



Programmierblatt 3.

Bearbeiten bis: **Sonntag, 10.05.2020**

Da auf den ersten zwei Blättern die Behandlung von Einfachschicht- und Doppelschichtpotentialen und -operatoren erfolgt ist, wenden wir uns nun auf diesem dritten Blatt der Greenschen Darstellungsformel zu.

Dazu sei $\Omega \subset \mathbb{R}^2$ ein beliebiges Gebiet mit $\text{diam}(\Omega) < 1$ und einem C^k -glatten und beschränkten Rand, wobei $k \geq 2$ ist. Seien weiter $\Gamma_1 \cup \dots \cup \Gamma_r = \Gamma := \partial\Omega$ die Zerlegung des Randes in dessen r Zusammenhangskomponenten und $I \cup J = \{1, \dots, r\}$ eine Zerlegung mit I nichtleer. Dann betrachten wir das Laplace-Problem

$$\begin{aligned} \Delta u(\mathbf{x}) &= 0 && \text{für } \mathbf{x} \in \Omega, \\ u(\mathbf{x}) &= g(\mathbf{x}) && \text{für } \mathbf{x} \in \Gamma_D := \bigcup_{\ell \in I} \Gamma_\ell, \\ \frac{\partial}{\partial \mathbf{n}} u(\mathbf{x}) &= h(\mathbf{x}, \mathbf{n}) && \text{für } \mathbf{x} \in \Gamma_N := \bigcup_{\ell \in J} \Gamma_\ell. \end{aligned}$$

Greensche Darstellungsformel

Die Lösung dieses Randwertproblems lässt sich mit der Fundamentallösung des Laplace-Operators,

$$\Phi(\mathbf{x}, \mathbf{y}) = -\frac{1}{2\pi} \log \|\mathbf{x} - \mathbf{y}\|,$$

mithilfe der *Greenschen Darstellungsformel* als

$$u(\mathbf{x}) = \int_{\Gamma} \Phi(\mathbf{x}, \mathbf{y}) \frac{\partial}{\partial \mathbf{n}} u(\mathbf{y}) d\sigma_{\mathbf{y}} - \int_{\Gamma} \frac{\partial}{\partial \mathbf{n}_{\mathbf{y}}} \Phi(\mathbf{x}, \mathbf{y}) u(\mathbf{y}) d\sigma_{\mathbf{y}} \quad \text{für } \mathbf{x} \in \Omega$$

darstellen. Hierzu ist die Kenntnis der vollständigen Cauchy-Daten, d.h. der Dirichlet- und Neumann-Daten, notwendig. Es lässt sich zeigen, dass diese der *Dirichlet-to-Neumann-Abbildung*

$$\mathcal{V} \frac{\partial}{\partial \mathbf{n}} u = \left(\frac{1}{2} I + \mathcal{K} \right) u|_{\Gamma} \quad (1)$$

genügen müssen. Dabei bezeichnet \mathcal{V} wie gehabt den *Einfachschicht-* und \mathcal{K} den *Doppelschichtoperator*.

Randparametrisierung

Analog zu den vorherigen Blättern beschreiben wir jede Randkomponente Γ_ℓ durch eine C^k -glatte Parametrisierung $\gamma_\ell: [0, 1] \rightarrow \Gamma_\ell$ mit $\gamma_\ell^{(m)}(0) = \gamma_\ell^{(m)}(1)$ für $m = 0, 1, \dots, k$ und $\gamma_\ell'(s) \neq \mathbf{0}$ für $s \in [0, 1]$. Durch Einsetzen der Fundamentallösung und der Parametrisierungen in (1) und Transformation aller Funktionen auf $[0, 1]$ erhalten wir für $\ell = 1, \dots, r$ die bekannte Darstellung der Integraloperatoren:

$$\sum_{m=1}^r \int_0^1 k_{\ell,m}^{\mathcal{V}}(s, t) \hat{\rho}_m^N(t) dt = \frac{1}{2} \hat{\rho}_\ell^D(s) + \sum_{m=1}^r \int_0^1 k_{\ell,m}^{\mathcal{K}}(s, t) \hat{\rho}_m^D(t) dt \quad (2)$$

mit den Integralkernen

$$k_{\ell,m}^{\mathcal{V}}(s,t) := -\frac{1}{4\pi} \log \|\gamma_\ell(s) - \gamma_m(t)\|^2 \|\gamma'_m(t)\|$$

und

$$k_{\ell,m}^{\mathcal{K}}(s,t) := \frac{\langle \gamma_\ell(s) - \gamma_m(t), \mathbf{n}_{\gamma_m(t)} \rangle}{2\pi \|\gamma_\ell(s) - \gamma_m(t)\|^2} \|\gamma'_m(t)\|$$

sowie den Dichten

$$\hat{\rho}_\ell^D(s) := u(\gamma_\ell(s)) \quad \text{und} \quad \hat{\rho}_\ell^N(s) := \frac{\partial}{\partial \mathbf{n}} u(\gamma_\ell(s)).$$

