Computational Physics(A)

Assignment 2

Chon Hei Lo*(罗俊熙)

School of Physics, Peking University

November 5, 2023

注 *1:* 此作业的解答如无说明，统一使用爱因斯坦求和约定。 *P.S.:* 这次作业为甚么题目没有标题和分值 *QQ*，强迫症受不了 *QQ*

1 Problems & Solutions

1.1 解方程组

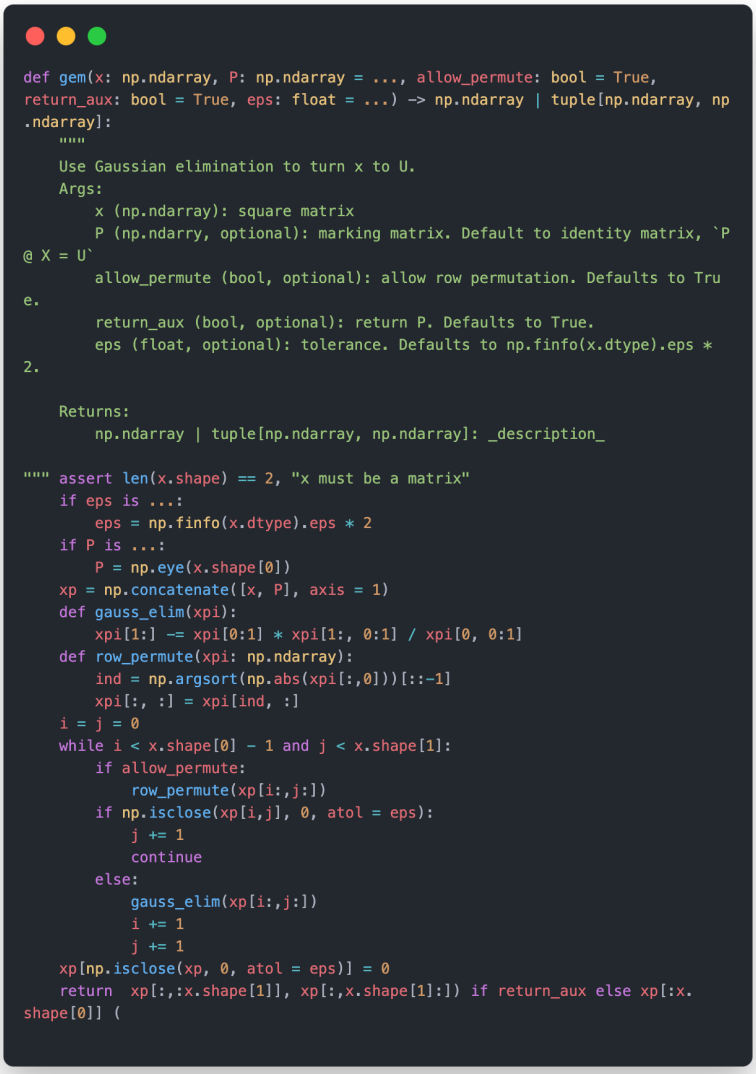
编写高斯消元法和 Cholesky 方法的代码，并求解如下线性方程组：

$$\begin{cases} 0.05x_1 + 0.07x_2 + 0.06x_3 + 0.05x_4 &= 0.23 \\ 0.07x_1 + 0.10x_2 + 0.08x_3 + 0.07x_4 &= 0.32 \\ 0.06x_1 + 0.08x_2 + 0.10x_3 + 0.09x_4 &= 0.33 \\ 0.05x_1 + 0.07x_2 + 0.09x_3 + 0.10x_4 &= 0.31 \end{cases}$$

