

14 Kuidas sebra triibuliseks saada?

Üheks bioloogia tähtsaks küsimuseks on, kuidas organismi rakud arenevad ja paljunevad, et lõpptulemusena tekivad organismis teatavad mustrid. Näiteks, kuidas sisaliku naharakud arenevad, et pealtnäha samadest rakkudest tekivad triibud, ringid ... Matemaatiliselt kirjeldas sarnast protsessi Alan Turing, kasutades reaktsiooni-difusiooni võrrandite süsteemi. Viimane kirjeldab kahe teatava keemilise aine reaktsiooni ja levikut organismi rakkudes, kus pärast stabiilsesse olekusse jõudmist tekivad jäädavad mustrid.

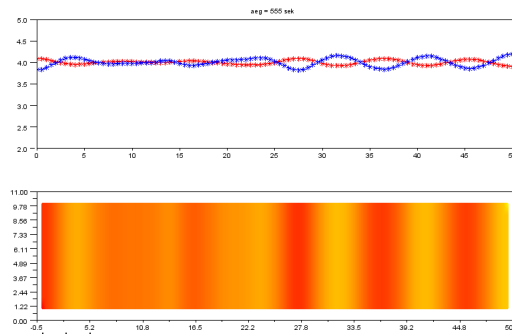


Foto: Wikipedia

Keerulisemal juhul peaksime vaatlema kolmemõõtmelist mudelit, kuid see on väga ajaminu- kas ettevõtmine. Lihtsamal juhul saame uurida ühemõõtmelist süsteemi, mille lahendid ei tule küll kahjuks nii põnevad, kuid iseloomustavad reaktsiooni-difusiooni võrrandite käitumist piisavalt. Märgime, et lahendusmeetodid jäävad mitmemõõtmelisel juhul väga sarnasteks, küll aga arvutusmaht kasvab olulisel määral.

14.1 Modelleerimine 1-mõõtmeliste reaktsiooni-difusioonivõrrandite süsteemiga

Vaatleme ühe ruumikoordinaadiga reaktsiooni-difusiooni võrrandite süsteemi (vt. [2]):

$$\left\{ \begin{array}{l} \frac{\partial u(x,t)}{\partial t} = D_u \cdot \frac{\partial^2 u(x,t)}{\partial x^2} + s \cdot \left(16 - u(x,t) \cdot w(x,t) \right) \\ \frac{\partial w(x,t)}{\partial t} = D_w \cdot \frac{\partial^2 w(x,t)}{\partial x^2} + s \cdot \left(u(x,t) \cdot w(x,t) - w(x,t) - \beta(x) \right) \end{array} \right. \quad (14.1)$$

- Ruumi- ja ajakoordinaat

$$x \in [0, b], \quad t \in [0, T].$$

- $u(x, t)$ ja $w(x, t)$ on kahe erineva keemilise aine kontsentratsioonid organismis hetkel t sekundit punktis x (ehk siis mingi riba peal).

- D_u ja D_w on vastavate ainete difusiooni konstandid, mis iseloomustavad nende ainete laiali hajumise kiirust.
- s on reaktsiooni kiiruse parameeter.
- $\beta(x)$ näitab rakkudes olevaid juhuslikke erinevusi hetkel $t = 0$.
- Meil on algtingimused

$$u(x, 0) = 4, \quad w(x, 0) = 4,$$

ja Neumann'i rajatingimused

$$u_t(0, t) = 0, \quad u_t(b, t) = 0, \quad w_t(0, t) = 0, \quad w_t(b, t) = 0.$$

Viited

- [1] T. Sauer. Numerical Analysis. 2nd ed. Pearson, 2012.
- [2] G. Turk. Generating Textures on Arbitrary Surfaces Using Reaction-Diffusion. Computer Graphics, Vol. 25, No. 4, pp. 289-298, 1991.

