

Iterative Löser für lineare Gleichungssysteme

Gregor Milicic
gregor.milicic@sbg.ac.at
Universität Salzburg

Abstrakt

Iterative Verfahren zum Lösen des Gleichungssystems $Ax = b$ mit der Matrix $A \in \mathbb{R}^{n \times n}$, der rechten Seite $b \in \mathbb{R}^n$ und der Lösung $x \in \mathbb{R}^n$ bieten sich im überfachlichen Bereich als Thema zwischen der Mathematik und der Informatik geradezu an. Nach einer kurzen theoretischen Vorstellung und Konvergenzbetrachtung der *Jacobi*-, *Gauß-Seidel*- und *SOR-Verfahren* erfolgt die Beschreibung eines Unterrichtskonzepts. Die Ausführungen sind dabei als Ergänzungen zu den online verfügbaren Foliensätzen und Handouts anzusehen. Eine didaktische Einbettung gefolgt mit weiteren, möglichen Vertiefungsthemen rundet die Arbeit ab.

1 Einleitung

Geringe mathematische Änderungen im Algorithmus bei iterativen Verfahren für lineare Gleichungssysteme können die Konvergenzeigenschaft erheblich beeinflussen und zu verbesserten Konvergenzeigenschaften führen. Zwingend zum Verstehen und Durchdringen der Verfahren nach einer theoretischen Betrachtung ist dann die Modellbildung im Sinne des Algorithmischen Denkens. Erst durch die konkrete Umsetzung am Computer werden scheinbar vernachlässigbare Ausnahmefälle relevant und kleinere Änderungen im Algorithmus nachvollziehbar. Das Thema der iterativen Verfahren für lineare Gleichungssysteme bietet daher für beide Fächer, der Mathematik und der Informatik, genügend Möglichkeiten um bekannte Sachverhalte in einem praxisnahen Kontext anzuwenden. Je nach Neigung der Lernenden und zeitlichem Rahmen sind zudem vertiefende Betrachtungen ohne Weiteres möglich. Im Folgenden wird dargestellt, wie die iterativen Jacobi-, Gauß-Seidel- und SOR-Verfahren im Mathematikunterricht eingeführt werden können. Einer kurzen theoretischen Betrachtung folgen konkrete Hinweise und Empfehlungen zur Erarbeitung im Unterricht. Diese Unterrichtssequenz wurde an der HTL in Braunau¹ im Rahmen des *EMMA-Experimentieren mit mathematischen Algorithmen*-Projekts² in ähnlicher Form wie im Abschnitt *Vorstellung der Verfahren im Unterricht* gehalten und,

¹<http://www.htl-braunau.at/htl-live.html>

²<http://www.uni-salzburg.at/emma>

einem iterativen Prozess folgend, auf Basis der dort gemachten Erfahrungen verändert. Der Aufbau kann natürlich angepasst und verändert werden und soll als Empfehlung verstanden werden. An die zugrunde liegenden didaktischen Prinzipien folgen weitere Vertiefungsmöglichkeiten und Ausblicke im Themenbereich Iterative Löser für lineare Gleichungssysteme. Die Ausführungen können als ergänzende und erklärende, sowie mit didaktischen Anmerkungen versehene Notizen zu den online verfügbaren Folien³ angesehen werden. Das in Java verfasste Programm wird auf Anfrage verschickt.

1.1 Benötigtes Vorwissen

Die Lernenden sollten mit Gleichungssystemen von 3 und mehr Variablen sowie in Grundzügen auch mit Matrizen vertraut sein. Das Wissen um die äquivalente Darstellung des Gleichungssystems in Gleichungsform und in Matrix-Vektor-Schreibweise ist dabei ausreichend:

$$\left. \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{array} \right\} \iff Ax = b. \quad (1)$$

Auf (theoretisch mögliche) Ausführungen anhand der Matrizen, eine Einführung der Iterationsmatrix o.ä. wurde im Hinblick auf die bereits bestehende fachliche Tiefe verzichtet. Des Weiteren sollten die Lernenden auch mit dem Summenzeichen \sum vertraut sein. Kenntnisse über iterative Verfahren, wie z.B. das Newton-Verfahren, sind natürlich hilfreich, werden aber nicht vorausgesetzt. Bezüglich der Informatik sind elementare Grundkenntnisse in der Programmierung einer beliebigen Programmiersprache ebenso wie die sichere Umsetzung von Pseudocode in eine Programmiersprache notwendig.

