Numerische Mathematik

§0 Einleitung

Prof. Dr. Georg Umlauf

H T W G
Hochschule Konstanz
Technik, Wirtschaft und Gestaltung

Wer bin ich?



Prof. Dr. Georg Umlauf

− Computergrafik, geometrische Algorithmen −

- Raum: E 407
- Tel.: 07531 / 206 702
- Sprechzeiten: Jederzeit, wenn die Türe auf ist…
- Email: **umlauf@htwg-konstanz.de**
- URL: **http://www.ios.htwg-konstanz.de/**

Vorlesung und Übung

Vorlesung

- Montag: 11:30 – 13:00
- Raum: F023

Übung

- Theoretische Übungen kapitelweise online
- Programmierübung, siehe nächste Seite

Vorlesung und Übung

Übung (cont.)

Zur **Übung** und **Vorbereitung der mündlichen Prüfungen** vier alternative Themen zur Implementierung in einer beliebigen Programmiersprache:

- **Newton-Iteration** zur Lösung von univariaten, nicht-linearen Nullstellen-problemen mit **Dämpfung**.
- **Gauß-Elemination** zur Lösung von quadratischen, linearen Gleichungs-systemen mit **Total-Pivotisierung**.
- **LR-Zerlegung** von quadratischen, linearen Gleichungssystemen mit **Nachiteration**.
- Kubische, univariate, nicht-uniforme **Spline-Interpolation** mit **Not-A-Knot-Randbedingungen**.

Bei der Implementierung müssen die **Kernalgorithmen** eigenständig, ohne Zuhilfenahme externer Bibliotheken implementiert werden.

Skriptum und Prüfung

Skriptum

- Skript von Helmuth Grieb (FH Esslingen) online:

  https://moodle.htwg-konstanz.de/moodle/course/view.php?id=794

- Stoff per Tafelanschrieb o.ä.

Prüfung

- Mündliche Prüfung mit individuell vereinbartem Termin in der Prüfungszeit.
  - Planung der Termine nach der Anmeldung im Dezember.

Inhalt der Vorlesung

§0. Einleitung
§1. Rundungsfehler
§2. Nichtlineare Gleichungen
§3. Lineare Gleichungssysteme
§4. Überbestimmte lineare Gleichungssysteme
§5. Interpolation
§6. Approximation

Literatur

Dahmen/Reusken:
Numerik für Ingenieure und Natruwissenschaflter, 2. Auflage, Springer, 2008.

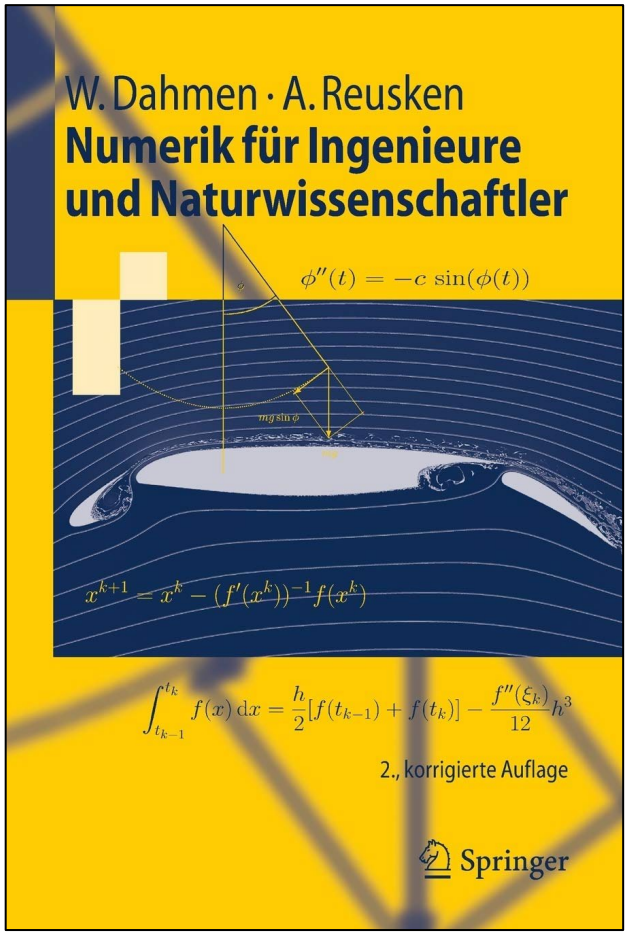
Preis: ca. 35,00€.



Wolfgang Dahmen



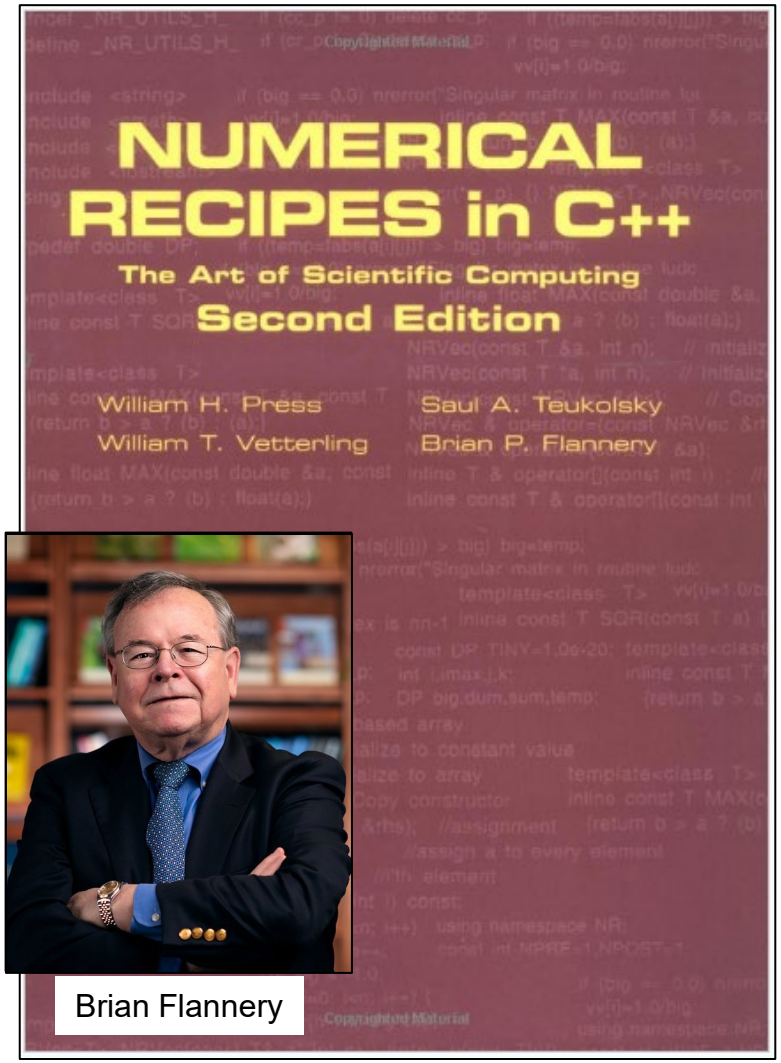
Arnold Reusken



Literatur

Press/Teukolsky/Vettering/Flannery:
Numerical Recipes in C++,
Cambridge University Press, 2002.

