Numerical methods in scientific computing 2023

Exercise 5
Return by Monday 20.2.2023 23:59 to Moodle
Exercise session: Friday 24.2.2023

Problem 1. (*computer*) (6 points)
Broyden's update is a method for solving systems of nonlinear equations. It is in a way extension of the secant method to multiple dimensions. In the method the exact inverse of the Jacobian is only needed in the beginning of the iteration. During the iteration the initial inverse Jacobian (B_i) is updated such that only matrix–vector or matrix–matrix multiplications are needed (i.e. no matrix decompositions and such).

The iteration goes like this

(1) Set the initial guess for the root x_0 and the initial inverse Jacobian $B_0 \leftarrow J^{-1}(x_0)$
(2) Loop over $i=1..i_{\max}$
   1. $x_i \leftarrow x_{i-1} - B_{i-1} f(x_{i-1})$
   2. $s_i \leftarrow x_i - x_{i-1}$
   3. $y_i \leftarrow f(x_i) - f(x_{i-1})$
   4. $B_i \leftarrow B_{i-1} + \dfrac{(s_i - B_{i-1} y_i) s_i^T B_{i-1}}{s_i^T B_{i-1} y_i}$

A) Write a function `broyden(x_init)` that finds and returns the root vector of the two-dimensional problem

$$f_1(x_1, x_2) = f_2(x_1, x_2) = 0,$$

with

$$f_1(x_1, x_2) = \exp\left(-(x_1^2 + x_2^2)\right) - \frac{1}{8},$$

$$f_2(x_1, x_2) = \sin(x_1) - \cos(x_2)$$

using the Broyden's method with initial guess vector `x_init`. Do not use any library implementations. Your function should return the resulting root vector and be placed in a file named “`broyden`”.

B) Show that your function gives correct results by solving the equations using some other tool (SciPy, Matlab, SLATEC,…)

Problem 2. *(pen and paper-computer)* (6 points)
By means of various identities, it is often possible to reduce the interval on which a function needs to be approximated.

A) Show how to reduce each of the following functions from $-\infty < x < \infty$ to the given interval.

(a) e^x, $-1/2 \leq x \leq 1/2$
(b) $\cos(x)$, $0 \leq x \leq \pi/4$
(c) $\arctan(x)$, $0 \leq x \leq 1$
(d) $\ln(x)$, $1 \leq x \leq 2$, reduce from $\ln(x)$, on $0 < x < \infty$.

B) Use this mapping to write in an efficient way your own exponential function `myexp(x)` that calculates and returns e^x for a double precision real number `x`. Put your function to a file named "`exp`".

Problem 3. (*pen and paper – you can use any symbolic math program, e.g. Maxima, Maple, Mathematica, Sagemath, SymPy etc.*) (6 points)
Determine the 2nd degree interpolating polynomial of function

$$f(x) = \frac{1}{1+x^2}$$

based on points $x_1 = 0$, $x_2 = 1$, and $x_3 = 3$ using
(a) Lagrange's polynomials (ch. 6, p. 7 in lecture notes)
(b) Newton's method (ch. 6, p. 23).

Show that the polynomials are the same. Plot the function and the polynomial in the interval $x \in [-0.5, 3.5]$.

Problem 4. *(pen and paper– you can use any symbolic math program, e.g. Maple, Mathematica, Maxima, Sagemath, SymPy etc.)* (6 points)
Let $x_{0,} x_{1,} \ldots, x_N$ be distinct real points, and consider the following interpolation problem: Choose a function

$$P_N(x) = \sum_{j=0}^{N} c_j e^{jx}$$

such that $P_N(x_i) = y_i$, $i = 0, 1, \ldots, N$, with the $y_{0,} y_{1,} \ldots, y_N$ given data. Show that there is a unique choice of coefficients $c_{0,} c_{1,} \ldots, c_N$. *Hint: The problems can be reduced to that of ordinary polynomial interpolation.*