Metody numeryczne II

Równania różniczkowe zwyczajne, część II

Janusz Szwabiński

szwabin@ift.uni.wroc.pl

Równania różniczkowe zwyczajne, część II

1. Zagadnienia początkowe (ciąg dalszy)
   - Metody niejawne
   - Algorytm Verleta
2. Zagadnienia brzegowe i własne
   - Prosta metoda strzałów
   - Równania liniowe
   - Jednowymiarowe równanie Schrödingera

Metody niejawne

$$y = (y_1, \ldots, y_n), \quad g(y,t) = [g_1(y,t), \ldots, g_n(y,t)]$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = g(y,t), \quad y(t_0) = y_0$$

Formalne rozwiązanie

$$y_{n+m} = y_n + \int_{t_n}^{t_{n+m}} g(y,t)\mathrm{d}t$$

$m = 1$ i metoda trapezów:

$$y_{n+1} \simeq y_n + \frac{\tau}{2}[g(y_n,t_n) + g(y_{n+1},t_{n+1})]$$

$\Rightarrow$ równanie względem y_{n+1}

$$0 = y_n - y_{n+1} + +\frac{\tau}{2}[g(y_n,t_n) + g(y_{n+1},t_{n+1})]$$

Metoda parabol

$\Rightarrow$ wzór Adamsa–Moultona

$$y_{n+1} = y_n + \frac{\tau}{15}[5g(y_{n+1},t_{n+1}) + 8g(y_n,t_n) - g(y_{n-1},t_{n-1})] + O(\tau^4)$$

Własności

- duży nakład obliczeń
- stabilność

Jeśli $g(y,t)$ jest liniową funkcją y

$$g(y,t) = \tilde{g}(t)y$$

$\Rightarrow$ równanie można rozwiązać analitycznie

$$y_{n+1} = \left[\frac{1+\frac{1}{2}\tilde{g}(t_n)\tau}{1-\frac{1}{2}\tilde{g}(t_{n+1})\tau}\right] y_n$$

Algorytm Verleta

$$\ddot{\vec{r}}_i = \frac{1}{m_i}\vec{F}_i$$

$\vec{r}_i$	-	wektor położenia i–tej cząstki
m_i	-	masa i–tej cząstki
$\vec{F}_i$	-	siła działająca na i–tą cząstkę

Jeśli liczba cząstek jest duża

$\Rightarrow$ układ wielu równań różniczkowych

$\Rightarrow$ wyliczenie siły może być skomplikowane

$\Rightarrow$ algorytm Rungego–Kutty raczej się nie nadaje ze względu na wydajność

Rozwijamy funkcje $\vec{r}_i(t+\Delta t)$ i $\vec{r}_i(t-\Delta t)$ w szereg Taylora wokół punktu t:

$$\begin{aligned}
\vec{r}_i(t+\Delta t) &= \vec{r}_i(t) + \Delta t\vec{v}_i(t) + \frac{1}{2}\Delta t^2\vec{a}_i(t) + \ldots \\
\vec{r}_i(t-\Delta t) &= \vec{r}_i(t) - \Delta t\vec{v}_i(t) + \frac{1}{2}\Delta t^2\vec{a}_i(t) + \ldots
\end{aligned}$$

Dodając stronami, otrzymamy

$$\vec{r}_i(t+\Delta t) = 2\vec{r}_i(t) - \vec{r}_i(t-\Delta t) + \Delta t^2\frac{\vec{F}_i(t)}{m_i} + O(\Delta t^4)$$

Odejmowanie stronami prowadzi do

$$\vec{v}_i(t) = \frac{\vec{r}_i(t+\Delta t) - \vec{r}_i(t-\Delta t)}{2\Delta t} + O(\Delta t^3)$$

