AM205 Midterm Writeup

Jiawen Tong

November 2017

Problem 1

(a)

Constructed the polynomial using monomial basis and computed the coefficients by solving the Vandermonde matrix.

The coefficients of the interpolant $\Gamma(x) \sim g(x) = \Sigma_{k=0}^{4} g_k x^k$ are:

$$g_0 = 9$$

$$g_1 = -16.583$$

$$g_2 = 11.625$$

$$g_3 = -3.417$$

$$g_4 = 0.375$$

$$g(x) = 0.375x^4 - 3.417x^3 + 11.625x^2 - 16.583x + 9$$

(b)

Convert function values of $\Gamma(x) = 1, 1, 2, 6, 24$ into log scale by **np.log()** and apply the same procedure as in (a). The coefficients of the polynomial $p(x)$ fitted on **log scale gamma** $\log(\Gamma(x))$ for x^4, x^3, x^2, x and the intercept are: 0.007, -0.119, 0.882, -1.921, 1.151. Finally, reconstruct the fitted $\Gamma(x) \sim h(x) = e^{p(x)}$ values by **np.exp()**.

(c)

I used `scipy.special.gamma` to compute the raw function $\Gamma(x)$. See plot of the three functions on the interval $1 \leq x \leq 5$ in Figure 1.

1. Direct polynomial fitting: $g(x)$

2. Polynomial fitted on log scale and converted to exponential: $h(x) = e^{p(x)}$

3. Python's `scipy.special.gamma`: $\Gamma(x)$

(d)

By definition **Relative Error = Absolute Error / True Value**,

maximum of $\frac{|\Gamma(x) - g(x)|}{\Gamma(x)} = \mathbf{0.285726}$,

maximum of $\frac{|\Gamma(x) - h(x)|}{\Gamma(x)} = \mathbf{0.011516}$.

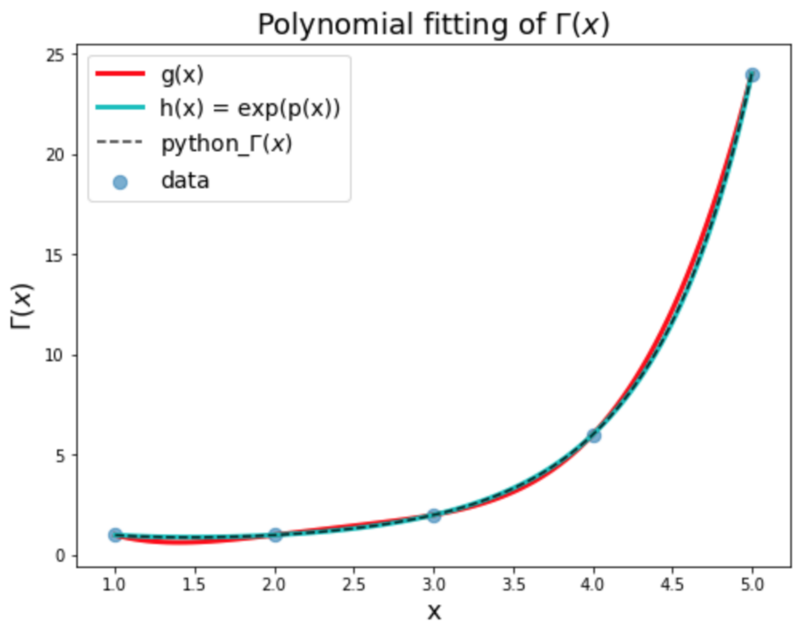

Figure 1: Problem 1: (c)

Since the maximum relative error between $\Gamma(x)$ and $h(x)$ is much smaller than that between $\Gamma(x)$ and $g(x)$, $h(x)$ is a more accurate approximation. In addition, we can also tell from the plot in Figure 1 that $h(x)$ is almost the same as $\Gamma(x)$, while $g(x)$ is a little bit off.