Nyström-Verfahren

Für n_m definieren wir die Quadraturknoten und Gewichte

$$t_{m,j} = \frac{j-1}{n_m} \quad \text{und} \quad \omega_{m,j} = \frac{1}{n_m} \quad \text{für } j = 1, \dots, n_m.$$

Dann ergibt sich das Nyström-Verfahren aus dem Gleichungssystem

$$\begin{bmatrix} \mathbf{V}_{1,1} & \mathbf{V}_{1,2} & \cdots & \mathbf{V}_{1,r} \\ \mathbf{V}_{2,1} & \mathbf{V}_{2,2} & \cdots & \mathbf{V}_{2,r} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{V}_{r,1} & \mathbf{V}_{r,2} & \cdots & \mathbf{V}_{r,r} \end{bmatrix} \begin{bmatrix} \rho_1^N \\ \rho_2^N \\ \vdots \\ \rho_r^N \end{bmatrix} = \begin{bmatrix} 1/2\mathbf{I} + \mathbf{K}_{1,1} & \mathbf{K}_{1,2} & \cdots & \mathbf{K}_{1,r} \\ \mathbf{K}_{2,1} & 1/2\mathbf{I} + \mathbf{K}_{2,2} & \cdots & \mathbf{K}_{2,r} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{K}_{r,1} & \mathbf{K}_{r,2} & \cdots & 1/2\mathbf{I} + \mathbf{K}_{r,r} \end{bmatrix} \begin{bmatrix} \rho_1^D \\ \rho_2^D \\ \vdots \\ \rho_r^D \end{bmatrix}. \quad (3)$$

Die Diagonalblöcke $\mathbf{K}_{\ell,\ell}$ bestimmen sich dabei, wie auf dem ersten Blatt beschrieben, und die Diagonalblöcke $\mathbf{V}_{\ell,\ell}$ werden, wie auf dem zweiten beschrieben, berechnet. Aufgrund der C^k -Glätte des Randes können sich die Randkomponenten Γ_ℓ nicht berühren, weswegen die Integralkerne für alle anderen Blöcke glatt sind und man also direkt

$$\mathbf{K}_{l,m} = [k_{\ell,m}^{\mathcal{K}}(t_{l,i}, t_{m,j}) \omega_{m,j}]_{i,j=1}^{n_\ell, n_m} \quad \text{und} \quad \mathbf{V}_{l,m} = [k_{\ell,m}^{\mathcal{V}}(t_{l,i}, t_{m,j}) \omega_{m,j}]_{i,j=1}^{n_\ell, n_m}$$

auswerten kann.

Für $\ell \in I$ ist

$$\rho_\ell^D = [g(\gamma_\ell(t_{l,i}))]_{i=1}^{n_\ell}$$

gegeben und ρ_ℓ^N die Unbekannte; dagegen ist für $\ell \in J$

$$\rho_\ell^N = [h(\gamma_\ell(t_{l,i}))]_{i=1}^{n_\ell}$$

gegeben und ρ_ℓ^D die Unbekannte.

Potentialauswertung

Nachdem nun eine Approximation der Cauchy-Daten bekannt ist, kann der Wert von $u(\mathbf{x})$ für ein $\mathbf{x} \notin \Gamma$ approximativ mittels der Quadraturformel als

$$u_n(\mathbf{x}) := - \sum_{m=1}^r \sum_{j=1}^{n_m} \omega_{m,j} \frac{1}{2\pi} \log \|\mathbf{x} - \gamma_m(t_{m,j})\| \|\gamma'_m(t_{m,j})\| \rho_{m,j}^N$$

$$- \sum_{m=1}^r \sum_{j=1}^{n_m} \omega_{m,j} \frac{\langle \mathbf{x} - \gamma_m(t_{m,j}), \mathbf{n}_{\gamma_m(t_{m,j})} \rangle}{2\pi \|\mathbf{x} - \gamma_m(t_{m,j})\|^2} \|\gamma'_m(t_{m,j})\| \rho_{m,j}^D \quad (4)$$

berechnet werden.

Aufgabe 1.

Die Parametrisierungen γ_ℓ der Randkomponenten Γ_ℓ speichern wir als ein **structure array** `mbp` ab. Dieses soll dabei die Felder `gamma`, `dgamma`, `d2gamma` und `ngamma` besitzen, welche **function handles** zu Funktionen sind, die wie folgt vektorwertig auswertbar sind:

$$\begin{aligned}\text{mbp}(\ell).\text{gamma}([s_1 \ \cdots \ s_n]) &= [\gamma_\ell(s_1) \ \cdots \ \gamma_\ell(s_n)] \\ \text{mbp}(\ell).\text{dgamma}([s_1 \ \cdots \ s_n]) &= [\gamma'_\ell(s_1) \ \cdots \ \gamma'_\ell(s_n)] \\ \text{mbp}(\ell).\text{d2gamma}([s_1 \ \cdots \ s_n]) &= [\gamma''_\ell(s_1) \ \cdots \ \gamma''_\ell(s_n)] \\ \text{mbp}(\ell).\text{ngamma}([s_1 \ \cdots \ s_n]) &= [\mathbf{n}_{\gamma_\ell(s_1)} \ \cdots \ \mathbf{n}_{\gamma_\ell(s_n)}].\end{aligned}$$