2 Theoretische Beschreibung der Verfahren

Warum sollten nun iterative Verfahren betrachtet werden? Kann das Gleichungssystem nicht auch exakt und direkt gelöst werden? Natürlich ist dies möglich, in der praktischen Anwendung werden jedoch zumeist iterative Verfahren verwendet. Dazu folgend ein Auszug aus einem Lehrbuch der Numerik für lineare Gleichungssysteme [4]:

Die [...] direkten Verfahren stellen bei einer kleinen Anzahl von Unbekannten oftmals eine effiziente Vorgehensweise dar. Praxisrelevante Problemstellungen [...] führen jedoch häufig auf große schwachbesetzte Gleichungssysteme. Die Speicherung derartiger Gleichungssysteme wird gewöhnlich erst durch die Vernachlässigung der Nullelemente der Matrix, die teilweise über 99% der Matrixkoeffizienten darstellen, ermöglicht. Betrachten wir die erläuterte

³<http://www.uni-salzburg.at/index.php?id=67971>

Diskretisierung der Konvektions-Diffusions-Gleichung mit $N = 500$, so benötigen wir bei einem vorausgesetzten Speicherplatzbedarf von 8 Byte für jede reelle Zahl etwa 10 Megabyte zur Speicherung der nichtverschwindenden Matrixkoeffizienten. Dagegen würde das Abspeichern der gesamten Matrix 500 Gigabyte beanspruchen.

Direkte Verfahren können in der Regel die besondere Gestalt der Gleichungssysteme nicht ausnutzen, wodurch vollbesetzte Zwischenmatrizen generiert werden, die einerseits den verfügbaren Speicherplatz überschreiten und andererseits zu unakzeptablen Rechenzeiten führen [...]. Desweiteren entstehen solche Gleichungssysteme zumeist durch eine Diskretisierung der zugrundeliegenden Aufgabenstellung, wodurch auch die exakte Lösung des Gleichungssystems nur eine Approximation an die gesuchte Lösung darstellt. Folglich erweist sich eine Näherungslösung für das Gleichungssystem mit einem Fehler in der Größenordnung des Diskretisierungsfehlers als ausreichend. Hierzu eignen sich iterative Methoden hervorragend.

2.1 Vorstellung der iterativen Verfahren

Vorgestellt werden sollen, wie bereits angeführt, im Einzelnen die Jacobi-, Gauß-Seidel- und SOR-Verfahren. Die zugrunde liegenden Ideen der Verfahren können vom Gauß-Verfahren und aufbauend aufeinander abgeleitet werden, sodass ein didaktisch sinnvoller Aufbau analog zum genetischen Prinzip möglich ist. Die jeweilige Vorschrift der Verfahren wird dabei so oft ausgeführt, d.h. es wird solange iteriert, bis eine gewünschte Genauigkeit erreicht wird.

2.2 Jacobi-Verfahren

Zur Berechnung der nächsten Iterierten $x^{(k+1)}$ wird beim Jacobi-Verfahren ausschließlich auf die alte Iterierte $x^{(k)}$ zugegriffen. Die zugrunde liegende Idee des Jacobi-Verfahrens ist es, die i -te Zeile des Systems (1) nach x_i umzustellen. Damit ergibt sich als Verfahrensvorschrift für $x^{(k+1)} = (x_1^{(k+1)}, x_2^{(k+1)}, \dots, x_n^{(k+1)})$:

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right) / a_{ii} \quad (2)$$

In letzter Zeit wurde dem Jacobi-Verfahren wieder mehr Beachtung geschenkt. Der ausschließliche Zugriff auf die vorherige und damit verfügbare Iterierte ermöglicht eine kanonische Parallelsierung der Berechnung, die Gleichungen können unabhängig voneinander auf verschiedenen Kernen gelöst werden⁴.