Zalety

- jedno wyliczenie siły w każdym kroku czasowym
- duża dokładność
- stabilność

Wady

- algorytm nie startuje z jednego punktu
- prędkość nie jest generowana automatycznie, trzeba ją dodatkowo wyliczać
- prędkość obarczona jest zwykle większym błędem, niż położenie

$\Rightarrow$ do wyliczenia prędkości użyj zmodyfikowanych wersji tego algorytmu

Zagadnienia brzegowe

Typowe fizyczne zagadnienie brzegowe ma postać równania różniczkowego drugiego rzędu

$$u''(x) = f(u, u', x)$$

z zadanymi na brzegach wartościami u lub u'

W 1D mamy cztery rodzaje warunków brzegowych:

1. $u(0) = u_0$ i $u(1) = u_1$
2. $u(0) = u_0$ i $u'(1) = v_1$
3. $u'(0) = v_0$ i $u(1) = u_1$
4. $u'(0) = v_0$ i $u'(1) = v_1$

Zagadnienia własne

Jeśli

- równanie różniczkowe zawiera parametr
- rozwiązanie równania spełnia zagadnienie brzegowe tylko dla pewnych wartości parametru
- musimy ten parametr wyznaczyć

$\Rightarrow$ zagadnienie własne

Prosta metoda strzałów

Rozważmy zagadnienie brzegowe

$$u'' = f(u, u', x), \quad u(0) = \alpha, \quad u(1) = \beta$$

Zagadnienie początkowe

$$u'' = f(u, u', x), \quad u(0) = \alpha, \quad u'(0) = s$$

ma wówczas na ogół dokładnie jedno rozwiązanie

$$u(x) \equiv u(x, s)$$

zależące od wyboru wartości początkowej s dla $u'(0)$

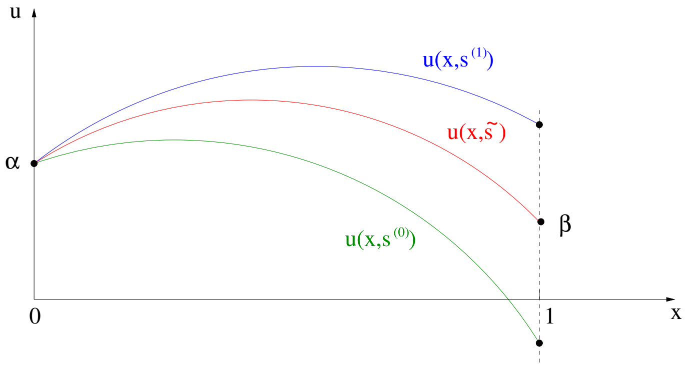
Szukamy takiego $s = \tilde{s}$, aby

$$u(0) = u(1, \tilde{s}) = \beta$$

Innymi słowy, szukamy miejsce zerowego $\tilde{s}$ funkcji

$$F(s) \equiv u(1, s) - \beta$$

Metoda bisekcji



Jeśli $F(s)$ jest różniczkowalne względem s

$\Rightarrow$ metoda Newtona

$$s^{(i+1)} = s^{(i)} - \frac{F(s^{(i)})}{F'(s^{(i)})}$$

Możemy zastąpić $F'(s)$ odpowiednim ilorazem różnicowym

$$\Delta F(s^{(i)}) = \frac{F(s^{(i)} + \Delta s^{(i)}) - F(s^{(i)})}{\Delta s^{(i)}}$$

Trudności:

- zbyt duże $\Delta s^{(i)} \Rightarrow$ przybliżenie niezbyt dokładne
- zbyt małe $\Delta s^{(i)} \Rightarrow$ znoszenie się składników

