

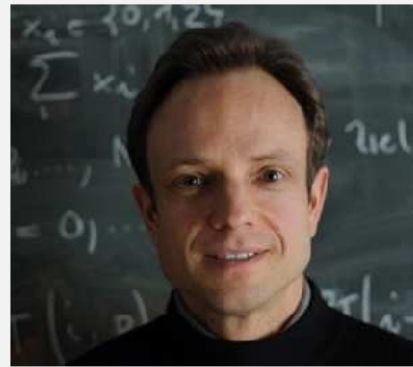
Course Video

Section 8.5.1: The Newton Iteration in $\mathbb{R}^n$ (I)

Prof. R. Hiptmair, SAM, ETH Zurich

Date: December 5, 2020

(C) Seminar für Angewandte Mathematik, ETH Zürich



Prerequisites.

- Multi-dimensional calculus: Jacobians

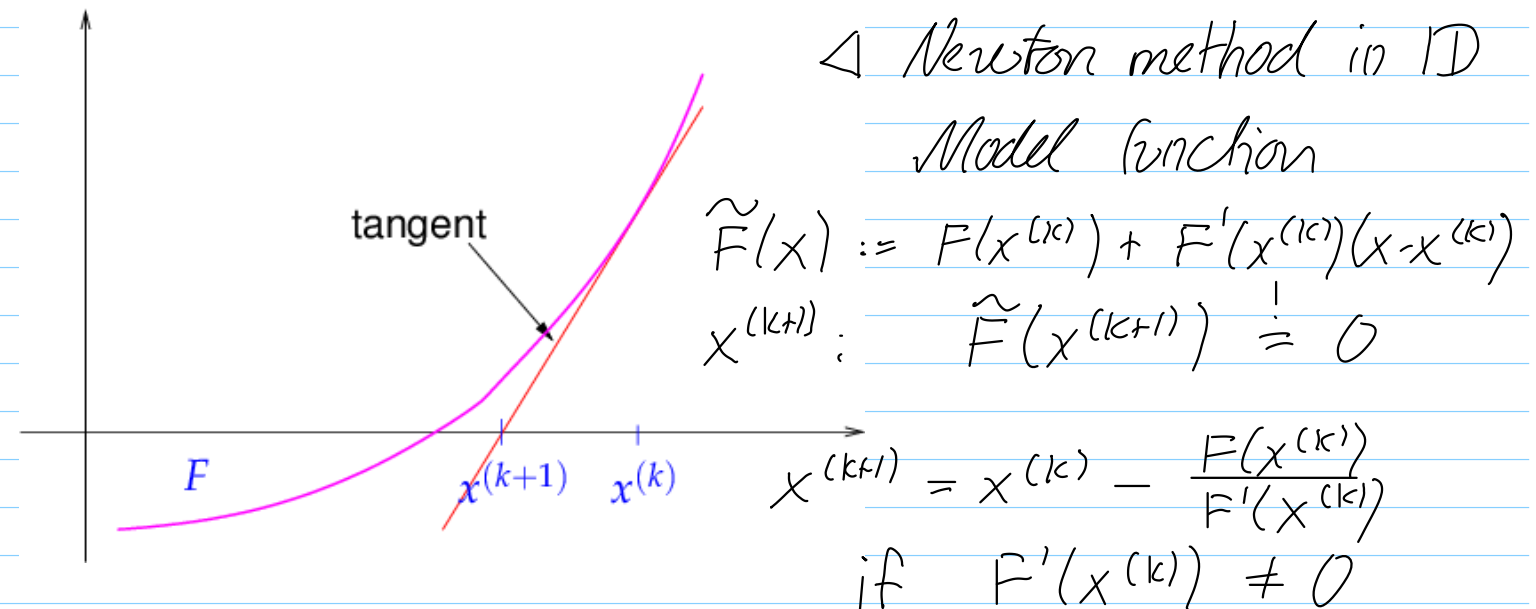
Dependencies. Section 8.4.2.1

Duration: 16 minutes

VIII Iterative Methods for Non-Linear Systems of Equations

8.5 Newton's Method in $\mathbb{R}^n$ NLSE : Find $x^* \in D \subset \mathbb{R}^n$: $F(x^*) = 0$ Assume $F : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ continuous & differentiable

8.5.1. The Newton Iteration



②

$\tilde{F} \triangleq$ truncated Taylor exp. of F around $x^{(k)}$:
a **local linearization**

▷ Same $\mathbb{R}^n$: $[F : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n]$

$$\tilde{F}(x) := F(x^{(k)}) + DF(x^{(k)})(x - x^{(k)})$$

↑ Jacobian $\in \mathbb{R}^{n,n}$: regular!

$$\tilde{F}(x^{(k+1)}) = 0 : x^{(k+1)} = x^{(k)} - DF(x^{(k)})^{-1} F(x^{(k)})$$

Idea ($\rightarrow$ Section 8.4.2.1):

local linearization:

Given $x^{(k)} \in D \supset x^{(k+1)}$ as zero of affine linear model function

$$F(x) \approx \tilde{F}_k(x) := F(x^{(k)}) + DF(x^{(k)})(x - x^{(k)}),$$

$$DF(x) \in \mathbb{R}^{n,n} = \text{Jacobian}, DF(x) := \left[\frac{\partial F_j}{\partial x_k}(x) \right]_{j,k=1}^n.$$



► **Newton iteration:** (generalizes (8.4.2.1) to $n > 1$)

$$x^{(k+1)} := x^{(k)} - DF(x^{(k)})^{-1} F(x^{(k)}), \quad [\text{if } DF(x^{(k)}) \text{ regular}] \quad (8.5.1.6)$$

Terminology: $\underline{s} := -DF(x^{(k)})^{-1} F(x^{(k)})$ is called the **Newton correction**

solve $n \times n$ LSE: $DF(x^{(k)}) \underline{s} = -F(x^{(k)})$

§ 8.5.1.8 (Newton's method in C++)

C++11-code 8.5.1.9: Newton's method in C++

```
2 template <typename FuncType, typename JacType, typename VecType>
3 VecType newton(FuncType &&F, JacType &&DFinv, VecType x, const double rtol,
4               const double atol) {
5     // Note that the vector x passes both the initial guess and also
6     // contains the iterates
7     VecType s(x.size()); // Vector for Newton corrections
8     // Main loop
9     do {
10         s = DFinv(x, F(x)); // compute Newton correction
11         x = x + s;           // compute next iterate
12     } while ((s.norm() > rtol * x.norm()) && (s.norm() > atol));
13     return x;
14 }
```

Newton

correction based termination (relative and absolute)

relative tolerance

absolute tolerance

Rem. 8.5.1.11 (Affine invariance of Newton method)

Given F consider the Newton iteration

$$G_A(x) = 0, \quad G_A(x) = A \cdot F(x), \quad A \in \mathbb{R}^{n,n} \text{ regular}$$

$$[DG_A(x) = A \cdot DF(x) : \text{product rule}]$$

③

$$\begin{aligned}
 \text{N.I. : } x^{(k+1)} &= x^{(k)} - [A \cdot DF(x^{(k)})]^{-1} A F(x^{(k)}) \\
 &= x^{(k)} - DF(x^{(k)})^{-1} \cancel{A^{-1} A} F(x^{(k)}) \\
 &\rightarrow \text{same as N.I. for } F \quad \forall A
 \end{aligned}$$

▷ Stopping rules should be affine invariant



True for code 8.5.1.9 ?

▷ Review questions 8.5.1.46

4

5

7