Implementieren Sie eine Funktion

```
function mbp = add_star_shaped_parametrisation(mbp, xc, no, r, dr, d2r),
```

welches dem **structure array** `mbp` die sternförmigen Parametrisierung

$$s \mapsto \mathbf{xc} + r(s) \begin{bmatrix} \cos(2\pi s) \\ \sin(2\pi s) \end{bmatrix}$$

anhängt. Hierbei bezeichnet `xc` den Aufpunkt und `no` ist die Orientierung der äusseren Normalen; für `no = 1` soll die äussere Normale vom Aufpunkt wegzeigen, für `no = -1` soll sie in Richtung des Aufpunkts zeigen.

In der Beilage finden Sie eine Funktion, welche mit Hilfe dieser Funktion verschiedene Gebiete als Randkomponentenbeschreibung per **structure array** `mbp` aufstellt. Ebenfalls vorhanden sind Funktionen, welche die so beschriebenen Gebiete dann meshen und weiter plotten können. Vergleichen Sie dazu ebenfalls den ersten Abschnitt des Skriptes `testmain`.

Aufgabe 2.

In der Beilage findet sich die Funktion

```
function dcd = interpolate_cauchy_data(mbp, ns, u, dnu),
```

welche die Randkomponentenbeschreibung per **structure array** `mbp`, einen Zeilenvektor `ns` der Anzahl an Randpunkten, die für jeden jeweiligen Rand benutzt werden sollen, und zwei **function handles** auf Funktionen, welche eine harmonische Funktion und ihre Normalableitung berechnen, nimmt. Der Rückgabewert der Funktion, ist dann ein **structure array**, welches die vollständigen, aber diskretisierten Cauchy-Daten für jede Randkomponente sowie die diskretisierte Beschreibung der Ränder beinhaltet.

Schreiben Sie dazu passend eine Funktion

```
function us = potential_cauchy(dcd, xs),
```

die mit Hilfe dieser diskreten Daten `dcd` per Potentialauswertung (4) den Wert der harmonischen Funktion approximativ bei den in den Spalten von `xs` stehenden Punkten rekonstruiert und als Zeilenvektor `us` zurückgibt.

Überprüfen Sie Ihre Potentialauswertung, indem Sie den zweiten Abschnitt des Skriptes `testmain` betrachten und daraus ein eigenes Skript schreiben, welches die Fehlerrate für wachsende Anzahlen an Randpunkten berechnet und darstellt.

Aufgabe 3.

Schreiben Sie eine Funktion

```
function dcd = nystroem_cauchy(mbp, ns, g, h, btf),
```

welche das Gleichungssystem (3) aufstellt und für die unbekanntenen Dirchlet- und Neumann-Daten löst.

Die Funktion nimmt dafür die Randkomponentenbeschreibung per **structure array** **mbp**, einen Zeilenvektor **ns** der Anzahl an Randpunkten, die für jeden jeweiligen Rand benutzt werden sollen, zwei function handles auf Funktionen, welche die Funktionen g für die Dirichletränder respektive die Funktion h für die Neumannränder berechnen, sowie einen **bool**-Zeilenvektor **btf**, als Eingabevariablen. Dabei ist **btf** an der ℓ -ten Stelle **true**, wenn die ℓ -te Randkomponente mit Dirichlet-Daten versehen wird; ansonsten sollen bei der ℓ -ten Randkomponente Neumann-Daten sein.

Der Rückgabewert der Funktion, soll dann ein **structure array** **dcd** sein, mit den gleichen Feldern, wie die der Funktion **interpolate_cauchy_data** berechnet.

Hinweis. Erstellen Sie in der Funktion zuerst **dcd** und setzen Sie darin alle schon bekannten Felder. Dann können Sie die Systemmatrizen blockweise berechnen, indem Sie separate "local functions" benutzen, um die vorkommenden Teilmatrizen zu berechnen. Es reicht dann diesen "local functions" die Variablen ℓ und m sowie das **dcd** zu übergeben.

Aufgabe 4.

Schreiben Sie ein Skript, welches die Fehlerrate des gesamten Verfahrens für wachsende Anzahlen an Randpunkten berechnet und darstellt. Wählen Sie dazu ein Gebiet Ihrer Wahl. Setzen Sie g und h so, dass die analytische Lösung $u(\mathbf{x}) = x_1^2 - x_2^2$ lautet.

Aufgabe 5 (freiwillig).

Schreiben Sie ein Skript, welches auf einem Gebiet Ihrer Wahl rechnet. Setzen Sie g und h so, dass keine analytische Lösung bestimmt werden kann.