⁴<http://phys.org/news/2014-06-19th-century-math-tactic-makeoverand.html>

2.2.1 Gauß-Seidel-Verfahren

Um die Konvergenzgeschwindigkeit (d.h. weniger Iterationen bei gleichbleibender Genauigkeit) zu erhöhen, kann für die Berechnung von $x_i^{(k+1)}$ auf die bereits berechneten und damit verfügbaren $x_1^{(k+1)}, x_2^{(k+1)}, \dots, x_{i-1}^{(k+1)}$ zugegriffen werden. Es ergibt sich damit:

$$x_i^{(k+1)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right) / a_{ii} \quad (3)$$

2.2.2 SOR-Verfahren

Das SOR-Verfahren (engl. *Successive Over-Relaxation, Überrelaxationsverfahren*) ist eine Modifizierung des Gauß-Seidel-Verfahrens, bei dem nicht der aus der Vorschrift erhaltene Wert, sondern ein um den Parameter $\omega \in (0, 2)$ vergrößerter Wert auf den vorherigen Schritt aufaddiert wird.

$$x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right) \quad (4)$$

Für $\omega = 1$ ergibt sich wieder das Gauß-Seidel-Verfahren. Im Gegensatz zum Gauß-Seidel-Verfahren wird beim Jacobi-Verfahren nur in speziellen Fällen eine Relaxation genutzt.

2.3 Konvergenz der Verfahren

Die Konvergenz der Verfahren ist abhängig von den Eigenschaften der Matrix A , genauer von der *Symmetrie*, *positiven Definitheit* und der *strikten Diagonaldominanz*.

Definition. Eine Matrix $A \in \mathbb{R}^{n \times n}$ heißt

- symmetrisch, falls $A = A^T$ gilt, also $a_{ij} = a_{ji}$ für $i, j = 1, \dots, n$ ist,
- positiv definit, falls $x^T A x > 0$ für alle $x \in \mathbb{R}^n$ ist,
- strikt diagonaldominant, falls $|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|$ ist.

Die Konvergenz aller Verfahren ist für strikt diagonaldominante (SDD) Matrizen sicher gestellt. Das Gauß-Seidel- und das SOR-Verfahren konvergieren zudem ebenfalls für symmetrische und positiv definite (SPD) Matrizen:

Verfahren	Konvergenzkriterium
Jacobi	SDD
Gauß-Seidel	SDD oder SPD
SOR	SDD (mit $0 < \omega \leq 1$) oder SPD

Die Konvergenzkriterien sind hinreichend, d.h. sie beweisen nicht die Divergenz bei Nichterfüllung. Während sowohl die Symmetrie als auch die strikte Diagonaldominanz direkt

von der Matrix A abgelesen werden können, muss die positive Definitheit mittels mathematischer Software (Mathematica, Wolfram Alpha, Matlab o.ä.) überprüft werden. Eine schwächere Bedingung zur Konvergenz ist die *irreduzibele Diagonaldominanz* der Matrix A . Weiterführende Informationen und die zugehörigen Beweise sind in jedem Fachwerk über Numerik linearer Gleichungssystem zu finden, z.B. [4]. Auf wikipedia.org ist zudem eine kurze und übersichtliche Einführung der Verfahren zu finden.

2.4 Abbruchkriterien

Wie bereits erwähnt wird als erster Schritt $x^{(0)} = \vec{0}$ gesetzt. Wie oft soll nun allerdings das iterative Verfahren durchgeführt werden? Gebräuchlich sind die folgenden 3 Kriterien.