Solution: 此题基本和上一次作业一样，所使用的代码片段如下：

两者结果一样，均为 $x = [1, 1, 1, 1]^T$

*Email: see.looooo@stu.pku.edu.cn; StudentID: 2000012508

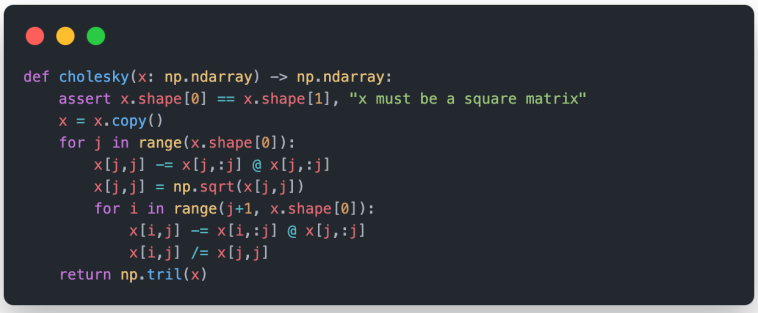


```
def gem(x: np.ndarray, P: np.ndarray = ..., allow_permute: bool = True,
return_aux: bool = True, eps: float = ...) -> np.ndarray | tuple[np.ndarray, np
.ndarray]:
    """
    Use Gaussian elimination to turn x to U.
    Args:
        x (np.ndarray): square matrix
        P (np.ndarry, optional): marking matrix. Default to identity matrix, `P
@ X = U`
        allow_permute (bool, optional): allow row permutation. Defaults to Tru
e.
        return_aux (bool, optional): return P. Defaults to True.
        eps (float, optional): tolerance. Defaults to np.finfo(x.dtype).eps *
2.

    Returns:
        np.ndarray | tuple[np.ndarray, np.ndarray]: _description_

""" assert len(x.shape) == 2, "x must be a matrix"
    if eps is ...:
        eps = np.finfo(x.dtype).eps * 2
    if P is ...:
        P = np.eye(x.shape[0])
    xp = np.concatenate([x, P], axis = 1)
    def gauss_elim(xpi):
        xpi[1:] -= xpi[0:1] * xpi[1:, 0:1] / xpi[0, 0:1]
    def row_permute(xpi: np.ndarray):
        ind = np.argsort(np.abs(xpi[:,0]))[::-1]
        xpi[:, :] = xpi[ind, :]
    i = j = 0
    while i < x.shape[0] - 1 and j < x.shape[1]:
        if allow_permute:
            row_permute(xp[i:,j:])
        if np.isclose(xp[i,j], 0, atol = eps):
            j += 1
            continue
        else:
            gauss_elim(xp[i:,j:])
            i += 1
            j += 1
    xp[np.isclose(xp, 0, atol = eps)] = 0
    return  xp[:,:x.shape[1]], xp[:,x.shape[1]:]) if return_aux else xp[:x.
shape[0]] (
```

Figure 1: 高斯消元法的代码片段



```
def cholesky(x: np.ndarray) -> np.ndarray:
    assert x.shape[0] == x.shape[1], "x must be a square matrix"
    x = x.copy()
    for j in range(x.shape[0]):
        x[j,j] -= x[j,:j] @ x[j,:j]
        x[j,j] = np.sqrt(x[j,j])
        for i in range(j+1, x.shape[0]):
            x[i,j] -= x[i,:j] @ x[j,:j]
            x[i,j] /= x[j,j]
    return np.tril(x)
```

