Równania Różniczkowe Cząstkowe: Fale

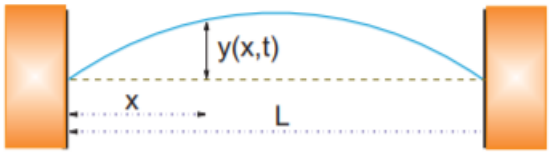
Agnieszka Mach, Robert Wojtaszek

September 2020

Spis treści

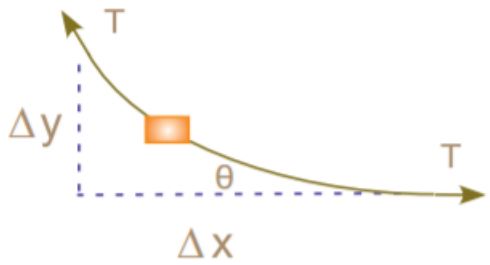
1 Równanie fal hiperbolicznych

Naszym rozważaniom poddajemy sznurek o długości L zawiązany na obu końcach (Rysunek 1.). Ciąg ma stałą gęstość ρ na jednostkę długości oraz stałe napięcie T (przy czym żadne siły tarcia nie działają ani na niego, ani na napięcie, które jest tak duże, że możemy zignorować ugięcie spowodowane grawitacją). Zakładamy, że przemieszczenie struny *y(x, t)* z pozycji spoczynkowej następuje jedynie w kierunku pionowym i jest funkcją poziomą położenia wzdłuż ciągu x i czasu t.



Rysunek 1: Rozciągnięty sznurek o długości L zawiązany na obu końcach. Pionowe zaburzenie struny od jej położenia równowagi wynosi *y(x, t)*.

Aby otrzymać proste liniowe równanie ruchu zakładamy, że przemieszczenie y(x, t) / L oraz nachylenie ∂ y / ∂ x struny jest małe. Wyodrębniamy nieskończenie mały przekrój Δ x (Rysunek 2.) i zauważamy, że różnica w pionowych składowych naprężeń na każdym końcu struny wytwarza siłę przywracającą, która przyspiesza ten odcinek sznurka w kierunku pionowym.



Rysunek 2: Element różnicowy struny pokazujący, w jaki sposób przemieszczenie struny prowadzi do siły przywracającej.

Dzięki zastosowaniu prawa Newtona otrzymujemy równanie falowem, w którym istnieniejają dwie niezależne zmienne (x oraz t), które sprawiają, że używamy tutaj terminu PDE (partial differential equation, i.e. równania różniczkowe cząstkowe):

$$\sum F_y = \rho\,\Delta x\,\frac{\partial^2 y}{\partial t^2},$$

$$= T\sin\theta(x+\Delta x) - T\sin\theta(x)$$

$$= T\left.\frac{\partial y}{\partial x}\right|_{x+\Delta x} - T\left.\frac{\partial y}{\partial x}\right|_{x} \simeq T\frac{\partial^2 y}{\partial x^2},$$

$$\Rightarrow \quad \frac{\partial^2 y(x,t)}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 y(x,t)}{\partial t^2}, \quad c = \sqrt{\frac{T}{\rho}},$$

2 Rozwiązanie poprzez rozszerzenie w trybie normalnym

Zakładamy, że rozwiązanie jest iloczynem funkcji przestrzeni i czasu:

$$y(x,t) = X(x)T(t).$$

Powyższą zależność podstawiamy do równania falowego, a następnie dzielimy przez y(x, t) i otrzymujemy równanie, które ma rozwiązanie tylko wtedy, gdy istnieją rozwiązania dla dwóch układów równań różniczkowych zwyczajnych:

$$\frac{d^2T(t)}{dt^2} + \omega^2 T(t) = 0, \quad \frac{d^2X(x)}{dx^2} + k^2 X(x) = 0, \quad k \stackrel{\text{def}}{=} \frac{\omega}{c}.$$

Częstotliwość kątowa ω i wektor falowy k są określone przez wymaganie rozwiązań spełniających warunki brzegowe.

$$X(x=0,t) = X(x=l,t) = 0$$

$$\Rightarrow \quad X_n(x) = A_n \sin k_n x, \quad k_n = \frac{\pi(n+1)}{L}, \quad n = 0, 1, \ldots.$$

W związku z powyższym naszym rozwiązaniem czasowym jest:

$$T_n(t) = C_n \sin\omega_n t + D_n \cos\omega_n t, \quad \omega_n = nck_0 = n\frac{2\pi c}{L}$$

W ów rozwiązaniu częstotliwość n-tego trybu normalnego jest stała. W rzeczywistości jest to pojedyncza częstotliwość oscylacji definiująca tryb normalny.

