1. (a)

For ridge regression: $\beta_{ridge} = (Z^TZ + \lambda I)^{-1} Z^T y \qquad (\lambda \geq 0)$

$$\|\beta_{ridge}\|_2^2 = \beta_{ridge}^T \beta_{ridge}$$

$$= y^T Z (Z^TZ + \lambda I)^{-2} Z^T y$$

$$\frac{\partial \|\beta_{ridge}\|_2^2}{\partial \lambda} = -2 I \left[y^T Z (Z^TZ + \lambda I)^{-3}) Z^T y \right]$$

Since $Z^TZ + \lambda I \succ 0$, $\frac{\partial \|\beta_{ridge}\|_2^2}{\partial \lambda}$ is negative, which means

$\|\beta_{ridge}\|_2$ increases as $\lambda \to 0$.

(b) For Lasso regression: $\hat{\beta}_{lasso} = \underset{\beta}{\operatorname{argmin}} \{ \|Y - X\beta\|_2^2 + \lambda \|\beta\|_1 \}$

Let $\lambda_1 < \lambda_2$. For λ_1, the solution $\hat{\beta}^{(1)}$ satisfy

$$\|Y - X\hat{\beta}^{(1)}\|_2^2 + \lambda_1 \|\hat{\beta}^{(1)}\|_1 \leq \|Y - X\hat{\beta}^{(2)}\|_2^2 + \lambda_1 \|\hat{\beta}^{(2)}\|_1 \qquad (1)$$

Also for λ_2, the solution $\hat{\beta}^{(2)}$ satisfy

$$\|Y - X\hat{\beta}^{(2)}\|_2^2 + \lambda_2 \|\hat{\beta}^{(2)}\|_1 \leq \|Y - X\hat{\beta}^{(1)}\|_2^2 + \lambda_2 \|\hat{\beta}^{(1)}\|_1 \qquad (2)$$

(1) + (2), then we have

$$\lambda_1 \|\hat{\beta}^{(1)}\|_1 + \lambda_2 \|\hat{\beta}^{(2)}\|_1 \leq \lambda_1 \|\hat{\beta}^{(2)}\|_1 + \lambda_2 \|\hat{\beta}^{(1)}\|_1$$

$$(\lambda_2 - \lambda_1)\left[\|\hat{\beta}^{(2)}\|_1 - \|\hat{\beta}^{(1)}\|_1 \right] \leq 0$$

Since $\lambda_2 - \lambda_1 > 0$, so it is satisfy

$$\|\hat{\beta}^{(2)}\|_1 < \|\hat{\beta}^{(1)}\|_1$$

Thus, when $\lambda \to 0$, $\|\hat{\beta}^{lasso}\|_1$ increases

2.

The solution of Ridge Regression is $\hat{\beta} = (Z^T Z + \lambda I)^{-1} Z^T y$

let $x_i := z$, and the equation of question is

$$\hat{f}(x) = \left(\sum_{i=1}^{n} c_i x_i\right)^T x \quad , \text{define } C = [c_1, c_2, \dots, c_n]^T$$

Set $\left(\sum_{i=1}^{n} c_i x_i\right)^T = \hat{\beta}^T$, $(C^T Z)^T = \hat{\beta}^T$

$$Z^T C = (Z^T Z + \lambda I)^{-1} Z^T y$$

According to the Hint:

$$Z^T C = Z^T (Z Z^T + \lambda I)^{-1} y$$

$$Z Z^T C = Z Z^T (Z Z^T + \lambda I)^{-1} y$$

$$\boxed{C = (Z Z^T + \lambda I)^{-1} y \text{, } c_i \text{ is the ith term in vector } C}$$

Question (a)

The original code as follows:

```
import numpy as np
np.random.seed(123)
def genData(n):
 mu = np.array([-2.0, 3.0])
 A = np.array([[1.0, 2.0], [3.0, 4.0]])
 X = np.random.normal(size = n * 2)
 X = X.reshape((2, n))
 X = (X + mu).dot(A)
 return(X.T)
```

```
genData(10)
```

```
---------------------------------------------------------------------------

ValueError                                Traceback (most recent call last)

Cell In[2], line 1
----> 1 genData(10)


Cell In[1], line 8, in genData(n)
      6 X = np.random.normal(size = n * 2)
      7 X = X.reshape((2, n))
----> 8 X = (X + mu).dot(A)
      9 return(X.T)


ValueError: operands could not be broadcast together with shapes (2,10) (2,)
```

Result:

Theoretically: Return a NumPy array of shape (10, 2) representing 10 samples, each with two features.