Preis: ca. 60,00€ bzw. 30,00€





William Press



Saul Teukolsky



William Vetterling

Brian Flannery

Literatur

Stoer/ Bulirsch:
Numerische Mathematik 1,
10. Auflage, Springer, 2007.

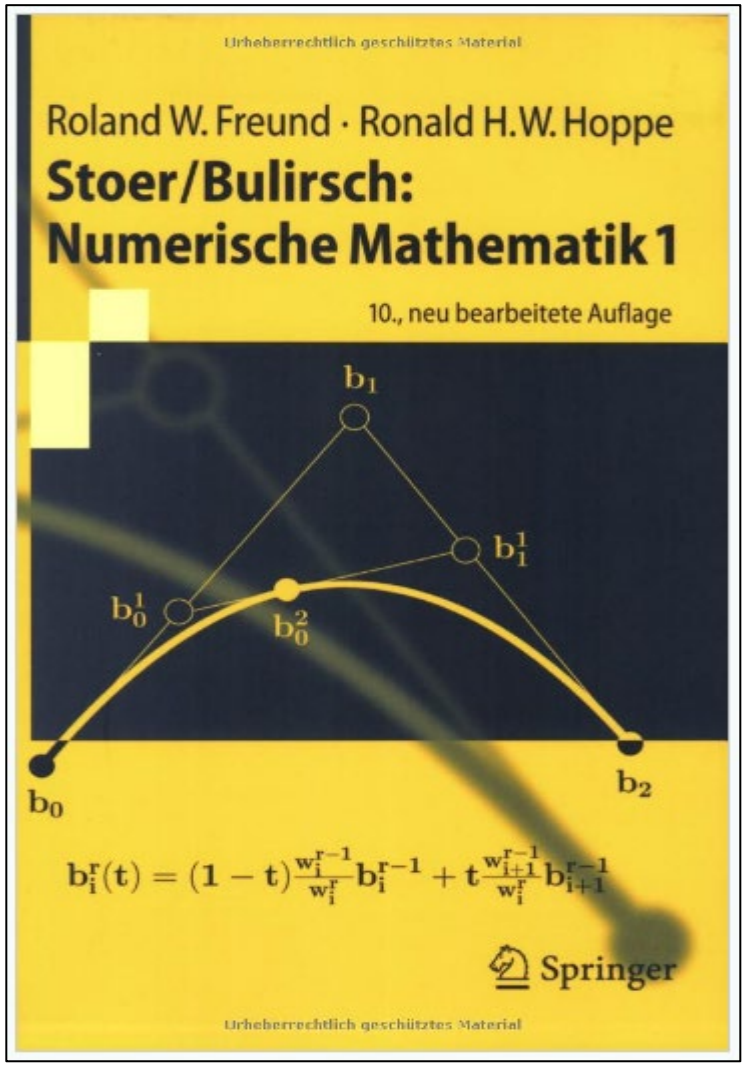
Preis: ca. 25,00€.



Josef Stoer



Roland Bulirsch



Inhalt der Vorlesung

§0. Einleitung
§1. Rundungsfehler
§2. Nichtlineare Gleichungen
§3. Lineare Gleichungssysteme
§4. Überbestimmte lineare Gleichungssysteme
§5. Interpolation
§6. Approximation

Inhalt der Vorlesung

§0. Einleitung
- Was ist numerische Mathematik?
- Forderungen an die numerische Mathematik
- Fehlerarten
- Formalisierung der Forderungen

§1. Rundungsfehler

§2. Nichtlineare Gleichungen

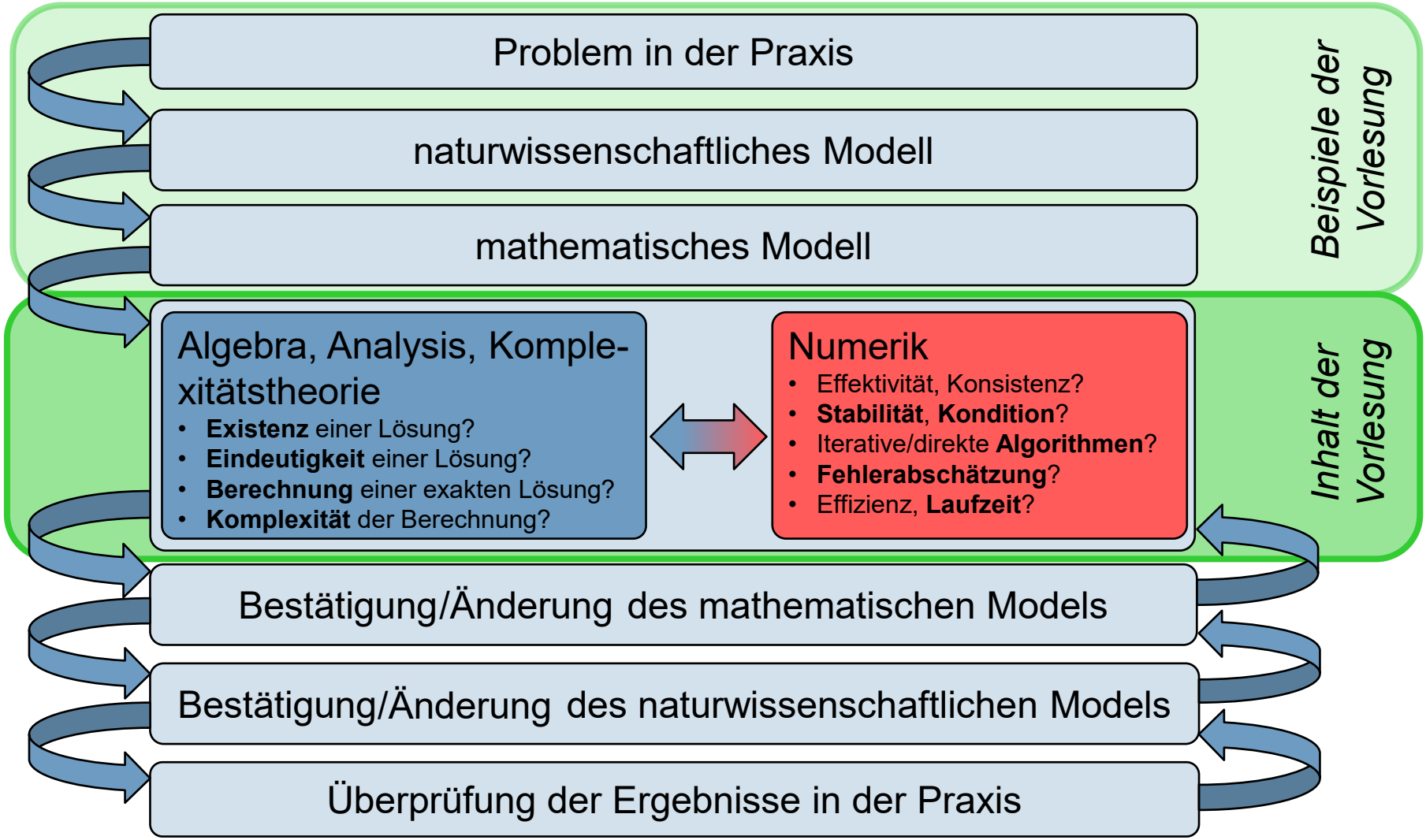
§3. Lineare Gleichungssysteme

§4. Überbestimmte lineare Gleichungssysteme

§5. Interpolation

§6. Approximation

Was ist Numerische Mathematik?



