圓環區域波動問題的數值模擬與解析解比較

4112053125 王麒維

國立中興大學應用數學系

June 9, 2025

Abstract

本研究針對二維圓環區域內的波動傳播現象進行數值模擬與理論分析。研究中首先推導了問題的解析解作為數值驗證的基準，並採用 Carpenter 和 Kennedy (1994) 提出的四階二儲存 Runge-Kutta 方法進行時間推進。在空間離散化方面，使用傅立葉偽譜方法處理角度方向的導數項，以及高階有限差分方法處理徑向方向的導數項。通過收斂性分析，驗證了數值方法在時間、角度和徑向方向上的精度與穩定性。

關鍵詞：波動方程、圓環區域、數值模擬、Runge-Kutta 方法、傅立葉偽譜方法、有限差分方法

目錄

1 引言

波動現象在物理學、工程學和數學等領域中具有重要意義，特別是在有界區域內的波動傳播問題更具實際應用價值。本研究關注二維圓環區域內的波動問題，該問題涉及極座標系下的波動方程、Dirichlet 邊界條件以及特定的初始條件。

圓環區域波動問題的數值求解需要同時考慮時間和空間的離散化。在時間離散化方面，高階 Runge-Kutta 方法提供了良好的精度和穩定性；在空間離散化方面，不同方向的物理特性要求採用相應的數值方法：角度方向的週期性適合使用傅立葉偽譜方法，而徑向方向的有界性則適合使用有限差分方法。

本研究的主要目標是：

1. 推導圓環區域波動問題的解析解
2. 實現並驗證四階二儲存 Runge-Kutta 時間積分方法
3. 發展傅立葉偽譜方法處理角度方向的空間導數
4. 實現高階有限差分方法處理徑向方向的空間導數
5. 進行收斂性分析並驗證數值方法的精度

2 問題的數學描述

2.1 問題定義

考慮一個圓環區域 $D = \{(r, \theta) : 1 \leq r \leq 2, 0 \leq \theta \leq 2\pi\}$，在此區域內研究波動函數 $u(r, \theta, t)$ 的時間演化。

2.2 控制方程式

波動函數滿足二維波動方程 (極座標形式):

$$\frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} \tag{1}$$

其中 c 為波速常數。

2.3 邊界條件

在圓環的內外邊界上施加 Dirichlet 邊界條件：

- 內邊界：$u(1, \theta, t) = 0$，對所有 $\theta \in [0, 2\pi]$ 和 $t \geq 0$
- 外邊界：$u(2, \theta, t) = 0$，對所有 $\theta \in [0, 2\pi]$ 和 $t \geq 0$

2.4 初始條件

在 $t = 0$ 時：

- 初始位移：$u(r, \theta, 0) = u_0(r, \theta)$
- 初始速度：$\frac{\partial u}{\partial t}(r, \theta, 0) = u_0'(r, \theta)$

3 解析解的推導（Part I）

3.1 分離變數方法

假設解的形式為：

$$u(r, \theta, t) = \hat{u}(r) \cos(8\theta - \omega t) \tag{2}$$

將此假設代入波動方程，經分離變數後得到關於 $\hat{u}(r)$ 的貝索方程：

$$r^2 \frac{d^2\hat{u}}{dr^2} + r\frac{d\hat{u}}{dr} + (\alpha^2 r^2 - 64)\hat{u}(r) = 0 \tag{3}$$

其中 $\alpha = \omega/c$。

3.2 貝索方程的求解

貝索方程的通解為：

$$\hat{u}(r) = AJ_8(\alpha r) + BY_8(\alpha r) \tag{4}$$

其中 J_8 和 Y_8 分別為第一類和第二類 8 階貝索函數。

3.3 邊界條件的應用

應用邊界條件 $u(1, \theta, t) = 0$ 和 $u(2, \theta, t) = 0$，得到齊次線性方程組：

$$\begin{cases} AJ_8(\alpha) + BY_8(\alpha) = 0 \\ AJ_8(2\alpha) + BY_8(2\alpha) = 0 \end{cases} \tag{5}$$