Figure 2: Cholesky 方法的代码片段

1.2 样条函数插值

对 $f(x) = \cos(x^2)$，采用三次样条插值。分别考虑如下两种边界条件：

(a) $x_0 = 0$ 和 $x_2 = 0.9$ 端点处的二次导数值为 0；

(b) 利用 $f(x)$ 得到 $x_0 = 0$ 和 $x_2 = 0.9$ 端点处的一次导数值。

Solution: 参考代码 `2-2.py` 和 `seelib.spline.py`，代码这里就不重复贴上了，结果如下

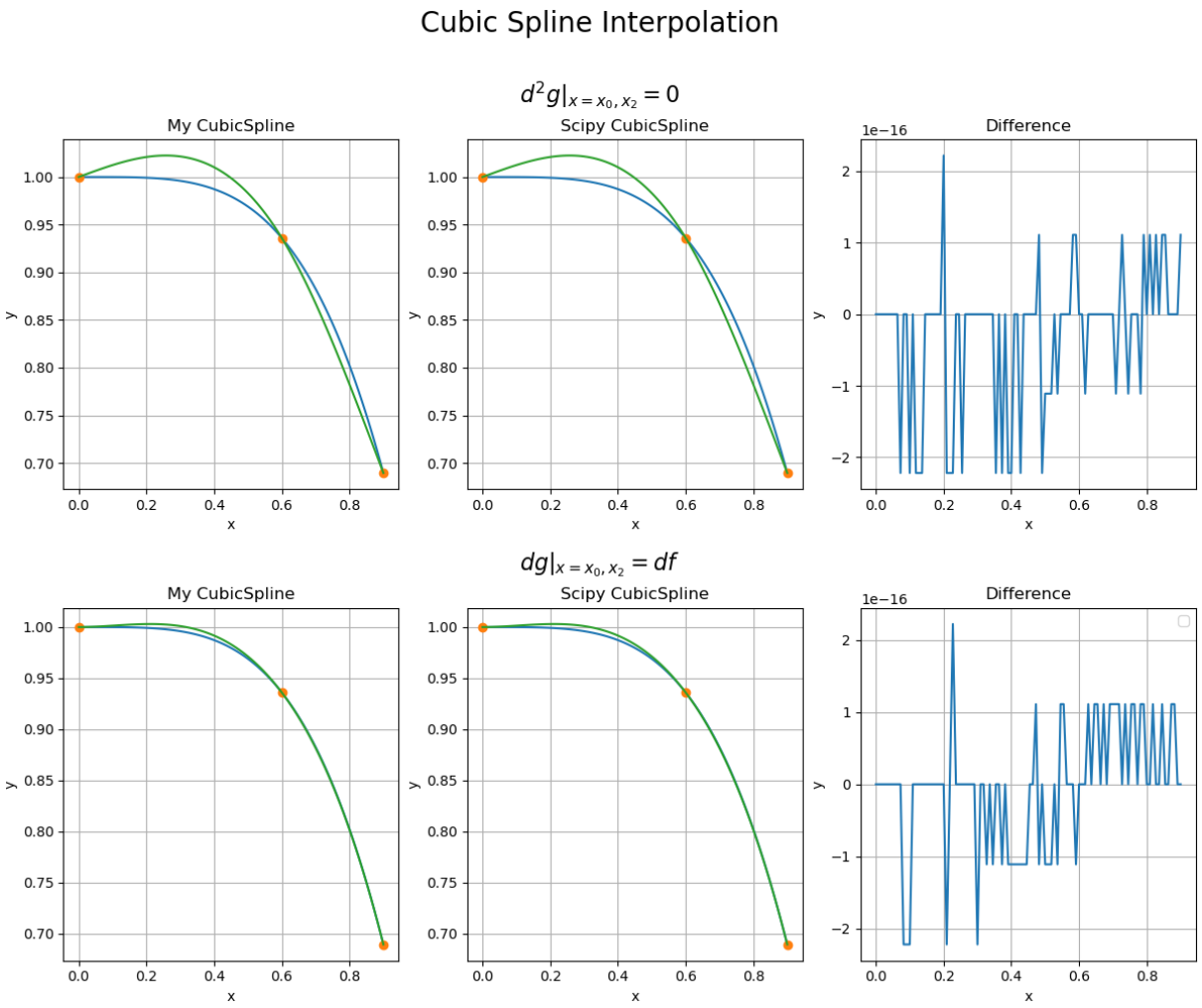


Figure 3: 两种边界条件下的样条函数插值

1.3 Runge 效应

考虑 Runge 函数 $f(x) = \frac{1}{1+25x^2}$ 在区间 $[-1, 1]$ 上的行为。本题中将分别利用等间距的多项式内插、Chebyshev 内插以及三次样条函数来近似 $f(x)$ 的数值。

(a) 考虑 $[-1, 1]$ 上的 21 个均匀分布的节点（包括端点，相隔 0.1 一个点）的 20 阶多项式 $P_{20}(x)$ 之内插（你可以利用各种方法，例如拉格朗日内插、牛顿内插或者 Neville 方法）。给出一个表分别列出 x，$f(x)$，$P_{20}(x)$ 以及两者差的绝对值。为了看出两者的区别请在这 21 个点分成的每个小段的中点也取一个数据点并一起列出 (因此共有 41 个点)，同时画图显示之。