Forderungen an die numerische Mathematik

- Verfahren müssen als Algorithmus darstellbar sein:
  - verwendet elementare Rechenoperationen,
  - liefert aus Eingabedaten nach endlich vielen Schritten die Ergebnisdaten,
  - logisch und arithmetisch korrekt, z.B. keine Division durch Null,
  - Fehlerschranken für die Resultate, um die Ergebnisse garantieren bzw. bewerten zu können.

  - Man unterscheidet zwischen
    - direkten Algorithmen, z.B. Gauß-Elemination, Mitternachtsformel,
    - iterativen Algorithmen, z.B. Newton-Verfahren, Back-propagation.

Forderungen an die numerische Mathematik

- Analyse numerischer Verfahren
  - Anzahl benötigter Rechenoperationen (**Laufzeit**, Aufwand, run-time),
  - benötigter **Speicherplatz**,
  - Parallelisierbarkeit (GPGPU, Berechnungszeit, Ausführungszeit, execution-time),
  - **Stabilität**, d.h. Empfindlichkeit gegenüber Fehlern in den Eingabedaten,
  - **Fehlerschranken** für die Resultate,
  - **Approximationsgüte**, insb. bei Iterationsverfahren.

Fehlerarten

- Fehler in den Eingabedaten — Problem in den Daten
  - Messungenauigkeiten, Erhebungsfehler, Abtastrate.
- Rundungsfehler
  - Dem Rechner stehen nur endlich viele Stellen zur Verfügung, aka Quantisierung.
- Abbruchfehler — Problem der Methode
  - Der Rechner muss die Berechnung nach endlich vielen Schritten abbrechen.
- Diskretisierungsfehler
  - Am Rechner werden stetige Funktionen oft durch endlich viele diskrete Werte ersetzt, aka Schrittweite.
- Menschlicher Irrtum, maschinelle Fehlfunktion
  - Kein Gegenstand der Numerik, Kontrollmaßnamen.

Hat Einfluss auf die Numerik

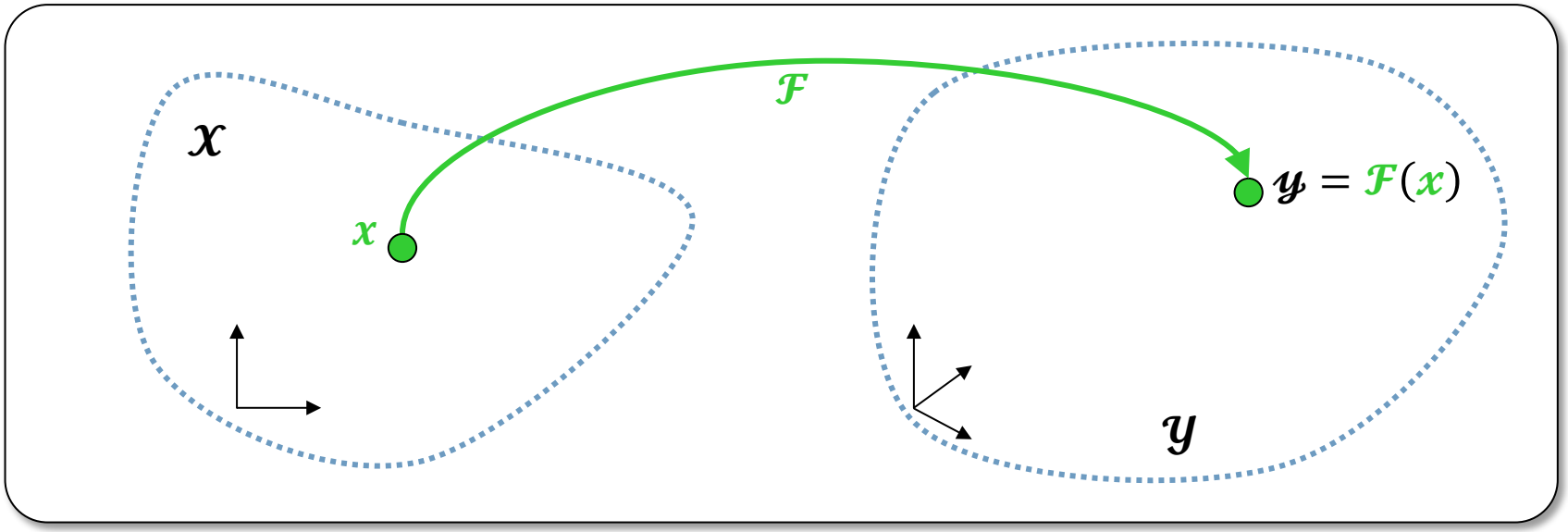
Zentrale Aufgabe der Numerik

Außerhalb der Numerik

Formalisierung der Forderungen

- Ein **mathematisches Modell** $\mathcal{F}: \mathcal{X} \to \mathcal{Y}$ hat eine
  - **Eingabe** (aka *input*, „*gegeben*“) $x \in \mathcal{X}$

  und liefert dafür eine
  - **Ausgabe** (aka *output*, „*gesucht*“) $y = \mathcal{F}(x) \in \mathcal{Y}$.