$\Rightarrow$ $\Delta s^{(i)} = \sqrt{\epsilon} s^{(i)}$

Równania liniowe

$$u''(x) + d(x)u'(x) + q(x)u(x) = s(x)$$

$$u(0) = u_0, \quad u(1) = u_1$$

Funkcje $d(x)$, $q(x)$ i $s(x)$ są gładkie

$\Rightarrow$ zasada superpozycji rozwiązań

$\Rightarrow$ nie musimy szukać dokładnie $\tilde{s}$ w metodzie strzałów

$u(x, s_1), \;\; u(x, s_2)$ - rozwiązania próbne

Szukamy rozwiązania postaci

$$u(x) = au(x, s_1) + bu(x, s_2)$$

Z warunków brzegowych

$$\begin{aligned} a + b &= 1 \\ u(1, s_1)a + u(1, s_2)b &= u_1 \end{aligned}$$

czyli

$$a = \frac{u(1, s_2) - u_1}{u(1, s_2) - u(1, s_1)}, \quad b = \frac{u_1 - u(1, s_1)}{u(1, s_2) - u(1, s_1)}$$

Zagadnienia Sturma–Liouville'a

$$[p(x)u'(x)]' + q(x)u(x) = s(x)$$

Przykłady

- równanie Legendre'a
- równanie Bessela

Ze wzorów trójpunktowych mamy:

$$\Delta_1 \equiv \frac{u_{n+1} - u_{n-1}}{2h} = u'_n + \frac{h^2}{6}u_n^{(3)} + O(h^4)$$
$$\Delta_2 \equiv \frac{u_{n+1} + u_{n-1} - 2u_n}{h^2} = u''_n + \frac{h^2}{12}u_n^{(4)} + O(h^4)$$

Stąd

$$p'_n\Delta_1 + p_n\Delta_2 = [p_n u'_n]' + \frac{h^2}{12}(p_n u_n^{(4)} + 2p'_n u_n^{(3)}) + O(h^4)$$

Zaniedbując drugi wyraz po prawej stronie otrzymamy

$$(2p_n + hp'_n)u_{n+1} + (2p_n - hp'_n)u_{n-1} = 4p_n u_n + 2h^2(s_n - q_n u_n)$$

Przykład Równanie Legendre'a

$$\frac{\mathrm{d}}{\mathrm{d}x}\left[(1-x^2)\frac{\mathrm{d}u}{\mathrm{d}x}\right] + l(l+1)u = 0, \quad x \in \langle -1, 1\rangle$$

Załóżmy, że

- nie znamy wartości l
- znamy za to wartości wielomianu $P_1(x) = x$ dla dwóch pierwszych punktów

Szukamy l odpowiadającego zadanym warunkom brzegowym

$\Rightarrow$ zagadnienie własne

Podprogram do całkowania równania Legendre'a najprostszym algorytmem Sturma–Liouville

```
SUBROUTINE SLL (N,H,L,U)
  IMPLICIT NONE
  INTEGER, INTENT (IN) :: N
  INTEGER :: I
  DOUBLE PRECISION, INTENT (IN) :: H,L
  DOUBLE PRECISION :: H2,Q,X,P,P1
  DOUBLE PRECISION, INTENT (OUT), DIMENSION (N) :: U
!
  H2 = 2.0*H*H
  Q = L*(1.0+L)
  DO I = 2, N-1
    X  =  (I-1)*H-1.0
    P  =  2.0*(1.0-X*X)
    P1 = -2.0*X*H
    U(I+1) = ((2.0*P-H2*Q)*U(I)+(P1-P)*U(I-1))/(P1+P)
  END DO
END SUBROUTINE SLL
```

Zróżniczkujmy równanie Sturma–Liouville'a dwukrotnie

$$pu^{(4)} + 2p'u^{(3)} = s'' - 3p''u'' - p^{(3)}u' - p'u^{(3)} - q''u - 2q'u' - qu''$$

