N7

Szymon Pach

Program napisany w Pythonie 2.7.14.
Rozwiązanie układu równań za pomocą gradientów sprzężonych.

Wyniki to liczby całkowite x1, x2 i x3 kolejno: 1, 2 i 1.

Kod programu:

```
1.  N = 3
2.  iterations = 0
3.
4.  def conjugateGradient(A, b, x):
5.      d = []
6.      g = []
7.      p = []
8.
9.      alpha = 0
10.     beta = 0
11.     dividend = 0
12.     divisor = 0
13.     temp0 = 0
14.     temp1 = 0
15.     abstand = 0
16.     stol = 0.0
17.
18.     i = 0
19.     j = 0
20.     k = 0
21.
22.     for i in range(N):
23.         x.append(0)
24.         p.append(0)
25.
26.     for i in range(N):
27.         temp0 = b[i]
28.         d.append(temp0)
29.         g.append(-temp0)
30.
31.
32.     for k in range(N, 0, -1):
33.         dividend = 0
34.         divisor  = 0
35.
36.         for i in range(N):
37.             dividend += d[i] * g[i]md
38.
39.             temp0 = 0
40.             for j in range(i):
41.                 temp0 += A[j][i] * d[j]
42.
43.             for j in range(i, N, 1):
44.                 temp0 += A[i][j] * d[j]
45.
46.
47.             p[i] = temp0
48.             divisor += d[i] * temp0
49.
50.         alpha = -dividend / divisor
51.         stol   = 0
52.         abstand = 0
53.
54.         for i in range(N):
55.             temp0 =  x[i]
```

```
            stol += pow(temp0, 2.0)
            temp1 =  alpha * d[i]
            abstand += pow(temp1, 2.0)
            x[i] =  temp0 + temp1


        for i in range(N):
            g[i] += alpha * p[i]

        dividend = 0

        for i in range(N):
            dividend += g[i] * p[i]


        beta = dividend / divisor

        for i in range(N):
            d[i] = -g[i] + beta * d[i]

        global iterations
        iterations += 1

A = [4.0, -1.0, 0.0], [-1.0, 4.0, -1.0], [0.0, -1.0, 4.0]
b = [2.0, 6.0, 2.0]

x = []

conjugateGradient(A, b, x);

print "Wyniki: \n"
for i in range(N):
    print "\t{}".format(x[i])
```