Iteracje proste

Witold Bołt

13 grudnia 2005

1 Treść zadania

Zbadać eksperymentalnie możliwości metody iteracji prostych dla układu równań:

$$\begin{cases} x = \frac{2x+x^2-y}{2} \\ y = \frac{2x-x^2+8}{9} + 4y - y^2 4 \end{cases}$$

2 Rozwiązanie

2.1 Uproszczenie układu

Przed zastosowaniem samej metody, możemy przekształcić nieco (uprościć) dany układ równań. Pierwsze równanie sprowadza się bowiem do warunku:

$$x^2 = y.$$

Drugie natomiast, do:

$$\frac{y^2}{4} = \frac{2x - x^2 + 8}{9}.$$

Z pierwszego równania wynika od razu, że jeśli jakiś y spełnia to równanie to na pewno $y \geqslant 0$. Z drugiego równania natomiast wynika, że $2x - x^2 + 8 \geqslant 0$, czyli $x \in [-2, 4]$. Jeśli założymy dodatkowo, że $x \geqslant 0$[1], to możemy spierwiastkować oba równania:

$$\begin{cases} x = \pm\sqrt{y} \\ \frac{y}{2} = \frac{\sqrt{2x-x^2+8}}{3} \end{cases}$$

Stąd otrzymujemy już „ostateczną" postać układu:

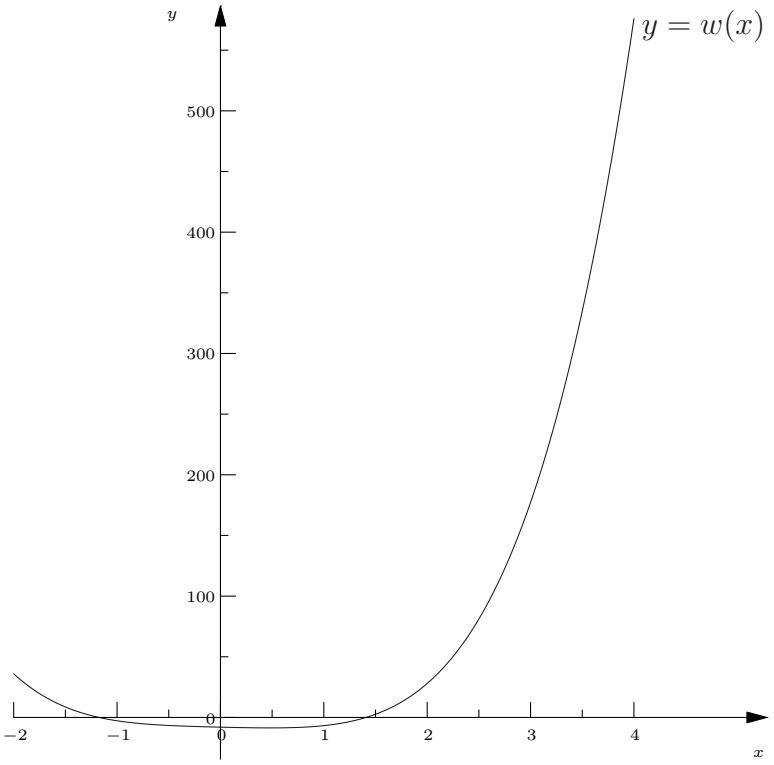
$$\begin{cases} x = \pm\sqrt{y} \\ y = \frac{2\sqrt{2x-x^2+8}}{3} \end{cases}$$

Łatwo zauważyć teraz, że układ ten moglibyśmy rozwiązać znajdując miejsca zerowe wielomianu (niezależnie od tego czy $x < 0$ czy też $x \geqslant 0$):

$$w(x) = \frac{9x^4}{4} + x^2 - 2x - 8, \quad x \in [-2, 4]$$

[1] W przeciwnym wypadku pierwsze równanie ma postać: $x = -\sqrt{y}$.

Rozwiązanie proponowane tutaj nie korzysta z tej metody. Wielomian ten może jednak posłużyć do sprawdzenia otrzymanego wyniku. Oto wykres tego wielomianu:



Z wykresu widać, że wielomian w w przedziale $[-2, 4]$ ma jedno miejsce zerowe dla $x > 0$ oraz jedna dla $x < 0$. Poniżej prezentowane rozwiązanie pozwala oszacować oba te rozwiązania z bardzo dużą dokładnością.

2.2 Iteracje

Szukamy rozwiązania układu (2.1). W tym celu będziemy wyliczać kolejne wyrazy ciągu (x_n, y_n) danego wzorem rekurencyjnym:

$$\begin{cases} x_{n+1} = \pm\sqrt{y_n} \\ y_{n+1} = \frac{2\sqrt{2x_n - x_n^2 + 8}}{3} \end{cases}$$

przy czym wyraz (x_0, y_0) jest pewnym ustalonym (podanym przez użytkownika) punktem, który spełnia założenia $x \in [-2, 4]$, $y \geqslant 0$.

