Programmierblatt 3.

Besprechungswoche: **07–11.12.2020**

Auf diesem Programmierblatt betrachten wir die numerische Lösung des Poisson-Problems

$$\begin{aligned} -\Delta\, u(\mathbf{x}) &= f(\mathbf{x}) \quad \text{für } \mathbf{x} \in \Omega, \\ u(\mathbf{x}) &= g(\mathbf{x}) \quad \text{für } \mathbf{x} \in \partial\Omega, \end{aligned} \tag{1}$$

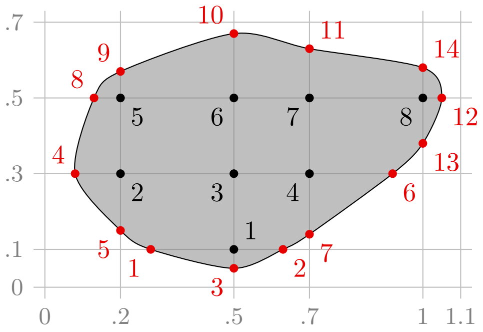
und eines verwandten Problems auf beliebigen beschränkten Gebieten $\Omega \subset \mathbb{R}^2$ mit Hilfe der Shortley-Weller-Approximation.

Diskretisierung des Gebietes

Um die Gleichung (1) mit der Shortley-Weller-Approximation diskretisieren zu können, benötigen wir eine rechtwinklige Gitternetzapproximation von Ω. Solche Gitter können kanonisch wie folgt durch drei Matrizen beschrieben werden:

- Eine Matrix $\mathbf{G} \in \mathbb{R}^{2\times n_d}$, welche die Koordinaten der inneren Gitterpunkte spaltenweise abspeichert,
$$\mathbf{G} = \begin{bmatrix} x_{d,1} & x_{d,2} & \cdots & x_{d,n_d} \\ y_{d,1} & y_{d,2} & \cdots & y_{d,n_d} \end{bmatrix},$$
- eine Matrix $\mathbf{B} \in \mathbb{R}^{2\times n_b}$, welche die Koordinaten der notwendigen Gitterpunkte am Gebietsrand spaltenweise abspeichert,
$$\mathbf{B} = \begin{bmatrix} x_{b,1} & x_{b,2} & \cdots & x_{b,n_b} \\ y_{b,1} & y_{b,2} & \cdots & y_{b,n_b} \end{bmatrix},$$
- und eine Matrix $\mathbf{C} \in \mathbb{Z}^{4\times n_d}$, welche in der j-ten Spalte Indizes auf den linken, den rechten, den unteren und den oberen Gitternachbarspunkt von dem j-ten inneren Gitterpunkt hat. Dabei wenden wir folgende Konvention an: Ist der Index positiv, so ist ein Punkt aus dem inneren gemeint, d.h. der Index weist auf die Spalte von **G**. Ist der Index negativ, so ist ein Punkt aus dem Gebietsrand gemeint, d.h. der Absolutbetrag des Indexes weist auf die Spalte von **B**.

Für ein einfaches Beispiel (siehe Beilage `bsp.m`) sieht dies dann wie folgt aus:



$$\mathbf{G} = \begin{bmatrix} .5 & .2 & .5 & .7 & .2 & .5 & .7 & 1 \\ .1 & .3 & .3 & .3 & .5 & .5 & .5 & .5 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} .28 & .63 & .5 & .08 & .2 & .92 & .7 & .13 & .2 & .5 & .7 & 1.05 & 1 & 1 \\ .1 & .1 & .05 & .3 & .15 & .3 & .14 & .5 & .57 & .67 & .63 & .5 & .38 & .58 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} -1 & -4 & 2 & 3 & -8 & 5 & 6 & 7 \\ -2 & 3 & 4 & -6 & 6 & 7 & 8 & -12 \\ -3 & -5 & 1 & -7 & 2 & 3 & 4 & -13 \\ 3 & 5 & 6 & 7 & -9 & -10 & -11 & -14 \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} .22 & .12 & .3 & .2 & .07 & .3 & .2 & .3 \\ .13 & .3 & .2 & .22 & .3 & .2 & .3 & .05 \\ .05 & .15 & .2 & .16 & .2 & .2 & .2 & .12 \\ .2 & .2 & .2 & .2 & .07 & .17 & .13 & .08 \end{bmatrix}$$

