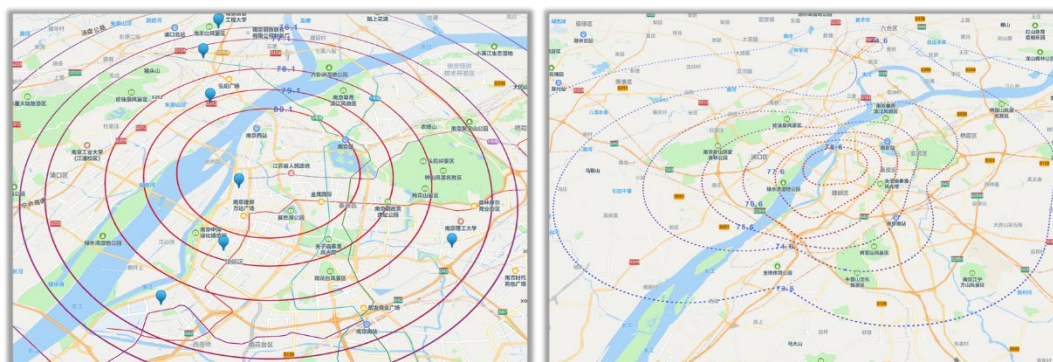


## 一. 可视化系统概述

本可视化系统可以用于显示地图界面，并根据输入测量数据标注出区域场强值，方便用户了解不同地点的场强分布情况，用户可以通过地图界面上的等值线，直观地了解每个位置的场强数值，从而对信号强度有一个清晰的认识。



Figx. 地图场强的两种代表性显示方式

由于用于展示经纬度和对应场强值的数据不连续且可能存在噪声，该可视化系统使用了多种数据预处理方法，然后进行三维或插值和绘制等场强线，用户可以通过该可视化系统实现以下功能：数据导入、数据清理、数据聚合、平滑处理、数据插值、单点场强查询、滑动显示场强等功能。

## 二. 可视化系统框架设计

将收集到的数据输入到可视化系统，由于收集到的数据具有一定的噪声，且在同一位置不同时间测量值会不断变化，系统提取出经纬度和场强大小之后将会做进一步的数据处理。

从数据的场强本身做数据清洗：通过绘制 Quantile-Quantile Plot (Q-Q 图)，我们可以发现数据近似于正态分布，因此根据数据的分布，使用  $3\sigma$  原则，认为数据处于距离均值超过 3 个标准差的值为异常并进行剔除。

从数据的时间空间方向进行数据清洗：由于在收集数据时可能在同一个地点进行了多次采样，不同的天气和温差可能对波的传播造成一定的影响，某点的多个场强值会在绘制过程中造成歧义，此外，若多个测量点距离相近且场强值差异过大，则会造成绘制曲线不平滑，因此我们可以采用 Density-based spatial clustering of applications with noise (DBSCAN)、Meanshift、k-means 等聚类方法，将时空上距离较近的点聚集在一起取平均存入数据库，便于下一步的处理。