Problem:

The shape of X is (2,10), however, the shape of mu is (2,). NumPy cannot directly add arrays of these two shapes. To solve this problem, we need to make sure that the shape of mu is correctly broadcast to X.

New errors:

Although I have fixed one error, it seems that a new problem has arisen. The error message should be that the matrix multiplication dimensions do not match. Next, I will fix this problem under the prompt of this error report.

```
import numpy as np
np.random.seed(123)
def genData(n):
 mu = np.array([-2.0, 3.0]).reshape(2,1) # Reshape mu into the shape of (2,1)
 A = np.array([[1.0, 2.0], [3.0, 4.0]])
 X = np.random.normal(size = n * 2)
 X = X.reshape((2, n))
 X = (X + mu).dot(A)
 return(X.T)
```

```
genData(10)
```

```
---------------------------------------------------------------------------

ValueError                                Traceback (most recent call last)

Cell In[4], line 1
----> 1 genData(10)


Cell In[3], line 8, in genData(n)
      6 X = np.random.normal(size = n * 2)
      7 X = X.reshape((2, n))
----> 8 X = (X + mu).dot(A)
      9 return(X.T)


ValueError: shapes (2,10) and (2,2) not aligned: 10 (dim 1) != 2 (dim 0)
```

```
import numpy as np
np.random.seed(123)
def genData(n):
 mu = np.array([-2.0, 3.0])
 A = np.array([[1.0, 2.0], [3.0, 4.0]])
 X = np.random.normal(size = n * 2)
 X = X.reshape((2, n))
 X = A @ X + mu.reshape(2, 1) # Reshape mu into the shape of (2,1) and fix the
problem of wrong dimensions
 return(X.T)
```

```
np.random.seed(123)
genData(10)
```

```
array([[-4.44340291, -2.97243642],
       [-1.19207249,  5.61320046],
       [ 1.26575775,  9.814494  ],
       [-4.78409871, -4.07449213],
       [-3.46656417, -0.51172859],
       [-1.21726601,  6.21690451],
       [-0.01481908,  4.5436826 ],
       [ 1.94465955, 10.46040647],
       [ 1.27404405, 10.81402437],
       [-2.0943676 ,  1.94452439]])
```

The main changes:

code line "X = (X + mu).dot(A)" change to "X = A @ X + mu.reshape(2, 1)"

Question (b)

The result of x1 and x2

x1 array([-4.44340291, -1.19207249, 1.26575775, -4.78409871, -3.46656417, -1.21726601, -0.01481908, 1.94465955, 1.27404405, -2.0943676])
x2: array([-2.97243642, 5.61320046, 9.814494 , -4.07449213, -0.51172859, 6.21690451, 4.5436826 , 10.46040647, 10.81402437, 1.94452439])

Question (c)

```
np.random.seed(123)
X = genData(10000)

mu_hat = np.mean(X, axis=0)
center = X - mu_hat
sigma_hat = (center.T @ center) / (X.shape[0] - 1)

print("Sample mean: ", np.round(mu_hat, 4))
print("Sample covariance matrix: ", np.round(sigma_hat, 4))
```

```
Sample mean:  [-1.9695  3.0706]
Sample covariance matrix:  [[ 5.0391 11.0854]
```

```
 [11.0854 25.1818]]
```