Maximale Anzahl der Iterationen Das Verfahren wird nach einer fixen Anzahl an Iterationen abgebrochen, $k < \text{max_iter}$, wobei max_iter die maximale Anzahl der Iterationen und k die jeweilige k -te Iteration.

Residuum Das Residuum für das System (1) und die k -te Iterierte $x^{(k)}$ ist definiert als $\text{res}(x^{(k)}) = \|Ax^{(k)} - b\|$. Für die exakte Lösung x^* gilt offensichtlich $\text{res}(x^*) = 0$. Die Lösung der iterativen Verfahren, sofern nicht ausiteriert, ist dabei allerdings nur eine Näherungslösung. Für viele folgende Schritte in aufeinander aufbauenden Verfahren in praktischen Anwendungen ist dies jedoch, wie beschrieben, vollkommen ausreichend. Als Abbruchbedingung wird daher auch oft $\text{res}(x) < \text{tol}$ mit z.B. $\text{tol} = 10^{-12}$ genutzt.

Relative oder absolute Änderung in den Iterationen Die Iteration kann ebenfalls abgebrochen werden, sobald ein weiterer Schritt nur noch eine kleine Änderung (bzw. Verbesserung) erzielt. Die absolute Änderung in den Iterationen ist definiert als $\text{error}_{abs} = \|x^{(k)} - x^{(k+1)}\|$, die relative Änderung als $\text{error}_{rel} = \|x^{(k)} - x^{(k+1)}\| / \|x^{(k)}\|$. Ist nun der relative oder der absolute Fehler kleiner als eine gewünschte Toleranz, kann das Verfahren terminiert werden.

3 Vorstellung der Verfahren im Unterricht

Am Beginn der Unterrichtseinheit wird die Verwendung von iterativen Verfahren motiviert, in dem auf große Systeme bei praxisrelevanten Problemstellungen und die späteren, von den Lernenden durchgeführten Experimenten verwiesen wird. Anknüpfend an das bekannte Wissen wird dann mit den Lernenden exemplarisch das Gauß-Verfahren als direktes Verfahren für eine konkrete 3×3 -Matrix durchgerechnet und wiederholt. Idealerweise sollte hierfür eine Matrix verwendet werden, die auch in den späteren Experimenten betrachtet wird, z.B.:

$$A = \begin{pmatrix} 4 & -2 & 0 \\ -2 & 4 & -2 \\ 0 & -2 & 4 \end{pmatrix} \quad \text{und} \quad b = \begin{pmatrix} -0.5 \\ -0.5 \\ -0.5 \end{pmatrix} . \quad (5)$$

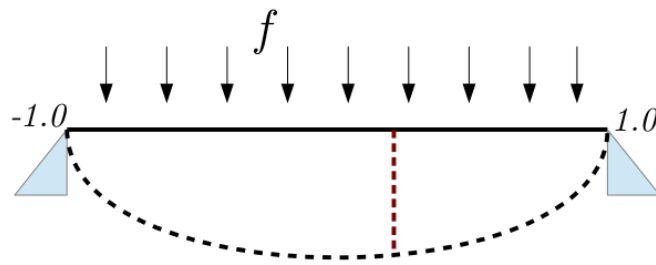


Abbildung 1: Der auf den Enden aufliegende Körper wird durch die Kraft f verformt.

Die Matrix A tritt bei der Diskretisierung des *Obstacle-Problems* auf. Dabei wirkt auf einen Körper eine Kraft von oben, sodass sich der Körper nach unten durchbiegt und verformt (1). Die Lösung der Gleichung beschreibt dann die Verformung.

Da die Matrix A symmetrisch und positiv definit ist, jedoch nicht strikt diagonaldominant, konvergieren das Gauß-Seidel- und das SOR-Verfahren, wohingegen über das Jacobi-Verfahren keine Aussage getroffen werden kann, da das Konvergenzkriterium hinreichend ist.