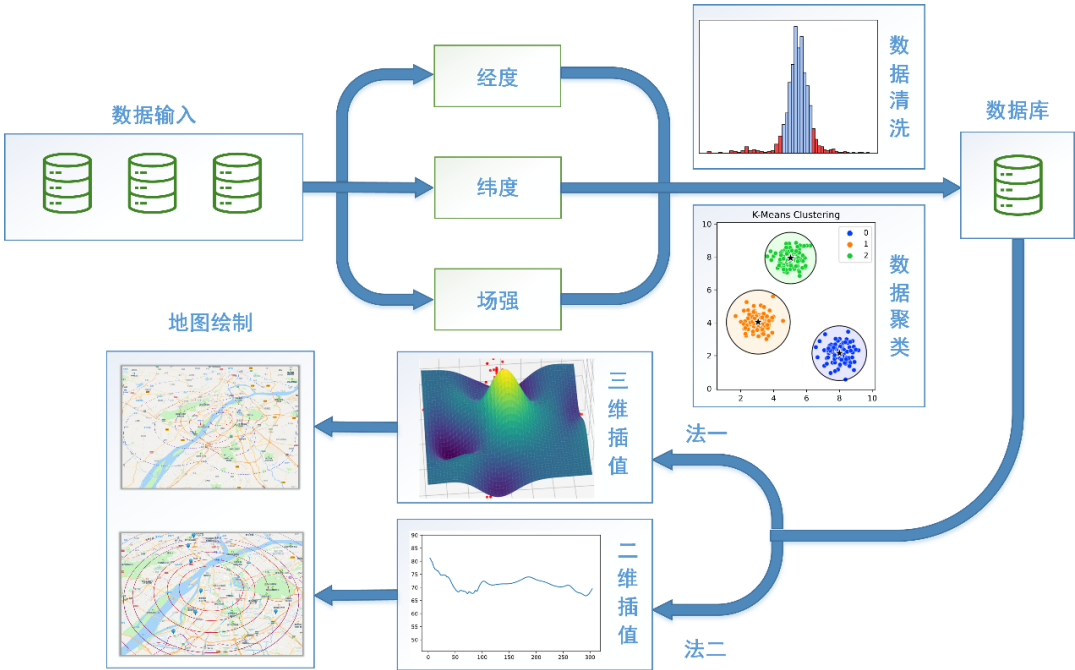
在收集到清洗后的数据之后，我们可以利用 Kriging 插值方法进行三维插值，Kriging 插值方法基于地理空间上的变量之间的空间相关性，通过利用已知数据点的空间分布来估计未知位置上的值，通过 Kriging 插值方法，我们可以利用已有的数据点的空间相关性来推断未知位置上的场强值。这种方法考虑了数据点之

间的空间距离和变量之间的空间相关性，通过计算权重来估计未知位置上的值。这样，我们可以生成连续的三维插值地图，帮助我们理解和分析地理空间上的场强分布情况。

进一步观察三维插值后的具体模型，我们可以按照距离将其简化为二维插值模型，以降低模型复杂度并提高数据利用率。根据理论，从中心发射塔发射的场强会随着距离逐渐衰减，因此，在二维插值方法中，我们可以计算各个测量点与发射塔之间的距离，并应用滑动平均窗口使得曲线尽量平滑。然后，我们可以使用拉格朗日、线性或三次样条等方法，根据距离-场强的关系进行插值。

通过这样的二维插值方法，我们可以得到更简化的模型，对场强进行估计和插值，这样的分析有助于我们更好地理解发射塔周围的场强分布情况，并为相关决策提供更准确的数据支持。

可视化系统框架设计结构图如下：



Figx. 可视化系统框架设计结构图

### 三. 可视化系统详细技术

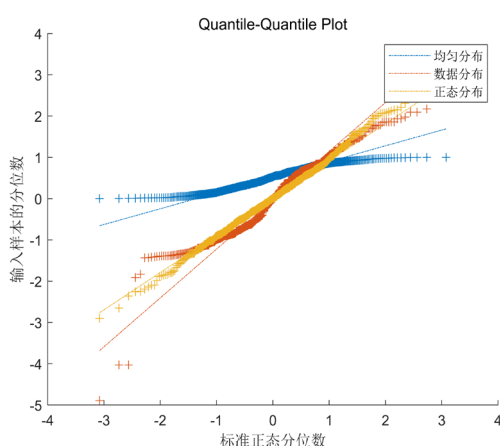
#### • 数据清洗

将收集到的数据按照场强值绘制出频率直方图,可观测到数据分布如下图所示,我们希望验证这些数据近似服从正态分布,然后就可以使用  $3\sigma$  原则剔除掉离群的异常值。

