Numerik Cheat Sheeto

1 Basics

1.1 Sortieren

1.2 FFT

2 Lineare Gleichungssysteme

2.1 Allgemeine Aufgabenstellung

Geg.: $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{b} \in \mathbb{R}^n$
Ges.: $\mathbf{x} \in \mathbb{R}^n$

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

2.2 Dreiecksmatrizen

Untere Dreiecksmatrix $\mathbf{L} \in \mathbb{R}^{n\times n}$ und obere Dreiecksmatrix $\mathbf{R} \in \mathbb{R}^{n\times n}$.

Reguläre/invertierbare/nicht-singuläre Matrix
Matrix $\mathbf{A}$ ist regulär, wenn $\det \mathbf{A} \neq 0$. Determinante einer ΔMatrix ist das Produkt ihrer Diagonalelemente. $\mathbf{L}$ und $\mathbf{R}$ sind regulär, wenn alle Diagonalelemente $\neq 0$.

Vorwärtseinsetzen

$$\mathbf{L}\mathbf{y} = \mathbf{b}$$

Rechenaufwand: n^2 AO
Um Speicher zu sparen $b_i \leftarrow y_i$.

`forward_subst`

---

for $j = 1 : n$
  $x_j \leftarrow b_j / l_{jj}$
  for $i = j + 1 : n$
    $b_i \leftarrow b_i - l_{ij} x_j$

Rückwärtseinsetzen

$$\mathbf{R}\mathbf{x} = \mathbf{y}$$

Rechenaufwand: n^2 AO
Um Speicher zu sparen $b_i \leftarrow x_i$.

`backward_subst`

---

for $j = n : 1$
  $x_j \leftarrow b_j / r_{jj}$
  for $i = 1 : j - 1$
    $b_i \leftarrow b_i - r_{ij} x_j$

2.3 LR-Zerlegung

Sei $\mathbf{A} \in \mathbb{R}^{n\times n}$ und $\mathbf{L}, \mathbf{R} \in \mathbb{R}^{n\times n}$

$$\mathbf{A} = \mathbf{L}\mathbf{R}$$

Ansatz:
Matrizen $\mathbf{A}$, $\mathbf{L}$, $\mathbf{R}$ in Teilmatrizen $\mathbf{A}_{**}$, $\mathbf{A}_{*1}$, $\mathbf{A}_{1*}$, $\mathbf{L}_{**}$, $\mathbf{L}_{*1}$, $\mathbf{R}_{**}$, $\mathbf{R}_{1*}$ zerlegen.

Es folgen 4 Gleichungen aus $\mathbf{A} = \mathbf{L}\mathbf{R}$:

$$\begin{aligned}
a_{11} &= l_{11} r_{11} && \\
\mathbf{A}_{*1} &= \mathbf{L}_{*1} r_{11} && \Leftrightarrow \mathbf{L}_{*1} = \mathbf{A}_{*1} / r_{11} \\
\mathbf{A}_{1*} &= l_{11} \mathbf{R}_{1*} && \Leftrightarrow \mathbf{R}_{1*} = \mathbf{A}_{1*} \\
\mathbf{A}_{**} &= \mathbf{L}_{*1}\mathbf{R}_{1*} + \mathbf{L}_{**}\mathbf{R}_{**} &&
\end{aligned}$$

Per Def. $l_{11} = 1$ und damit $r_{11} = a_{11}$, sodass

$$\mathbf{A}_{**} - \mathbf{L}_{*1}\mathbf{R}_{1*} = \mathbf{L}_{**}\mathbf{R}_{**} \tag{1}$$

Praktische Umsetzung
Elemente von $\mathbf{A}$ überschreiben, sodass:

$$\begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ l_{21} & r_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & r_{n-1,n} \\ l_{n1} & \cdots & l_{n,n-1} & r_{nn} \end{pmatrix} \tag{2}$$

Kriterium
Sei $\mathbf{A}$ regulär, $\mathbf{A}$ besitzt eine LR-Zerlegung $\Leftrightarrow$ Alle Hauptuntermatrizen regulär.

Modellproblem: Bandmatrix
Irgendwas bzgl Effizienz.

LR-Decomp
Aufwand: *kubisch*
$\downarrow$ Aufwand: Nur 1x für jede Matrix betreiben. Sobald LR-Decomp vorliegt nur noch *quadratischer* Aufwand
$\Downarrow$ Aufwand: Tridiagonalmatrix. Erster Schritt mit 3 AOPs. Restmatrix bleibt tridiagonal. Aufwand $3n + 6n$ für R-und F-Einsetzen.

