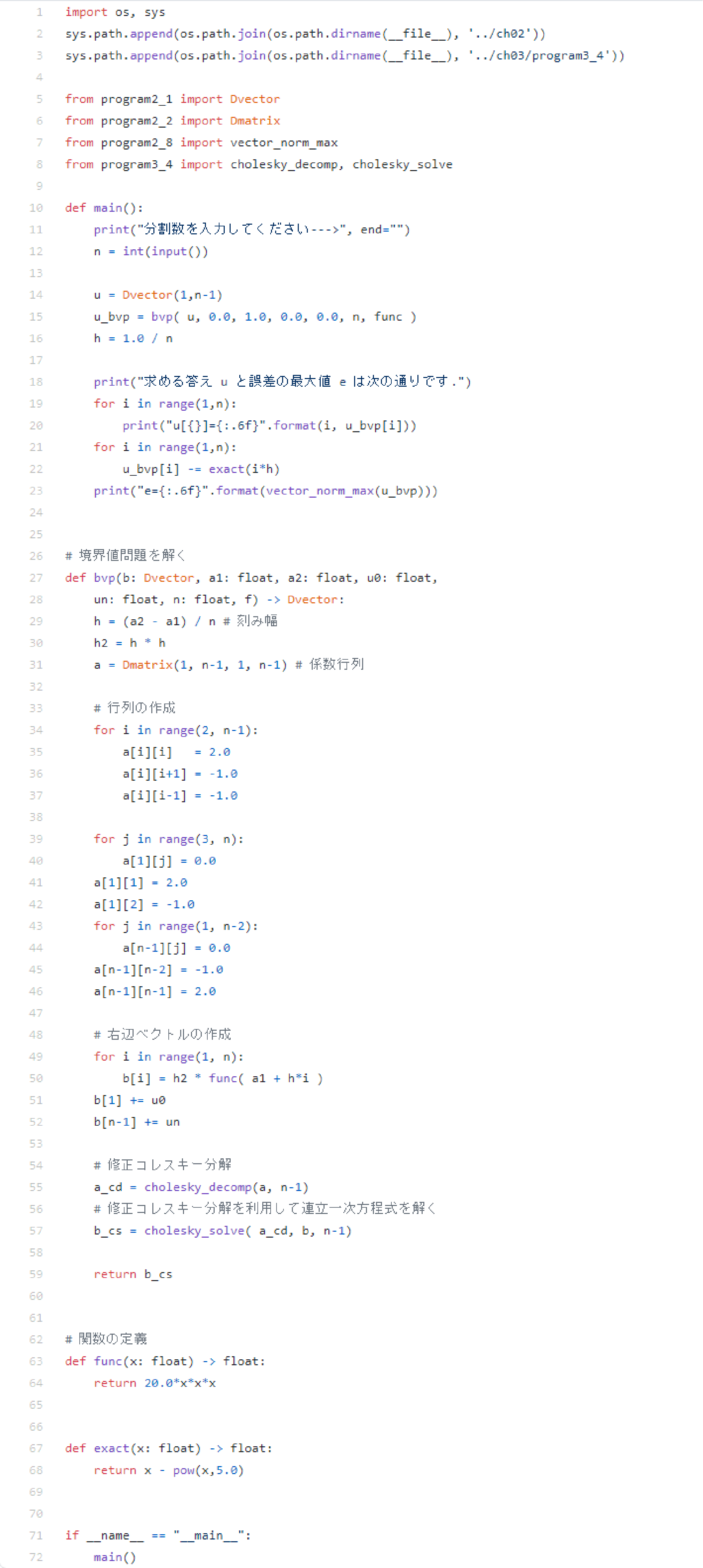

```python
import os, sys
sys.path.append(os.path.join(os.path.dirname(__file__), '../ch02'))
sys.path.append(os.path.join(os.path.dirname(__file__), '../ch03/program3_4'))

from program2_1 import Dvector
from program2_2 import Dmatrix
from program2_8 import vector_norm_max
from program3_4 import cholesky_decomp, cholesky_solve

def main():
    print("分割数を入力してください--->", end="")
    n = int(input())

    u = Dvector(1,n-1)
    u_bvp = bvp( u, 0.0, 1.0, 0.0, 0.0, n, func )
    h = 1.0 / n

    print("求める答え u と誤差の最大値 e は次の通りです.")
    for i in range(1,n):
        print("u[{}]={:.6f}".format(i, u_bvp[i]))
    for i in range(1,n):
        u_bvp[i] -= exact(i*h)
    print("e={:.6f}".format(vector_norm_max(u_bvp)))


# 境界値問題を解く
def bvp(b: Dvector, a1: float, a2: float, u0: float,
    un: float, n: float, f) -> Dvector:
    h = (a2 - a1) / n # 刻み幅
    h2 = h * h
    a = Dmatrix(1, n-1, 1, n-1) # 係数行列

    # 行列の作成
    for i in range(2, n-1):
        a[i][i]   = 2.0
        a[i][i+1] = -1.0
        a[i][i-1] = -1.0

    for j in range(3, n):
        a[1][j] = 0.0
    a[1][1] = 2.0
    a[1][2] = -1.0
    for j in range(1, n-2):
        a[n-1][j] = 0.0
    a[n-1][n-2] = -1.0
    a[n-1][n-1] = 2.0

    # 右辺ベクトルの作成
    for i in range(1, n):
        b[i] = h2 * func( a1 + h*i )
    b[1] += u0
    b[n-1] += un

    # 修正コレスキー分解
    a_cd = cholesky_decomp(a, n-1)
    # 修正コレスキー分解を利用して連立一次方程式を解く
    b_cs = cholesky_solve( a_cd, b, n-1)

    return b_cs


# 関数の定義
def func(x: float) -> float:
    return 20.0*x*x*x


def exact(x: float) -> float:
    return x - pow(x,5.0)


if __name__ == "__main__":
    main()
```