Es folgt ein Abstraktionsschritt von einer konkreten Matrix A hin zu einer allgemeinen 3×3 -Matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{11} \\ a_{21} & a_{22} & a_{22} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}.$$

Als erste Idee wird nun ein iteratives Verfahren konstruiert, bei dem die erste Gleichung nach x_1 , die zweite Gleichung nach x_2 und die dritte Gleichung nach x_3 aufgelöst wird, welches dem Jacobi-Verfahren entspricht. Um das iterative Vorgehen der Verfahren nun zu vermitteln können die ersten zwei, drei Schritte des Jacobi-Verfahrens zum Lösen des vorgestellten Systems (5) von den Lernenden oder gemeinsam an der Tafel gelöst werden, bevor schrittweise das Gauß-Seidel- und das SOR-Verfahren vorgestellt werden.

Es folgt die Verallgemeinerung von einem System mit 3 Gleichungen auf ein System mit $n \in \mathbb{N}$ Gleichungen, also auf Matrizen $A \in \mathbb{R}^{n \times n}$.

Als erste Iteration wird für gewöhnlich $x^{(0)} = \vec{0}$ gewählt. Die Lernenden werden anschließend gefragt, wie lange das Verfahren durchgeführt werden soll. Im Lehrer-Schüler-Gespräch werden mögliche Abbruchbedingungen erarbeitet (im Programmgerüst und Pseudocode wird eine Kombination von maximaler Iteration und absoluter Änderung der Iterierten verwendet). Anschließend wird der Pseudocode schrittweise vorgestellt und besprochen. Zusätzliche Anmerkungen und Hinweise sind im online verfügbaren Foliensatz enthalten.

Die Lernenden sind dann angehalten, den Pseudocode in Einzel- oder Gruppenarbeit möglichst selbstständig am Computer umzusetzen.

3.1 Unterrichtsmaterialien

Für den Unterricht stehen

- Handouts mit den Formeln, Pseudocode und kurzen Hinweisen,
- grobes Gerüst für das Program in Java,
- Folien mit Kommentaren für die Lehrperson

bereit. Die Folien können ergänzend zur Tafel für Erklärungen und die Vorstellung der Verfahren genutzt werden. Nach der Besprechung des Pseudocodes können die Handouts ausgegeben und ebenfalls kurz vorgestellt und besprochen werden. Auf den Handouts sind die einzelnen Verfahren mit den jeweiligen Vorschriften kurz beschrieben, der Pseudocode ist ebenfalls zur vereinfachten Umsetzung auf dem Computer enthalten. Die Aufgabe der Schüler ist es nun, den Pseudocode in das Programmgerüst umzusetzen. Im Programm sind zudem 6 Beispielmatrizen mit jeweils unterschiedlichen Eigenschaften implementiert.

3.2 Arbeitsformen

Der Unterricht gliedert sich in zwei große Blöcke, der gemeinsamen Herleitung der Verfahren und die Modellierung der Verfahren am Computer. Die Vorstellung der Verfahren geschieht hauptsächlich in Präsentationsform der Lehrperson, wobei ein Lehrer-Schüler-Gespräch zur gemeinsamen Erarbeitung zielführend erscheint. Das Nachvollziehen der Algorithmen kann in Einzel- oder Gruppenarbeit geschehen. Je nach Vorliebe können anschließend die Lernenden in kleineren Gruppen oder allein in Projektarbeit mit Unterstützung der Lehrperson die Verfahren modellieren. Es ist dabei ratsam, den Lernenden ausreichend Zeit zur selbständigen Fehlersuche zu geben und am Beginn auf Fragen von Seite der Lernenden zu warten anstatt in den Lernprozess einzugreifen. Zum einen können Fehler das Verständnis der Arbeitsweise des Computers und des Algorithmus fördern, zum anderen können die Lernenden selbstständig zuerst an der Fehlerbehebung im Sinne eines selbstgesteuerten Lernprozesses in dieser Phase arbeiten.