14.2 1-mõõtmelise difusiooni võrrandite süsteemi lahendamine võrgumeetodiga

1. Valime lõigul $[0, b]$ osalõikude arvu N_x , sammude pikkuse $h_x = \frac{b}{N_x}$ ja moodustame sõlmed

$$x_i = b \cdot \frac{i}{N_x}, \quad i = 0, \dots, N_x.$$

2. Valime ajalõigul $[0, T]$ osalõikude arvu N_t , sammude pikkuse $h_t = \frac{T}{N_t}$ ja moodustame sõlmed

$$t_j = T \cdot \frac{j}{N_t}, \quad j = 0, \dots, N_t.$$

3. Otsime lahendite $u(x, t)$ ja $w(x, t)$ lähisväärtusi u_{ij} ja w_{ij} sõlmedes x_0, \dots, x_{N_x} ja t_0, \dots, t_{N_t} . Selleks moodustame $(N_x + 1) \times (N_t + 1)$ -mõõtmelised maatriksid

$$u = \left(\begin{array}{c|ccc} \mathbf{u}_{00} & u_{01} & \cdots & u_{0,N_t} \\ \mathbf{u}_{10} & u_{11} & \cdots & u_{1,N_t} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_{N_x-1,0} & u_{N_x-1,1} & \cdots & u_{N_x-1,N_t} \\ \mathbf{u}_{N_x,0} & u_{N_x,1} & \cdots & u_{N_x,N_t} \end{array} \right), \quad w = \left(\begin{array}{c|ccc} \mathbf{w}_{00} & w_{01} & \cdots & w_{0,N_t} \\ \mathbf{w}_{10} & w_{11} & \cdots & w_{1,N_t} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{w}_{N_x-1,0} & w_{N_x-1,1} & \cdots & w_{N_x-1,N_t} \\ \mathbf{w}_{N_x,0} & w_{N_x,1} & \cdots & w_{N_x,N_t} \end{array} \right).$$

4. Edaspidi kasutame lühiduse mõttes u_j ja w_j vastavate maatriksite j -nda veeru tähistamiseks. Funktsiooni argumentides olevald u ja w tähistavad veeruvektoreid.

5. Ülesande lahendamise Newton'i meetodiga vektorkujul,

$$x_{j+1} = x_j - J^{-1} \cdot F(x_j),$$

mille kirjutame arvutuslikult ökonoomsemalt (lineaarvõrrandite süsteemi kaudu):

$$\left\{ \begin{array}{l} J(u_{j_k}, w_{j_k}) \cdot v = -F(u_j, w_j) \\ \begin{pmatrix} u_{j_{k+1}} \\ w_{j_{k+1}} \end{pmatrix} = \begin{pmatrix} u_{j_k} \\ w_{j_k} \end{pmatrix} + v \end{array} \right\}, \quad j = 0, \dots, N_t - 1, \quad k = 0, 1, \dots$$

Siin

$$u_{j_0} = u_j, \quad w_{j_0} = w_j$$

ja

$$u_{j+1} := u_{j_k}, \quad w_{j+1} := w_{j_k} \quad \text{mingi } k \text{ korral.}$$

Märgime, et SciLab'is tuleks omistamised u ja w osa jaoks teha eraldi. Järgnevalt selgitame täpsemalt, millised on Jacob'i maatriks J ja vabaliige F .

6. Tähistame

$$\sigma_u = D_u \cdot \frac{h_t}{h_x^2}, \quad \sigma_w = D_w \cdot \frac{h_t}{h_x^2}.$$

Jacob'i maatriksi veidi ülevaatlikumaks konstrueerimiseks toome sisse uued maatrikid. Olgu

$$J_{1,u} = \begin{pmatrix} -3 & 4 & -1 & 0 & \dots & 0 & 0 & 0 & 0 \\ -\sigma_u & 1 + 2 \cdot \sigma_u & -\sigma_u & 0 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 & -\sigma_u & 1 + 2 \cdot \sigma_u & -\sigma_u \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & -4 & 3 \end{pmatrix},$$

$$J_{1,w} = \begin{pmatrix} -3 & 4 & -1 & 0 & \dots & 0 & 0 & 0 & 0 \\ -\sigma_w & 1 + 2 \cdot \sigma_w + h_t \cdot s & -\sigma_w & 0 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 & -\sigma_w & 1 + 2 \cdot \sigma_w + h_t \cdot s & -\sigma_w \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & -4 & 3 \end{pmatrix},$$

