MA3227 Numerical Analysis II

Assignment 1

Deadline: 25 February 2021, 12 noon
Total marks: 20

1 Newton's method for computing $1/y$ [10 marks]

This assignment illustrates how we can use Newton's method to compute $1/y$ using only addition and multiplication. Being able to do so is occasionally useful because addition and multiplication are often easier to implement than inversion.

The key challenge in this endeavour is to determine a function $f : \mathbb{R} \to \mathbb{R}$ such that $f(\frac{1}{y}) = 0$. A simple example of such a function is $f(x) = xy - 1$.

1. [1 mark] Show that Newton's method applied to $f(x) = xy - 1$ is given by
$$x_{k+1} = \tfrac{1}{y}. \tag{1}$$

2. [2 mark] The formula derived in Task 1 means that Newton's method immediately discards the initial guess x_0 and jumps straight to the exact solution $x = \frac{1}{y}$. Explain in one or two sentences why Newton's method exhibits this behaviour for all linear functions $f(x) = ax + b$. Your answer should be based mainly on a verbal description rather than formulae.

While convergent in a single step, the iteration (1) cannot be evaluated without computing $\frac{1}{y}$ and is therefore not useful for our purposes. Hence, let us try another function $f(x)$.

3. [1 mark] Show that Newton's method applied to the function $f(x) = \frac{1}{x} - y$ is given by
$$x_{k+1} = x_k\,(2 - y\,x_k). \tag{2}$$

This iteration indeed requires only addition and multiplication as desired. We therefore move on to studying the convergence properties of (2).

4. [1 mark] Show that (2) implies the error recursion
$$x_{k+1} - \tfrac{1}{y} = -y\,\big(x_k - \tfrac{1}{y}\big)^2. \tag{3}$$

If we assume an initial error $|x_0 - \frac{1}{y}|$ independent of y, then (3) implies that the error $|x_1 - \frac{1}{y}|$ after one iteration will be smaller for smaller y. In order to avoid this unequal error reduction, we multiply both sides of (3) by y to obtain
$$y\,x_{k+1} - 1 = -\big(y\,x_k - 1\big)^2.$$
This shows that (2) converges as long as the weighted initial error $|y\,x_0 - 1| = |y|\,|x_0 - \frac{1}{y}|$ is less than one.

One can show that $|y\,x_0(y) - 1| \le \frac{1}{17}$ for all $y \in [1, 2]$ if $x_0(y)$ is given by
$$x_0(y) = \tfrac{1}{17}\,(24 - 8y).$$

Under these assumptions, Newton's method is hence guaranteed to converge and we can easily compute the maximal number of iterations required to reach machine precision.

5. [2 mark] Determine the smallest integer K such that for all $y \in [1, 2]$ we have

$$\left|x_K - \tfrac{1}{y}\right| \leq 10^{-15}$$

where

$$x_0 = \tfrac{1}{17}\,(24 - 8y) \qquad \text{and} \qquad x_{k+1} = x_k\,(2 - y\,x_k).$$

Finally, let us compare the above method for computing $1/y$ against the bisection method and the function `inv(y)` provided by Julia.

6. [1 mark] Determine the smallest bracketing interval $[a_0, b_0]$ for the function $f(x) = xy - 1$ assuming all we know is that $y \in [1, 2]$.
7. [1 mark] Determine the minimal number of bisection steps K required to reduce the initial bracketing interval $[a_0, b_0]$ determined in Task 6 to a bracketing interval $[a_K, b_K]$ such that

$$|b_K - a_K| \leq 2 \times 10^{-15}.$$

8. [unmarked] Fill in the blanks in the provided functions `newton_inv()` and `bisection_inv()` and run `benchmark()` to measure the runtime of these functions. On my machine, this leads to the following output:

```
      inv runtime:   1.37 nanoseconds
   Newton runtime:   2.70 nanoseconds
Bisection runtime: 206.67 nanoseconds
```

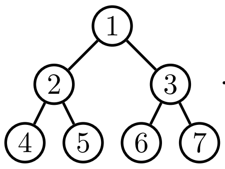
   Note how we have just derived an algorithm which comes within a factor two of Julia's built-in `inv(y)` function. This is an incredible achievement given that `inv(y)` is based on a hardware instruction on your CPU! (You can see this by typing `@code_native inv(1.0)` on the REPL. The `vmovsd` instruction loads `1.0` into a CPU register, `vdivsd` executes the division.)

9. [1 mark] We observe that Newton's method is roughly $206/2.7 \approx 75$ times faster than the bisection method. Relate this observation to what we have seen above. (This is an open-ended question. Try to answer as comprehensively yet concisely as possible.)

— Continued on next page —

2 Fill path theorem and nested dissection [10 marks]

Consider a sparse symmetric matrix A whose associated graph is given by

$G(A) =$ 

.

Note that each undirected edge $\circ\!-\!\circ$ represents a pair of directed edges $\circ\!\rightleftarrows\!\circ$ in this picture.

1. [3 marks] Determine the sparsity pattern of the LU factorisation of A. State your result as a single matrix where the diagonal entries are numbered 1 to 7, nonzero entries in A are marked with $\bullet$ and fill-in entries in $L + U$ are marked with `x`.
   You do not have to motivate your answer.
2. [3 marks] Explain why $V_{\text{sep}} = \{1\}$ is a good separator for the first step of the nested dissection recursion. (This is an open-ended question. Try to answer as comprehensively yet concisely as possible.)
3. [2 marks] Draw a copy of the graph of A where the vertices are numbered such that LU factorisation of the corresponding matrix does not incur any fill-in.
   You do not have to motivate your answer.
4. [2 marks] Write down the sparsity pattern of PAP^T where P is the matrix associated with the permutation determined in Task 3.