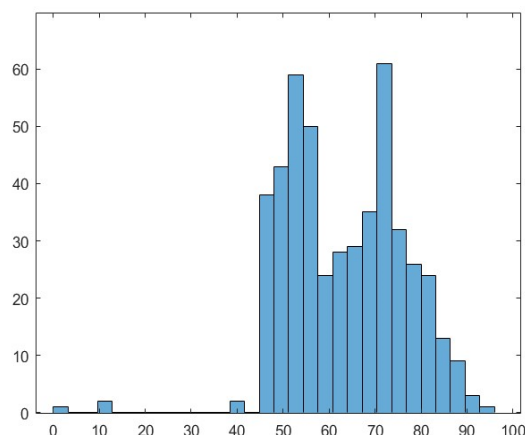
一种验证数据服从正态分布的方法是使用 Quantile-Quantile Plot (Q-Q 图),在 Q-Q 图中,数据点几乎落在  $y = x$  直线上,则说明数据服从正态分布, Q-Q 图的横坐标代表标准正态分布中高斯函数的积分上限  $x$ , 即

$$F(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{1}{2}z^2} dz$$

纵坐标代表该组数的分布范围,如图中生成的正态分布均值为 0, 方差为 1, 大部分落在了  $(-4,4)$  之间, 而生成的均匀分布则落在  $(0,1)$  之间。



Figx. 验证正态分布 Q-Q 图



Figx. 数据分布频率直方图

使用蒙特卡洛模拟,生成均匀分布和标准正态分布的数据,并将我们的数据进行  $z$ -score 标准化之后绘制 Q-Q 图,图中数据分布和标准正态分布近似,数据点均落在了  $y = x$  附近,而均匀分布则和正态分布相差较远,这证明将我们数据按照正态分布的数据清洗方法有一定的可行性。

标准正态分布公式和计算  $\sigma$  的方法为:

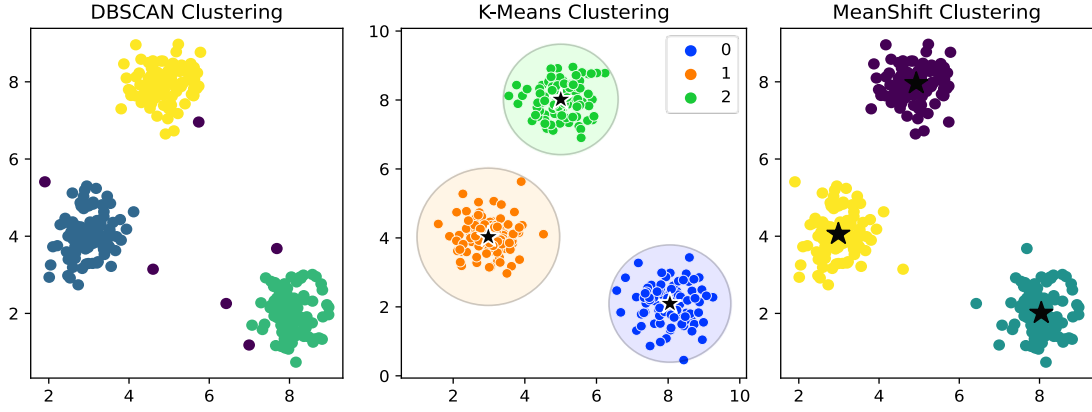
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}$$

#### • 数据聚类

聚类是按照空间特征(如经纬度)把数据集分割成不同的类或簇,使得同一个簇内的数据对象的相似性尽可能大,同时不在同一个簇中的数据对象的差异性也尽可能地大,为方便演示,本例随机生成了一组数据并使用聚类方法进行比较。

在此我们的需求是将距离相近的数据采集点聚合在一起作为一个新的数据

点,新的数据点代表了此簇内的平均场强大小,下图是三种不同聚类方法效果图:



Figx. 多种聚类方法效果比较示意图

**Density-Based Spatial Clustering of Applications with Noise (DBSCAN)** 是一种基于密度的空间聚类方法,它通过将数据点组织成具有高密度区域的群组,并根据低密度区域中的噪声点进行分类,从而发现具有不同密度的数据簇,优点是发现任意形状的簇,对噪声点具有鲁棒性,并且不需要指定簇的数目。

---

**Algorithm 1** DBSCAN

---

**Input:**  $DB$ : Dataset of points,  $\varepsilon$ : Epsilon distance threshold,  $MinPts$ : Minimum number of points required for a cluster.