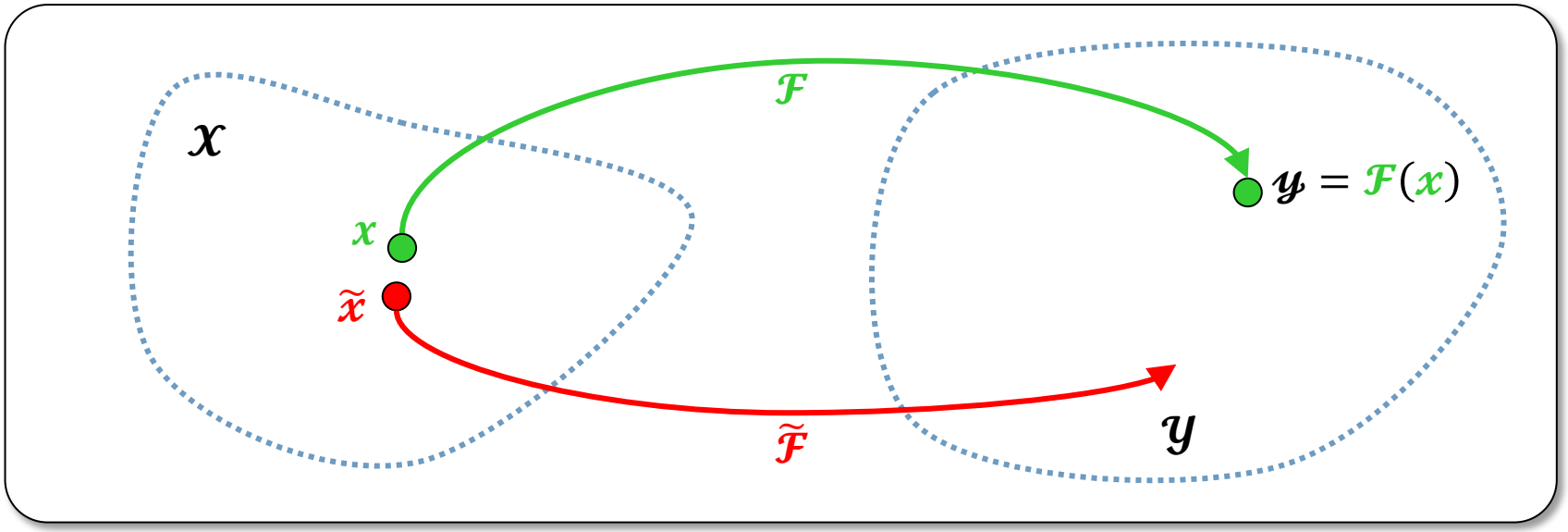
Formalisierung der Forderungen

ABER

- Die Eingabe x ist nur fehlerbehaftet (aka *gestört*) gegeben

und

- das mathematische Modell $\mathcal{F}$ ist u.U. nicht exakt, nicht effizient oder nicht geschlossen (effektiv) lösbar.

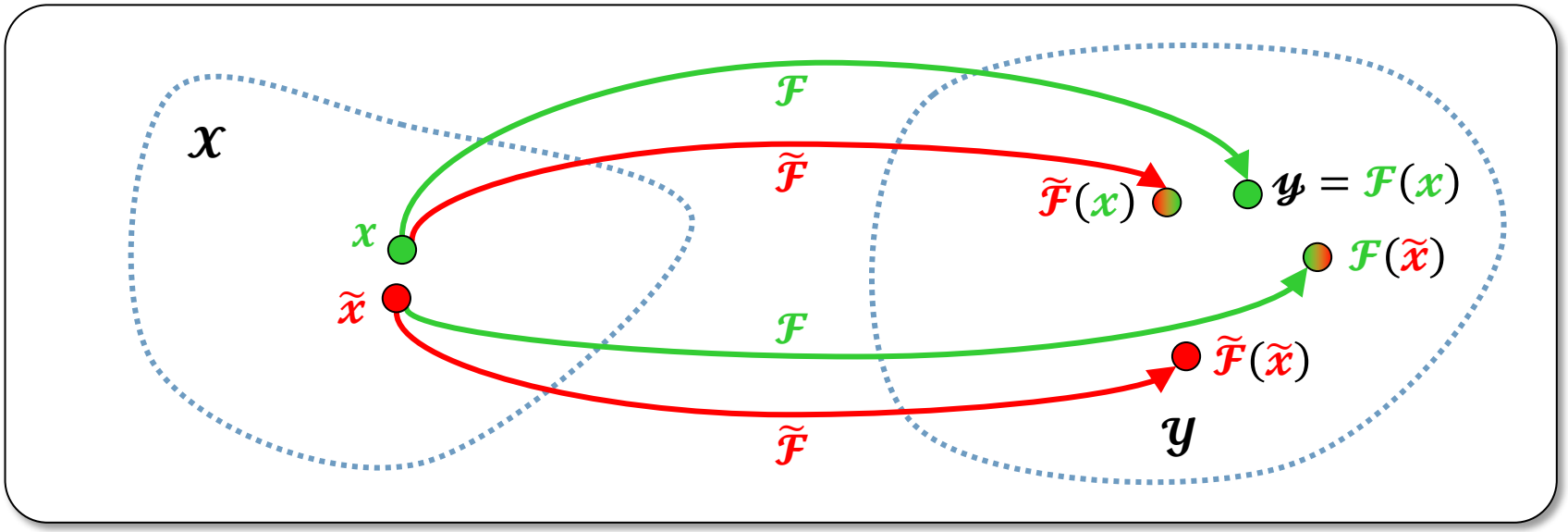


Formalisierung der Forderungen

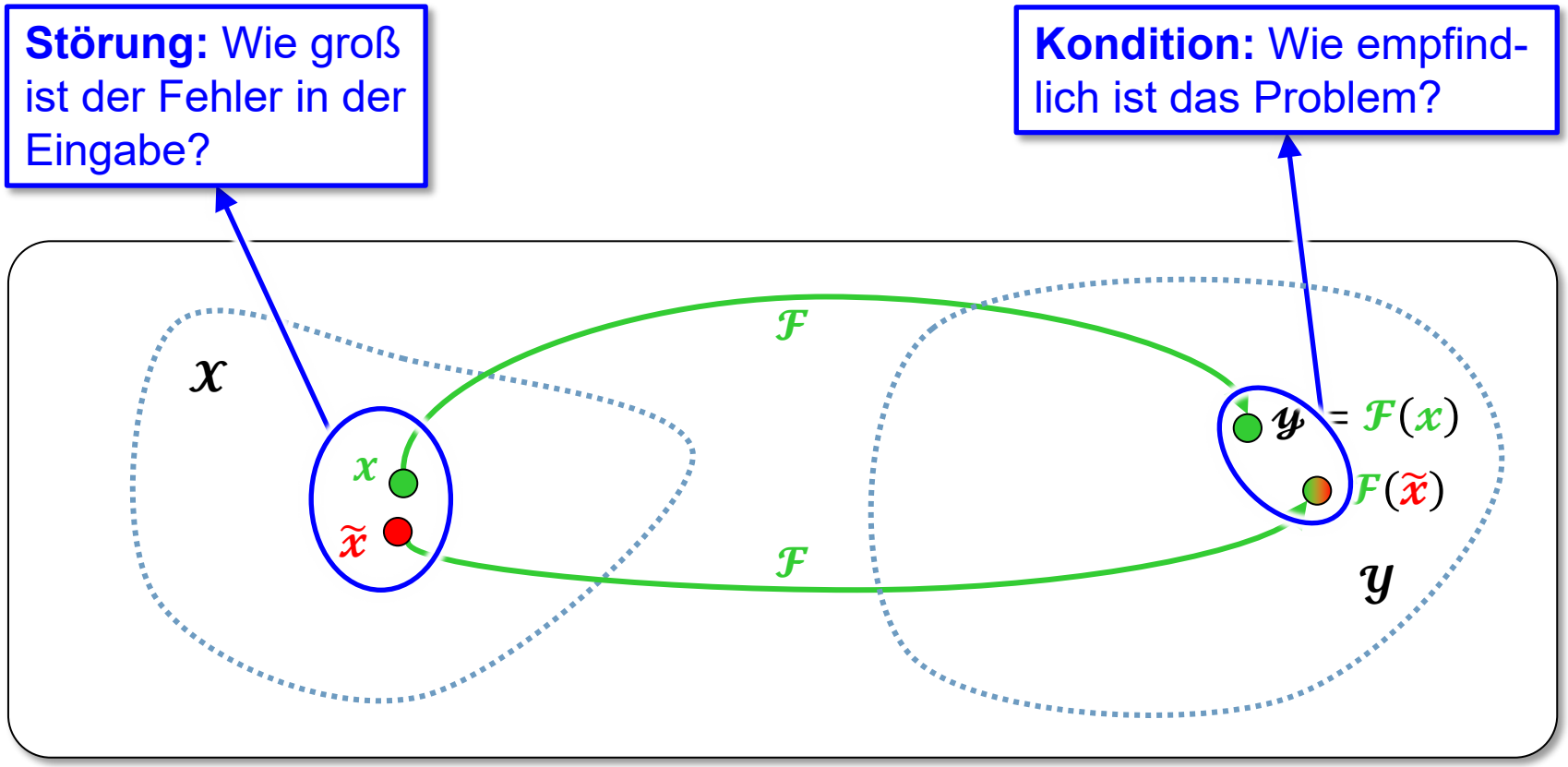
- **ZIEL:** Zur Berechnung von $\mathcal{F}$ verwende
  - ein **numerisches Verfahren/Algorithmus** $\tilde{\mathcal{F}}$
  - basierend auf einer **gestörten Eingabe** $\tilde{x}$.