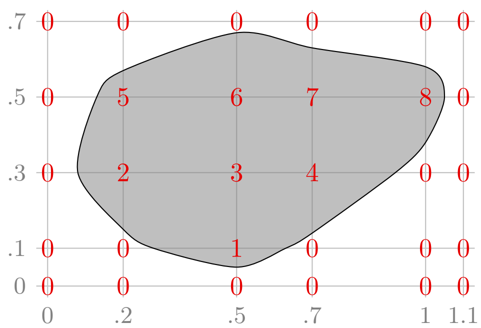
Für die Shortley-Weller-Approximation ist es äusserst praktisch, wenn man noch zusätzlich eine Matrix $\mathbf{H} \in \mathbb{R}^{4\times n_d}$ speichert, welche die gleiche Struktur wie **C** hat, aber statt den Indizes den Abstand zu dem jeweiligen Gitternachbarspunkt speichert.

Im Allgemeinen ist das Gebiet Ω nicht durch ein schon diskretisiertes, rechtwinkliges Gitternetz gegeben, sondern durch eine andersartige Beschreibung. Wir betrachten hierzu für dieses Blatt folgende Art der Beschreibung von Ω: Das Gebiet sei durch eine *Level-set* Funktion $\varphi\colon \mathbb{R}\times\mathbb{R}\to\mathbb{R}$ und zugehöriger Schranke $c\in\mathbb{R}$ als

$$\Omega := \left\{(x,y)\in\mathbb{R}^2 \,:\, \varphi(x,y) > c\right\}$$

gegeben. Dabei nehmen wir an, dass jeweils $\partial\Omega \subset \{(x,y)\in\mathbb{R}^2 \,:\, \varphi(x,y)\leq c\}$ gilt, also dass Ω offen ist. Der Einfachheit halber betrachten wir im Folgenden nur die Bestimmung von achsenparallelen Gittern. Seien dafür Partitonen $x_1 < x_2 < \ldots < x_r$ der x-Achse und $y_1 < y_2 < \ldots < y_s$ der y-Achse vorgegeben, wobei wir annehmen, dass diese $f(x_1,y), f(x_r,y), f(x,y_1), f(x,y_s) \leq c$ für alle $x_1\leq x\leq x_r$ und $y_1\leq y\leq y_s$ erfüllen. Dann berechnet man wie folgt das zugehörige Gitter:

- Zuerst berechnet man $\mathbf{I} = \left[i_{j,k}\right]_{j,k=1}^{r,s}$, wobei $i_{j,k}=0$ für alle i,j mit $\varphi(x_j,y_k)\leq c$ ist. Die anderen Einträge enumeriert man mit den paarweise verschiedenen Zahlen $1,2,\ldots,n_d$. Für das Beispiel ergibt sich bei einem spaltenorientierten Vorgehen also:



$$\mathbf{I} = \begin{bmatrix} 0&0&0&0&0\\ 0&0&2&5&0\\ 0&1&3&6&0\\ 0&0&4&7&0\\ 0&0&0&8&0\\ 0&0&0&0&0 \end{bmatrix}$$

- Als nächstes definiert man $\mathbf{C} = \left[c_{\ell,i}\right]_{\ell,i=1}^{4,n_d}$ und $\mathbf{G} = \left[g_{\ell,i}\right]_{\ell,i=1}^{2,n_d}$. Dafür iteriert man wieder über $\mathbf{I}$: Ist $i_{j,k}\neq 0$ dann setzt man $c_{1,i_{j,k}} = i_{j-1,k}$, $c_{2,i_{j,k}} = i_{j+1,k}$, $c_{3,i_{j,k}} = i_{j,k-1}$ und $c_{4,i_{j,k}} = i_{j,k+1}$, sowie $g_{1,i_{j,k}} = x_j$ und $g_{2,i_{j,k}} = y_k$. Für das Beispiel ergibt sich also:

$$\mathbf{C} = \begin{bmatrix} 0&0&2&3&0&5&6&7\\ 0&3&4&0&6&7&8&0\\ 0&0&1&0&2&3&4&0\\ 3&5&6&7&0&0&0&0 \end{bmatrix} \qquad \mathbf{G} = \begin{bmatrix} .5&.2&.5&.7&.2&.5&.7&1\\ .1&.3&.3&.3&.5&.5&.5&.5 \end{bmatrix}$$