**Output:** Clustering result.

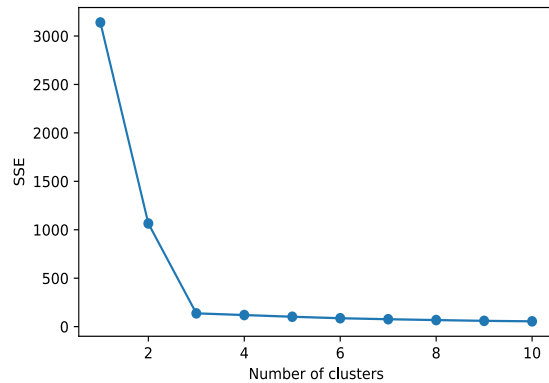
```

1: Initialize all data points as unclassified.
2: for each data point  $P$  in dataset  $DB$  do
3:   if  $P$  is already classified then
4:     Continue to the next point.
5:   end if
6:   Find the  $\varepsilon$ -neighborhood of  $P$  in dataset  $DB$ .
7:    $neighbors \leftarrow$  points in  $\varepsilon$ -neighborhood of  $P$ 
8:   if number of  $neighbors < MinPts$  then
9:     Mark  $P$  as a noise point.
10:  else
11:    Create a new cluster  $C$ .
12:    Add  $P$  to cluster  $C$ .
13:    Expand the cluster by calling  $expandCluster(P, C, neighbors, \varepsilon, MinPts)$ .
14:  end if
15: end for
16: procedure  $EXPANDCLUSTER(P, C, neighbors, \varepsilon, MinPts)$ 
17:   for each neighbor point  $N$  in  $neighbors$  do
18:     if  $N$  is unclassified then
19:       Mark  $N$  as a point belonging to cluster  $C$ .
20:       Find the  $\varepsilon$ -neighborhood of  $N$  in dataset  $DB$ .
21:        $N\_neighbors \leftarrow$  points in  $\varepsilon$ -neighborhood of  $N$ .
22:       if number of  $N\_neighbors \geq MinPts$  then
23:         Add  $N\_neighbors$  to  $neighbors$ .
24:       end if
25:     end if
26:     if  $N$  does not belong to any cluster then
27:       Add  $N$  to cluster  $C$ .
28:     end if
29:   end for
30: end procedure

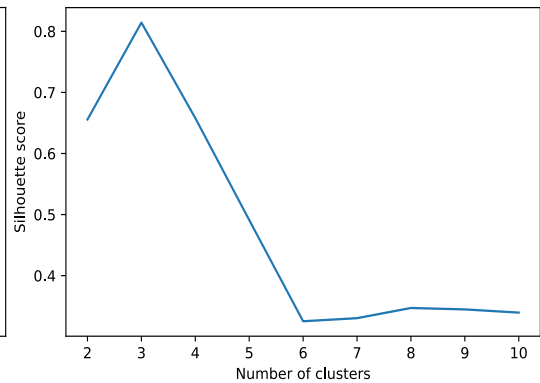
```