(b) 现在选取 $n = 20$ 并将上问中均匀分布的节点换为标准的 Chebyshev 节点:

$$x_k = \cos\left(\frac{k + 1/2}{20}\pi\right), \quad k = 0, 1, \cdots, 19$$

然后构造 $f(x)$ 在 $[-1, 1]$ 上的近似式，

$$f(x) \approx C(x) \equiv -\frac{c_0}{2} + \sum_{k=0}^{20} c_k T_k(x)$$

其中在各个 Chebyshev 的节点处我们要求它严格等于 $f(x)$ 。同样列出上问的表并画图，与上问结果比较。

(c) 仍然考虑第一问中均匀分布的 21 个节点的内插。但这次利用 21 点的三次样条函数。重复上面的列表、画图并比较。

Solution: 参考代码 `2-3.py`，插值结果如??，它们值的差异在Table 1所示，可以看见在两边出现了巨大的差异。

Table 1: Runge 函数的插值结果

x	$f(x)$	$P_{20}(x)$	$\lvert P_{20}(x) - f(x)\rvert$	$T(x)$	$\lvert T(x) - f(x)\rvert$	$S(x)$	$\lvert S(x) - f(x)\rvert$
-1.00	0.04	0.04	0.00	0.04	0.01	0.04	0.00
-0.95	0.04	-39.95	39.99	0.05	0.01	0.04	0.00
-0.90	0.05	0.05	0.00	0.04	0.01	0.05	0.00
-0.85	0.05	3.45	3.40	0.06	0.00	0.05	0.00
-0.80	0.06	0.06	0.00	0.06	0.00	0.06	0.00
-0.75	0.07	-0.45	0.51	0.06	0.01	0.07	0.00
-0.70	0.08	0.08	0.00	0.07	0.00	0.08	0.00

Table 1: Runge 函数的插值结果

x	$f(x)$	$P_{20}(x)$	$\lvert P_{20}(x)-f(x)\rvert$	$T(x)$	$\lvert T(x)-f(x)\rvert$	$S(x)$	$\lvert S(x)-f(x)\rvert$
-0.65	0.09	0.20	0.12	0.09	0.01	0.09	0.00
-0.60	0.10	0.10	0.00	0.11	0.01	0.10	0.00
-0.55	0.12	0.08	0.04	0.11	0.00	0.12	0.00
-0.50	0.14	0.14	0.00	0.13	0.01	0.14	0.00
-0.45	0.16	0.18	0.01	0.16	0.00	0.16	0.00
-0.40	0.20	0.20	0.00	0.21	0.01	0.20	0.00
-0.35	0.25	0.24	0.01	0.26	0.01	0.25	0.00
-0.30	0.31	0.31	0.00	0.31	0.00	0.31	0.00
-0.25	0.39	0.40	0.00	0.38	0.01	0.39	0.00
-0.20	0.50	0.50	0.00	0.49	0.01	0.50	0.00
-0.15	0.64	0.64	0.00	0.64	0.00	0.64	0.00
-0.10	0.80	0.80	0.00	0.81	0.01	0.80	0.00
-0.05	0.94	0.94	0.00	0.95	0.01	0.94	0.00
0.00	1.00	1.00	0.00	1.00	0.00	1.00	0.00
0.05	0.94	0.94	0.00	0.95	0.01	0.94	0.00
0.10	0.80	0.80	0.00	0.81	0.01	0.80	0.00
0.15	0.64	0.64	0.00	0.64	0.00	0.64	0.00
0.20	0.50	0.50	0.00	0.49	0.01	0.50	0.00
0.25	0.39	0.40	0.00	0.38	0.01	0.39	0.00
0.30	0.31	0.31	0.00	0.31	0.00	0.31	0.00
0.35	0.25	0.24	0.01	0.26	0.01	0.25	0.00
0.40	0.20	0.20	0.00	0.21	0.01	0.20	0.00
0.45	0.16	0.18	0.01	0.16	0.00	0.16	0.00
0.50	0.14	0.14	0.00	0.13	0.01	0.14	0.00
0.55	0.12	0.08	0.04	0.11	0.00	0.12	0.00
0.60	0.10	0.10	0.00	0.11	0.01	0.10	0.00
0.65	0.09	0.20	0.12	0.09	0.01	0.09	0.00
0.70	0.08	0.08	0.00	0.07	0.00	0.08	0.00
0.75	0.07	-0.45	0.51	0.06	0.01	0.07	0.00
0.80	0.06	0.06	0.00	0.06	0.00	0.06	0.00