Anstatt $\mathcal{F}(x)$ wird $\tilde{\mathcal{F}}(\tilde{x})$ berechnet!

- Zum besseren Verständnis was bei der Berechnung von $\tilde{\mathcal{F}}(\tilde{x})$ passiert, untersuche zusätzlich auch $\mathcal{F}(\tilde{x})$ und $\tilde{\mathcal{F}}(x)$.

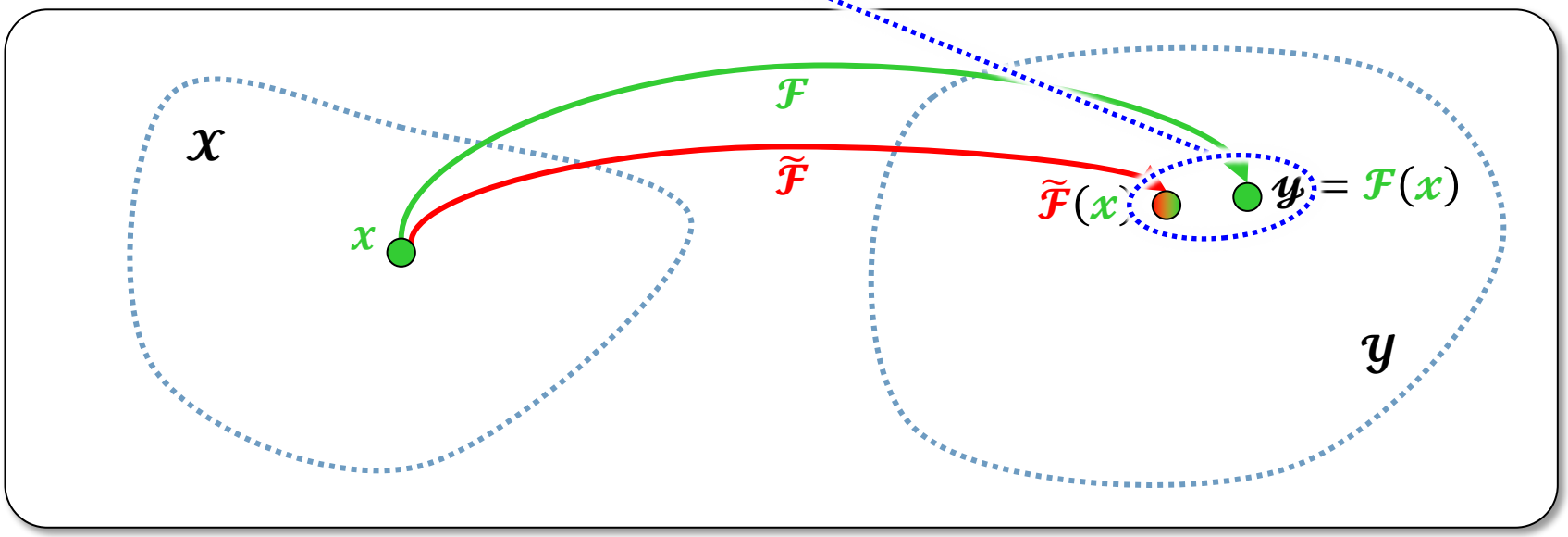


Formalisierung der Forderungen



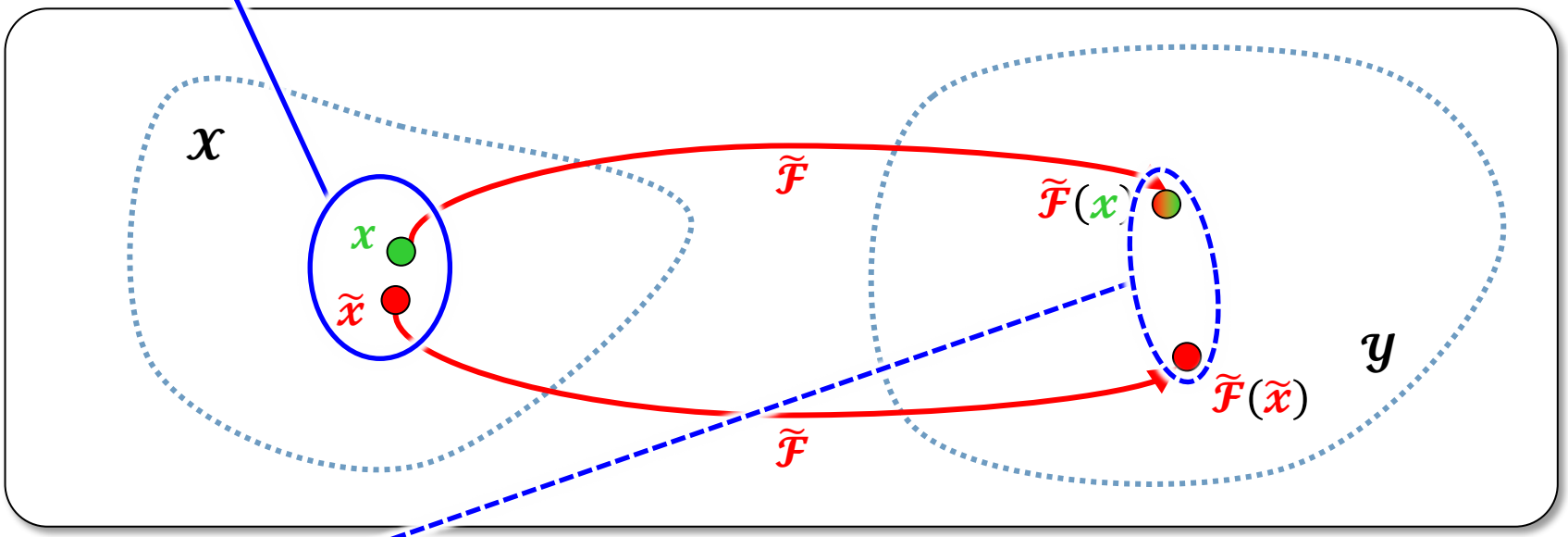
Formalisierung der Forderungen

Konsistenz: Wie gut passt das numerische Verfahren zum Problem?