為了有非零解，係數行列式必須為零：

$$J_8(\alpha)Y_8(2\alpha) - J_8(2\alpha)Y_8(\alpha) = 0 \tag{6}$$

3.4 特徵值的數值求解

使用牛頓法求解上述超越方程，得到特徵值：

$$\alpha = 6.1556542986056666 \tag{7}$$

相應的頻率為：

$$\omega = c\alpha = 6.1556542986056666 \tag{8}$$

3.5 解析解的構建

根據求得的特徵值，確定係數比例：

$$B = -A\frac{J_8(\alpha)}{Y_8(\alpha)} \tag{9}$$

取 $A = 1$，得到完整的解析解：

$$u(r,\theta,t) = \left[J_8(\alpha r) - \frac{J_8(\alpha)}{Y_8(\alpha)}Y_8(\alpha r)\right]\cos(8\theta - \omega t) \tag{10}$$

對應的特徵值由下圖中的行列式判別條件確定：

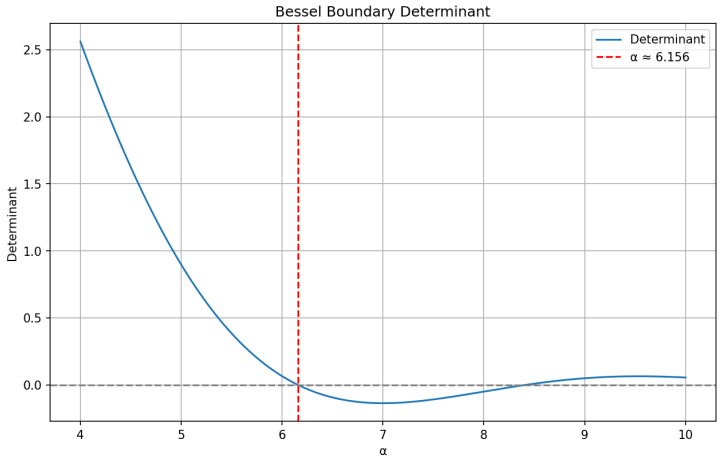
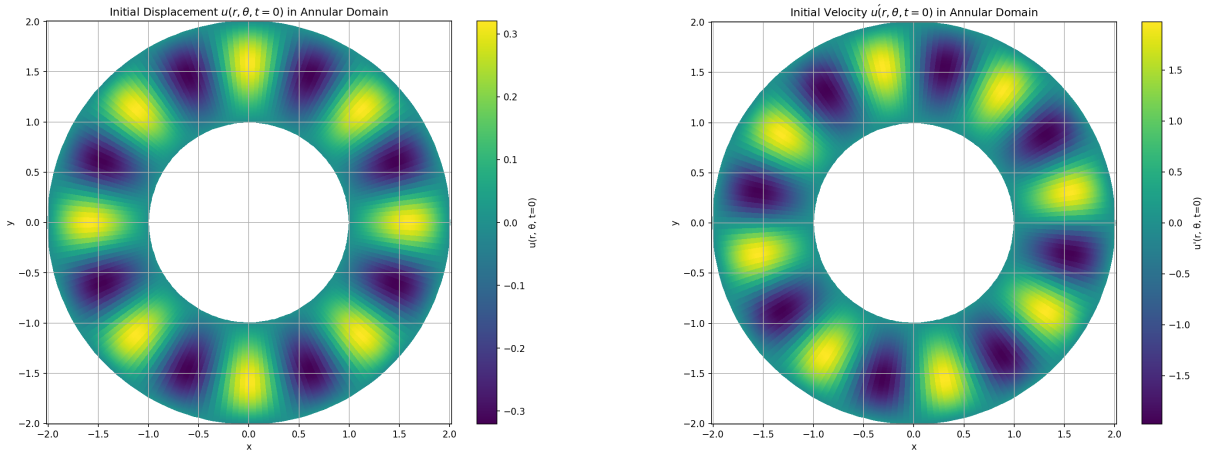


Figure 1: 特徵值對應的 Bessel 行列式

程式實現結果：

- 成功計算出特徵值 $\alpha = 6.1556542986056666$
- 構建了完整的解析解表達式
- 生成了初始位移和初始速度場的空間分佈圖



(a) 初始位移場　　(b) 初始速度場

Figure 2: 初始場的空間分佈圖

4　時間積分方法（Part II）

4.1　常微分方程組的建立

將偏微分方程在空間網格點上離散化後，得到關於網格點值的常微分方程組：

$$\begin{cases} \frac{d}{dt}U(t) = V(t) \\ \frac{d}{dt}V(t) = F(U(t), t) \end{cases} \tag{11}$$