- Die Anzahl Nullen in $\mathbf{C}$ ist nun genau die Anzahl n_b. Damit kann man $\mathbf{B} = \left[b_{\ell,i}\right]_{\ell,i=1}^{2,n_b}$ ansetzen. Per Iteration über die Einträge von $\mathbf{I}$ berechnet man nun die notwendigen Gitterpunkte am Gebietsrand. (Man kann hier gleichzeitig das $\mathbf{H}$ berechnen.) Ist $i_{j,k}\neq 0$ dann schaut man man zuerst nach links; ist $i_{j-1,k}=0$ und daher auch $c_{1,i_{j,k}}=0$, so muss man einen Randpunkt in Richtung links bestimmen. Ist dies das m-te Mal, wo man einen Randpunkt sucht, so setzt man $c_{1,i_{j,k}} = -m$. Dann sucht man per Bisektion die Stelle links vom Punkt (x_j,y_k), d.h. ein (ξ,y_k) mit $\xi\in(x_{j-1},x_j)$ wo φ von $>c$ zu $\leq c$ wird und setzt schliesslich $b_{1,m}=\xi$ und $b_{2,m}=y_k$. Analoges macht man nach rechts, wenn $i_{j+1,k}=0$; nach unten, wenn $i_{j,k-1}=0$, und oben, wenn $i_{j,k+1}=0$ ist. Für das Beispiel ergibt sich bei einem spaltenorientierten Vorgehen also:

$$\mathbf{C} = \begin{bmatrix} -1&-4&2&3&-8&5&6&7\\ -2&3&4&-6&6&7&8&-12\\ -3&-5&1&-7&2&3&4&-13\\ 3&5&6&7&-9&-10&-11&-14 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} .28&.63&.5&.08&.2&.92&.7&.13&.2&.5&.7&1.05&1&1\\ .1&.1&.05&.3&.15&.3&.14&.5&.57&.67&.63&.5&.38&.58 \end{bmatrix}$$

Aufgabe 1 (Erstellen des Gitters).

Schreiben Sie die Funktion

```
function [C, H, G, B] = makeGrid(phi, c, xs, ys, p),
```

welche die oben beschriebene Gittererstellung umsetzt. Die Funktion φ wird dabei als *Function Handle* `phi` übergeben werden. Die übergebenen Partitionen sind der Form `xs` $= \begin{bmatrix} x_1 & x_2 & \cdots & x_r \end{bmatrix}$ und `ys` $= \begin{bmatrix} y_1 & y_2 & \cdots & y_s \end{bmatrix}$. Für die Bestimmung der Gitterpunkte am Gebietsrand sollen jeweils `p` Bisektionsschritte benutzt werden.

Diskretisierung der Gleichung

Mit dieser Gitterbeschreibung zuhanden widmen wir uns der Diskretisierung der Gleichung (1) mit der Shortley-Weller-Approximation. Betrachten wir den i-ten inneren Gitterpunkt, so haben wir mit der i-ten Spalte von $\mathbf{G}$ Zugriff auf dessen Koordinaten, mit der i-ten Spalte von $\mathbf{C}$ auf die Indizes (und damit dank $\mathbf{G}$ und $\mathbf{B}$ auch Koordinaten) von seinem linken, rechten, unteren und oberen Gitternachbarspunkt. Schliesslich sind in der i-ten Spalte von $\mathbf{H}$ die Abstände zu seinem linken, rechten, unteren und oberen Gitternachbarspunkt.