---

**k-means** 算法通过迭代优化数据点与其所属簇中心点之间的距离来划分数据点为  $k$  个簇，与 DBSCAN 相比，**k-means** 需要预先指定簇的数目，这可能会对聚类结果产生一定的影响。选择一个较小的  $k$  值可能导致簇的数目过少，无法捕捉到数据集中的所有结构和模式，选择一个较大的  $k$  值可能导致簇的数目过多，其中一些簇可能只包含很少的数据点或不具有实际含义，为了解决这个问题，可以使用肘部法则和轮廓系数等方法来选择合适的  $k$  值，肘部法选择 SSE 开始下降不明显的点，平均轮廓法选择向下的拐角点，因此仿真可证明均能找到符合实际情况的  $k$  值 3。



Figx. 肘部法



Figx. 平均轮廓法

**Meanshift** 算法的核心思想是通过计算数据点的梯度向量来调整其位置，使其向梯度方向上的密度最大的区域移动。具体而言，算法首先为每个数据点选择一个初始位置，然后在每次迭代中，计算每个数据点在当前位置的梯度向量，即指向密度增加最快的方向。然后，将数据点移动到梯度方向上的下一个位置，并继续迭代，直到收敛为止。**Meanshift** 算法的计算复杂度较高，尤其是在处理大规模数据集时，但聚类效果略优于 DBSCAN。

### • 三维插值

三维插值在该可视化系统中的应用是为了生成连续的三维插值地图，以使用户能够更好地理解和分析地理空间上的场强分布情况。由于用于展示地图界面的经纬度和对应场强值的数据可能是不连续的，这意味着在某些位置可能没有直接的测量值，通过进行三维插值，可以填补这些缺失的位置并生成连续的地图，使用户能够更全面地了解场强的分布情况。

**Kriging** 插值方法是一种基于地理空间上变量之间的空间相关性的插值方法。它利用已知数据点的空间分布来估计未知位置上的值，考虑了数据点之间的空间距离和变量之间的空间相关性。通过 **Kriging** 插值，我们可以更准确地推断未知位置上的场强值，并生成连续的三维插值地图。

**Kriging** 插值法将估计未知点的值表示为  $\hat{z}_0 = \sum_{i=0}^n \lambda_i z_i$ ，其中  $\lambda_i$  是权重系数，假设空间属性  $z$  是均一的，满足每个点期望值相同方差也相同。



$$E[z(x, y)] = E[z] = c, Var[(x, y)] = \sigma^2$$

当满足无偏约束条件时，估计的误差期望为 0，即  $E(\hat{z}_o - z_o) = 0$ ，导致权重约束满足条件  $\sum_{i=1}^n \lambda_i = 1$ 。然后我们定义优化目标：

$$\begin{aligned} J &= Var(\sum_{i=1}^n \lambda_i z_i - z_o) \\ &= Var(\sum_{i=1}^n \lambda_i z_i) + Var(z_o) - 2Cov(\sum_{i=1}^n \lambda_i z_i, z_o) \\ &= \sum_{i=1}^n \sum_{j=1}^n [E(x_i x_j) - E(x_i)E(x_j)] + Cov(z_o, z_o) - 2Cov(\sum_{i=1}^n \lambda_i z_i, z_o) \\ &= \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j Cov(z_i, z_j) - 2 \sum_{i=1}^n \lambda_i Cov(z_i, z_o) + Cov(z_o, z_o) \end{aligned}$$

