Numerik 1 21.12.2018 Wahid Far, Okhtay

2.Projekt:Iterative Löser für LGSe

Bei diesem Projekt soll mithilfe der Splitting Verfahren (Jacobi, Gauß-Seidel,SOR) der Ergebnisvektor eines LGS iterativ also näherungsweise bestimmt werden. Sie stellen alle indirekte Verfahren dar.

Theorie :

Im Allgemenein orientieren sich die Splitting Verfahren am banachschen Fixpunksatz, wo die Fixpunktiterationen gegen einen Wert konvergieren.
Das heißt, das iterative Lösen eines LGS beruht auf die näherungsweisen Bestimmung des Lösungsvektors $x \in \boldsymbol{R}^{n,1}$ bzgl des LGS Ax= b mit $A \in \boldsymbol{R}^{n,n}$ und $b \in \boldsymbol{R}^{n,1}$. Hierbei wird zuallerst der Lösungsvektor x erraten,wodurch man diesbezüglich einen Ergebnisvektor $b^n mit n \in \boldsymbol{R}$ erhält.
Durch das iterative Verfahren nähert sich dieser Ergebnisvektor b^n immer mehr dem tatsächlichen Ergebnisvektor b an, wodurch näherungsweise der Lösungsvektor x bestimmt wird. Das unterscheidet sich stark zum direkten Verfahren (Gaußelimination) zur Lösung eines LGS, wo durch äquivalente Matrixumformungen der Lösungsvektor direkt bestimmt wird.
Die expliziten Verfahrensvorschriften der Splitting Verfahren zeichnen sich durch folgende Formeln in Matrixschreibweise(E= Einheitsmatrix, D=Diagonalmatrix, A=gegebene Matrix, L = linke untere Dreiecksmatrix, U= rechte obere Dreiecksmatrix , $k \in \boldsymbol{R}$,) aus:

1) Jacobi Verfahren: $x^{k+1} = (E - D^{-1} A) x^k + D^{-1} b$
2) Gauß-Seidel Verfahren: $x^{k+1} = (E - (D-L)^{-1}) x^k + (D-L)^{-1} b$
3) SOR-Verfahren: $x^{k+1} = (E - (\frac{1}{\omega} D - L)^{-1} A) x^k + (\frac{1}{\omega} D - L)^{-1} b$ mit $\omega > 0$ und $\omega \in \boldsymbol{R}$

Implementierung

In Aufgabe 1 wurden die Splitting-Verfahren Jacobi, Gauß-Seidel und SOR-Verfahren mithilfe der Summenschreibweise umgesetzt, um ein effektives Implementieren zu ermöglichen. Zudem wurde eine Toleranz tol also || $b - Ax^k$ || / ||b|| <= tol festgelegt, die den Abstand zwischen dem iterativen und dem tatsächlichen Ergebnisvektor sinnvoll festlegt. (it= Anzahl der Iterationen, itMax= vorgegebene Iterationenanzahl)

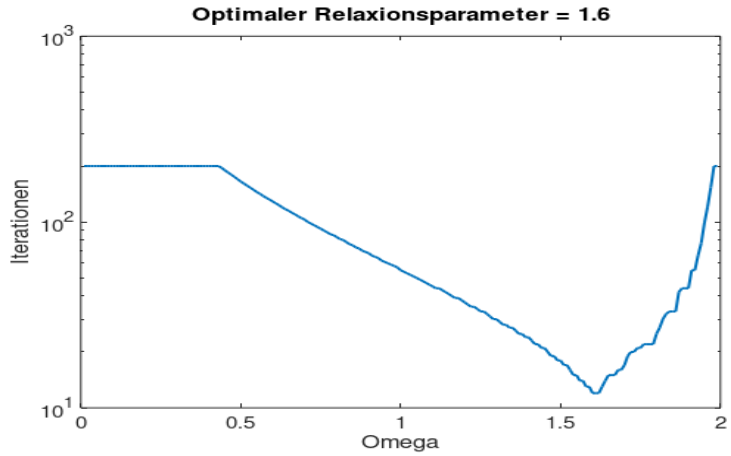
Code:

```
%Abbruchbedingung
while ( (err/norm(b)) > tol && it < itMax )
%x_neu ( x ) berechnen
  for i = 1:n
    x(i) = (b(i) - A(i,[1:i-1, i+1:n] )*x_alt([1:i-1,i+1:n]) )/ A(i,i);
  end
end
```

Die Verfahren unterscheiden sich in den Abläufen folgendermaßen:
i) Beim Jacobi Verfahren wird der zuvor erratende Ergebnisvektor x bei allen Iterationen weiterverwendet.
ii) Beim Gauß-Seidel Verfahren wird der zuvor erratende Ergebnisvektor x sukzessiv

überschrieben und bei den nächsten Iterationen weiterverwendet.
iii) Das SOR-Verfahren verhält sich genauso wie das Gauß-Seidel Verfahren bloß das für den Relaxationsparameter ω folgendes gilt : $\omega > 0$ und $\omega \in \boldsymbol{R}$

In der Aufgabe 2 sollten wir anhand einer gegebenen Tridiagonalmatrix bestimmen, wie groß der Relaxationsparameter ω sein müsste, um die kleinste Anzahl von Iterationen zu erreichen. Hierbei bestimmen wir den Index mit den geringsten Iterationen und erkennen so den optimalen Relaxationsparameter ω . Der optimale Relaxationsparameter liegt bei ω = 1.6 .



In der Aufgabe 3 sollte eine Funktion f(x) bestimmt werden für die folgendes gilt:
-u''(x) = f(x) , 0<x<1 mit u(0)=u(1)=0 , f(x) = 1, $A = h^{-2} * tridiag_{n-1}(1,2,-1)$ mit Au=f und f(x) = $1/x^2$. Anhand der Indexfindung im Code können wir die Randpunkte im Grafen und somit die Lösung bestimmen. Je kleiner h ist desto exakter ist hierbei der Graf.

Code:

```
%X-Werte in die Funktion einsetzen
for i = 1:n-1
  b(i) =  f(x(i+1))
end
%Ergebnis berechnen
[u,~]=SOR(A,b,u,1.6,1e-10,200);
u = [0;u;0]       %Randbpunkte bestimmen
```

In der Abbildung erkennt man die Lösung für h=0.01.