

```

1 import os, sys
2 sys.path.append(os.path.join(os.path.dirname(__file__), '../ch02'))
3
4 from program2_1 import Dvector
5 from program2_2 import Dmatrix
6 from program2_3 import input_vector, input_matrix
7 from program2_4 import inner_product
8 from program2_8 import vector_norm1
9
10 N = 10 # N元方程式
11 EPS = 10.0 ** -8.0 # epsilon の設定
12 KMAX = 100 # 最大反復回数
13
14 def main():
15     global N
16
17     a = Dmatrix(1, N, 1, N) # 行列 a[1...N][1...N]
18     b = Dvector(1, N) # b[1...N]
19     x0 = Dvector(1, N) # x[1...N]
20
21     # ファイルのオープン
22     with open("input_sp.dat", "r") as fin:
23         with open("output_sp.dat", "w") as fout:
24             input_matrix( a, 'A', fin, fout ) # 行列 A の入出力
25             input_vector( b, 'b', fin, fout ) # ベクトル b の入出力
26             input_vector( x0, 'x0', fin, fout ) # 初期ベクトル x0 の入出力
27             x = cg( a, b, x0 ) # 共役勾配法(CG法)
28
29             # 結果の出力
30             fout.write("Ax=b の解は次の通りです\n")
31             for i in range(1, N+1):
32                 fout.write("{:.6f}\n".format(x[i]))
33
34 # 共役勾配法 (CG法)
35 def cg(a: Dmatrix, b: Dvector, x0: Dvector):
36     k = 0
37
38     r = Dvector(1, N) # r[1...N]
39     p = Dvector(1, N) # p[1...N]
40     x = x0.copy()
41
42     tmp = matrix_vector_product( a, x ) # tmp <- A b
43
44     for i in range(1, N+1):
45         p[i] = b[i] - tmp[i]
46         r[i] = p[i]
47
48     while True:
49         # alpha の計算
50         tmp = matrix_vector_product( a, p ) # tmp <- A p_k
51         work = inner_product(p, tmp) # work <- (p, Ap_k)
52         alpha = inner_product(p, r) / work
53
54         # x_{k+1} と r_{k+1} の計算
55         for i in range(1, N+1):
56             x[i] += alpha * p[i]
57         for i in range(1, N+1):
58             r[i] -= alpha * tmp[i]
59
60         # 収束判定
61         eps = vector_norm1(r)
62         k += 1 # 反復回数の更新
63         if eps < EPS:
64             break
65
66         # beta と p_{k+1} の計算
67         beta = - inner_product(r, tmp) / work
68         for i in range(1, N+1):
69             p[i] = r[i] + beta * p[i]
70
71         if k >= KMAX:
72             break
73
74     if k == KMAX:
75         print("答えが見つかりませんでした")
76         exit(1)
77     else:
78         print(f"反復回数は{k}回です") # 反復回数を画面に表示
79         return x
80
81 # 行列 a[1...N][1...N] と ベクトルb[1...N] との積
82 def matrix_vector_product(a: Dmatrix, b: Dvector):
83     N = b.last_idx - b.head_idx + 1
84     c = Dvector(1, N)
85
86     for i in range(1, N+1):
87         c[i] = sum( ( a[i][j] * b[j] for j in range(1, N+1) ) )
88
89     return c
90
91 if __name__ == "__main__":
92     main()

```