现在目标则为寻找使  $J$  最小的一组  $\lambda_i$ ，我们通过求偏导使得如下等式成立：

$$\frac{\partial J}{\partial \lambda_i} = 0; i = 1, 2, \dots, n$$

---

#### Algorithm 2 Kriging Interpolation

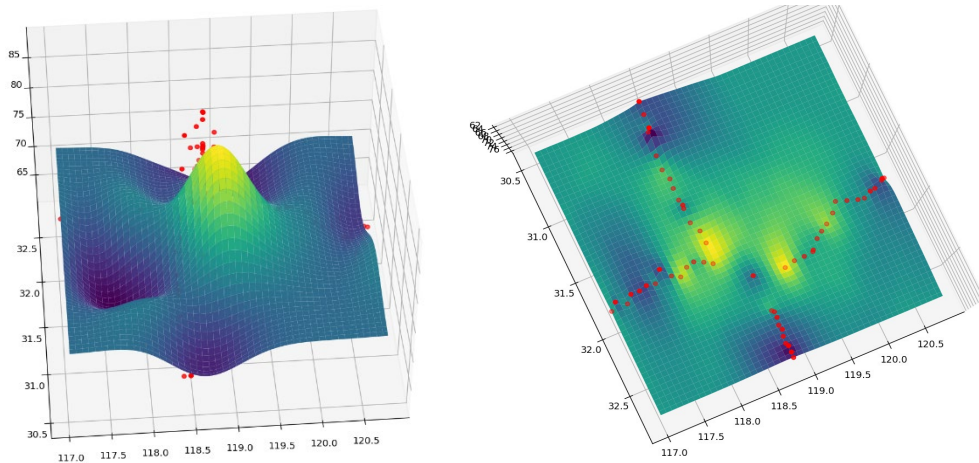
---

**Input:**  $P$ : Set of known data points,  $P_o$ : Unknown point for interpolation,  $N$ : Number of neighboring points to consider.

**Output:** Interpolated value at point  $P_o$ .

- 1: Compute the spatial distance  $d_i$  between  $P_o$  and each point  $p_i$  in  $P$ .
  - 2: Select the  $N$  points in  $P$  with the smallest distances to  $P_o$ .
  - 3: Compute the semivariance  $\gamma_i$  between each pair of selected points.
  - 4: Fit a semivariogram model to the computed semivariance values.
  - 5: Solve the Kriging system of equations to obtain the weights  $w_i$  for each selected point.
  - 6: Compute the interpolated value  $z_o$  at point  $P_o$  using the weights and known values.
  - 7: Return  $z_o$ .
- 

这是一个带有约束条件的最优化问题，我们可以通过拉格朗日乘数法进行解决，综上所述，Kriging 插值法通过最小化代价函数来估计未知点的值，并使用拉格朗日乘数法解决带约束的最优化问题，以得到最优的权重系数，这样可以进行空间属性的插值估计，下面是使用 Kriging 算法对数据点进行拟合的效果图，其中红点代表在该点测量有场强。