Sel juhul $(N_x + 1) \times (N_x + 1)$ -mõõtmelise nullmaatriksi $\mathbf{0}$ korral

$$J_1 := \begin{pmatrix} J_{1,u} & \mathbf{0} \\ \mathbf{0} & J_{1,w} \end{pmatrix}.$$

Analoogiliselt

$$J_2(u, w) = h_t \cdot s \cdot \begin{pmatrix} d_0(w) & d_0(u) \\ -d_0(w) & -d_0(u) \end{pmatrix},$$

kus $d_0(x)$ tähistab diagonaalmaatriksit, mille diagonaalil on vektori x elemendid, kusjuures esimesel ja viimasel real on kõik nullid. Jacob'i maatriks avaldub siis summana

$$J(u, w) = J_1 + J_2(u, w).$$

7. Defineerime vektori

$$F_0(u, w) = \begin{pmatrix} \tilde{u} \\ \tilde{w} \end{pmatrix}, \quad \tilde{u} = \begin{pmatrix} 0 \\ -h_t \cdot s \cdot 16 - u_1 \\ \vdots \\ -h_t \cdot s \cdot 16 - u_{N_x} \\ 0 \end{pmatrix}, \quad \tilde{w} = \begin{pmatrix} 0 \\ h_t \cdot s \cdot \beta_1 - w_1 \\ \vdots \\ h_t \cdot s \cdot \beta_{N_x} - w_{N_x} \\ 0 \end{pmatrix}.$$

Sel juhul Newton'i meetodi vabaliige F avaldub eelneva kaudu järgnevalt:

$$F(u_{j_k}, w_{j_k}) = F_0(u_j, w_j) + \left(J_1 + \frac{1}{2} \cdot J_2(u_{j_k}, w_{j_k}) \right) \cdot \begin{pmatrix} u_{j_k} \\ w_{j_k} \end{pmatrix},$$

kus u_j ja w_j on eelmise veeru väärtused (ehk Newton'i iteratsiooni algväärtused).

14.3 Praktikumi ülesanne

1. Olgu löigu pikkus $b = 50$ ja aeg $T = 2000$ sekundit (nii suureks võtke see alles lõpus, kui programm töötab veatult).
2. Rakkude häiritused β_i võtame väärtuse 12 lähedalt juhuslike arvude generaatori abil.

```
beeta = 12 + (rand(x) - 0.5) / 10 // x on ruumisõlmede vektor
```

3. Lisaks,

$$D_u = 0.25, \quad D_w = 0.0625, \quad s = 0.03125.$$

4. Kasutades olemasolevat SciLab'i tööfaili, realiseerida Newton'i iteratsioonimeetod lähislahendi u ja w leidmiseks. Märkime, et fail sisaldab vajalikke protseduure Jacob'i maatriksi ja vabaliikme leidmiseks.
5. Kasutades failis olevat graafiku protseduuri, väljastada vahetulemuste graafikud (graafiku protseduuris j tähistab maatriksi veeru numbrit).

```
if modulo( j , 51 ) == 0 then
    graafik( uu , ww , j )
end
```

6. Mida võib öelda lahendi käitumise kohta aja kulgedes? Kas lahend tuleb alati sama?
7. Kuidas mõjub häirituste β_i suuremaks või väiksemaks muutmine ning kui arvutusvead peaksid võimenduma, siis kas on abi ruumi- või ajasammu lühendamisest või siis Newton'i meetodi iteratsioonide arvu suurendamisest (või hoopis vähendamisest)?