`lr_decomp`

---

for $k = 1 : n$
  for $i = k + 1 : n$
    $a_{ik} \leftarrow a_{ik} / a_{kk}$
    for $j = k + 1 : n$
      $a_{ij} \leftarrow a_{ij} - a_{ik} a_{kj}$

Problem der Existenz einer LR-Z
Falls $\mathbf{A}$ oder $\mathbf{A}_{**}$ eine 0 auf der Diagonalen hat, existiert keine LR-Zerlegung. Lösung: Permutiere die Zeilen von $\mathbf{A}$ so, dass das Ergebnis eine LR-Z besitzt.

Permutationsmatrix
Sei $\mathbf{P} \in \mathbb{R}^{n\times n}$. Falls in jeder Zeile und Spalte von $\mathbf{P}$ genau ein Eintrag 1 und alle anderen 0, dann ist $\mathbf{P}$ eine Permutationsmatrix. $\mathbf{P}$ ist orthogonal. Ein Produkt zweier Permutationsmatrizen $\mathbf{PQ}$ ist auch eine Permutationsmatrix.

Permutation
Bijektive Abbildung $\pi : \{1, \cdots, n\} \to \{1, \cdots, n\}$.

LR-Z mit Pivotsuche
$\mathbf{A}$ regulär. Es existiert $\mathbf{P} \in \mathbb{R}^{n\times n}$ sodass $\mathbf{PA} = \mathbf{LR}$ gilt.

Pivotisierung: Finde betragsmaximalstes Element in der aktuellen Spalte, welches unter dem aktuellen Diagonalelement von $\mathbf{A}$ liegt und tausche die aktuelle Zeile mit der Zeile in der das betragsmaximalste Element ist, mit Hilfe von $\mathbf{P}$. $a_{11} \neq 0$, da betragsgrößtes Element.

Lösen eines Gleichungssystems mit Pivotsuche
$\mathbf{Ax} = \mathbf{b} \Leftrightarrow \mathbf{PAx} = \mathbf{Pb} \Leftrightarrow \mathbf{LRx} = \mathbf{Pb}$.
1.) $\mathbf{Ly} = \tilde{\mathbf{b}}$ 2.) $\mathbf{Rx} = \mathbf{y}$

`lr_pivot`

---

for $k = 1 : n$
  $i_* \leftarrow k$ // Finde max. Element
  for $i = k + 1 : n$
    if $|a_{ik}| > |a_{i_* k}|$: $i_* \leftarrow i$
  $p_k \leftarrow i_*$
  for $j = 1 : n$ // Tausche Zeilen
    $\gamma \leftarrow a_{kj}$, $a_{kj} \leftarrow a_{i_* j}$, $a_{i_* j} \leftarrow \gamma$
  for $i = k + 1 : n$
    $a_{ik} \leftarrow a_{ik} / a_{kk}$
    for $j = k + 1 : n$
      $a_{ij} \leftarrow a_{ij} - a_{ik} a_{kj}$

$\mathbf{p}$ protokolliert, welche Vertauschungen durchgeführt wurden, um sie später auf $\mathbf{b}$ anwenden zu können.
Aufwand: $\frac{2}{3} n^3$.

Sonderfall: $\mathbf{A}$ positiv definit
TODO.

2.4 Fehlerverstärkung

Norm des Matrix-Vektor-Produkts
Wie stark ändert sich die Länge eines Vektors wenn er mit $\mathbf{A}$ multipliziert wird. Mapping von Einheitskreis auf Ellipse..
Für $\mathbf{A} \in \mathbb{R}^{n\times n}$ gilt:

$$\begin{aligned}
\alpha_2(\mathbf{A}) &= \min\{\|\mathbf{Ay}\|_2 : \mathbf{y} \in \mathbb{R}^n, \|\mathbf{y}\|_2 = 1\} \\
\beta_2(\mathbf{A}) &= \max\{\|\mathbf{Ay}\|_2 : \mathbf{y} \in \mathbb{R}^n, \|\mathbf{y}\|_2 = 1\}
\end{aligned}$$

und

$$\alpha_2(\mathbf{A})\|\mathbf{z}\|_2 \leq \|\mathbf{Az}\|_2 \leq \beta_2(\mathbf{A})\|\mathbf{z}\|_2$$

Eigenschaften der Norm:

$$\begin{aligned}
\|\mathbf{x}\| = 0 &\Leftrightarrow \mathbf{x} = 0 \\
\|\lambda \mathbf{x}\| &= |\lambda| \|\mathbf{x}\| \\
\|\mathbf{x} + \mathbf{y}\| &\leq \|\mathbf{x}\| + \|\mathbf{y}\|
\end{aligned}$$

TODO

2.5 QR-Zerlegung

Für jede Matrix gibt es eine QR-Z.