Table 1: Runge 函数的插值结果

x	$f(x)$	$P_{20}(x)$	$\lvert P_{20}(x) - f(x)\rvert$	$T(x)$	$\lvert T(x) - f(x)\rvert$	$S(x)$	$\lvert S(x) - f(x)\rvert$
0.85	0.05	3.45	3.40	0.06	0.00	0.05	0.00
0.90	0.05	0.05	0.00	0.04	0.01	0.05	0.00
0.95	0.04	-39.95	39.99	0.05	0.01	0.04	0.00
1.00	0.04	0.04	0.00	0.04	0.01	0.04	0.00

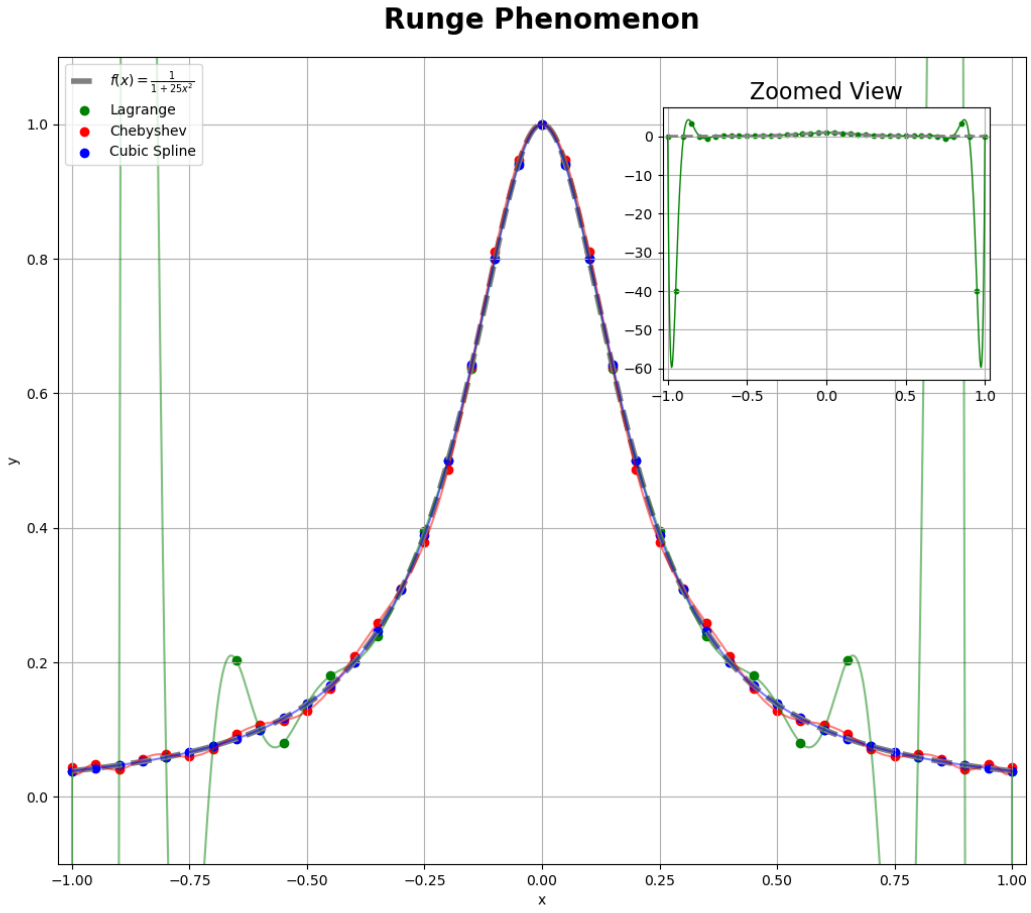


Figure 4: Runge 函数的插值结果

1.4 样条函数在计算机绘图中的运用

本题中我们考虑 Cubic spline 在计算机绘图中的广泛运用。我们将尝试用三次样条函数平滑地连接若干个二维空间中已知的点。考虑二维空间的一系列点

$(x_i, y_i), \quad i = 0, 1 \cdots n$。我们现在希望按照顺序（由 0 到 n）将它们平滑地连接起来。一个方便的办法是引入一个连续参数 $t \in [0, n]$，取节点为 $t_i = 0, 1, \cdots, n$ ，然后分别建立两个样条函数：$S_\Delta(X; t)$ 和 $S_\Delta(Y; t)$ 它们分别满足