Question (d)

```
np.random.seed(123)
Y = genData(1000)
Y_sq = np.sum(Y**2, axis=1)
dist = Y_sq[:, None] + Y_sq[None, :] - 2 * Y @ Y.T
K = np.exp(-dist)

print(K)
```

```
[[1.00000000e+00 5.75702727e-68 5.73668714e-52 ... 5.15168023e-03
  2.17360147e-01 1.54036900e-05]
 [5.75702727e-68 1.00000000e+00 7.84664864e-02 ... 1.81480964e-45
  4.59198152e-57 6.51844702e-37]
 [5.73668714e-52 7.84664864e-02 1.00000000e+00 ... 1.32160916e-32
  1.20647538e-42 1.97968563e-25]
 ...
 [5.15168023e-03 1.81480964e-45 1.32160916e-32 ... 1.00000000e+00
  1.35123889e-01 3.39922950e-01]
 [2.17360147e-01 4.59198152e-57 1.20647538e-42 ... 1.35123889e-01
  1.00000000e+00 2.94659118e-03]
 [1.54036900e-05 6.51844702e-37 1.97968563e-25 ... 3.39922950e-01
  2.94659118e-03 1.00000000e+00]]
```

```
eigenvalues = np.linalg.eigvals(K)
largest_3 = np.sort(eigenvalues)[-3:][::-1]
print("The largest 3 eigenvalues are: ", np.round(largest_3, 4))
```

```
The largest 3 eigenvalues are:  [105.5731+0.j  91.1879+0.j  78.3936+0.j]
```

4.

(a) let $f(x)=(x-1)^2$ $g(x)=\lambda|x|$. Both $f(x)$, $g(x)$ are convex.

Hence, $(x-1)^2+\lambda|x|$ is convex, $x^*(\lambda)$ is global minimizer.

For any $x\neq x^*(\lambda)$, $(x^*(\lambda)-1)^2+\lambda|x^*(\lambda)|<(x-1)^2+\lambda|x|$ holds

(b)

For $x>0$, we have $h(x)=(x-1)^2+\lambda x$.

$h'(x)=2(x-1)+\lambda=0$, $x=\frac{2-\lambda}{2}\geq 0$, $\lambda<2$

For $x<0$, we have $h(x)=(x-1)^2-\lambda x$

$h'(x)=2(x-1)-\lambda=0$, $x=\frac{2+\lambda}{2}<0$, $\lambda<-2$ which has a contradiction with $\lambda>0$. omit it