Setzen wir also u_i als Approximation von u und f_i als Auswertung von f beim i-ten inneren Gitterpunkt sowie weiter g_i als Auswertung von g beim i-ten Gitterpunkte am Gebietsrand, dann nimmt die i-te Gleichung der Shortley-Weller-Approximation von (1) gemäss dem Vorlesungsskript die Form

$$\begin{aligned}\left(\frac{2}{h_{1,i}h_{2,i}}+\frac{2}{h_{3,i}h_{4,i}}\right)u_i-\sum_{\substack{\ell=1\\c_{\ell,i}>0}}^{2}\frac{2}{h_{\ell,i}(h_{1,i}+h_{2,i})}u_{c_{\ell,i}}-\sum_{\substack{\ell=3\\c_{\ell,i}>0}}^{4}\frac{2}{h_{\ell,i}(h_{3,i}+h_{4,i})}u_{c_{\ell,i}}\\-\sum_{\substack{\ell=1\\c_{\ell,i}<0}}^{2}\frac{2}{h_{\ell,i}(h_{1,i}+h_{2,i})}g_{-c_{\ell,i}}-\sum_{\substack{\ell=3\\c_{\ell,i}<0}}^{4}\frac{2}{h_{\ell,i}(h_{3,i}+h_{4,i})}g_{-c_{\ell,i}}=f_i\end{aligned}$$

an. Setzt man $\mathbf{u}=\big[u_i\big]_{i=1}^{n_d}$, $\mathbf{f}=\big[f_i\big]_{i=1}^{n_d}$ und $\mathbf{g}=\big[g_i\big]_{i=1}^{n_b}$, so lässt sich dies als lineares Gleichungssystem

$$\mathbf{Au}+\mathbf{A}_b\mathbf{g}=\mathbf{f}\quad\text{respektive}\quad\mathbf{Au}=\mathbf{f}-\mathbf{A}_b\mathbf{g}$$

schreiben. Hierbei sind die Matrizen $\mathbf{A}$ und $\mathbf{A}_b$ generell gross aber auch *dünn-besetzt* — in einer Zeile stehen in $\mathbf{A}$ und $\mathbf{A}_b$ zusammen jeweils nur fünf nicht-null Einträge — und müssen daher als `sparse`-Matrizen gespeichert werden. Dafür geht man wie folgt vor:

- Man definiert die Spaltenvektoren

$$\mathbf{a}_i=\begin{bmatrix}1\\1\\1\\1\\1\\\vdots\\n_d\\n_d\\n_d\\n_d\\n_d\end{bmatrix},\quad\mathbf{a}_j=\begin{bmatrix}1\\c_{1,1}\\c_{2,1}\\c_{3,1}\\c_{4,1}\\\vdots\\n_d\\c_{1,n_d}\\c_{2,n_d}\\c_{3,n_d}\\c_{4,n_d}\end{bmatrix}\quad\text{und}\quad\mathbf{a}_v=\begin{bmatrix}\frac{2}{h_{1,1}h_{2,1}}+\frac{2}{h_{3,1}h_{4,1}}\\-\frac{2}{h_{1,1}(h_{1,1}+h_{2,1})}\\-\frac{2}{h_{2,1}(h_{1,1}+h_{2,1})}\\-\frac{2}{h_{3,1}(h_{3,1}+h_{4,1})}\\-\frac{2}{h_{4,1}(h_{3,1}+h_{4,1})}\\\vdots\\\frac{2}{h_{1,n_d}h_{2,n_d}}+\frac{2}{h_{3,n_d}h_{4,n_d}}\\-\frac{2}{h_{1,n_d}(h_{1,n_d}+h_{2,n_d})}\\-\frac{2}{h_{2,n_d}(h_{1,n_d}+h_{2,n_d})}\\-\frac{2}{h_{3,n_d}(h_{3,n_d}+h_{4,n_d})}\\-\frac{2}{h_{4,n_d}(h_{3,n_d}+h_{4,n_d})}\end{bmatrix}.$$

- Man erstellt aus $\mathbf{a}_i$, $\mathbf{a}_j$ und $\mathbf{a}_v$ die Vektoren $\hat{\mathbf{a}}_i$, $\hat{\mathbf{a}}_j$ und $\hat{\mathbf{a}}_v$ indem man in allen drei Vektoren die Zeilen streicht, wo $\mathbf{a}_j$ negative Einträge hat. Ebenso erstellt man aus $\mathbf{a}_i$, $\mathbf{a}_j$ und $\mathbf{a}_v$ die Vektoren $\tilde{\mathbf{a}}_i$, $\tilde{\mathbf{a}}_j$ und $\tilde{\mathbf{a}}_v$ indem man in allen drei Vektoren die Zeilen streicht, wo $\mathbf{a}_j$ positive Einträge hat.
- Mit dem MATLAB `sparse`-Befehl[1] gelten nun

$$\mathbf{A} = \texttt{sparse(}\hat{\mathbf{a}}_i, \hat{\mathbf{a}}_j, \hat{\mathbf{a}}_v\texttt{)} \quad \text{und} \quad \mathbf{A}_d = \texttt{sparse(}\tilde{\mathbf{a}}_i, -\tilde{\mathbf{a}}_j, \tilde{\mathbf{a}}_v\texttt{)}.$$