Problem 2

(a)

Given function values at $x = 0$, 1, ..., 8, first and second derivatives continuous at internal points and the natural end points condition $s_k^{''}(0) = 0$, $s_k^{''}(8) = 0$, we can solve the coefficients of the cubic basis polynomials exactly. I used the 4 basis cubics in the lecture note:

$$c_0(x) = x^2(3 - 2x)$$

$$c_1(x) = -x^2(1 - x)$$

$$c_2(x) = (x - 1)^2 x$$

$$c_3(x) = 2x^3 - 3x^2 + 1$$

To solve the coefficients of these basis for a spline, we can firstly write out the conditions and arrange the unknowns in a matrix form. The solution of these linear systems are the coefficients of these basis functions for each spline for all subintervals [0,1], [1,2], ..., [7, 8].

For example, by $s_0(0) = 1$, $s_0(i) = 0$ (i=1,2,...,8) and first derivatives are continuous at internal points, we have

$$0 \le x \le 1, \quad s_0(x) = c_3(x) + ac_1(x) + ic_2(x)$$

$$\begin{aligned}
1 \le x \le 2, &\quad s_0(x) = bc_1(x) + ac_2(x) \\
2 \le x \le 3, &\quad s_0(x) = cc_1(x) + bc_2(x) \\
3 \le x \le 4, &\quad s_0(x) = dc_1(x) + cc_2(x) \\
4 \le x \le 5, &\quad s_0(x) = ec_1(x) + dc_2(x) \\
5 \le x \le 6, &\quad s_0(x) = fc_1(x) + ec_2(x) \\
6 \le x \le 7, &\quad s_0(x) = gc_1(x) + fc_2(x) \\
7 \le x \le 8, &\quad s_0(x) = hc_1(x) + gc_2(x)
\end{aligned}$$

Apply to these equations that second derivatives are continuous at internal points and zero the 2 end points, we will get

$$6 + 4a + 2i = -2b - 4a$$

$$\begin{aligned}
4b + 2a &= -2c - 4b \\
4c + 2b &= -2d - 4c \\
4d + 2c &= -2e - 4d \\
4e + 2d &= -2f - 4e \\
4f + 2e &= -2g - 4f \\
4g + 2f &= -2h - 4g \\
f^{''}(0) &= -6 - 2a - 4i = 0 \\
f^{''}(8) &= 4h + 2g = 0
\end{aligned}$$

Rearrange the above 9 unknowns a, b, ..., i to a matrix form:

$$\begin{bmatrix}
8 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 8 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 & 8 & 2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 8 & 2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2 & 8 & 2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & 8 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 & 8 & 2 & 0 \\
-2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & 4 & 0
\end{bmatrix}
\begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \end{bmatrix}
=
\begin{bmatrix} -6 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 6 \\ 0 \end{bmatrix}$$

Repeat this procedure for $s_1(x)$, $s_2(x)$, ..., $s_8(x)$, we will get all coefficients of the 9 splines. Then, we can find the linear combination coefficients $b = (b_0, b_1, ..., b_8)$ by linear least square fitting of the 9 splines on all data points (x_i, y_i) for i=1,...,N.

$$\begin{bmatrix} s_0(x_1) & s_1(x_1) & \dots & s_8(x_1) \\ s_0(x_2) & s_1(x_2) & \dots & s_8(x_2) \\ \vdots & \vdots & \vdots & \vdots \\ s_0(x_N) & s_1(x_N) & \dots & s_8(x_N) \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_8 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

Using data from `sdata1.txt`, the fitted values of b are

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ b_8 \end{bmatrix} = \begin{bmatrix} 0.51856191 \\ -0.02703424 \\ -0.87616979 \\ -0.34414149 \\ 0.60963002 \\ -0.2735107 \\ -1.63220254 \\ -2.94993774 \\ -1.50771464 \end{bmatrix}$$

See plot of the points from `sdata1.txt` and the fitted spline in Figure 2.