LR-Z: Schlecht konditioniertes Problem

$\kappa_2(\mathbf{A}) \gg 1$, $\kappa_2(\mathbf{A}) \leq \kappa_2(\mathbf{L})\kappa_2(\mathbf{R})$

Kritisch falls $\kappa_2(\mathbf{L})\kappa_2(\mathbf{R}) \gg \kappa_2(\mathbf{A})$.

Ziel: Suche Transformationen $\mathbf{Q} \in \mathbb{R}^{n\times n}$, die die Norm unverändert lassen:

$$\|\mathbf{Qy}\|_2 = \|y\|_2$$

Mit Hinzunahme des Skalarprodukts:

$$\langle \mathbf{y},\mathbf{y}\rangle_2 = \|\mathbf{y}\|_2^2 = \|\mathbf{Qy}\|_2^2 = \langle \mathbf{Qy},\mathbf{Qy}\rangle_2 = \langle \mathbf{y},\mathbf{Q}^*\mathbf{Qy}\rangle_2$$

muss $\mathbf{Q}^*\mathbf{Q} = \mathbf{I}$ gelten.

Gesucht: $\mathbf{A} = \mathbf{QR}$.

Konditionszahl bzw. Fehlerverstärkung wird nicht verschlechtert: $\alpha_2(\mathbf{A}) = \alpha_2(\mathbf{R})$, $\beta_2(\mathbf{A}) = \beta_2(\mathbf{R})$
$\kappa_2(\mathbf{A}) = \kappa_2(\mathbf{R})$.

Givens-Rotation

Mit Hilfe von Givens-Rotationen können wir beliebige $\mathbf{A} \in \mathbb{R}^{m\times n}$ auf obere Δgestalt bringen.

$$\mathbf{Q} = \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \text{ und } \mathbf{Qy} = \begin{pmatrix} cy_1 + sy_2 \\ 0 \end{pmatrix}$$

Konsekutiv Givens-Rotationen $\mathbf{Q}_{ij}$ i-te und j-te Zeile anwenden um Eintrag a_{ij} zu beseitigen. Bsp. $\mathbf{A} \in \mathbb{R}^{4\times 3}$:

$$\mathbf{R} = \mathbf{Q}_{43}\mathbf{Q}_{32}\mathbf{Q}_{42}\mathbf{Q}_{21}\mathbf{Q}_{31}\mathbf{Q}_{41}\mathbf{A}$$
$$\underbrace{\mathbf{Q}_{41}^*\mathbf{Q}_{31}^*\mathbf{Q}_{21}^*\mathbf{Q}_{42}^*\mathbf{Q}_{32}^*\mathbf{Q}_{43}^*}_{\mathbf{Q}}\mathbf{R} = \mathbf{A}$$

Kompakte Darstellung

Verwende Nulleinträge von $\mathbf{A}$ bzw. $\mathbf{R}$ um $\mathbf{Q}_{ij}$ zu beschreiben.

Finde Givens-Rotation:

$$\rho = \begin{cases} s = \rho, c = \sqrt{1-s^2} & \text{falls } |\rho| < 1 \\ c = 1/\rho, s = \sqrt{1-c^2} & \text{falls } |\rho| > 1 \\ c = 1, s = 0 & \text{falls } \rho = 1 \end{cases}$$

Speichern der QR-Z in $\mathbf{A}$:

$$\begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ \rho_{21} & r_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & r_{n-1,n} \\ \rho_{n1} & \cdots & \rho_{n,n-1} & r_{nn} \end{pmatrix} \tag{3}$$