Aufgabe 2 (Erstellen der Systemmatrizen)**.**
Schreiben Sie die Funktion

```
function [A, Ab] = shortleyWeller(C, H),
```

welche die oben beschriebene Systemmatrizenerstellung umsetzt.

Aufgabe 3 (Erstellen der Systemvektoren)**.**
Schreiben Sie die Funktionen

```
function fv = evaluateOnGridDomain(fh, G),
function gv = evaluateOnGridBoundary(gh, B),
```

welche die oben beschriebenen Systemvektoren erstellen. Dabei wird die Funktion f, respektive g, als *Function Handle* `fh`, respektive `gh`, übergeben. Die Function Handles sollen dabei (im Gegensatz zu der Definition von φ) ein vektorwertiges Argument nehmen. $\mathbf{f}$, respektive $\mathbf{g}$, werden als Spaltenvektor `fv`, respektive `gv`, zurückgegeben.

Numerische Beispiele

Mit den implementierten Funktionen zuhanden wollen wir diese nun anwenden um die Konvergenz des Verfahrens testen.

Aufgabe 4 (Testbeispiel)**.**
Setzen wir synthetisch $u(\mathbf{x}) = |{}^1\!/_2 x_1 + x_2|({}^1\!/_2 x_1 + x_2) + 4x_1^2 x_2$ als exakte Lösung an, so ist dieses u die Lösung von (1) mit $f(\mathbf{x}) = -{}^5\!/_2 \chi_{(.5x_1+x_2)>0} + {}^5\!/_2 \chi_{(.5x_1+x_2)<0} - 8x_2$ und $g(\mathbf{x}) = u(\mathbf{x})$.
Betrachten Sie ein Gebiet Ihrer Wahl und schreiben Sie ein Skript, welches für sukzessive feinere Achsen-Partitionen die Gleichung (1) mit der Shortley-Weller-Approximation löst und den maximalen, absoluten Fehler bei den inneren Gitterpunkten berechnet. Messen sie dabei ebenfalls die Laufzeiten, die jeweils benötigt werden
(a) um das Gitter zu erstellen,
(b) um die Systemmatrizen zu erstellen,
(c) um die Systemvektoren zu erstellen
(d) und um das lineare Gleichungssystem mit dem \-Solver zu lösen,
sowie die Anzahl innerer Gitterpunkte.

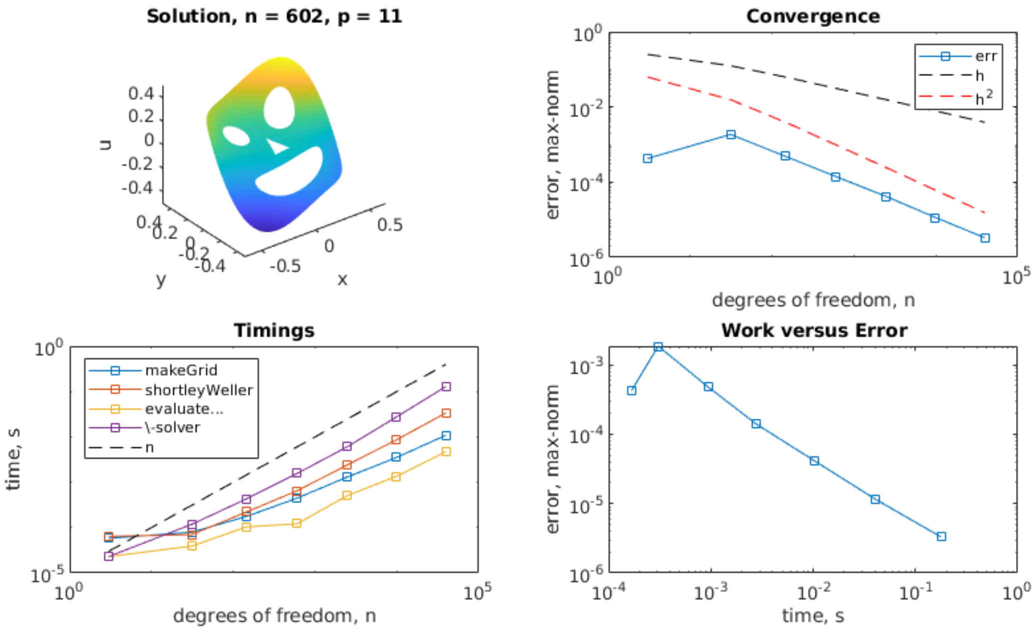
Hinweis. *In der Beilage finden Sie MATLAB-Funktionen, welche das Plotten der Lösungen ermöglichen. Sind `u`, respektive `v`, die Werte der Funktion bei den inneren Gitterpunkten, respektive den Gitterpunkten am Gebietsrand, eines Gitters, welches durch `C`, `H`, `G` und `B` gegeben ist, dann lässt sich die Lösung durch*

```
[F, P] = makeMesh(C, H, G, B);
plotMeshFunction(F, P, u, v);
```

