

Obliczanie pierwiastków wielomianu metodą Newtona

Zastosowanie:

Funkcja *Newton_i* oblicza przybliżoną wartość pierwiastka wielomianu **poli_i** o współczynnikach w postaci przedziałów liczb zmiennopozycyjnych zapisanego w postaci $a_0 x^n + a_1 x^{n-1} + \dots + a_{n-1} x + a_n$ przy pomocy algorytmu Newtona, zaczynając od przedział startowego [**fgl**; **fgr**] z dokładnością do **eps**, wykonując maksymalnie **max_iter** iteracji

Opis metody

Przybliżona wartość pierwiastka wielomianu poli_i jest obliczana metodą iteracyjną Newtona na podstawie wzoru:

$$x_{k+1} = x_k - \frac{p(x_k)}{p'(x_k)}, \text{ dla } k = 0, 1, 2, \dots$$

gdzie wartość przybliżenia początkowego x_0 jest dana, a wartości $p(x_k)$ oraz $p'(x_k)$ wyliczane są za pomocą schematu Hornera. Iteracyjny proces obliczania kończy się, gdy różnica bezwzględna między dwoma kolejnymi przybliżeniami jest mniejsza od zadanej dokładności, lub wartość bezwzględna funkcji w znalezionym punkcie jest w przybliżeniu równa 0.

Wywołanie funkcji

Newton_i(fgl, fgr, eps, max_iter, p)

Dane

fgr, fgl - (first guess left/right) zmienne określające końce przedziału pierwszego przypisania

eps - zmienna określająca dokładność obliczeń

max_iter - zmienna określająca maksymalną liczbę wykonywanych iteracji algorytmu

p - tablica przechowująca współczynniki wielomianu, indeksowana od 0, w której element

$p_{2i} = \text{lewy kraniec przedziału współczynnika } a_i, i = 0, 1, 2, \dots, n, a$

$p_{2i+1} = \text{prawy koniec przedziału współczynnika } a_i, i = 0, 1, 2, \dots, n;$

Wynik

ans - ciąg znaków będący przekonwertowanym przedziałem, reprezentującym pierwiastek wielomianu, wraz z dopiskiem "Steps: " i liczbą wykonanych operacji. Zmienna definiowana wewnątrz ciała funkcji.

Parametry pomocnicze

Wszystkie poniższe zmienne definiowane są wewnątrz ciała funkcji

p_i - tablica przechowująca przedziały będące współczynnikami wielomianu opisanego w tablicy **p**, indeksowana od 0, w której element $p_i = a_i, i = 0, 1, 2, \dots, n$;

dfi - tablica przechowująca przedziały będące współczynnikami pochodnej funkcji w postaci $p = 0$, indeksowana od 0, w której element $dfi_i = a_i, i = 0, 1, 2, \dots, n$;

wyk - wykładnik potęgi, stosowany przy obliczaniu wartości funkcji (wielomianu) w punkcie;

fg - przedział startowy (pierwsze przybliżenie);

res - przedział wynikowy;

f - wartość funkcji w punkcie;

prim - wartość pochodnej w punkcie;

steps - licznik iteracji;

sg - wartość signum;

comp - przedział przesunięcia (zmienna dodatkowa wykorzystywana przy obsłudze błędów);

l, r - ciągi znaków zawierające końce przedziału wynikowego;

Typy parametrów

integer: *max_iter, steps, sg, wyk*

double: *fgl, fgr, eps*

Interval: *fg, res, f, prim, comp*

vector: *p_i, dfi*

string: *l, r*

QVector: *p*

QString: *ans*

Identyfikatory nielokalne

Interval: zmienna dwuelementowa typu krotkowego, przechowująca parę liczb będących końcami przedziału, reprezentowanymi przez zmienne zmiennoprzecinkowe typu double.

vector: zmienna dynamiczna typu tablicowego, przechowująca zmienne typu Interval, indeksowana od 0.

string: zmienna dynamiczna typu tablicowego, przechowująca zmienne typu character, indeksowana od 0, w której element o indeksie $i = 0, 1, 2, \dots$ odpowiada $i + 1$ znakowi w ciągu.

QString: zmienna dynamiczna typu tablicowego, przechowująca zmienne typu character, kompatybilna z systemem QML, indeksowana od 0, w której element o indeksie $i = 0, 1, 2, \dots$ odpowiada $i + 1$ znakowi w ciągu.