`Qr Decomp` von $\mathbf{A} \in \mathbb{R}^{m\times n}$

---

for $k = 1 : \min(m,n)$ // Loop über Diagonale
  for $i = k+1 : m$ // Loop über Elemente unter Diagonalen
    if $a_{ik} = 0$
      $\rho \leftarrow 1$, $c \leftarrow 1$, $s \leftarrow 0$
    else if $|a_{kk}| \geq |a_{ik}|$ // Vgl. mit Diag.element
      $\tau \leftarrow a_{ik}/a_{kk}$, $\rho \leftarrow \tau/\sqrt{\tau^2+1}$, $s \leftarrow \rho$, $c \leftarrow \sqrt{1-s^2}$
    else // Vgl. mit Diag.element
      $\tau \leftarrow a_{kk}/a_{ik}$, $\rho \leftarrow \sqrt{\tau^2+1}/\tau$, $c \leftarrow 1/\rho$, $s \leftarrow \sqrt{1-c^2}$
    // Diag.element aktual., Giv.-Rot. in aktueller It. speichern
    $a_{kk} \leftarrow ca_{kk} + sa_{ik}$, $a_{ik} \leftarrow \rho$
    for $j = k+1 : n$ // Loop über Elemente in der k-ten Zeile
      // Giv-Rot auf Zeile anwenden
      $\alpha \leftarrow a_{kj}$, $a_{kj} \leftarrow c\alpha + sa_{ij}$, $a_{ij} \leftarrow -s\alpha + ca_{ij}$

Aufwand: $6n^2 + 2n^3$ (quadratische Matrix) 3x mehr als LR-Z.

Lösen Gleichungssystem $\mathbf{Ax} = \mathbf{b}$

$\mathbf{b} = \mathbf{Ax} = \mathbf{QRx} = \mathbf{Qy}$ $\Leftrightarrow$ 1.) $\mathbf{y} = \mathbf{Q}^*\mathbf{b}$ 2.) $\mathbf{y} = \mathbf{Rx}$.

1.) Qr_transform: Über einzelne G_{ij} (oben links angefangen) iterieren und auf b multiplizieren.

2.) Rückwärtseinsetzen.

Effizientere QR-Z

Householder-Spiegelungen: Aufwand 2x mehr als LR-Z.

Mit Optimierungen bei Speicherzugriffen bei QR-Z ähnlich schnell wie LR-Z.

2.6 Ausgleichsprobleme

Wir suchen $\mathbf{x}$ so, dass alle Gleichungen möglichst gleich gut erfüllt werden.

3 Nichtlineare Gleichungssysteme

Wir untersuchen nichtlineare Gleichungssysteme der Form

Gegeben eine stetige Funktion $f : \mathbb{R}^n \to \mathbb{R}^n$, finde $\mathbf{x}^* \in \mathbb{R}^n$ mit $f(\mathbf{x}^*) = \mathbf{0}$

Transformieren in ein Nullstellenproblem.

3.1 Bisektionsverfahren

Einfache Technik, das in jedem Schritt den Fehler mindestens halbiert. Basierend auf dem Zwischenwertsatz für stetige Funktionen.

Zwischenwertsatz

Eine reele Funktion f, die in $[a,b]$ stetig ist, nimmt jeden Wert zwischen $f(a)$ und $f(b)$ an. Haben $f(a)$ und $f(b)$ verschiedene Vorzeichen, so ist eine Existenz mindestens einer Nullstelle in $[a,b]$ garantiert.

Verfahren in mathematischer Notation

$$(a^{(0)}, b^{(0)}) = (a,b)$$
$$x^{(m)} = \frac{a^{(m)} + b^{(m)}}{2}$$
$$(a^{(m+1)}, b^{(m+1)}) = \begin{cases} (a^{(m)}, x^{(m)}) & \text{if } f(a^{(m)})f(x^{(m)}) < 0 \\ (x^{(m)}, b^{(m)}) & \text{sonst.} \end{cases}$$

`Bisection`

$f_a \leftarrow f(a)$, $f_b \leftarrow f(b)$
`while` $b - a > \epsilon$
  $x \leftarrow (a+b)/2$
  $f_x \leftarrow f(x)$
  `if` $f_a f_x < 0$
    $b \leftarrow x$, $f_b \leftarrow f_x$
  `else`
    $a \leftarrow x$, $f_a \leftarrow f_x$

3.2 Allgemeine Fixpunktiterationen

3.3 Newton-Verfahren

4 Eigenwertprobleme

4.1 Vektoriteration

4.2 Inverse Iteration

4.3 Orthogonale Iteration

4.4 QR-Iteration

4.5 Praktische QR-Iteration

5 Approximation von Funktionen

5.1 Polynominterpolation

5.2 Neville-Aitken-Verfahren

5.3 Newtons dividierte Differenzen

5.4 Approximation von Funktionen

6 Numerische Integration

6.1 Quadraturformeln

6.2 Fehleranalyse