2.3 Realizacja programowa

Program realizujący wyliczanie kolejnych wyrazów ciągu (x_n, y_n) jest bardzo prosty. Został napisany w języku C99. Oto kod źródłowy procedury głównej:

```
void przyb(double x, double y, int ite) {
  double xa,ya;
  double dir = 1;

  if(y<0 ||x<-2 ||x>4) {
    printf("Dane z poza dziedziny:\n\t -2 <= x <= 4, y>=0\n");
    return;
  }

  if (x<0) dir = -1;

  printf("0: x = %.30lf\t y = %.30lf\n",x,y);

  for(int n=1; n<=ite; n++) {
    ya = 2*sqrt((-1)*x*x + 2*x + 8)/3;
    xa = dir*sqrt(y);
    x=xa;
    y=ya;
    printf("%d: x = %.30lf\t y = %.30lf\n",n,x,y);
  }

// podstawiamy wyliczone wartosci do prawych stron „oryginalncyh" rownan
  printf("sprawdzenie:\n");
  x = 0.5*(2*xa+xa*xa-ya);
  y = (2*xa-xa*xa+8)/9 + (4*ya-ya*ya)/4;
  printf(" x = %.30lf\t y = %.30lf\n",x,y);

// badamy roznice miedzy tym co wyszlo z prawej i z lewej strony
  x = x-xa;
  y = y-ya;
  printf(" dx = %.30lf\t dy = %.30lf\n",x,y);
}
```

W programie zakładamy, że jeśli użytkownik poda jako wartość początkową x liczbę z przedziału $[-2, 0)$ to szukamy rozwiązania ujemnego, w przeciwnym wypadku dodatniego (do ustalenie tego służy zmienna `dir`).

2.4 Wyniki

Oto przykładowe wyniki działania programu.
Przykład dla $x \geqslant 0$:

```
ile iteracji? 22
x=0
y=0
0: x = 0.000000000000000         y = 0.000000000000000
1: x = 0.000000000000000         y = 1.885618083164127
2: x = 1.373178095938079         y = 1.885618083164127
3: x = 1.373178095938079         y = 1.984466131254004
4: x = 1.408710804691298         y = 1.984466131254004
5: x = 1.408710804691298         y = 1.981352565869934
6: x = 1.407605259250595         y = 1.981352565869934
7: x = 1.407605259250595         y = 1.981453781963722
8: x = 1.407641212086276         y = 1.981453781963722
9: x = 1.407641212086276         y = 1.981450494764864
10: x = 1.407640044459117        y = 1.981450494764864
11: x = 1.407640044459117        y = 1.981450601526666
12: x = 1.407640082381383        y = 1.981450601526666
13: x = 1.407640082381383        y = 1.981450598059254
14: x = 1.407640081149743        y = 1.981450598059254
15: x = 1.407640081149743        y = 1.981450598171869
16: x = 1.407640081189744        y = 1.981450598171869
17: x = 1.407640081189744        y = 1.981450598168212
18: x = 1.407640081188445        y = 1.981450598168212
19: x = 1.407640081188445        y = 1.981450598168331
20: x = 1.407640081188487        y = 1.981450598168331
21: x = 1.407640081188487        y = 1.981450598168327
22: x = 1.407640081188486        y = 1.981450598168327
sprawdzenie:
 x = 1.407640081188486    y = 1.981450598168327
 dx = 0.000000000000000   dy = 0.000000000000000
```

Przykład dla $x < 0$:

```
ile iteracji? 22
x=-1
y=0
0: x = -1.000000000000000        y = 0.000000000000000
1: x = -0.000000000000000        y = 1.490711984999860
2: x = -1.220947167161569        y = 1.885618083164127
3: x = -1.373178095938079        y = 1.344518696316765
4: x = -1.159533827155019        y = 1.223478778812415
5: x = -1.106109749894835        y = 1.388270490674269
6: x = -1.178248908624264        y = 1.424281763852756
```

```
7: x = -1.193432764697181        y = 1.375214196177325
8: x = -1.172695269956064        y = 1.364445790096621
9: x = -1.168094940532070        y = 1.379113284189124
10: x = -1.174356540488928       y = 1.382327237160296
11: x = -1.175724133102785       y = 1.377949141914738
12: x = -1.173860784724807       y = 1.376989388587232
13: x = -1.173451911493280       y = 1.378296740930736
14: x = -1.174008833412567       y = 1.378583296709780
15: x = -1.174130868647009       y = 1.378192954116066
16: x = -1.173964630692112       y = 1.378107392363600
17: x = -1.173928188759261       y = 1.378223943180290
18: x = -1.173977829083791       y = 1.378249490415349
19: x = -1.173988709662639       y = 1.378214690363689
20: x = -1.173973888280182       y = 1.378207062375586
21: x = -1.173970639486178       y = 1.378217453101561
22: x = -1.173975064940291       y = 1.378219730691583
sprawdzenie:
 x = -1.173976203735302  y = 1.378217592732228
 dx = -0.000001138795011         dy = -0.000002137959356
```