$$S_\Delta(X; t_i) = x_i$$
$$S_\Delta(Y; t_i) = y_i$$

这两个样条函数可以看作是 $(x(t), y(t))$ 的内插近似。因此绘制参数曲线 $(x(t), y(t))$ 的问题就化为求出两个样条函数并将它们画出的问题。我们考虑的函数是着名的心形线（cardioid）。它的极坐标方程是：

$$r(\phi) = 2a(1 - \cos\phi)$$

为了方便起见我们取了 $2a = 1$。（请利用上一题中关于样条函数内插的相应代码来处理本题）

(a) 选取 $\phi = t\pi/4, \quad t = 0, 1, \cdots, 8$ 这九个点，结出 $x_t = r(\phi)\cos\phi$ 和 $y_t = r(\phi)\cos\phi$ 的数值。将这些数值作为精确的数值列在一个表里。

(b) 给出过这 8 个点的两个三次样条函数 $S_\Delta(X; t)$ 和 $S_\Delta(Y; t)$。

(c) 画出参数形式的曲线 $(x_t, y_t) = (S_\Delta(X; t), S_\Delta(Y; t))$ 同时画出它所内插的严格的曲线进行比较，请标出相应的节点。

(d) 简要说明为什么这个算法可以平滑地连接所有的点 (这实际上是很多画图软件中 spline 曲线所采用的算法)。

Solution: 参考代码 `2-4.py`，可以给出这九个点的数据，如下表：

Table 2: 心形线的数据

t	0	1	2	3	4	5	6	7	8
x_t	0.00	0.21	0.00	-1.21	-2.00	-1.21	0.00	0.21	0.00
y_t	0.00	0.21	1.00	1.21	0.00	-1.21	-1.00	-0.21	0.00