Formalisierung der Forderungen

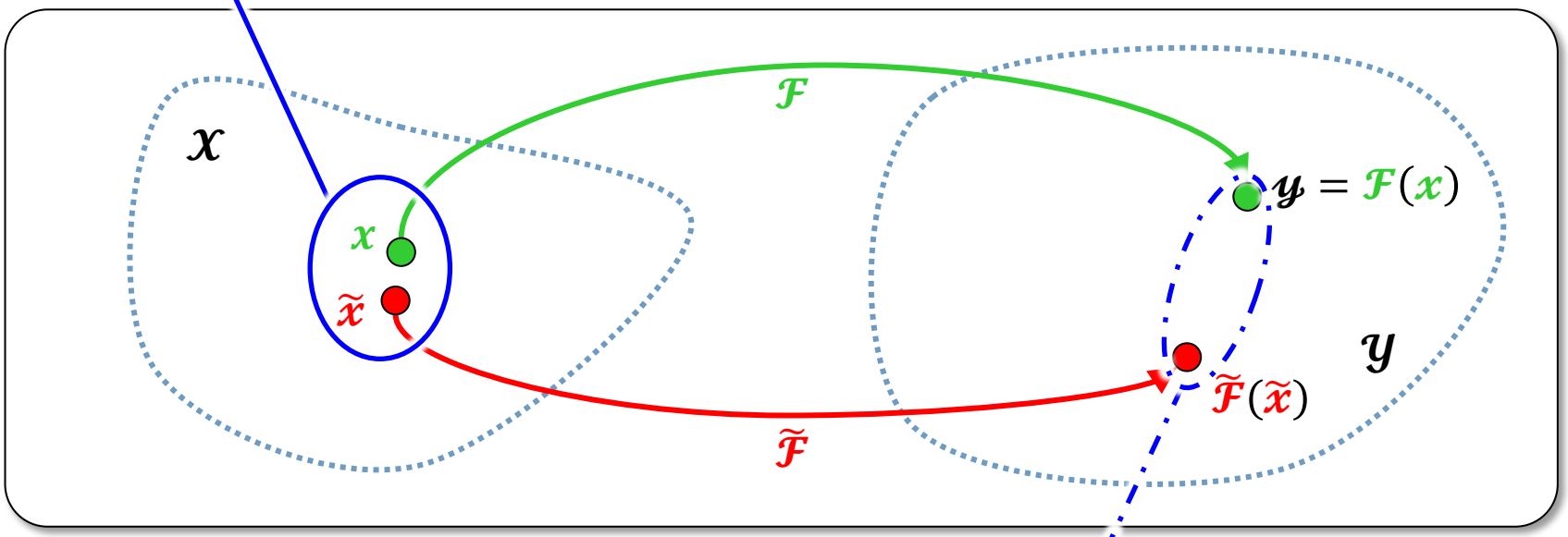
Störung: Wie groß ist der Fehler in der Eingabe?



Stabilität: Wie robust ist das numerische Verfahren?

Formalisierung der Forderungen

Störung: Wie groß ist der Fehler in der Eingabe?



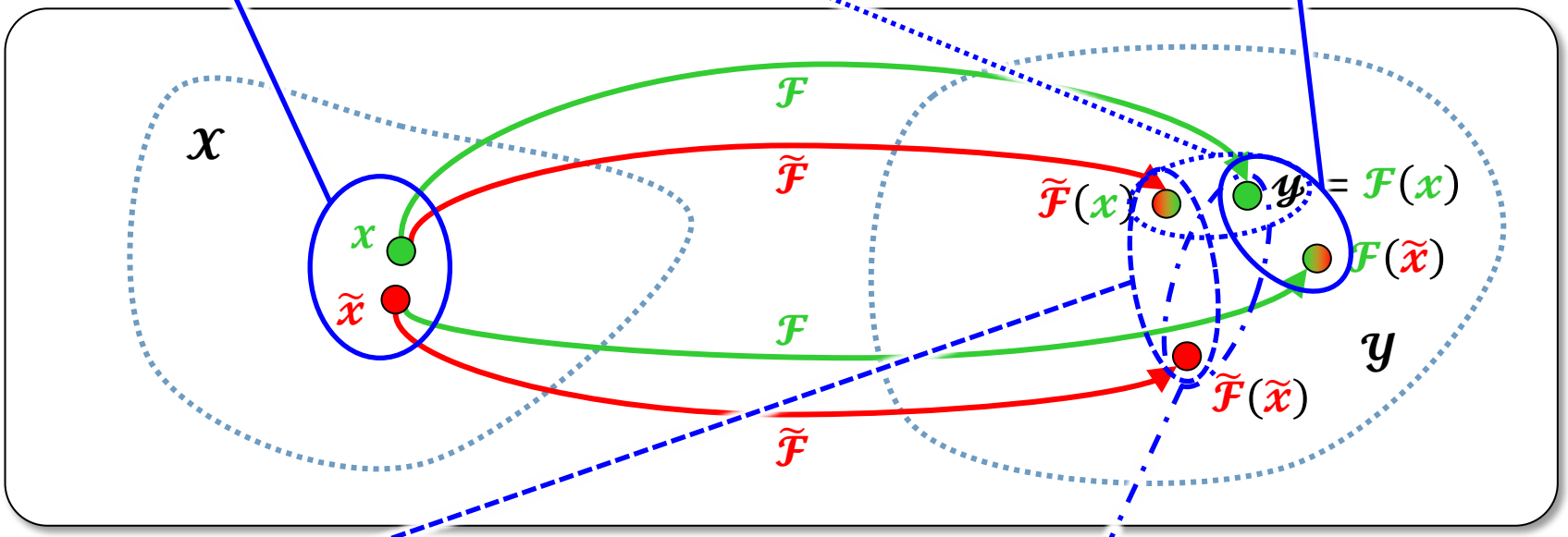
Konvergenz: Wie robust ist das numerische Verfahren für das gegebene Problem?

Formalisierung der Forderungen

Störung: Wie groß ist der Fehler in der Eingabe?

Konsistenz: Wie gut passt das numerische Verfahren zum Problem?

Kondition: Wie empfindlich ist das Problem?