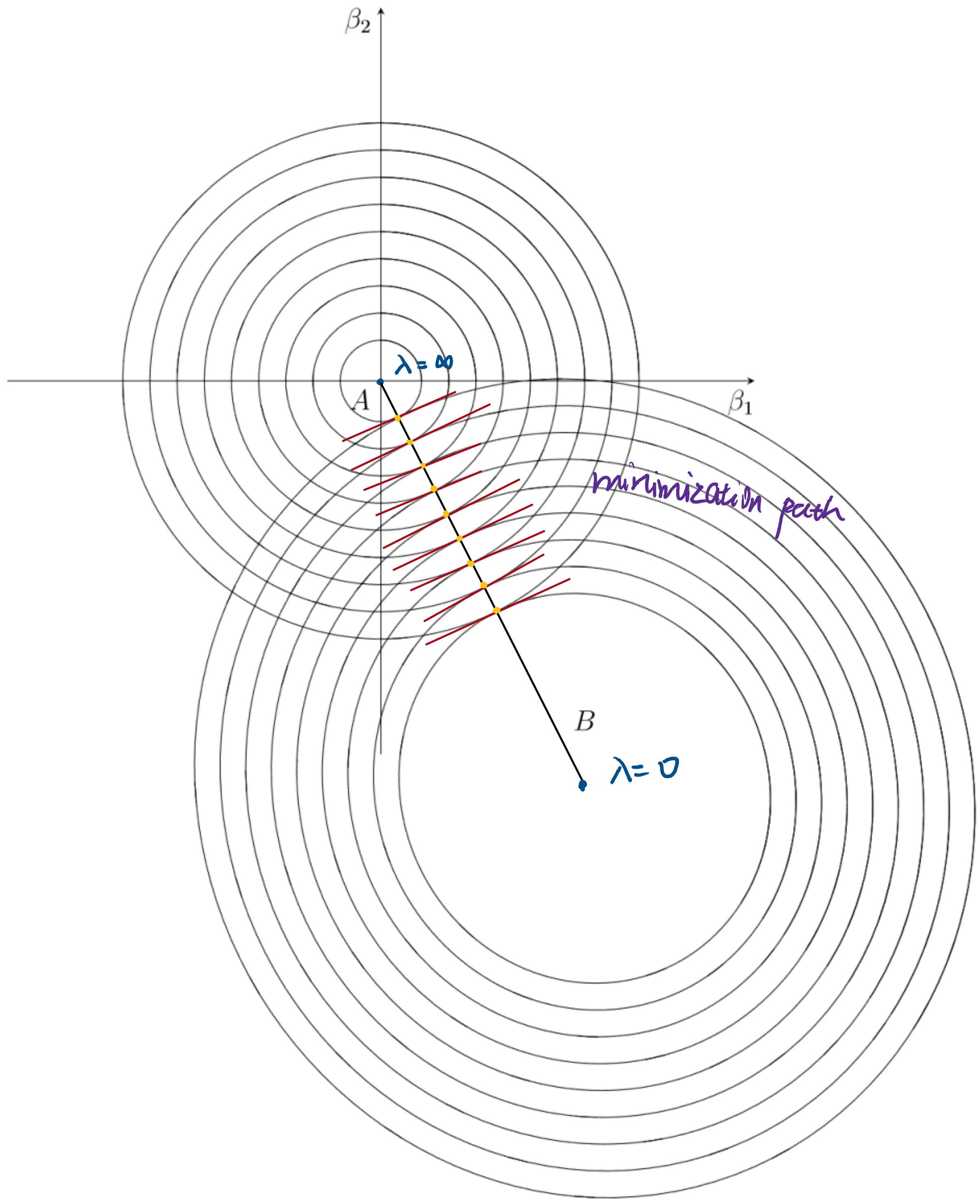
For $x=0$ $\lim\limits_{x\to 0^+}h'(x)=-2+\lambda$, $\lim\limits_{x\to 0^-}h'(x)=-2-\lambda<0$

If $x=0$ is the global minimizer, then have $\lim\limits_{x\to 0^+}h'(x)=-2+\lambda\geq 0$