Jednokrotne różniczkowanie daje

$$u^{(3)} = \frac{1}{p}(s' - p''u' - 2p'u'' - q'u - qu')$$

Stąd

$$c_{n+1}u_{n+1} + c_{n-1}u_{n-1} = c_n u_n + d_n + O(h^6)$$

gdzie

$$\begin{aligned}
c_{n+1} &= 24p_n + 12hp'_n + 2h^2q_n + 6h^2p''_n - 4h^2\frac{(p'_n)^2}{p_n} + h^3p_n^{(3)} \\
&\quad +2h^3q'_n - h^3q_n\frac{p'_n}{p_n} - h^3p''_n\frac{p'_n}{p_n} \\
c_n &= 48p_n - 20h^2q_n - 8h^2\frac{(p'_n)^2}{p_n} + 12h^2p''_n + 2h^4q'_n\frac{p'_n}{p_n} - 2h^4q''_n \\
c_{n-1} &= 24p_n - 12hp'_n + 2h^2q_n + 6h^2p''_n - 4h^2\frac{(p'_n)^2}{p_n} - h^3p_n^{(3)} \\
&\quad -2h^3q'_n + h^3q_n\frac{p'_n}{p_n} + h^3p''_n\frac{p'_n}{p_n} \\
d_n &= 24h^2s_n + 2h^4s''_n - 2h^4s'_n\frac{p'_n}{p_n}
\end{aligned}$$

Algorytm Numerowa

$$u'' + q(x)u = s(x)$$

Ze wzoru trójpunktowego mamy

$$\frac{u_{n+1} - 2u_n + u_{n-1}}{h^2} = u''_n + \frac{h^2}{12}u_n^{(4)} + O(h^4)$$

Z drugiej strony

$$\begin{aligned}
u_n^{(4)} &= \frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(s(x) - q(x)u(x)\right)\Big|_{x=x_n} \\
&= \frac{s_{n+1} - 2s_n + s_{n-1}}{h^2} - \frac{q_{n+1}u_{n+1} - 2q_nu_n + q_{n-1}u_{n-1}}{h^2} + O(h^2)
\end{aligned}$$

Stąd

$$c_{n+1}u_{n+1} + c_{n-1}u_{n-1} = c_n u_n + d_n + O(h^6)$$

gdzie

$$\begin{aligned}
c_{n+1} &= 1 + \frac{h^2}{12}q_{n+1} \\
c_n &= 2 - \frac{5}{6}h^2q_n \\
c_{n-1} &= 1 + \frac{h^2}{12}q_{n-1} \\
d_n &= \frac{h^2}{12}(s_{n+1} + 10s_n + s_{n-1})
\end{aligned}$$

```
SUBROUTINE NUMEROV (N,H,Q,S,U)
  IMPLICIT NONE
  INTEGER, INTENT (IN) :: N
  INTEGER :: I
  DOUBLE PRECISION,INTENT (IN) :: H
  DOUBLE PRECISION :: G,C0,C1,C2,D,UTMP
  DOUBLE PRECISION, INTENT (IN), DIMENSION (N)    :: Q,S
  DOUBLE PRECISION, INTENT (INOUT), DIMENSION (N) :: U
!
  G = H*H/12.0
!
  DO I = 2, N-1
    C0 = 1.0+G*Q(I-1)
    C1 = 2.0-10.0*G*Q(I)
    C2 = 1.0+G*Q(I+1)
    D  = G*(S(I+1)+S(I-1)+10.0*S(I))
    UTMP   = C1*U(I)-C0*U(I-1)+D
    U(I+1) = UTMP/C2
  END DO
END SUBROUTINE NUMEROV
```


Jednowymiarowe równanie Schrödingera

$$-\frac{1}{2}\frac{\mathrm{d}^2\psi(x)}{\mathrm{d}x^2} + V(x)\psi(x) = E\psi(x)$$