(b)

Using data from `sdata2.txt`, the fitted values of b are

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ b_8 \end{bmatrix} = \begin{bmatrix} -0.49281047 \\ -1.67616518 \\ -1.50517194 \\ -1.07335582 \\ -1.33135505 \\ -0.25321862 \\ -1.26349519 \\ -1.22293233 \\ -1.5130342 \end{bmatrix}$$

See plot of the points from `sdata2.txt` and the fitted spline in Figure 3.

(c)

The plot of `sdata1.txt` looks reasonable while that of `sdata2.txt` does not. This could be explained by the discontinuity of data points in `sdata2.txt`. Since splines require 1-st and 2-nd derivatives being continuous at internal points, they model smooth functions better. We can see in Figure 3, the spline is trying to fit the linearly spaced points to be continuous by using the discontinuous data points from `sdata2.txt`.

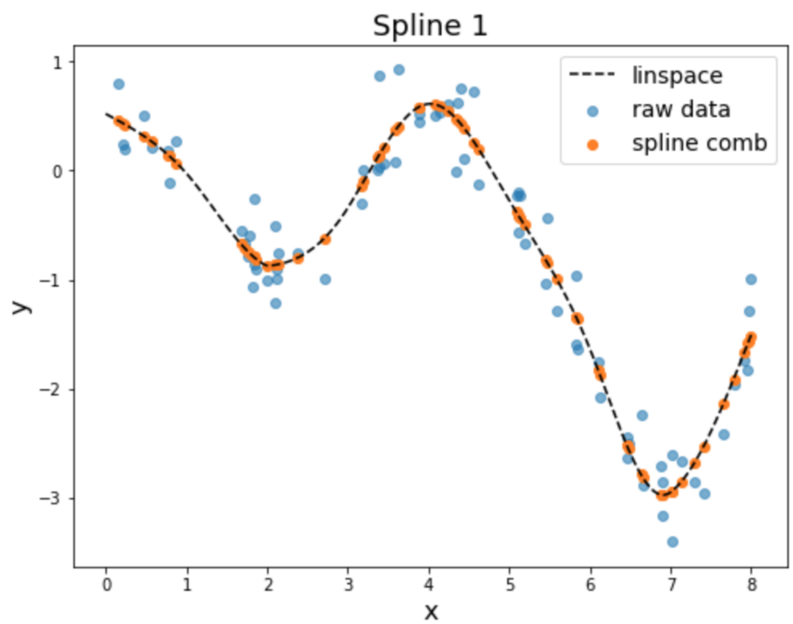

Figure 2: Problem 2: (a)

Problem 3

(a)

Expanding the $f(x+h)$ and $f(x-h)$ by Taylor series, we have:

$$f(x+h) = f(x) + hf^{'}(x) + \frac{h^2}{2}f^{''}(x) + \frac{h^3}{6}f^{(3)}(x) + O(h^4)f^{(4)}(x)$$

$$f(x-h) = f(x) - hf^{'}(x) + \frac{h^2}{2}f^{''}(x) - \frac{h^3}{6}f^{(3)}(x) + O(h^4)f^{(4)}(x)$$

$$f(x+h) + f(x-h) = 2f(x) + h^2 f^{''}(x) + O(h^4)f^{(4)}(x)$$

Similarly, expanding $f(x+2h)$ and $f(x-2h)$ by Taylor series, we have:

$$f(x+2h) + f(x-2h) = 2f(x) + (2h)^2 f^{''}(x) + O(h^4)f^{(4)}(x)$$

Plug in these results into the finite difference formula:

$$f^{''}_{diff}(x) = \frac{[a + 2b(a) + 2c(a)]f(x) + [b(a) + 4c(a)]h^2 f^{''}(x) + O(h^4)f^{(4)}(x)}{h^2}$$