3.3 Herausforderungen für die Lernenden

Die zwei wesentlichen Abstraktionsschritte von einer Matrix mit konkreten Zahlen hin zu einer allgemeinen 3×3 -Matrix und anschließend zu einer allgemeinen Matrix des $\mathbb{R}^{n \times n}$ können die Lernenden vor Herausforderungen stellen. Wiederholen, Nachfragen und Paraphrasieren der Lernenden können dabei helfen, das Verstehen der Sachverhalte sicher zu stellen. Der Wissenstransfer von der bekannten Formel für das Gauß-Verfahren hin zu den iterativen Verfahren ist ebenfalls ein für die Lernenden anspruchsvoller Übergang, falls nicht schon iterative Verfahren für andere Problemstellungen bekannt sind. Die Vorstellung eines vermeintlich ungenauen Verfahrens, das zudem nach einer nicht vorher bekannten Rechenzeit abgebrochen werden muss, steht für viele Lernende im Gegensatz zum Bild der exakten Wissenschaft der Mathematik, zumal ja auch das exakte Gauß-Verfahren benutzt werden könnte. Es kann dann auf die anstehenden Experimente

verwiesen werden, die Effizienz der Verfahren wird ja erst im nächsten Schritt untersucht und verglichen. Hilfreich kann auch ein Hinweis auf andere, etwas ungenaue Verfahren im Alltag sein, wie z.B. das Runden anstelle der exakten Berechnung zur Ermittlung der Größenordnung des Ergebnisses.

Überraschenderweise wurde das zur Verfügung gestellte Programmgerüst von den Lernenden als restriktiv wahrgenommen, sollte es doch eigentlich eine hilfreiche Anleitung und Unterstützung sein, sodass nicht alle Hilfsfunktionen (Normberechnung, Ausgabe-funktion) von den Schülern selbst implementiert werden müssen.

Die Begriffe der Konvergenz und Divergenz können die Lernenden ebenfalls vor eine Herausforderung stellen. Ein kleines Residuum (d.h. nahe 0) ist dabei kein ausreichendes Kriterium für die Konvergenz, nur bei einem mit fortschreitender Iterationszahl fallenden Residuum kann von Konvergenz gesprochen werden. Die Beobachtungen mit dem Programm bieten eine gute Möglichkeit die Begriffe Konvergenz und Divergenz von den Verfahren zu thematisieren und am praktischen Beispiel und den gemachten Experimenten durchzusprechen. Gerade divergierende Fälle legen auch für die Lernenden das Eingehen auf diese Begriffe nahe.

4 Erfahrungen bei der Durchführung

Die vorgestellten Inhalte wurden zusammen mit den Materialien an der HTL in Braunau im Rahmen des Freifachs *Universitätsmathematik* thematisiert und genutzt. Bereits erwähnt wurde die eher negative Aufnahme des Programmgerüsts von Seiten der Lernenden. Genutzt wurde zudem die Programmiersprache C. Besser geeignet für eine erneute Durchführung erscheint, aufgrund der Plattformunabhängigkeit, jedoch Java.

Das Experimentieren mit den Verfahren bereiteten den Schülerinnen und Schülern große Freude und erschloss ihnen ein anderes, bisher unbekanntes Bild auf die Mathematik. Es gab nicht "das eine, richtige" Verfahren, sondern jedes Verfahren hat seine Vor- und Nachteile. Abhängig von der Problemstellung, d.h. der auftretenden Matrix, konnten - und unter Umständen mussten - sich die Schülerinnen und Schüler für ein anderes Verfahren entscheiden. Nach einer ersten Implementierung und kurzen Hinweisen durch die Lehrperson zu möglichen Verbesserung wurde zudem der Ehrgeiz der Schülerinnen und Schüler entfacht, es gab regelrechte Wettberechnungen welches Verfahren mit der jeweiligen Implementation schneller war und für noch größere Systeme funktionierte. Ein besonderer Reiz lag natürlich auch im Wissen über die Aktualität dieser Verfahren für die Forschung und Praxis. Für gewöhnlich werden die iterativen Verfahren erst im zweiten oder dritten Semester des Mathematikstudiums eingeführt, die Schülerinnen und Schüler waren dementsprechend auch stolz bereits in der Schule mit diesen Verfahren umzugehen. Zudem wurde auch eine Diplomarbeit an der HTL Braunau in diesem Themenbereich angefertigt.