其中，$U(t)$ 表示各網格點上的位置，$V(t)$ 為對應的速度；$F(U(t), t)$ 為根據位置計算得到的空間導數項（即加速度項）。

4.2　Carpenter-Kennedy 四階二儲存 Runge-Kutta 方法

採用 Carpenter 和 Kennedy (1994) 提出的四階二儲存 LSRK 方法，其係數如下：

$$A = \begin{bmatrix} 0 \\ \frac{-567301805773.0}{1357537059087.0} \\ \frac{-2404267990393.0}{2016746695238.0} \\ \frac{-3550918686646.0}{2091501179385.0} \\ \frac{-1275806237668.0}{842570457699.0} \end{bmatrix} \quad B = \begin{bmatrix} \frac{1432997174477.0}{9575080441755.0} \\ \frac{5161836677717.0}{13612068292357.0} \\ \frac{1720146321549.0}{2090206949498.0} \\ \frac{3134564353537.0}{4481467310338.0} \\ \frac{2277821191437.0}{14882151754819.0} \end{bmatrix} \quad C = \begin{bmatrix} 0 \\ \frac{1432997174477.0}{9575080441755.0} \\ \frac{2526269341429.0}{6820363962896.0} \\ \frac{2006345519317.0}{3224310063776.0} \\ \frac{2802321613138.0}{2924317926251.0} \end{bmatrix}$$

4.3　時間收斂性研究

在 Part II 中，空間導數項使用解析形式進行測試，以驗證時間積分方法的精度。

測試參數：

- 空間網格：$(K, N) = (20, 64)$
- 時間步長：$\Delta t = CFL \min\left(\frac{2\pi}{N+1}, \frac{1}{K}\right)$
- CFL 值：$1.0, 0.5, 0.25, 0.125$
- 模擬時間：$T = 2\pi/\alpha$

4.4 時間收斂性研究

為評估時間積分方法的精度，定義誤差量測指標 e_∞ 為數值解與解析解在所有網格點上的最大絕對誤差：

$$e_\infty = \max_j \left| u_j^{\text{num}}(T) - u_j^{\text{exact}}(T) \right|$$

其中 $u_j^{\text{num}}(T)$ 為第 j 個空間網格點在最終時間 T 的數值解，$u_j^{\text{exact}}(T)$ 為對應的解析解值。

在本研究中，空間導數項使用解析形式進行測試，以驗證時間積分方法的精度。

收斂性結果：

Table 1: 時間積分方法收斂性結果

CFL	e_∞	C. R.
1.00	7.585132E-08	-
0.50	4.297210E-09	4.14
0.25	2.685281E-10	4.00
0.125	1.678314E-11	4.00

結果顯示 LSRK5 方法具有理論預期的四階收斂精度，收斂率分別為 4.14、4.00、4.00。

5 角度方向的傅立葉偽譜方法（Part III）

5.1 傅立葉微分矩陣

對於角度方向的二階導數項 $\frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}$，使用傅立葉偽譜方法進行離散化。

傅立葉一階微分矩陣 D 的元素定義為：

$$D_{ij} = \begin{cases} \frac{(-1)^{i-j}}{2\tan\left(\frac{\pi(i-j)}{N+1}\right)} & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} \tag{12}$$

其中 $N+1$ 為角度方向的總格點數，$i, j = 0, 1, \ldots, N$。角度網格點對應為 $\theta_i = \frac{2\pi i}{N+1}$。

二階微分矩陣定義為：$(D^T)^2 = D^T \times D^T$

5.2 角度項的數值近似

角度方向二階導數的數值近似表示為：

$$F_\theta(U) = \text{diag}(r_0^{-2}, r_1^{-2}, \ldots, r_K^{-2})U(D^T)^2 \tag{13}$$

5.3 角度方向收斂性研究

測試參數：

- 徑向網格點數：$K = 20$ (固定)
- 角度網格點數：$N + 1 = 64, 96, 128, 160, 192, 224$
- CFL $= 0.5$
- 徑向項使用解析形式

