Problem Sheet 1

Numerical Methods for CSE

David Bimmler

1. Arrow matrix-vector multiplication

1.a. Matrix A

$$A = \begin{bmatrix} d_1 & 0 & 0 & \dots & a_1 \\ 0 & d_2 & 0 & \dots & a_2 \\ 0 & 0 & d_3 & \dots & a_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & a_3 & \dots & d_n \end{bmatrix}$$

1.b. `tic-toc` Results

We can clearly see that the asymptotic complexity of the function `arrowmatvec` lies in $O(n^3)$. In the first few measurements, the actual multiplication is being dominated by other factors, such as the calls to length and the creation of the matrix A. With increasing input size n, though, these factors become have a lesser impact, and most of the time is spend calculating the product, which is why the runtime approaches the graphed line.

From the code we can see why $O(n^3)$ is the expected asymptotic complexity: the matrix A gets multiplied with itself, which is an operation in $O(n^3)$ as the matrices are all square and of size n.

1.c. Efficient Reimplementation

1.d. Complexity of Reimplementation

1.e. Runtime Comparison

1.f. Eigen implementations

2. Avoiding Cancellation

2.a. Behaviour of the Error

2.a.1. Derivation of f_2

$$f_1(x_0, h) \coloneqq \sin(x_0 + h) - \sin(x_0) \tag{1}$$

$$\sin(x_0 + h) - \sin(x_0) = 2\cos(x_0 + \frac{h}{2})\sin(\frac{h}{2}) =: f_2(x_0, h) \tag{2}$$

2.a.2. Approximation for $f'(x)$

$$f'(x) \approx \frac{f(x_0 + h) - f(x_0)}{h} = \frac{f_2(x_0, h)}{h} \tag{3}$$

The following code implements this formula:

```
function res = nme1p2a
    x = 1.2;
    res = zeros(21,2);
    for i = -20:0
        h = 10^i;
        res(i+21,:) = [f2(x,h)/h,f1(x,h)/h];
    end

function y = f1(x0,h)
y = sin(x0 + h) - sin(x0);

function y = f2(x0, h)
y = 2 * cos(x0 + (h/2)) * sin(h/2);
```

`f1` is only included here to make the plotting easier afterwards.