To make $f^{''}_{diff}(x) \sim f^{''}(x)$, we set the coefficients of $f(x)$ to 0 and the coefficients of $f^{''}(x)$ to 1:

$$\begin{cases} a + 2b(a) + 2c(a) = 0 \\ b(a) + 4c(a) = 1 \end{cases}$$

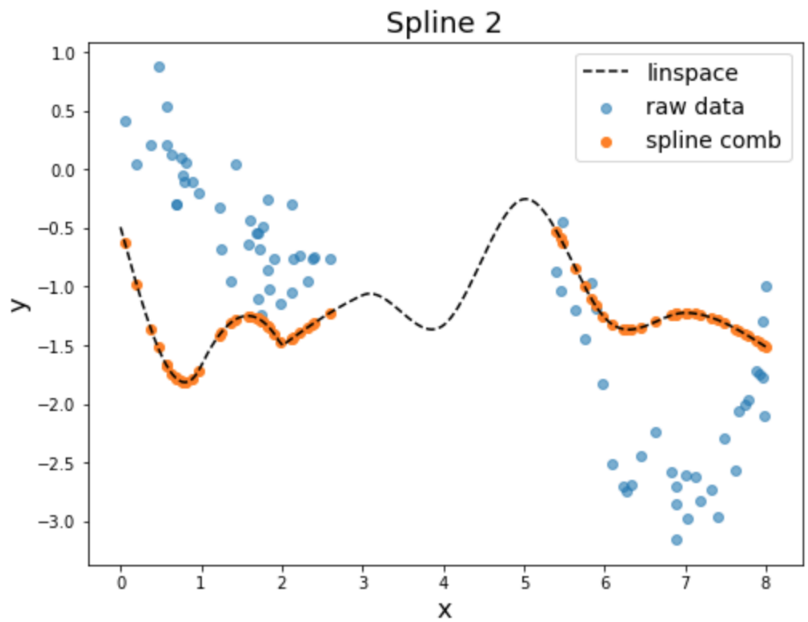

Figure 3: Problem 2: (b)

$$\begin{cases} b(a) = -\frac{2}{3}a - \frac{1}{3} \\ c(a) = \frac{1}{6}a + \frac{1}{3} \end{cases}$$

Then, we can verify the truncation error of this formula:

$$f^{''}_{diff}(x) = f^{''}(x) + O(h^2)f^{(4)}(x)$$

$$T = f^{''}(x) - f^{''}_{diff}(x) = O(h^2)f^{(4)}(x)$$

Since we have h to the **2nd order** in the truncation error, this scheme is upto $O(h^2)$ **(second-order)** accurate.

(b)

Stability analysis:

$$U_j^{n+1} - U_j^n = \frac{\Delta t}{h^2}[aU_j^b + b(U_{j+1}^n + U_{j-1}^n) + c(U_{j+1}^n + U_{j-2}^n)]$$

$$= \mu[aU_j^b + (-\frac{2}{3}a - \frac{1}{3})(U_{j+1}^n + U_{j-1}^n) + (\frac{1}{6}a + \frac{1}{3})(U_{j+1}^n + U_{j-2}^n)]$$

$$U_j^{n+1} = (1 + \mu a)U_j^n + \mu(-\frac{2}{3}a - \frac{1}{3})(U_{j+1}^n + U_{j-1}^n) + \mu(\frac{1}{6}a + \frac{1}{3})(U_{j+1}^n + U_{j-2}^n)$$

Simplify this formula by:

$$\begin{cases} U_j^n(k) = [\lambda(k)]^n e^{ikjh} \\ e^{ikh} = cos(kh) + i * sin(kh) \\ e^{-ikh} = cos(kh) - i * sin(kh) \\ cos(kh) = 1 - 2sin^2\frac{kh}{2} = 1 - 2s \\ cos(2kh) = 2cos^2(kh) - 1 = 2(1-2s)^2 - 1 = 8s^2 - 8s + 1 \end{cases}$$