分别画出 x 和 y 的三次样条函数，如Figure 5所示。

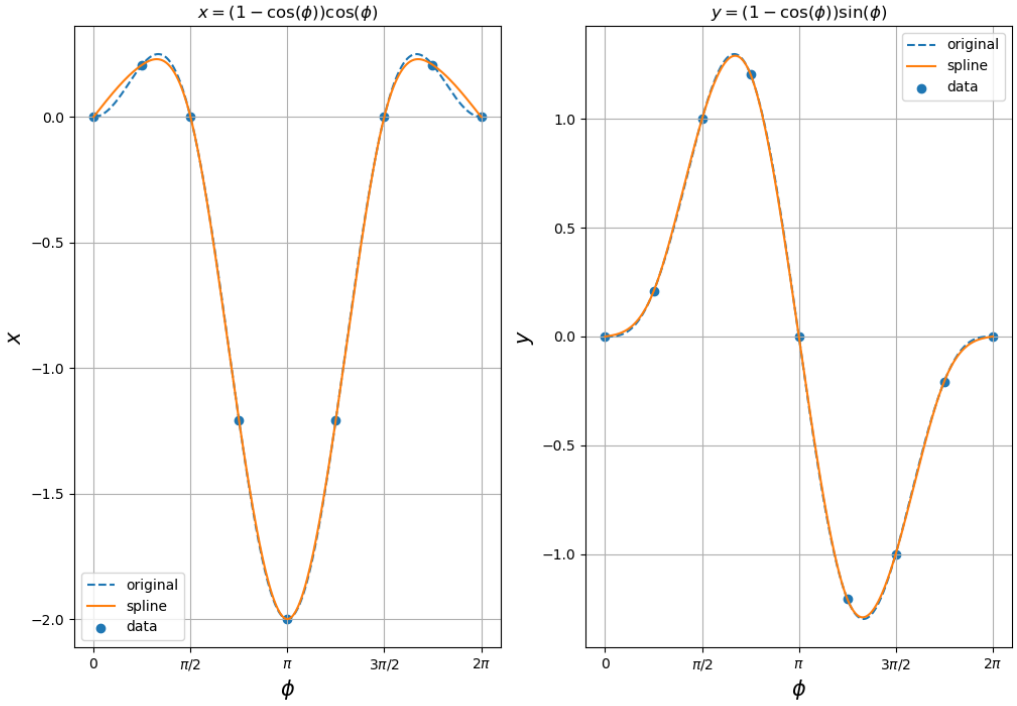


Figure 5: 两个参数方程分别的插值结果

通过三次样条函数，可以得到如下的结果：

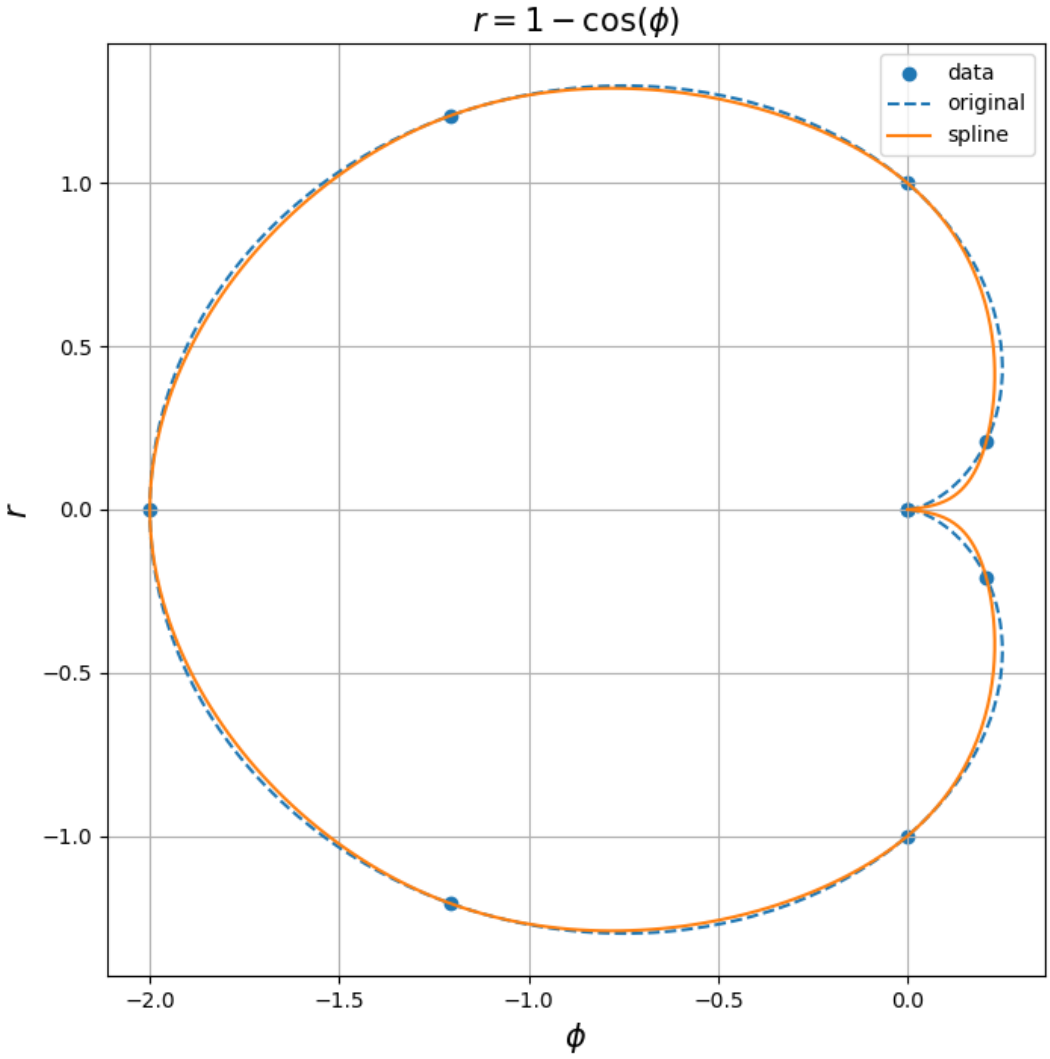


Figure 6: 心形线的插值结果

因为样条函数是连续可导的，因此 $f(x(t), y(t))$ 也是可导的，因此可以平滑地连接所有的点。

1.5 对称矩阵特征值问题

考虑一个对称矩阵

$$H = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}$$

(a) 通过消元法求出三角矩阵 L 和对角矩阵 D，使 $H = LDL^T$。

(b) 可以将矩阵 H 的一个角元 $H_{33} = 1$ 替换为 q ，使矩阵 H 为半正定，求出最小的 q。

(c) 当 $H_{33} = 2$ 时，求解其本征值。进而求出扩展到 4×4 矩阵 H' 时的本征值。

$$H' = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}$$

Solution:

(a) 使用高斯消元法进行消元，可以得到：

$$\left[\begin{array}{ccc} & H & \\ & & \\ \hline 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right] \Rightarrow \left[\begin{array}{ccc} 2 & 0 & 0 \\ 0 & \frac{3}{2} & -1 \\ 0 & -1 & 1 \\ \hline 1 & \frac{1}{2} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right]$$

$$\Rightarrow \left[\begin{array}{ccc} 2 & 0 & 0 \\ 0 & \frac{3}{2} & 0 \\ 0 & 0 & \frac{1}{3} \\ \hline 1 & \frac{1}{2} & \frac{1}{3} \\ 0 & 1 & \frac{2}{3} \\ 0 & 0 & 1 \end{array}\right]$$

记 $U^{-1} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ 0 & 1 & \frac{2}{3} \\ 0 & 0 & 1 \end{bmatrix}$ 和 $D = \begin{bmatrix} 2 & & \\ & \frac{3}{2} & \\ & & \frac{1}{3} \end{bmatrix}$，求解 U：