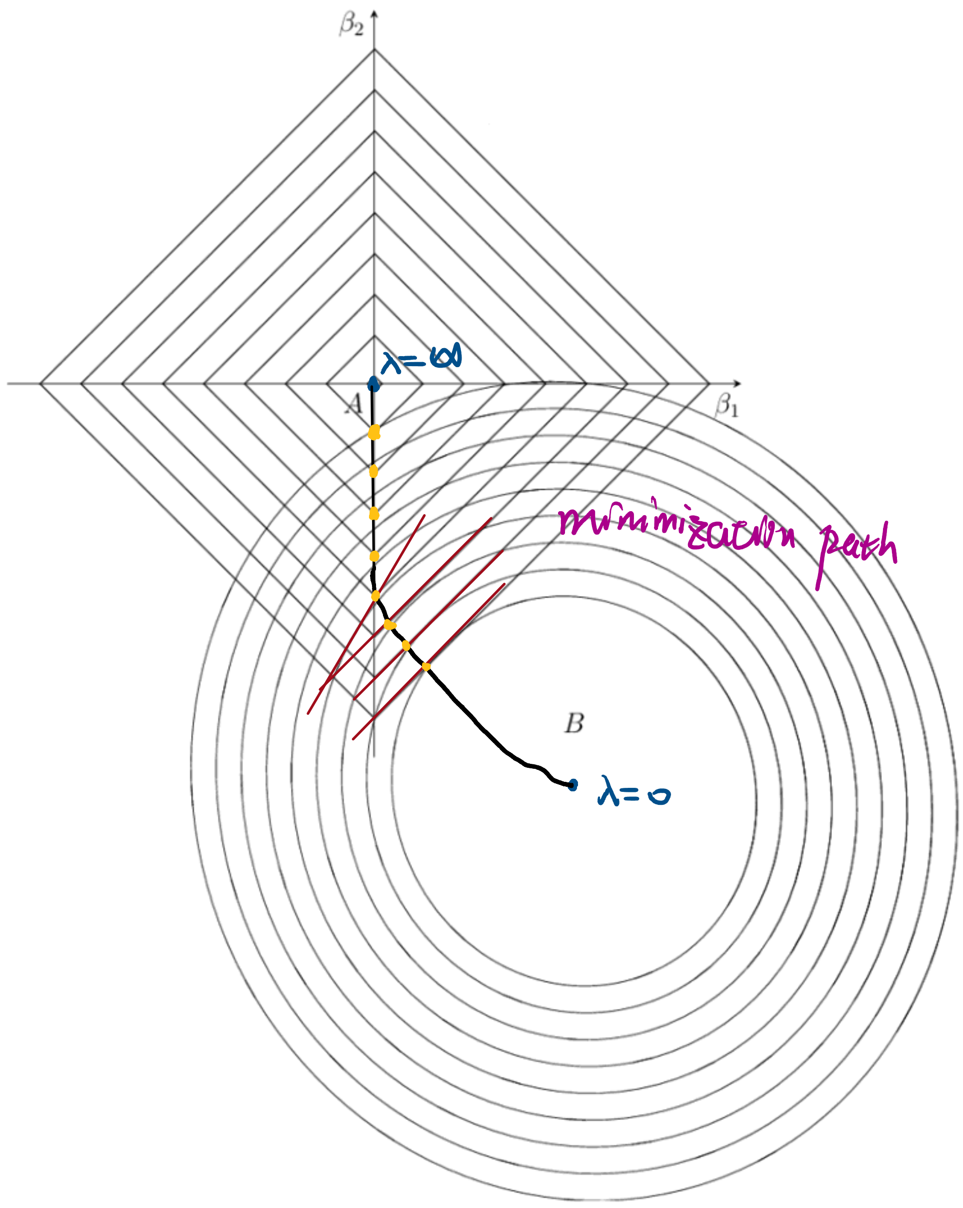
$\lambda\geq 2$, $\lim\limits_{x\to 0^-}h'(x)=-2-\lambda<0$ always holds

$$\boxed{\begin{array}{l}\text{In conclusion:}\\ \text{① If } 0<\lambda<2,\ x^*(\lambda)=\frac{2-\lambda}{2}\\ \text{② If } \lambda\geq 2,\ x^*(\lambda)=0\end{array}}$$

5.



β_2
β_1
$\lambda=\infty$
A
minimization path
B
$\lambda=0$



β_2
β_1
$\lambda=\infty$
A
minimization path
B
$\lambda=0$

6.

(a) Let $K(x,y) = x^3y^3 + xy := K_1(x,y) + K_2(x,y)$

For $K_1(x,y)$, let $x_1 \cdots x_n, C_1 \cdots C_n \in R$. We have

obviously, $K_1(x,y) = K_1(y,x)$. For any positive integer n, let

$x_1 \cdots x_n, C_1 \cdots C_n \in R$, we have

$$\boxed{\sum_{i=1}^{n}\sum_{j=1}^{n} C_i C_j x_i^3 x_j^3 = \left[\sum_{i=1}^{n} C_i x_i^3\right]^2 \geq 0}$$, $K_1(x,y)$ is a semi-positive kernel.

For $K_2(x,y)$, Similarly, $K_2(x,y) = K_2(y,x)$

$$\boxed{\sum_{i=1}^{n}\sum_{j=1}^{n} C_i C_j xy = \left[\sum_{i=1}^{n} C_i x_i\right]^2 \geq 0,}$$ $K_2(x,y)$ is a semi-positive kernel

$K(x,y) = K_1(x,y) + K_2(x,y)$ is a semi-positive kernel

(b) Let $G_1(x,y) = \langle x,y\rangle^3$, $G_2(x,y) = \langle x,y\rangle$ let $x_1 \cdots x_n, C_1 \cdots C_n \in R$.