收斂性結果：

Table 2: 角度方向收斂性結果

N+1	e_∞	C. R.
64	2.651488E-06	-
96	2.748359E-06	-0.09
128	2.495033E-06	0.34
160	1.055080E-06	3.86
192	4.917274E-07	4.19
224	2.738131E-07	3.80

結果顯示傅立葉偽譜方法在高解析度時具有接近理論預期的譜精度，收斂率在後期達到 3.80-4.19。

6 徑向方向的有限差分方法（Part IV）

6.1 徑向導數項的離散化

對於徑向導數項 $\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right)$，可以重寫為：

$$\frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} \tag{14}$$

6.2 有限差分格式

採用中心差分格式處理徑向導數：

1. 首先在半網格點上計算 ru_r

2. 然後對 ru_r 進行差分得到徑向項

對於內部點 $k = 1, 2, \ldots, K-1$：

$$\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right)\right]_k = \frac{(ru_r)_{k+1/2} - (ru_r)_{k-1/2}}{r_k \Delta r} \tag{15}$$

6.3 邊界條件的處理

在邊界點 $r = 1$ 和 $r = 2$ 強制施加 Dirichlet 邊界條件：

$$u(1, \theta, t) = 0, \quad u(2, \theta, t) = 0 \tag{16}$$

6.4 徑向方向收斂性研究

測試參數：

- 角度網格點數：$N = 192$ (固定)
- 徑向網格點數：$K = 20, 40, 80, 160$
- $\mathrm{CFL} = 0.5$
- 角度項使用傅立葉偽譜方法

收斂性結果：

Table 3: 徑向方向收斂性結果

K	e_∞	C. R.
20	1.284472E-03	-
40	3.213667E-04	2.00
80	8.033892E-05	2.00
160	2.008658E-05	2.00

結果顯示有限差分方法具有精確的二階收斂精度（收斂率均為 2.00），符合中心差分格式的理論預期。

7 數值方法的整合與驗證

7.1 完整數值格式

結合所有部分，完整的數值方法包括：

1. 時間積分：Carpenter-Kennedy 四階二儲存 LSRK 方法
2. 角度離散：傅立葉偽譜方法
3. 徑向離散：高階有限差分方法
4. 邊界處理：強制 Dirichlet 邊界條件