$V(x)$ - pewien (znany) potencjał

E - szukana wartość własna

$\psi(x)$ - szukana funkcja własna

Zachodzi

$$\psi''(x) + 2[E - V(x)]\psi(x) = 0$$

$\Rightarrow$ do całkowania równanie Schrödingera możemy użyć algorytmu Numerowa

Szukamy stanów związanych

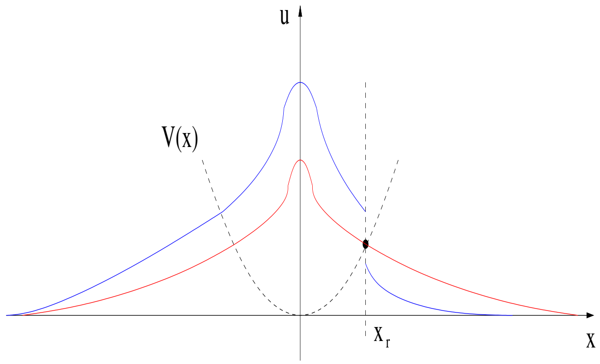
$\Rightarrow$ funkcja falowa powinna spełniać warunek

$$\lim_{|x|\to\infty} \psi(x) = 0$$

Wady prostej metody strzałów:

- całkowanie w kierunku wykładniczego spadku funkcji falowej może prowadzić do wzrostu błędu obliczeń

Modyfikacja metody strzałów



Warunek ciągłości dla obu rozwiązań

$$\begin{aligned} \psi_l(x_r) &= \psi_r(x_r) \\ \psi_l'(x_r) &= \psi_r'(x_r) \end{aligned}$$

czyli

$$\frac{\psi_l'(x_r)}{\psi_l(x_r)} = \frac{\psi_r'(x_r)}{\psi_r(x_r)}$$

Korzystając ze wzoru trójpunktowego otrzymamy

$$\begin{aligned} F(E) &= \frac{[\psi_l(x_r+h) - \psi_l(x_r-h)] - [\psi_r(x_r+h) - \psi_r(x_r-h)]}{2h\psi_r(x_r)} \\ &= 0 \end{aligned}$$

Zarys postępowania:

(i) wybieramy obszar całkowania tak, aby wpływ wyboru granic był zaniedbywalny

(ii) podajemy rozsądną wartość startową dla wartości własnej E

(iii) wyznaczamy funkcję $\psi_l(x)$, całkując równanie od lewej strony do punktu $x_r + h$, oraz funkcję $\psi_r(x)$, całkując od prawej strony do punktu $x_r - h$. Wybieramy zero jako pierwszą wartość funkcji $\psi_l(x)$ i $\psi_r(x)$ (warunki brzegowe), oraz pewną niewielką liczbę jako wartość tych funkcji dla drugiego punktu całkowania

(iv) aby dopasować do siebie rozwiązania, jedno należy przeskalować, np.:

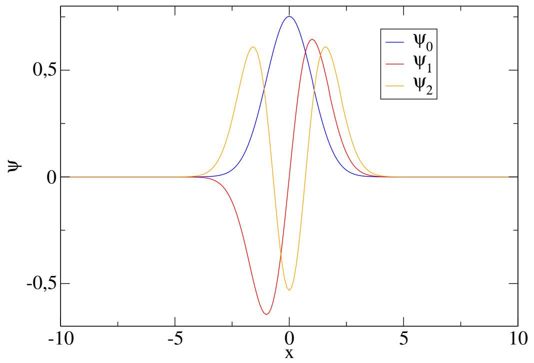
$$\psi_l(x) \to \psi_l(x)\frac{\psi_r(x_r)}{\psi_l(x_r)}$$

(v) wyznaczamy $F(E)$

(vi) stosujemy jedną z metod poszukiwania pierwiastka, aby znaleźć wartość własną z równania $F(E) = 0$ z wymaganą dokładnością

Przykład Wartości i funkcje własne oscylatora harmonicznego ($m = \hbar = \omega = 1$)

E_0	E_1	E_2	E_3	E_4
0,5000000	1,5000000	2,5000000	3,5000000	4,4999999



ψ
0,5
0
-0,5
-10
-5
0
5
10
x
ψ0
ψ1
ψ2