Warunek początkowy prędkości zerowej, ∂ y / ∂ t (t = 0) = 0, wymaga, aby

wartości Cn w powyższym działaniu były równe zero. Podsumowując wszystkie te składowe rozwiązaniem w trybie normalnym są:

$$y_n(x,t) = \sin k_n x \, \cos \omega_n t, \quad n = 0, 1, \ldots.$$

Ponieważ równanie falowe jest liniowe względem y, obowiązuje zasada liniowej superpozycji a najbardziej ogólne rozwiązanie dla fal na strunie z ustalonymi końcami można zapisać jako sumę trybów normalnych:

$$y(x,t) = \sum_{n=0}^{\infty} B_n \, \sin k_n x \, \cos \omega_n t.$$

3 Rozwiązanie równania falowego

Rozwiązanie równania falowego przedstawia funkcję przemieszczenia u (x, t) zdefiniowaną dla wartości x od 0 do l oraz dla t od 0 do ∞ , która spełnia warunki początkowe i brzegowe. Równanie falowe wynika z konwekcyjnego typu problemów w drganiach, mechanice fal i dynamice gazów. To rozwiązanie nie jest tak proste, jak rozwiązanie zwykłego równania różniczkowego.

Równanie falowe jest najprostszym przykładem hiperbolicznego równania różniczkowego. Przedstawiony program do rozwiązywania równań falowych wykorzystuje następujące warunki brzegowe do rozwiązania problemów:

u (x, 0) = f (x)

u1 (x, 0) = ϕ (x)

u (0, t) = ψ 1 (t)

u (1, t) = ψ 2 (t)

dla $0 \leqslant t \leqslant T$

Po uruchomieniu program rozwiązuje równanie falowe, wykonując następujące czynności:

- Program pyta o wartość kwadratu c
- Użytkownik musi wprowadzić wartość warunków początkowych i brzego-

wych, czyli u (0, t) i u (5, t). W tym przypadku u (0, t) i u (5, t) są odpowiednio warunkami początkowymi i brzegowymi. Można je zmieniać w zależności od charakteru problemu i określonych kryteriów rozwiązania.

- Po wprowadzeniu tych parametrów program wyświetla dane wyjściowe na ekranie.

W tym przypadku zdefiniowana funkcja to f (x) = x2 (5-x).

```c
#include<stdio.h>
#include<math.h>
#define X 5
#define T 5

float fun(int x)
{
    return x * x * (5 - x);
}
main()
{
    float u[X + 1][T + 1], square_of_c, ut, ue;
    int i, j;
    printf("\n Wprowadź kwadrat c: ");
    scanf("%d", &square_of_c);
    printf(" Wprowadź wartość u[0][t]:");
    scanf("%f", &ut);
    printf(" Wprowadź wartość u[%d][t]:", X);
    scanf("%f", &ue);
    for (j = 0; j <= T; j++)
```

```c
    {
        u[0][j] = ut;
        u[X][j] = ue;
    }
    for (i = 1; i <= X - 1; i++)
        u[i][1] = u[i][0] = fun(i);
    for (j = 1; j <= T - 1; j++)
        for (i = 1; i <= X - 1; i++)
            u[i][j + 1] = u[i - 1][j] + u[i + 1][j] - u[i][j - 1];
    printf(" Wartości u[i][j] to: \n");
    for (j = 0; j <= T; j++)
    {
        for (i = 0; i <= X; i++)
            printf("%6.1f", u[i][j]);
        printf("\n");
    }
}
```