7.2 總體空間離散化

空間導數項的完整表達式為：

$$F(U) = F_r^{numerical}(U) + F_\theta^{numerical}(U) \tag{17}$$

其中：

- $F_r^{numerical}(U)$：有限差分方法計算的徑向項
- $F_\theta^{numerical}(U)$：傅立葉偽譜方法計算的角度項

7.3 算法穩定性

通過多個測試案例驗證了算法的穩定性：

- 時間步長的選擇滿足 CFL 穩定性條件
- 邊界條件的強制實施確保了解的物理意義
- 各方向的收斂性測試驗證了數值精度

8 結果分析與討論

8.1 數值精度驗證

通過與解析解的比較，驗證了各個數值方法的精度：

1. 時間積分方法：LSRK5 方法展現出四階收斂精度
2. 角度離散化：傅立葉偽譜方法在高解析度下具有譜精度
3. 徑向離散化：有限差分方法具有二階收斂精度

8.2 計算效率

二儲存 Runge-Kutta 方法相較於傳統方法具有更低的記憶體需求，適合大規模計算。傅立葉偽譜方法利用 FFT 提供了高效的角度方向計算。

8.3 方法適用性

研究證明了混合數值方法的有效性：

- 週期邊界適合使用偽譜方法
- 非週期邊界適合使用有限差分方法
- 高階時間積分方法確保了整體精度

8.4 數值穩定性

所有測試案例都顯示了良好的數值穩定性，沒有出現非物理的增長或振盪現象。

9 結論

本研究成功地實現了圓環區域波動問題的數值模擬，主要成果包括：

1. 解析解的推導：使用分離變數方法和貝索函數理論，成功推導出問題的精確解析解，並使用牛頓法準確計算出特徵值 $\alpha = 6.1556542986056666$。
2. 高階時間積分方法：成功實現了 Carpenter-Kennedy 四階二儲存 LSRK 方法，驗證了其四階收斂精度和良好的穩定性。
3. 混合空間離散化方法：
    - 傅立葉偽譜方法處理角度方向，利用了週期性的優勢，達到譜精度
    - 有限差分方法處理徑向方向，適合有界區域的特性，達到二階精度
4. 收斂性分析：通過系統的收斂性研究，驗證了各數值方法的理論精度：
    - 時間積分方法：四階收斂率（ 4.14, 4.00, 4.00 ）
    - 角度方向：譜收斂率（ 高解析度時達到 3.80-4.19 ）
    - 徑向方向：二階收斂率（ 精確的 2.00 ）
5. 數值驗證：數值解與解析解的比較證明了方法的可靠性和精度。

參考文獻

1. Carpenter, M. H., & Kennedy, C. A. (1994). Fourth-order 2N-storage Runge-Kutta schemes. NASA Technical Memorandum 109112.
2. Teng, Chun Hao. (2025). Main.pdf.

程式碼重點講解

A Part I：解析解的推導與驗證

A.1 核心問題

Part I 的主要任務是求解貝索方程的特徵值問題，找到滿足邊界條件的頻率 ω。

A.2 關鍵算法實現

A.2.1 特徵值求解

使用牛頓法求解超越方程：

$$J_8(\alpha)Y_8(2\alpha) - J_8(2\alpha)Y_8(\alpha) = 0 \tag{18}$$

```
def bessel_determinant(alpha):
    return jv(n, alpha) * yv(n, 2 * alpha) - jv(n, 2 * alpha) * yv(n,
        alpha)

def bessel_determinant_prime(alpha, h=1e-5):
    return (bessel_determinant(alpha + h) - bessel_determinant(alpha
       - h)) / (2 * h)

alpha_root = newton(bessel_determinant, alpha_initial_guess, fprime=
   bessel_determinant_prime)
```

Listing 1: 貝索函數行列式定義

重要技術點：

- 使用數值微分計算導數，步長 $h = 10^{-5}$ 提供足夠精度
- 初始猜測值設為 6.0，基於函數圖形分析
- 得到特徵值 $\alpha = 6.1556542986056666$

A.2.2 解析解構建

確定係數比例並建立完整解：

```
A = 1.0
B = -A * jv(n, alpha * 1) / yv(n, alpha * 1)
hat_u = A * jv(n, alpha * R) + B * yv(n, alpha * R)
u = hat_u * np.cos(n * Theta)
```

Listing 2: 解析解構建

B Part II：時間積分方法

B.1 LSRK5 方法實現

Part II 採用 Carpenter-Kennedy 四階二儲存 Runge-Kutta 方法，其主要優勢是記憶體效率高。

B.1.1 係數定義

```
A = np.array([0.0,
              -567301805773.0 / 1357537059087.0,
              -2404267990393.0 / 2016746695238.0,
              -3550918686646.0 / 2091501179385.0,
              -1275806237668.0 / 842570457699.0])

B = np.array([1432997174477.0 / 9575080441755.0,
              5161836677717.0 / 13612068292357.0,
              1720146321549.0 / 2090206949498.0,
              3134564353537.0 / 4481467310338.0,
              2277821191437.0 / 14882151754819.0])
```

Listing 3: LSRK5 係數

B.1.2 時間推進循環

```
for step in range(num_steps):
    t_n = step * dt
    u_stage, v_stage = u.copy(), v.copy()
    du, dv = np.zeros_like(u), np.zeros_like(v)

    for j in range(5):
        t_stage = t_n + C[j] * dt
        du_dt, dv_dt = rhs_system(u_stage, v_stage, R, Theta, t_stage
            ,
                                  n, alpha, omega, c, r_arr, hat_u)
        du = A[j] * du + dt * du_dt
        dv = A[j] * dv + dt * dv_dt
        u_stage += B[j] * du
```

```
        v_stage += B[j] * dv

    u, v = u_stage, v_stage
```