Consider $G_2(x,y)$. $G_2(x,y) = G_2(y,x)$

$$\boxed{\sum_{i=1}^{n}\sum_{j=1}^{n} C_i C_j \langle x_i, x_j\rangle = \left\langle \sum_{i=1}^{n} C_i x_i, \sum_{j=1}^{n} C_j x_j \right\rangle = \left\|\sum_{i=1}^{n} C_i x_i\right\|^2 \geq 0}$$

$G_2(x,y)$ is a semi-positive kernel

Thus, the Gram matrix G_2 of $G_2(x,y)$ is semi-positive.

Suppose the $G_1(x,y)$'s Gram matrix is G_1.

$$\boxed{G_1 = G_2 \circ G_2 \circ G_2 \text{ is also semi-positive (Schur theorem)}}$$

where $\circ$ is Hadamard product. $G_1(x,y)$ is a semi-positive kernel

Hence, $G(x,y) = G_1(x,y) + G_2(x,y)$ is a semi-positive kernel

(C) In $(-1,1)$, using Geometric series expansion $\frac{1}{1-xy} = \sum_{k=0}^{\infty} (xy)^k$.

$k(x,y) = k(y,x)$. $k(x,y) = \frac{1+xy}{1-xy} = (1+xy)\sum_{k=0}^{\infty}(xy)^k = \sum_{k=0}^{\infty}(xy)^k + \sum_{k=0}^{\infty}(xy)^{k+1}$

For any positive integer n, let $x_1 \cdots x_n \in (-1,1)$, $C_1 \cdots C_n \in R$

$$\boxed{\begin{aligned}\sum_{i=1}^{n}\sum_{j=1}^{n} C_i C_j \left(\sum_{k=0}^{\infty}(x_i x_j)^k + \sum_{k=0}^{\infty}(x_i x_j)^{k+1}\right) &= \sum_{i=1}^{n}\sum_{j=1}^{n} C_i C_j \left(1 + 2\sum_{k=1}^{\infty}(x_i x_j)^k\right) \\ &= \left(\sum_{i=1}^{n} C_i\right)^2 + 2\sum_{k=1}^{\infty}\left(\sum_{i=1}^{n} C_i x_i^k\right)^2 \geq 0\end{aligned}}$$

Hence, $k(x,y)$ is a semi-positive kernel

(d) $k(x,u) = \cos(x-u) = \dfrac{e^{i(x-u)} + e^{-i(x-u)}}{2}$ for all $x \in C$

$k(u,x) = \cos(u-x) = \cos(-(x-u)) = \cos(x-u) = k(x,u)$

For any positive integer n. Let $x_1 \dots x_n \in C$, $c_1 \dots c_n \in R$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j \cdot \frac{e^{i(x_i-x_j)} + e^{-i(x_i-x_j)}}{2} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j e^{i(x_i-x_j)} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j e^{-i(x_i-x_j)}$$

$$= \frac{1}{2}\left(\sum_{i=1}^{n} c_i e^{ix_i}\right)\left(\sum_{j=1}^{n} c_j e^{-ix_j}\right) + \frac{1}{2}\left(\sum_{i=1}^{n} c_i e^{-ix_i}\right)\left(\sum_{j=1}^{n} c_j e^{ix_j}\right)$$

$$= \frac{1}{2}\left(\sum_{i=1}^{n} c_i e^{ix_i}\right)\overline{\left(\sum_{j=1}^{n} c_j e^{ix_j}\right)} + \overline{\frac{1}{2}\left(\sum_{i=1}^{n} c_i e^{ix_i}\right)}\left(\sum_{j=1}^{n} c_j e^{ix_j}\right)$$

$$\boxed{= \left|\sum_{i=1}^{n} c_i e^{ix_i}\right|^2 \geq 0}$$

Hence, $k(u,x)$ is a semi-positive kernel