Figx. Kriging 算法侧视俯视图

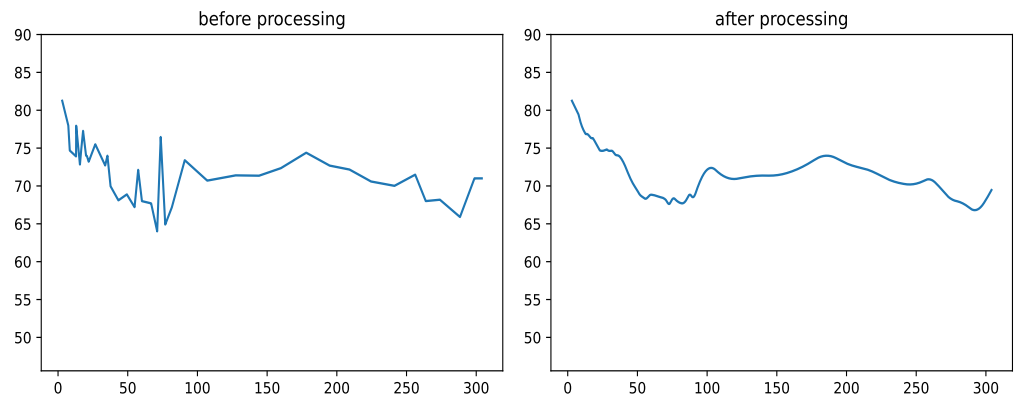
最终绘制结果如下：



Figx. Kriging 三维插值算法绘制图

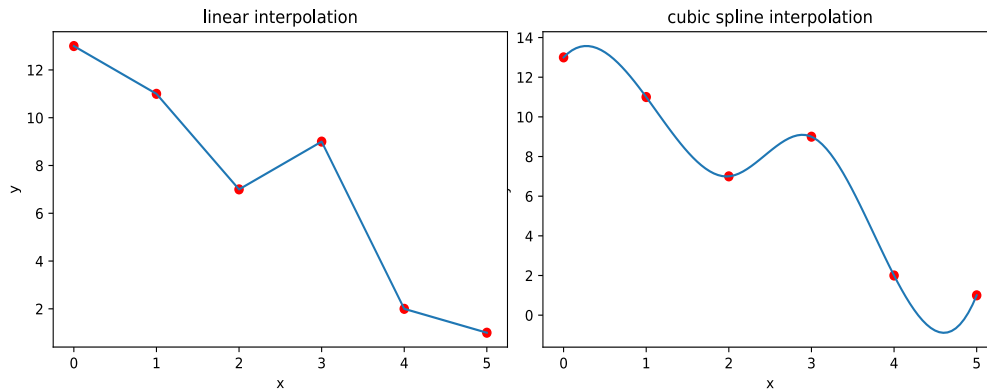
• 二维插值

根据 Kriging 算法和实际理论结果，场强是从中心向外递减，且三维上观察呈单峰状，我们可以将三维插值简化到二维插值，将测量的数据点进行计算，计算距离中心发射台的距离和场强，绘制 $f(\text{距离})=\text{场强}$ 的二维曲线图，将此映射函数按照 y 轴进行旋转，则可以转化成为三维的模型。



Figx. 数据平滑处理前后效果图

在得到函数 $f$ 前，我们需要先对数据进行处理，去掉数据中不平滑的尖刺，在这里使用了移动平均对数据进行平滑处理，去除了数据中的尖刺和噪声，使数据变得更加平滑。在得到平滑的数据后，可以根据测量数据点的位置和其对应的场强值计算距离中心发射台的距离，然后，可以绘制一个以距离为横坐标、场强为纵坐标的二维曲线图，表示距离与场强之间的关系。



Figx. 两种插值方法图

然后再使用三次样条插值或线性插值来拟合函数，其中三次样条插值是一种基于多个数据点的插值方法，它使用三次多项式来逼近数据点之间的曲线，以获得平滑的插值结果。线性插值是一种简单的插值方法，它使用两个相邻数据点之间的直线来估计未知点的值，适用于数据变化较为线性的情况。

### Algorithm 3 Cubic Spline Interpolation

**Input:**  $X$ : Array of x-coordinates of data points,  $Y$ : Array of y-coordinates of data points.

**Output:** Interpolated function.

- 1: Sort data points by  $X$  in ascending order.
- 2: Compute second derivatives  $D$  at each data point using *computeSecondDerivatives*( $X$ ,  $Y$ ).
- 3: Initialize arrays  $h$ ,  $a$ ,  $b$ ,  $c$ , and  $d$  with appropriate sizes.
- 4: Compute  $h_i = X_{i+1} - X_i$  for  $i = 0$  to  $n - 1$ , where  $n$  is the number of data points.
- 5: Compute  $a_i = Y_i$  for  $i = 0$  to  $n - 1$ .
- 6: Compute  $c_i = D_i$  for  $i = 0$  to  $n - 1$ .
- 7: **for**  $i = 0$  to  $n - 2$  **do**
- 8:     Compute  $b_i = \frac{Y_{i+1} - Y_i}{h_i} - \frac{h_i}{3}(2c_i + c_{i+1})$ .
- 9:     Compute  $d_i = \frac{c_{i+1} - c_i}{3h_i}$ .
- 10: **end for**
- 11: Interpolate each segment between  $X_i$  and  $X_{i+1}$  using the cubic polynomial equation.
- 12: **procedure** COMPUTESECONDDERIVATIVES( $X$ ,  $Y$ )
- 13:     Initialize array  $D$  with appropriate size.
- 14:     Solve the tridiagonal system of equations  $Ax = b$  to find  $D$ .
- 15:     Return  $D$ .
- 16: **end procedure**