Listing 4: LSRK5 時間推進

重要技術點：

- 使用解析形式的空間導數項進行時間收斂性測試
- 達到理論四階收斂精度（收斂率：4.14, 4.00, 4.00）
- 二儲存結構減少記憶體使用量

C Part III：角度方向的傅立葉偽譜方法

C.1 傅立葉微分矩陣

Part III 的核心是構建傅立葉偽譜微分矩陣處理角度方向的二階導數。

C.1.1 一階微分矩陣構建

```
def construct_D_matrix(N_plus_1):
    D = np.zeros((N_plus_1, N_plus_1))
    for i in range(N_plus_1):
        for j in range(N_plus_1):
            if i != j:
                diff = i - j
                D[i, j] = ((-1)**diff) / (2 * np.tan(np.pi * diff /
                    N_plus_1))
    return D
```

Listing 5: 傅立葉微分矩陣

C.1.2 角度項數值計算

```
def f_theta_numerical(u, R_inv2, D2):
    return R_inv2 @ u @ D2  # 矩陣乘法，形狀 (K, N+1)
```

Listing 6: 角度項數值計算

重要技術點：

- 二階微分矩陣：$D^{(2)} = (D^{(1)})^T (D^{(1)})^T$
- 利用矩陣乘法實現高效的空間導數計算
- 對角矩陣 $R^{-2} = \text{diag}(1/r_k^2)$ 處理徑向權重

C.2 收斂性分析

測試不同角度網格解析度：$N+1 = 64, 96, 128, 160, 192, 224$，觀察傅立葉方法的譜精度特性。

D Part IV：徑向方向的有限差分方法

D.1 徑向導數的數值處理

Part IV 是最複雜的部分，需要處理徑向方向的有限差分離散化。

D.1.1 預計算優化

為提高計算效率，預先計算時間無關的徑向導數係數：

```
def precompute_radial_derivatives(r_arr, Theta, n, alpha):
    K = len(r_arr) - 1
    r_half = np.zeros(K)
    for k in range(K):
        r_half[k] = (r_arr[k] + r_arr[k+1]) / 2

    A_coef = 1.0
    B_coef = -A_coef * jv(n, alpha * 1) / yv(n, alpha * 1)

    radial_coeff = np.zeros(K)
    for k in range(K):
        r_val = r_half[k]
        djv_dr = alpha * (jv(n-1, alpha * r_val) - jv(n+1, alpha *
           r_val)) / 2
        dyv_dr = alpha * (yv(n-1, alpha * r_val) - yv(n+1, alpha *
           r_val)) / 2
        hat_u_r = A_coef * djv_dr + B_coef * dyv_dr
        radial_coeff[k] = r_val * hat_u_r

    return radial_coeff
```

Listing 7: 預計算徑向導數係數

D.1.2 矩陣化徑向計算

```
def f_r_numerical_optimized(u, r_arr, radial_coeff, Theta, t, n,
   omega):
    K = len(r_arr) - 1
```

```
    dr = 1.0 / K

    # 使用預計算係數快速計算半網格點上的 ru_r
    cos_term = np.cos(n * Theta[0, :] - omega * t)
    ru_r_half = radial_coeff[:, np.newaxis] * cos_term[np.newaxis, :]

    # 向量化計算內部點的徑向項
    r_k = r_arr[1:K].reshape(-1, 1)
    numerator = ru_r_half[1:K, :] - ru_r_half[0:K-1, :]
    f_r_inner = numerator / (r_k * dr)

    f_r = np.zeros_like(u)
    f_r[1:K, :] = f_r_inner
    f_r[0, :] = 0.0   # 邊界條件
    f_r[K, :] = 0.0   # 邊界條件

    return f_r
```

Listing 8: 矩陣化徑向項計算

D.1.3 邊界條件處理

```
def enforce_boundary_conditions(u, v):
    u[0, :] = 0.0   # r=1 邊界
    u[-1, :] = 0.0  # r=2 邊界
    v[0, :] = 0.0   # r=1 邊界
    v[-1, :] = 0.0  # r=2 邊界
```

Listing 9: 邊界條件強加

重要技術點：

- 使用半網格點計算徑向導數，提高數值穩定性
- 貝索函數導數遞推關係：$J_n'(x) = \frac{1}{2}[J_{n-1}(x) - J_{n+1}(x)]$
- 預計算與矩陣化操作大幅提升計算效率
- 每個時間步都強加 Dirichlet 邊界條件