INPUT: $A \in \mathbb{R}^{n \times n}$ (invertierbar)
$b \in \mathbb{R}^n$

OUTPUT: $x \in \mathbb{R}^n$ mit $Ax = b$

ALGORITHMUS: rel. Richardson:

$$x^{k+1} = x^k - \Theta(Ax^k - b)$$

↳ **PARAMETER:** conf.py
- A: Matrix
- b: Vektor
- x^0: Startvektor
- Θ: Schrittweite
- tol: Fehlertoleranz
- maxiter: max. Iterationszahl
- (*kwargs: für callback?)

↳ WAS MUSS IMPLEMENTIERT WERDEN?

Wir gehen lediglich von Basisdatentypen aus.

- FORMAT FÜR MATRIZEN & VEKTOREN
  - Vektoren: List()
  - Matrix: CSR-Klasse
- LA-ROUTINEN
  - Skalarmultiplikation: float × List
  - Addition: list + list
  - Euklidische Norm: $(\text{float}^2 + \dots + \text{float}^2)^{1/2}$
  - Matrix-Vektor Produkt: CSR × list

→ lalib.py

- rel. Richardson

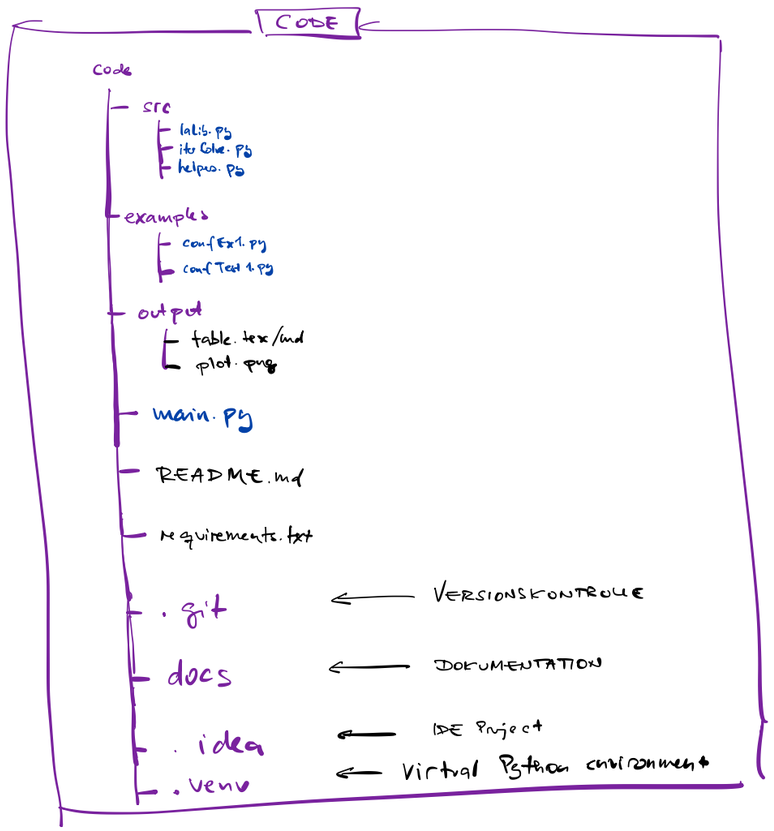
→ iterSolver.py

- ERGEBNISSE AUSLESEN
  - Tabellen speichern
  - Grafiken erstellen

→ helpers.py

SINNVOLLE MODULARISIERUNG: [siehe Kursnotizen]

- Metadaten
- Inhaltliche Bausteine
- Medien
- Literatur
- LaTeX Konfiguration
- ...



- Grafiken
- Tabellen

↑

[kontinuierliche Integration von Code & Text]