MA3227 Numerical Analysis II

Lab Session 4

Simon Etter, 2019/2020

1 Newton's method for complex square roots

Let $f : \mathbb{R}^N \to \mathbb{R}^N$ be a continuously differentiable function. Newton's method for finding $x^\star \in \mathbb{R}^N$ such that $f(x^\star) = 0$ iteratively improves an initial guess $x_0 \in \mathbb{R}^N$ using the updates

$$x_{k+1} = x_k - \nabla f(x_k)^{-1} f(x_k)$$

where $\nabla f(x)$ denotes the Jacobian matrix

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x) & \dots & \frac{\partial f_1}{\partial x_N}(x) \\ \vdots & & \vdots \\ \frac{\partial f_N}{\partial x_1}(x) & \dots & \frac{\partial f_N}{\partial x_N}(x) \end{pmatrix}.$$

Use Newton's method to compute the complex square root of $w + \iota v \in \mathbb{C}$, i.e. to find $x + \iota y \in \mathbb{C}$ such that

$$(x + \iota y)^2 = w + \iota v \qquad \iff \qquad \begin{cases} x^2 - y^2 = w \\ \quad 2xy = v \end{cases}.$$

Use $(x_0, y_0) = (w, v)$ as initial guess, and terminate the iteration once

$$\big|(x + \iota y)^2 - (w + \iota v)\big| \le 10\, \varepsilon_{\text{mach}}\, |x + \iota y|$$

where $\varepsilon_{\text{mach}}$ denotes machine precision. Throw an error if the iteration does not terminate after 20 steps.

Hint. The proposed algorithm does not converge if $w + \iota v$ is close to the negative real axis.

Julia syntax.

- `im` denotes the imaginary unit. Example: `im^2 -> -1+0im`.
- `eps()` denotes machine precision. Example: `eps() -> 2.220446049250313e-16`.

2 Guaranteed convergence of Newton's method for a single equation

Prove pictorially that Newton's method converges monotonically to a root $x^\star \in \mathbb{R}$ if the function $f : \mathbb{R} \to \mathbb{R}$ and initial guess $x_0 \in \mathbb{R}$ are such that there exists a root $x^\star \in \mathbb{R}$ and

$$\operatorname{sign}(f(x_0)) = \operatorname{sign}(f''(x)) \text{ for all } \begin{cases} x \in (x_0, x^\star) & \text{if } x_0 < x^\star, \\ x \in (x^\star, x_0) & \text{if } x^\star < x_0. \end{cases}$$