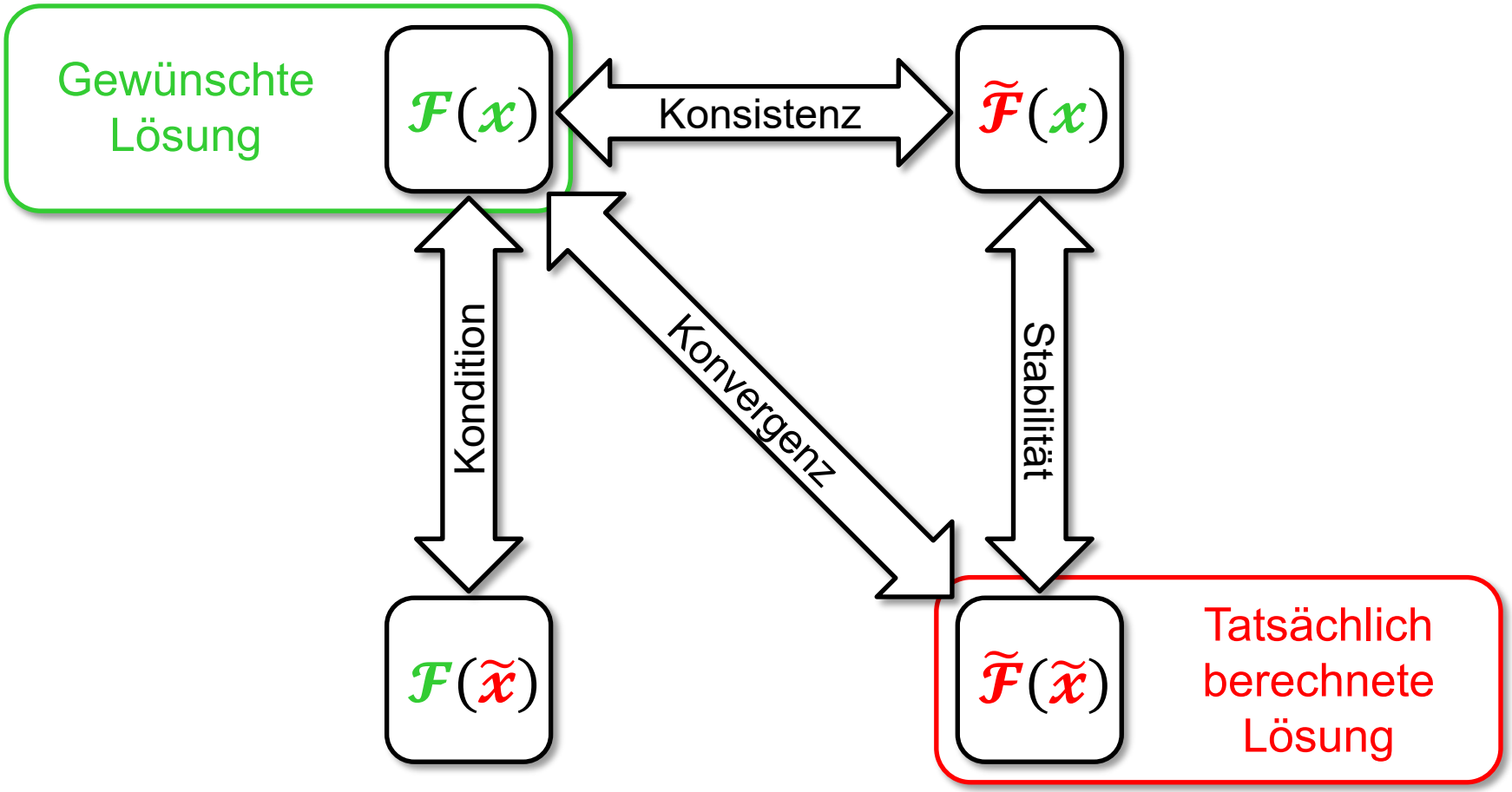


Stabilität: Wie robust ist das numerische Verfahren?

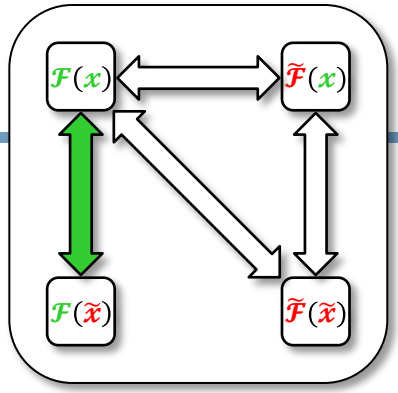
Konvergenz: Wie robust ist das numerische Verfahren für das gegebene Problem?

Formalisierung der Forderungen

- Was sind die Zusammenhänge zwischen $\mathcal{F}$, $\tilde{\mathcal{F}}$, x und $\tilde{x}$?



Formalisierung der Forderungen

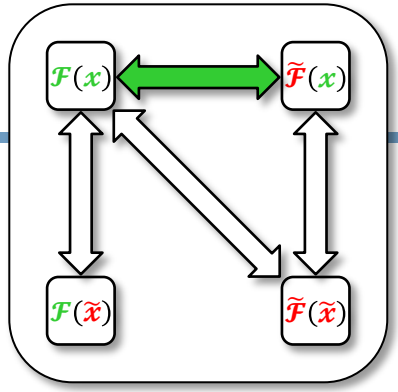


1. **Kondition:** Wie robust ist das mathematische Modell $\mathcal{F}$ bzgl. Fehlern in der Eingabe?
   - Formalisiert: $\|\mathcal{F}(\tilde{x}) - \mathcal{F}(x)\|_y$
   - **Relative Kondition κ** (aka *Konditionszahl*): Kleinste Konstante $\kappa \geq 0$ so, dass

$$\frac{\|\mathcal{F}(x) - \mathcal{F}(\tilde{x})\|_y}{\|\mathcal{F}(x)\|_y} \leq \kappa \cdot \frac{\|x - \tilde{x}\|_x}{\|x\|_x}.$$

   - Gut konditioniert: $\kappa \leq 1$, aka *Fehlerdämpfung*
   - Schlecht konditioniert: $\kappa \gg 1$, aka *Fehlerverstärkung*
   - Eigenschaft des Problems: unvermeidbarer Problemfehler
   - Verbesserung nur durch Änderung des Modells möglich!

Formalisierung der Forderungen



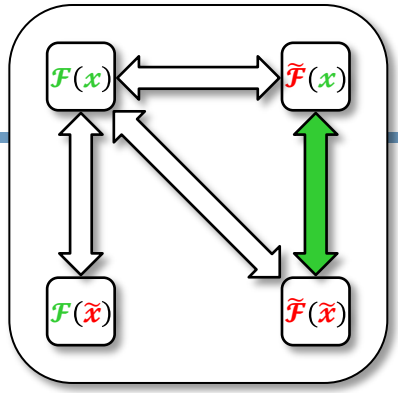
2. **Konsistenz:** Wie gut löst das numerische Verfahren $\tilde{\mathcal{F}}$ das mathematische Modell $\mathcal{F}$?

- Formalisiert: $\left\|\tilde{\mathcal{F}}(x) - \mathcal{F}(x)\right\|_y$
- Effektivität des numerischen Verfahrens $\tilde{\mathcal{F}}$.
- **Konsistenzordnung p** $\in \mathbb{N}$ für $\mathcal{F} \in C^0$:

  $\left\|\tilde{\mathcal{F}}(x_h) - \mathcal{F}(x_h)\right\|_y \leq O(h^p)$ für $h = \|x - x_h\|_x \to 0$.

- Eigenschaft des Zusammenspiels von mathematischem Modell und numerischem Verfahren.
  - Ist das Verfahren geeignet das Problem zu lösen?
- Verbesserung durch Auswahl des Verfahrens möglich.