最终绘制效果如下：



Figx. 二维插值算法绘制图



## 四. 主要技术创新点

- 流畅优美的动画呈现

1. 计算速度优化：通过采用预计算和仅计算屏幕刷新范围内图形的方法，实现了动画计算速度的显著优化，这一创新点使得动画的运行效率大幅提升，同时降低了计算资源的消耗。

2. 基于 Python Flask 后台的三维复杂计算：我们将三维复杂计算部分迁移到 Python Flask 后台进行处理，通过利用后台服务器进行计算，我们能够充分发挥 Python 语言的计算能力和丰富的科学计算库，提供更为准确和高效的计算结果。

3. JavaScript 中的完全二维集成：我们将二维计算完全集成在 JavaScript 中，这也是一项创新。通过在前端利用 JavaScript 进行计算，我们实现了快速的图形渲染和交互，为用户提供了流畅而高效的使用体验。

- 智能科学的数据处理

1. 数据清洗创新：我们采用了综合的数据清洗方法，包括从数据场强本身和时间空间方向进行清洗。通过使用 Quantile-Quantile Plot (Q-Q 图) 和  $3\sigma$  原则，我们能够准确地识别和剔除异常值，提高数据的质量和准确性。同时，通过聚类方法如 DBSCAN、Meanshift 和 k-means，我们能够将距离较近的数据点聚集在一起取平均，消除数据的冗余性，使得绘制的曲线更加平滑和可靠。

2. 三维与二维插值创新：引入地理学上 Kriging 插值方法到通信，进行了三维插值，考虑数据点之间的空间相关性，能够准确推断未知位置上的场强值，并进一步简化为二维插值模型，降低了模型复杂度并提高了数据利用率，提供连续且可靠的场强分布图，为相关决策提供准确的数据支持。

- 灵活多变的可选参数

可支持屏幕刷新精度 (Pixel accuracy)、刷新闻隔时间 (throttleInterval)、滑动窗口大小 (slider k)、插值变差函数 (variogram model) 以及各类插值、拟合和绘制方法。用户可根据具体需求，选择较高的刷新精度、复杂度更高的模型以获得更详细的绘图结果，或选择较低的精度、简单的模型以提高性能和加载速度，这种灵活性使得应用能够适应不同场景和需求。

- 跨平台的开发框架 uniapp

1. 可移植性强：uni-app 的应用程序可以在 iOS、Android 和 Web 等多个平台上运行，可编译导出为微信、百度、抖音、快手等小程序，具有很强的可移植性，使用同一套代码开发应用程序，并在不同平台上进行部署，减少重复开发和

维护的工作量。

2. 生态系统支持：uni-app 拥有庞大的开发者社区和丰富的插件生态系统，生态系统市场非常庞大，即插即用的地图绘制组件实用性强、应用空间广阔。

## 五. 推广应用方向

**信号优化和规划：**帮助用户在特定区域内优化信号覆盖和强度。通过对模拟和优化算法进行可视化，产品可以帮助决策不同场景下的信号优化方案，包括天线位置优化、功率调整、频谱分配等，在电信运营商、网络规划师和设备厂商等行业非常有价值。

**故障诊断和维护：**帮助用户快速定位和解决信号问题，通过实时监测和分析场强数据，产品可以提供故障报警、异常检测和故障定位等功能，帮助用户及时发现和解决信号异常或故障，提高网络的可靠性和稳定性。

**多维度据可视化：**产品可以支持输入离散点并进行插值补全，使用户能够更全面地理解和分析网络数据。通过使用热力图、动态可视化、等值线等方式，产品可以呈现不同参数、时间段和地理位置上的网络数据，帮助用户发现数据之间的关联关系和趋势，做出更准确的决策。