plotten.

[1] Lesen Sie die MATLAB-Hilfe, um zu verstehen was der Befehl `A = sparse(I, J, V)` ergibt.

Sie sollten dann einen Plot der folgenden Form erstellen können, der die gleichen asymptotischen Verhalten aufzeigt:



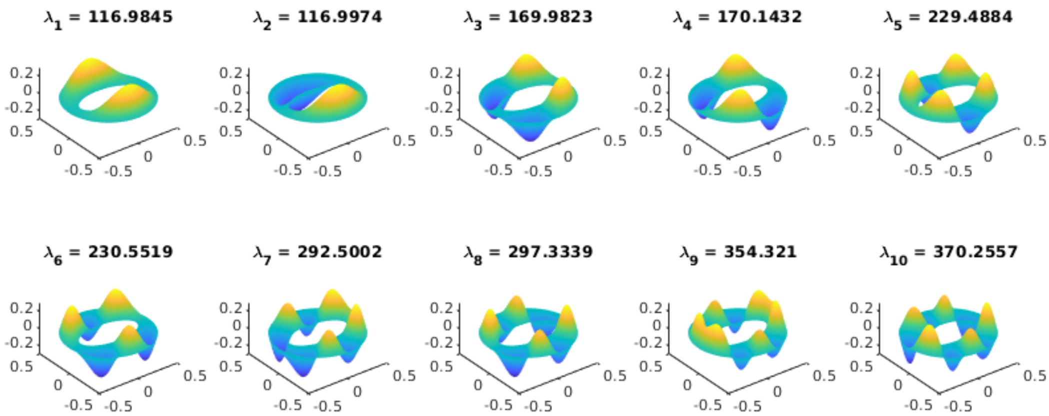
Eine Anwendung, die sich damit nun approximativ lösen lässt, ist die Bestimmung der Frequenzen und Auslenkungen der Eigenschwingunen einer 2-dimensionalen, idealen Membran Ω. Es lässt sich zeigen, dass die Auslenkung der Membran eine Eigenfunktion des Laplace-Operators ist, währenddem der zugehörige Eigenwert die zugehörige Frequenz darstellt. Das heisst wir betrachten

$$\begin{aligned} -\Delta\, u(\mathbf{x}) &= \lambda u(\mathbf{x}) && \text{für } \mathbf{x} \in \Omega, \\ u(\mathbf{x}) &= 0 && \text{für } \mathbf{x} \in \partial\Omega. \end{aligned} \tag{2}$$

Die zugehörige Shortley-Weller-Approximation ist dann durch $\mathbf{A}\mathbf{u} = \lambda\mathbf{u}$ gegeben.

Aufgabe 5 (Eigenschwingungen)**.**

Berechnen Sie damit approximativ die ersten 10 Eigenschwingungen — d.h. die Eigenfunktionen zu den betragsmässig kleinsten Eigenwerten — eines Gebietes Ihrer Wahl und stellen Sie die wie folgt dar:



Hinweis. *Benutzen Sie hierfür den MATLAB-Befehl* `eigs` *mit der Option* `'smallestabs'`*, siehe auch die MATLAB-Hilfe.*