Formalisierung der Forderungen

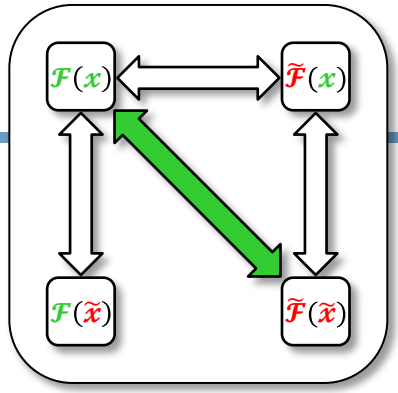


3. **Stabilität:** Wie robust ist das numerische Verfahren $\tilde{\mathcal{F}}$ bzgl. Fehlern in der Eingabe?
   - Formalisiert: $\left\|\tilde{\mathcal{F}}(x) - \tilde{\mathcal{F}}(\tilde{x})\right\|_y$
   - Rundungsfehler in den Rechenschritten.
   - Das numerische Verfahren $\tilde{\mathcal{F}}$ heißt **stabil** (aka *numerisch stabil*, *Rundungsfehler-stabil*), wenn es ein $\sigma \in \mathbb{R}$ gibt mit

$$\left\|\tilde{\mathcal{F}}(x) - \tilde{\mathcal{F}}(\tilde{x})\right\|_y \leq \kappa \cdot \sigma \cdot \|x - \tilde{x}\|_X.$$

     - Bei stabilen Verfahren übersteigt der akkumulierte Rundungsfehler den Fehler durch Kondition des Problems nicht wesentlich.
   - Eigenschaft des numerischen Verfahrens/Algorithmus.
   - Verbesserung durch Optimierung des Algorithmus.

Formalisierung der Forderungen



4. **Konvergenz:** Wie gut löst das numerische Verfahren $\tilde{\mathcal{F}}$ bei gestörter Eingabe $\tilde{x}$ das mathematische Modell $\mathcal{F}$?
   - Formalisiert: $\left\|\tilde{\mathcal{F}}(\tilde{x}) - \mathcal{F}(x)\right\|_y$
   - Konvergenz folgt i.d.R. aus Stabilität und Konsistenz

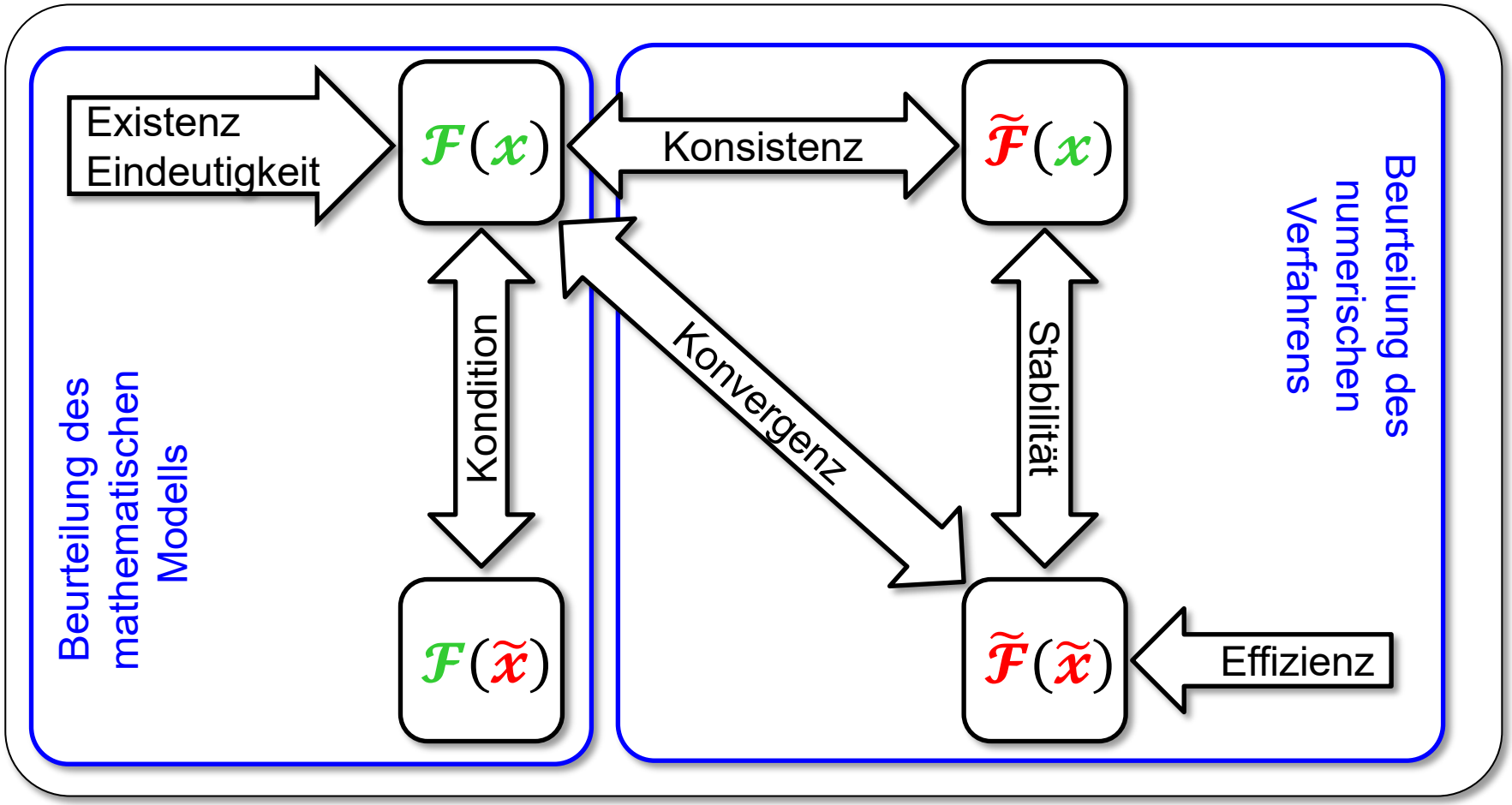
$$\left\|\tilde{\mathcal{F}}(\tilde{x}) - \mathcal{F}(x)\right\|_y \leq \underbrace{\left\|\tilde{\mathcal{F}}(\tilde{x}) - \tilde{\mathcal{F}}(x)\right\|_y}_{\text{Stabilität}} + \underbrace{\left\|\tilde{\mathcal{F}}(x) - \mathcal{F}(x)\right\|_y}_{\text{Konsistenz}}$$

$$\leq \overbrace{\kappa \cdot \sigma \cdot \left\|\tilde{x} - x\right\|_X}^{\text{Stabilität}} + \overbrace{c \cdot \left\|\tilde{x} - x\right\|_X^p}^{\text{Konsistenz}}.$$

   - Wenn gut konditioniert und $\left\|x - \tilde{x}\right\|_X \to 0$, dann auch $\left\|\tilde{\mathcal{F}}(\tilde{x}) - \mathcal{F}(x)\right\|_y \to 0$, aka „*reality check*“

Formalisierung der Forderungen

Überblick



Formalisierung der Forderungen

5. **Effizienz:** Wie schnell kann das numerische Verfahren $\widetilde{\mathcal{F}}$ berechnet werden?
   - *Direkte Verfahren:*
     - Algorithmische Laufzeit: Anzahl relevanter Rechenoperationen gemessen im O-Kalkül.
   - *Iterative Verfahren:*
     - Konvergenzgeschwindigkeit, oft gemessen im O-Kalkül.
     - Algorithmische Laufzeit der einzelnen Iterationsschritte.