QVector: zmienna dynamiczna typu tablicowego, przechowująca zmienne typu double, indeksowana od 0, kompatybilna z systemem QML

Tekst funkcji

```
QString ean_class::newton_i(double fgl, double fgr, double eps, int max_iter, QVector<double> p)
{
    vector<Interval<double>> p_i;
    for(int i = 0; i < p.size(); i+=2)
    {
        p_i.push_back({p[i], p[i+1]});
    }
    vector<Interval<double>> dfi = {};
    int wyk = p_i.size()-1;
    for(int i = 0; i < p_i.size()-1; i++)
    {
        dfi.push_back(p_i[i]*wyk);
        wyk--;
    }
    Interval<double> fg = {fgl, fgr};
    Interval<double> res = fg;
    Interval<double> f;
    Interval<double> prim;
    int steps = 0;
    int sg = 0;
    Interval<double> comp;
    //schemat Hornera
    f = p_i[0];
    for(int i = 1; i < p_i.size(); i++)
    {
        f = res * f;
        f = p_i[i] + f;
    }
    prim = dfi[0];
    for(int i = 1; i < dfi.size(); i++)
    {
        prim = res * prim;
        prim = dfi[i] + prim;
    }

    QString ans;
    while (steps < max_iter && (f.a * f.b > 0 || fabs(f.a) > eps || fabs(f.b) > eps))
    {
        while(prim.a * prim.b <= 0 || fabs(prim.a) < eps || fabs(prim.b) < eps)
        {
            if(res.b < fg.a)
            {
                sg = 1;
            }
        }
    }
}
```

```

    }
    else{
        sg = -1;
    }
    comp = {sg*0.01, sg*0.01};
    res = fg - comp;
    fg = res;
    f = p_i[0];
    for(int i = 1; i < p_i.size(); i++)
    {
        f = res * f;
        f = p_i[i] + f;
    }
    prim = dfi[0];
    for(int i = 1; i < dfi.size(); i++)
    {
        prim = res * prim;
        prim = dfi[i] + prim;
    }
}
comp = f/prim;
if(comp.a*comp.b <= 0 && fabs(comp.a) < eps && fabs(comp.b) < eps)
{
    break;
}
res = res - f/prim;
f = p_i[0];
for(int i = 1; i < p_i.size(); i++)
{
    f = res * f;
    f = p_i[i] + f;
}
prim = dfi[0];
for(int i = 1; i < dfi.size(); i++)
{
    prim = res * prim;
    prim = dfi[i] + prim;
}
steps++;
}
cout << "steps: " << steps << endl;
ans = "[";
string l, r;
res.lEndsToStrings(l,r);
ans.append(QString::fromStdString(l));
ans.append(" ; ");
ans.append(QString::fromStdString(r));
ans.append("] Steps: ");

```

```

ans.append(QString::number(steps));
return ans;
}

```

Przykłady

- wielomian $W(x) = x^2 - 2$

dane: poli_i = { {1,1}, {0,0}, {-2,-2} }

eps = 1e-05

max_iter = 5

fgl = 1.0

fgr = 1.0

wynik: ans = [1.4142135623730922E0 ; 1.4142135623730979E0]

szerokość przedziału: 5.7e-15

st = 3
- wielomian

$$W(x) = [1.01; 1.02]x^3 + [0.98; 1.02]x^2 + [-3.01; -3]x + [-3.1; -2.9]$$

dane: poli_i = { {1.01,1.02}, {0.98,1.02}, {-3.01,-3}, {-3.1,-2.9} }

eps = 1e-3

max_iter = 100

fgl = 2.0

fgr = 2.12

wynik: ans = [1.7261692827875627E0 ; 2.0066650191013116E0]

szerokość przedziału = 2.8e-1

st = 100
- wielomian $W(x) = [-2.1; -2]x^3 + [1.11; 1.14]x^2 + [3.15; 3.17]$

dane: poli_i = { {-2.1, -2}, {1.11,1.14}, {0,0}, {3.15,3.17} }

eps = 1e-5

max_iter = 1000

fgl = 2.11

fgr = 2.12

wynik: ans = [1.0152308263061085E0 ; 1.8555566923096210E0]

szerokość przedziału = 8.403e-1

st = 1000