5 Didaktische Anmerkungen

Bei jedem Modellierungsprozess kommt der Beschreibung eines Lösungsalgorithmus in Form eines Pseudocodes besondere Bedeutung zu. Daher stehen die Pseudocodeformulierungen der einzelnen iterativen Verfahren bzw. Methoden im Mittelpunkt des Unterrichtskonzepts. Damit folgt die skizzierte Unterrichtssequenz einem Konzept, das bereits 1988 in Form eines fünfstufigen Strukturierungsplans sowie zweier Hypothesen formuliert wurde:

“... These 1:

... ein Unterricht, der an der Übersetzung (= Kodierungsebene) beginnt, muss versagen, da er ein stufenweises System der Problemlösung auf den Kopf stellt ...” [1]

In These 2 wird die Bedeutung der grafischen Darstellung von Lösungsverfahren zur Diskussion gestellt. Flussdiagramme nach DIN 66001 [3] oder Struktogramme [5] werden als Beschreibungsformen zur Erzielung höherer Übersichtlichkeit vorgestellt. Auf die Zwischenstufe II im Problemlöseplan (2) wurde im Unterrichtskonzept verzichtet. Sie bietet sich aber als Übungs- / Ergänzungseinheit zum vorliegenden Konzept an. Hier besitzt die Lehrperson völlige Entscheidungsfreiheit. Je nach Verlauf der Unterrichtseinheiten können einzelne grafische Repräsentationen ergänzt werden. In einem weiteren Beitrag zur Didaktik der Informatik wurde von Fuchs [2, S. 44] noch einmal auf die fundamentale Bedeutung von Beziehungsstrukturen als Bausteine von Algorithmen hingewiesen.

Bei der Übersetzung in die Sprache C++ bzw. Java werden die Schritte III, IV und V (Handsimulation, Kodierung und Testen) aus dem Strukturplan realisiert. Effizienzprüfungen der einzelnen Verfahren erfolgt über Rechenzeitvergleiche.

In einer modernen Informatikdidaktik würde wohl das Wort *Computer* durch das Wort *Informatiksystem* ersetzt werden. Computer - auch in Schulen - sind mittlerweile nicht nur “einfache Rechner”, sondern “ ... Informatiksysteme als spezifische Zusammenstellung von Hardware, Software und Netzverbindungen zur Lösung eines Anwenderproblems ...” [6].

6 Vertiefungsmöglichkeiten und Ausblick

In folgenden Unterrichtseinheiten können mit den modellierten Verfahren Experimente unter diversen Gesichtspunkten (Rechenzeit, Genauigkeit, Anzahl der benötigten Iterationen, Speicherbedarf) durchgeführt werden. Exemplarische Matrizen sind im Programmgerüst implementiert und können beim Programmstart ausgewählt werden.

Das vorgestellte Unterrichtskonzept bietet darüber hinaus viele Möglichkeiten zur vertieften Betrachtung in Form von weiteren Unterrichtseinheiten, Diplomarbeiten an berufsbildenden höheren Schulen oder auch Bachelorarbeiten an Hochschulen. Denkbar ist auch eine Betrachtung von iterativen Verfahren im Allgemeinen (Bspw. Newton-Verfahren), ein Vergleich der verschiedenen Abbruchkriterien und ihrer Anwendbarkeit oder die Möglichkeiten zur Parallelisierung der Berechnungen. Außerdem könnte das *Sparse-Format*,