$$\left[\begin{array}{c|c} U^{-1} & I \end{array}\right]$$

$$
\begin{aligned}
&= \left[\begin{array}{ccc|ccc} 1 & \frac{1}{2} & \frac{1}{3} & 1 & 0 & 0 \\ & 1 & \frac{2}{3} & 0 & 1 & 0 \\ & & 1 & 0 & 0 & 1 \end{array}\right] \\
&\Rightarrow \left[\begin{array}{ccc|ccc} 1 & \frac{1}{2} & 0 & 1 & 0 & -\frac{1}{3} \\ & 1 & 0 & 0 & 1 & -\frac{2}{3} \\ & & 1 & 0 & 0 & 1 \end{array}\right] \\
&\Rightarrow \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & -\frac{1}{2} & 0 \\ & 1 & 0 & 0 & 1 & -\frac{2}{3} \\ & & 1 & 0 & 0 & 1 \end{array}\right]
\end{aligned}
$$

得到 $L = \begin{bmatrix} 1 & & \\ -\frac{1}{2} & 1 & \\ & -\frac{2}{3} & 1 \end{bmatrix}$。容易验证 $H = LDL^T$（或者使用 `2-5.py`）进行检验。

(b) 在进行变换的过程中，我们轻松地发现了角元 $H_{33} = q$ 的改变对 L 不会影响，那么 $D_{33} = q - \frac{2}{3}$，若要求 H 半正定，那么 $D_{33} \geq 0$，可知 $q \geq \frac{2}{3}$。

(c) 直接展开其特征多项式：

$$|\lambda I - H| = (\lambda - 2)^4 - 3(\lambda - 2)^2 + 1$$

得到 $(\lambda - 2)^2 = \frac{3 \pm \sqrt{5}}{2}$，故特征值有 4 个，分别为：

$$
\begin{aligned}
\lambda_1 &= 2 + \sqrt{\frac{3 + \sqrt{5}}{2}} \\
\lambda_2 &= 2 - \sqrt{\frac{3 + \sqrt{5}}{2}} \\
\lambda_3 &= 2 + \sqrt{\frac{3 - \sqrt{5}}{2}} \\
\lambda_4 &= 2 - \sqrt{\frac{3 - \sqrt{5}}{2}}
\end{aligned}
$$