4 Fale dla zmiennego napięcia i gęstości

Prędkość propagacji fal na strunie wyznaczyliśmy jako c = T / ρ. Oznacza to tyle, że fale poruszają się wolniej w obszarach o dużej gęstości i szybciej w obszarach o wysokim napięciu. Jeśli gęstość struny zmienia się, na przykład ma ona grubsze końce (aby wspierać ciężar środka), to c nie będzie już stałą i nasze równanie falowe będzie musiało być rozszerzone. Ponadto, jeśli gęstość wzrośnie, to i napięcie wzrośnie (większe napięcie w celu przyspieszenia większej masy). Jeśli

działa grawitacja, to będziemy się również spodziewać napięcia przy końcach sznurka (końce powinny być wyżej niż środek, ponieważ muszą podtrzymywać wagę całego sznurka).

W celu wyprowadzenia równania ruchu fal ze zmienną gęstością i napięciem, rozważamy raz jeszcze element struny (Rysunek 2.) użyty do wyprowadzenia równania falowego. Jeśli nie założymy, że napięcie T jest stałe to używamy drugiego prawa Newtona:

$$F = ma$$

$$\Rightarrow \quad \frac{\partial}{\partial x}\left[T(x)\,\frac{\partial y(x,t)}{\partial x}\right]\Delta x = \rho(x)\Delta x\,\frac{\partial^2 u(x,t)}{\partial t^2}$$

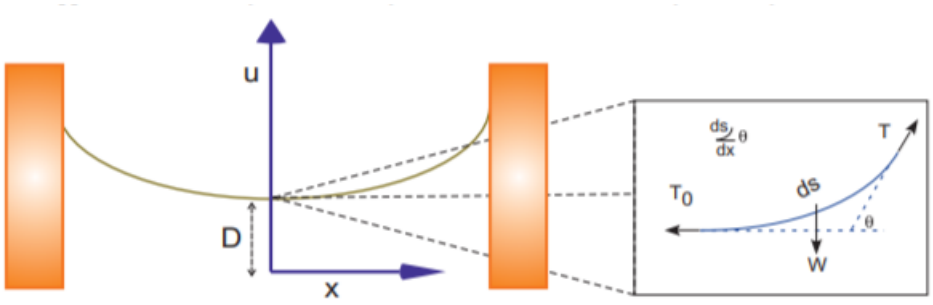
$$\Rightarrow \quad \frac{\partial T(x)}{\partial x}\,\frac{\partial y(x,t)}{\partial x} + T(x)\,\frac{\partial^2 y(x,t)}{\partial x^2} = \rho(x)\,\frac{\partial^2 y(x,t)}{\partial t^2}.$$

5 Fale w sieci

Aż do teraz ignorowaliśmy wpływ grawitacji na kształt naszej struny i jej napięcie. Jest to o tyle dobre przybliżenie, jeśli rozpatrujemy przypadek bardzo małego ugięcie struny (czyli w sytuacji, gdy np. napięcie jest bardzo wysokie, a struna lekka). Jeśli jednak struna jest masywna (mówimy o łancuchu, albo też bardzo cieżkiej linie), wtedy ugięcie w środku spowodowane grawitacją może być całkiem spore (Rysunek 3.), a wynikająca z tego różnica w kształcie i naprężeniu należy uwzględnić w równaniu falowym. Ponieważ napięcie nie jest już jednolite, fale przemieszczają się szybciej w pobliżu końców struny, które są pod większym naprężeniem, ponieważ muszą utrzymać cały ten ciężar.

6 Wyprowadzenie kształtu łańcucha

Rozważmy ciąg o jednolitej gęstości, na który działa grawitacja. Aby uniknąć nieporozumień z użyciem y (x) w celu opisania zaburzenia struny nazywamy u (x) równowagowym kształtem struny (Rysunek 3.). Problemem statycznym, który musimy rozwiązać jest określenie kształtu u (x) i napięcie T (x). Wstawka na Rysunku 3. jest diagramem swobodnego ciała środka struny i pokazuje, że ciężar W tej sekcji długości łuku s jest równoważony składową pionową naprężenia T. Naciąg poziomy T0 jest równoważony przez składową poziomą T:



