

```

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_8 import vector_norm_max
8
9 N = 10 # N元方程式
10 EPS = 10.0 ** -8.0 # epsilon の設定
11 KMAX = 100 # 最大反復回数
12
13 def main():
14     a = Dmatrix(1, N, 1, N) # 行列 a[1...N][1...N]
15     b = Dvector(1, N) # b[1...N]
16     x = Dvector(1, N) # x[1...N]
17
18     # ファイルのオープン
19     with open("input_sp.dat", "r") as fin:
20         with open("output_sp.dat", "w") as fout:
21             input_matrix(a, 'A', fin, fout) # 行列 A の入出力
22             input_vector(b, 'b', fin, fout) # ベクトル b の入出力
23             input_vector(x, 'x', fin, fout) # 初期ベクトル x0 の入出力
24             x = jacobi_lin(a, b, x) # ヤコビ法
25
26             # 結果の出力
27             fout.write("Ax=b の解は次の通りです\n")
28             for i in range(1, N+1):
29                 fout.write("{:.6f}\n".format(x[i]))
30
31 # ヤコビ法
32 def jacobi_lin(a: Dmatrix, b: Dvector, x: Dvector) -> Dvector:
33     k = 0
34
35     xn = Dvector(1, N) # xn[1...N]
36
37     # x <- x_k, xn <- x_{k+1}
38     while True:
39         for i in range(1, N+1):
40             xn[i] = b[i]
41             for j in range(1, N+1):
42                 xn[i] -= a[i][j] * x[j]
43             xn[i] += a[i][i] * x[i] # 余分に引いた分を加える
44             xn[i] /= a[i][i]
45         for i in range(1, N+1):
46             x[i] = xn[i] - x[i]
47         eps = vector_norm_max(x) # 最大値ノルム∞の計算
48         for i in range(1, N+1):
49             x[i] = xn[i] # 値を更新
50         k += 1
51
52         if eps <= EPS or k >= KMAX:
53             break
54
55     if k == KMAX:
56         print("答えが見つかりませんでした")
57     else:
58         print(f"反復回数は{k}回です")
59
60     return x
61
62 if __name__ == "__main__":
63     main()

```