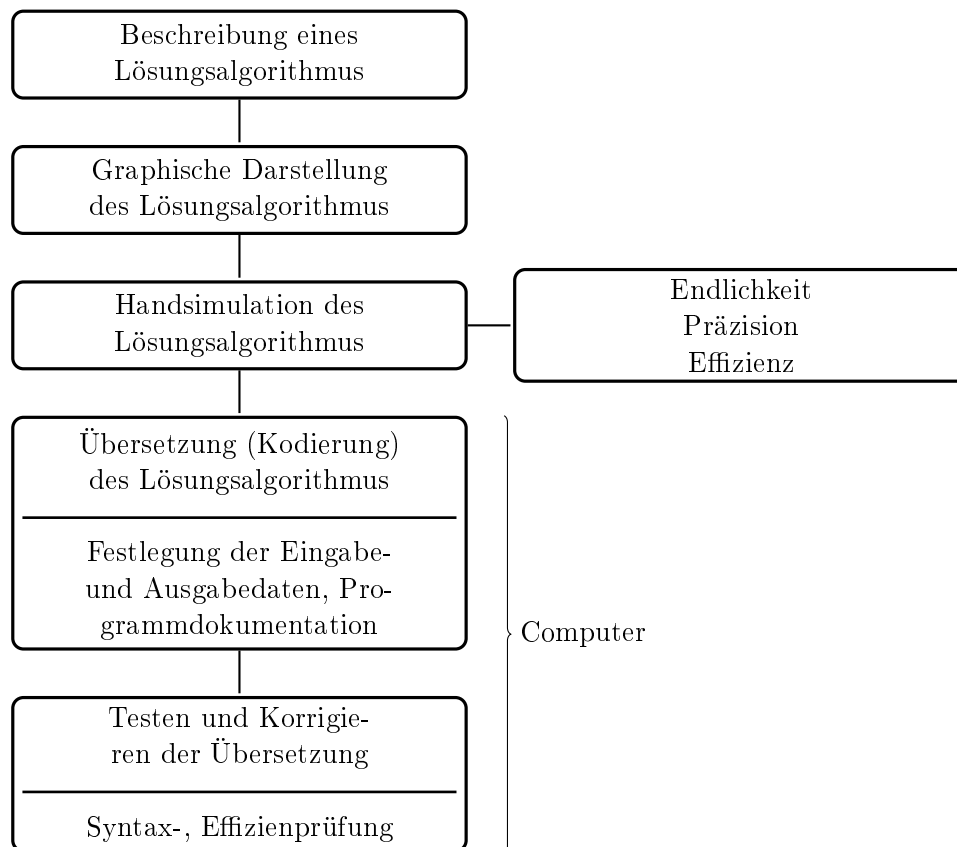


Abbildung 2: Stufenweiser Problemlöseplan nach Fuchs [1, S. 249]

bei dem nur die Nicht-Null-Einträge und deren Position in den Matrizen und Vektoren abgespeichert werden, mit dem vollen Abspeichern der Matrix bezüglich des Speicherbedarfs und der Rechenzeit verglichen werden.

Literatur

- [1] K. Fuchs. “Erfahrungen und Gedanken zu Computern im Unterricht”. In: *Journal für Mathematik-Didaktik* 2/3 (1988), S. 247–256.
- [2] K.J. Fuchs. “Didaktik der Informatik: Die Logik fundamentaler Ideen”. In: *Medien und Bildung* 4+5 (1994), S. 42–45.
- [3] Klein. *Einführung in die DIN – Normen*. 1980.
- [4] Andreas Meister. *Numerik linearer Gleichungssysteme*. Springer Fachmedien Wiesbaden. Springer, 2015. ISBN: 978-3-658-07200-1.

- [5] I. Nassi und B. Shneiderman. “Flowchart Techniques for Structured Programming”. In: *SIGPLAN Not.* 8.8 (Aug. 1973), S. 12–26. ISSN: 0362-1340. DOI: 10.1145/953349.953350. URL: <http://doi.acm.org/10.1145/953349.953350>.
- [6] S. Schubert und A. Schwill. *Didaktik Der Informatik*. Spektrum Akademischer Verlag, 2011. ISBN: 9783827426536. URL: <https://books.google.at/books?id=z8E1BAAAQBAJ>.