Rysunek 3: Po lewej: jednolita struna zawieszona na końcach w polu grawitacyjnym przybiera kształt liny nośnej. Po prawej: wykres siły odcinka sieci trakcyjnej w jej najniższym punkcie. Ponieważ końce sznurka muszą utrzymać cały ciężar struny, napięcie zmienia się teraz wzdłuż struny.

$$T(x)\sin\theta = W = \rho g s, \quad T(x)\cos\theta = T_0,$$
$$\Rightarrow \quad \tan\theta = \rho g s / T_0.$$

Sposób ten polega na zamianie powyższego równania na równanie różniczkowe, które możemy rozwiązać. Robimy to poprzez zastępienie nachylenia tg θ $\frac{\partial}{\partial x}\left(\frac{du}{dx}\right)$:

$$\frac{du}{dx} = \frac{\rho g}{T_0} s, \quad \Rightarrow \quad \frac{d^2u}{dx^2} = \frac{\rho g}{T_0}\frac{ds}{dx}.$$

Musimy jednak pamiętać, że:

$$ds = \sqrt{dx^2 + du^2}$$

przez co nasze równanie wygląda w następujący sposób:

$$\frac{d^2u}{dx^2} = \frac{1}{D}\frac{\sqrt{dx^2+du^2}}{dx} = \frac{1}{D}\sqrt{1+\left(\frac{du}{dx}\right)^2},$$

$$D = T_0/\rho g,$$

gdzie D jest połączeniem stałych z wymiarem długości.
Równanie to jest równaniem dla sieci trakcyjnej o rozwiązaniu:

$$u(x) = D\cosh\frac{x}{D}.$$

7 Równanie Schrödingera zależne od czasu

Ponieważ obszar uwięzienia naszej cząstki ma rozmiar atomu, musimy rozwiązać ten problem kwantowy mechanicznie. Zaczynamy z cząstką o określonym pędzie i położeniu elektronu, co oznacza, że rozwiązaniem jest pakiet falowy, który nie jest stanem własnym z jednorodną zależnością (exp (- i ω t)) od czasu. W konsekwencji musimy teraz rozwiązać zależne od czasu równanie Schrodingera. W tym celu musimy określić funkcję falową dla wszystkich późniejszych czasów. Ewolucję czasową i przestrzenną cząstki kwantowej opisuje zależne od czasu równanie Schrodingera 1-D.

$$i\frac{\partial\psi(x,t)}{\partial t} = \tilde{H}\psi(x,t)$$

$$i\frac{\partial\psi(x,t)}{\partial t} = -\frac{1}{2m}\frac{\partial^2\psi(x,t)}{\partial x^2} + V(x)\psi(x,t),$$

gdzie podstawiliśmy 2m = 1. Ponieważ początkowa funkcja falowa jest złożona to rozkładamy ją na funkcję rzeczywistą oraz części urojone (ułatwi to nasze obliczenia):

$$
\begin{aligned}
\psi(x,t) &= R(x,t) + i\,I(x,t), \\
\psi(x,t) &= R(x,t) + i\,I(x,t), \\
\Rightarrow \quad \frac{\partial R(x,t)}{\partial t} &= -\frac{1}{2m}\frac{\partial^2 I(x,t)}{\partial x^2} + V(x)I(x,t), \\
\frac{\partial I(x,t)}{\partial t} &= +\frac{1}{2m}\frac{\partial^2 R(x,t)}{\partial x^2} - V(x)R(x,t),
\end{aligned}
$$

gdzie V (x) jest potencjałem działającym na cząstkę.