$$\lambda(k) = (1 + \mu a) + \mu(-\frac{2}{3}a - \frac{1}{3})(e^{ikh} + e^{-ikh}) + \mu(\frac{1}{6}a + \frac{1}{3})(e^{2ikh} + e^{-2ikh})$$

$$\lambda(k) = (1 + \mu a) + \mu(-\frac{2}{3}a - \frac{1}{3})2cos(kh) + \mu(\frac{1}{6}a + \frac{1}{3}) + 2cos(2kh)$$

$$\lambda(k) = 1 - 4\mu s + \frac{8}{3}\mu s^2(a + 2)$$

This gives the amplification factor $\lambda(k)$ in terms of $\mu, a, s = sin^2\frac{kh}{2}$

(c)

$$a = -2$$

$$b = -\frac{2}{3}a - \frac{1}{3} = 1$$

$$c = \frac{1}{6}a + \frac{1}{3} = 0$$

Therefore,

$$f_{diff}^{''}(x) = \frac{f(x-h) - 2f(x) + f(x+h)}{h^2}$$

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \frac{U_{j-1}^n - 2U_j^n + U_{j+1}^n}{h^2}$$

This is the same with the forward Euler discretization for heat equation discussed in `lecture 16`.

$$\lambda(k) = 1 - 4\mu s + \frac{8}{3}\mu s^2(a + 2) = 1 - 4\mu s$$

Setting $\theta = 0$ in the stability analysis of the θ-method in `lecture 16` gives

$$\lambda(k) = \frac{1 - 4(1-\theta)\mu s}{1 + 4\theta\mu s} = 1 - 4\mu s$$

(d)

The method is always unstable when $|\lambda(k)| > 1$ can hold for some s regardless of the value of μ. When $\lambda(k) > 1$, we will have:

$$1 - 4\mu s + \frac{8}{3}\mu s^2(a + 2) > 1$$

$$-4\mu s + \frac{8}{3}\mu s^2(a+2) > 0$$

$$\frac{2}{3}s(a+2) > 1$$

$$a+2 > \frac{3}{2s}$$

If $a+2 > min(\frac{3}{2s})$, this method is always unstable regardless of μ. That is,

$$s = 1, a+2 > min(\frac{3}{2s}) = \frac{3}{2}$$

$$a > -\frac{1}{2}$$

$$a_* = -\frac{1}{2}$$

(e)

When $a \leq -\frac{1}{2}$, by analyzing λ as a function of $s = sin^2\frac{kh}{2}$, we can find the range of μ as a function of a.

- **Case 1: a + 2 = 0, a = -2**

$$\lambda = -4\mu s + 1$$

$$-1 \leq -4\mu s + 1 \leq 1$$

$$-2 \leq -4\mu s \leq 0$$

$$0 \leq \mu s \leq \frac{1}{2}$$

$$\mu \leq min(\frac{1}{2s}) = \frac{1}{2}$$

- **Case 2: a + 2 < 0, a < -2**

$$\lambda(s) = \frac{8}{3}\mu(a+2)s^2 - 4\mu s + 1$$

is a parabolic function on the interval $0 \leq s \leq 1$

$$\lambda_{max} = \lambda(s=0) = 1 \in [-1, 1]$$

$$\lambda_{min} = \lambda(s=1) = \frac{8}{3}\mu(a+2) - 4\mu + 1$$

To satisfy $|\lambda_{min}| \leq 1$, we set $-1 \leq \lambda_{min} \leq 1$, which gives

$$\mu \leq \frac{-3}{2(2a+1)}$$

- **Case 3: a + 2 > 0, -2 < a ≤ -1/2**

  If $\frac{3}{4(a+2)} < 1$, $a > \frac{5}{4}$,

$$\lambda_{max} = \lambda(s=0) = 1 \in [-1, 1]$$

$$\lambda_{min} = \lambda(s = \frac{3}{4(a+2)}) = \frac{a+2-\frac{3}{2}\mu}{a+2}$$

