$

$$n\xi' \left[\xi(a)\right]^{n-1} + a\xi'(a) = -\xi(a)$$

$$\xi' = -\frac{\xi(a)}{\left[\xi(a)\right]^{n-1} + a}$$

Put in (5)

$$\frac{\frac{\xi(a)a}{\left[\xi(a)\right]^{n-1}+a}}{\xi(a)} = \frac{\xi(a)a}{\left[\xi(a)\right]^{n} + a\xi(a)} \tag{9}$$

For $a > 0$ and $n \geq 2$ we have proved that ξ is well-conditioned. In the next step we look at the derivative of the function to investigate if this is the only root,

$$\frac{d}{dx}x^n + ax + 1 = \frac{nx^n}{x} + x \tag{10}$$

For $a > 0$ and $n \geq 2$ the derivative will always be positive and therefor we have only one root: $x = 0$.

3 Workout 6.8

Show that

$$0.1 = \sum_{k=1}^{\infty} 2^{-4k} + 2^{-4k-1} \tag{11}$$

and conclude $(0.1)_{10} = (0.000\overline{1111})_2$**. The last four binary digits are repeated. In the IEEE standard with single precision show that if** $\hat{x} = fl\,(0.1)$ **then** $\frac{x-\hat{x}}{x} = \frac{1}{4}u$

$$\sum_{n=4}^{\infty} \frac{1}{2}^{4k} + \frac{1}{2}^{4k+1} = \tag{12}$$

$$\sum_{k=1}^{\infty} \frac{1}{2^4}^{k} + \frac{1}{2}\sum_{k=1}^{\infty} \frac{1}{2^4}^{k} = \frac{1}{1-\frac{1}{16}} + \frac{1}{2}\frac{1}{1-\frac{1}{16}} = \frac{1}{15} + \frac{1}{30} = 0.1$$

4 Workout 6.6

In the IEEE standard with double precision determine an interval when an interval there at the distance between the floating-point numbers is exactly = 1.

We start with interval of the floating point numbers: $2^e - 2^{e-1} = 2^e$. This interval got gaps that are determined with the precision: 2^{p-1}, subtracting one because we count the spaces. We now divide the interval with the number of gaps to get the spacing between the floating point numbers: $\frac{2^e}{e^p-1} = 2^{e-p+1}$ setting this expression to the distance of 1 that we were looking for: $1 = 2^{e-p+1}$. Knowing the precision of IEEE is equal to 53 yields: $1 = 2^{e-53+1}$. Clearly e is 52 and by plugging this into the first interval calculation we have the interval $[2^{52}, 2^{53})$

5 Lab 1 exercise 4

The inverse of the hyperbolic cosine is defined as:

$$\cosh^{-1} x = \log\left(x \pm \sqrt{x^2 - 1}\right) \tag{13}$$

Show by computing in python that this formula suffers from serious cancellation when the minus sign is used and x is large. A better formula is:

$$\cosh^{-1} x = 2\log(\sqrt{\frac{x+1}{2}} + \sqrt{\frac{x-1}{2}}) \tag{14}$$

Derive this formula and show by computing in python that it is well behaved.

$$(5) = \log(\sqrt{\frac{x+1}{2}} + \sqrt{\frac{x-1}{2}})^2$$

$$= \log(\frac{x+1}{2} + 2\sqrt{\frac{x+1}{2}}\sqrt{\frac{x-1}{2}} + \frac{x-1}{2})$$

$$= \log(x + \sqrt{x+1}\sqrt{x-1})$$

$$= \log(x + \sqrt{x^2-1})$$

5.1 Python script

```
import matplotlib.pyplot as plt
import numpy as np
import math

def cosh(x):
    return (np.exp(x)+np.exp(-x))/2


def coshinv(x):
    return np.log(x - np.sqrt((x**2) - 1))


def coshinv_new(x):
    return 2*np.log(np.sqrt((x+1)/2)+np.sqrt((x-1)/2))

print(coshinv(cosh(10)))
-10.000000013503529
print(coshinv_new(cosh(10)))
10.0
```