8 Rozpraszanie pakietu fal kwantowych za pomocą tridiagonalnego rozwiązania do równania Schrödingera dla 1D

```
#include <complex>
#include "memalloc.h"
#include "pde.h"
#define pi 3.141592653589793

using namespace std;
typedef complex<double> dcmplx;

double Pot(double x, double a, double V0)                    // Potencjał
{
    return fabs(x) <= 0.5e0 * a ? V0 : 0e0;
}
void Init(dcmplx Psi[], double x[], int nx, double x0, double sig, double k0)
{
    double a, b, dx, f;
    int i;

    a = 1e0 / sqrt(sqrt(2e0 * pi) * sig);
    b = -1e0 / (4e0 * sig * sig);
    for (i = 1; i <= nx; i++) {
        dx = x[i] - x0;
        f = a * exp(b * dx * dx); if (f < 1e-10) f = 0e0;
        Psi[i] = f * dcmplx(cos(k0 * x[i]), sin(k0 * x[i]));
    }
}
void ProbDens(dcmplx Psi[], double Psi2[], int nx, double hx, double& PsiNorm)
//Obliczanie gęstości prawdopodobieństwa Psi2[] funkcji falowej Psi[]
```

```
{
    int i;

    for (i = 1; i <= nx; i++) {
        Psi2[i] = norm(Psi[i]);             // nieznormalizowana gęstość prawdopodobieństwa
        if (Psi2[i] < 1e-10) Psi2[i] = 0e0;
    }

    PsiNorm = 0.5e0 * (Psi2[1] + Psi2[nx]);      // całka za pomocą trapezu
    for (i = 2; i <= (nx - 1); i++) PsiNorm += Psi2[i];
    PsiNorm *= hx;

    for (i = 1; i <= nx; i++) Psi2[i] /= PsiNorm;      // znormalizowane prawd. gęstość
}
int main()
{
    dcmplx* Psi;
    double* Psi2, * V, * x;
    double a, sig, ht, hx, k0, PsiNorm, t, tmax, V0, x0, xmax;
    int i, it, nout, nt, nx, nx2;
    char fname[80];
    FILE* out;

    a = 5e0;                                // szerokość potencjalnej bariery
    V0 = 50e0;                              // wysokość potencjalnej bariery
    x0 = -20e0;                           // początkowa pozycja fali
    sig = 1e0;                                      // połowa szerokości pakietu
```

```
    sig = 1e0;                                      // połowa szerokości pakietu
    k0 = 10e0;                            // średnia liczba fali pakietu
    xmax = 100e0;                                       // max x
    hx = 5e-2;                                  // rozmiar kroku
    tmax = 5e0;                               // maksymalny czas propagacji
    ht = 5e-3;                                      // krok czasowy
    nout = 40;

    nx = 2 * (int)(xmax / hx + 0.5) + 1;           // nieparzysta liczba węzłów przestrzennych
    nt = (int)(tmax / ht + 0.5);                        // liczba kroków w czasie
    nx2 = nx / 2;

    Psi = CVector(1, nx);                                  // funkcja fali
    Psi2 = Vector(1, nx);                               // gęstość prawdopodobieństwa
    V = Vector(1, nx);                                          // potencjał
    x = Vector(1, nx);                                  // współrzędna przestrzenna

    for (i = 1; i <= nx; i++) {
        x[i] = (i - nx2 - 1) * hx;
        V[i] = Pot(x[i], a, V0);
    }

    Init(Psi, x, nx, x0, sig, k0);

    for (it = 1; it <= nt; it++) {                                  // pętla czasowa
        t = it * ht;
        PropagQTD(Psi, V, nx, hx, ht);

        ProbDens(Psi, Psi2, nx, hx, PsiNorm);              // gęstość prawdopodobieństwa
```

```

        if (it % nout == 0 || it == nt) {
            sprintf(fname, "scatter_%4.2f.txt", t);
            out = fopen(fname, "w");
            fprintf(out, "t = %4.2f\n", t);
            fprintf(out, "      x          V        PsiR      PsiI      Psi2\n");
            for (i = 1; i <= nx; i++)
                fprintf(out, "%10.5f%10.5f%10.5f%10.5f%10.5f\n",
                    x[i], V[i], real(Psi[i]), imag(Psi[i]), Psi2[i]);
            fclose(out);
        }
    }
}
```

9 Bibliografia

- Materiały dydaktyczne udostępnione przez prowadzącego: 2Fale.pdf
- http://www.if.pw.edu.pl/ pluta/pl/dyd/lekcje/lekcja12/segment3/main.htm
- https://www.codewithc.com/category/numerical-methods/numerical-methods-c/
- http://phys.ubbcluj.ro/ tbeu/INP/programs.html
- http://www.if.pw.edu.pl/ agatka/numeryczne/wyklad09.pdf
- https://pl.wikipedia.org/wiki/Rownaniefalowe