  Similarly, setting $-1 \leq \lambda_{min} \leq 1$ will give

$$\mu \leq \frac{4a+8}{3}$$

  If $\frac{3}{4(a+2)} \geq 1$, $a \leq \frac{5}{4}$,

$$\lambda_{max} = \lambda(s=0) = 1 \in [-1, 1]$$

$$\lambda_{min} = \lambda(s=1) = \frac{8}{3}\mu(a+2) - 4\mu + 1$$

  Similarly, setting $-1 \leq \lambda_{min} \leq 1$ will give

$$\mu \leq \frac{-3}{4a+2}$$

In conclusion,

$$\begin{cases} \mu \leq \frac{-3}{4a+2} & (a \leq -\frac{5}{4}) \\ \mu \leq \frac{4a+8}{3} & (-\frac{5}{4} < a \leq -\frac{1}{2}) \end{cases}$$

(f)

We can see from the plot in Figure 4 that the maximum value of μ for stability grows as a increases. Hence, setting $a = a_* = -\frac{1}{2}$ allows for the largest stable value of $\mu = \frac{\Delta t}{h^2} = 2$ to be chosen.

When $a = -\frac{1}{2}$, $b = 0$, $c = \frac{1}{4}$,

$$U_j^{n+1} - U_j^n = 2[-\frac{1}{2}U_j^n + \frac{1}{4}(U_{j+2}^n + U_{j-2}^n)]$$

$$U_j^{n+1} = \frac{1}{2}(U_{j+2}^n + U_{j-2}^n)$$

We can see that the 1-st derivative terms are cancelled out. Each update of U_j^{n+1} would use 2 steps in space as 1 while the time-step restriction is strict: $\Delta t \leq 2h^2$. Given the initial condition $U_j^0 = (-1)^j$, the odd points are always -1 while the even points are always 0. Thus, the discontinuity in the initial conditions cannot be smoothed out by this scheme.

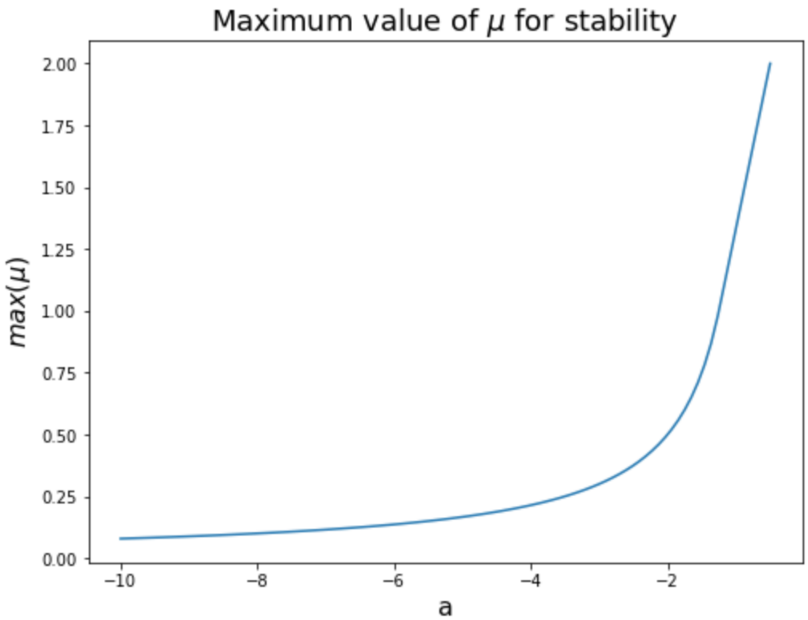

Figure 4: Problem 3: (f)

Problem 4

(a)

Expand at (jh, kh) using Taylor series, we will have

$$(\nabla^2 v)_{jk} \sim \alpha V(jh,kh) + 4\beta V(jk,kh) + 5\beta h^2 \frac{\partial^2 V}{\partial x^2}(jk,kh) + \beta O(h^4)\frac{\partial^4 V}{\partial x^4}(jh,kh)$$

$$+4\beta V(jk,kh) + 5\beta h^2 \frac{\partial^2 V}{\partial y^2}(jk,kh) + \beta O(h^4)\frac{\partial^4 V}{\partial y^4}(jh,kh)$$

$$= (\alpha+8\beta)V(jk,kh)+5\beta h^2(\frac{\partial^2 V}{\partial x^2}+\frac{\partial^2 V}{\partial y^2})(jk,kh)+\beta O(h^4)\frac{\partial^4 V}{\partial x^4}(jk,kh)+\beta O(h^4)\frac{\partial^4 V}{\partial y^4}(jk,kh)$$

$$\begin{cases} \alpha + 8\beta = 0 \\ 5\beta = \frac{1}{h^2} \end{cases}$$

$$\begin{cases} \alpha = -\frac{8}{5h^2} \\ \beta = \frac{1}{5h^2} \end{cases}$$

We can verify the truncation error to be **2-nd order accurate**:

$$T_{jk} = (\nabla^2 v)_{jk} - (\frac{\partial^2 V}{\partial x^2}+\frac{\partial^2 V}{\partial y^2})(jk,kh) = O(h^2)\frac{\partial^4 V}{\partial x^4}(jk,kh) + O(h^2)\frac{\partial^4 V}{\partial y^4}(jk,kh)$$

(b)
Boundary condition: $V_{j+1,k} = 0$
Apply ghost node approach at exterior points: $V_{j+2,k} = -V_{j,k}$
Then, we have

$$(\nabla^2 v)_{jk} = (\alpha - \beta)V_{j,k} + \beta(V_{j-2,k} + V_{j-1,k} + V_{j,k-2} + V_{j,k-1} + V_{j,k+1} + V_{j,k+2})$$

(c)

$$-\nabla^2 v + v = f$$

$$(1-\alpha)V_{j,k} - \beta(V_{j-2,k} + V_{j-1,k} + V_{j+1,k} + V_{j+2,k} + V_{j,k-2} + V_{j,k-1} + V_{j,k+1} + V_{j,k+2}) = F_{j,k}$$

Discretized as above, by using the scheme in part (a) and handling missing points as in part (b), we can construct the matrix A by looping through the interior points of the region Ω. See the plot of the numerical solution for $V_{j,k}$ over the region Ω in Figure 5.

Conversion between 2D and 1D indexing is done using dictionary loop-up. Each interior point in the region Ω has a unique 1D index, determined by the looping sequence over the M by M grid.

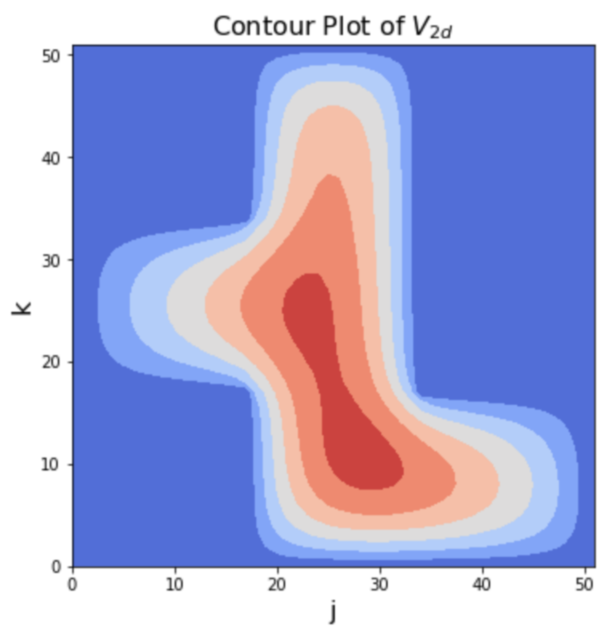

Figure 5: Problem 4: (c)

(d)
Maximum of $V_{j,k}$ = **0.5147104874225262**. At **j = 28, k = 11**