**有限差分法上机报告**

1计算长直接地金属槽中的电场分布。金属槽横截面如图所示，其侧壁与底面电位均为0，顶盖电位相对值为10。槽内电位函数满足拉普拉斯方程，计算槽内电位分布。

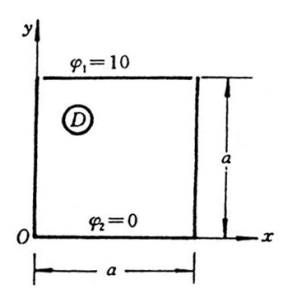


图1 金属槽横截面

* 1. **有限差分法解题步骤**

对于槽内点位分布，可理想化为二维问题。选定直角坐标系，槽内电位函数满足拉普拉斯方程。

1.1.1离散化处理

对计算区域取正方形网格离散，步长，在方向的节点数都为5。

1.1.2 超松弛迭代法的差分方程形式

其中称为加速收敛因子或超松弛因子，它的数值决定超松弛程度，影响迭代解收敛的速度。其取值范围为。

对于第一类边值问题，若一方形场域由正方形网格划分，每边的节点数为，则可由下式计算：

所以这里可以得到。

1.1.3 给出边界条件

本题属于第一类边界条件，可以利用直接赋值的方式，即

1.1.4 内部节点赋初值

对于内部条件赋值，可知场内各处电位必然介于和之间，采用等差递增的方式来赋初值，即

1.1.5 检查迭代解收敛的条件

计算精度为千分之一，即对于场域内任一节点，两次相邻迭代值需满足。

1.1.6 计算框图

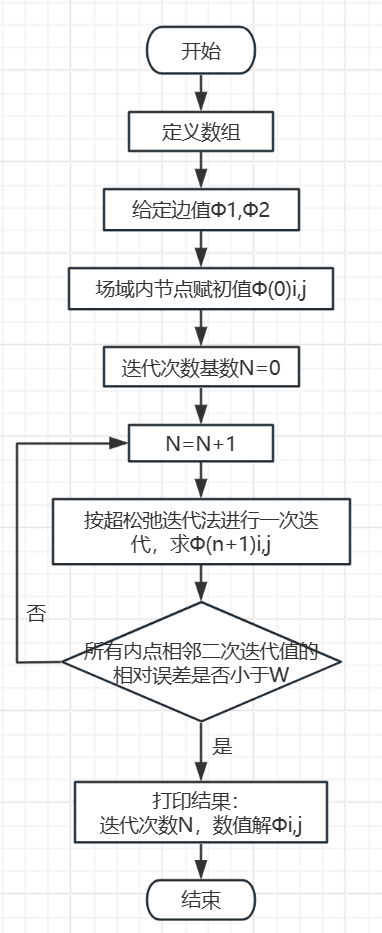


图2 计算框图

源代码请见附录1

* 1. **计算结果**

0.00 10.00 10.00 10.00 0.00

0.00 4.29 5.27 4.29 0.00

0.00 1.88 2.50 1.88 0.00

0.00 0.71 0.98 0.71 0.00

0.00 0.00 0.00 0.00 0.00

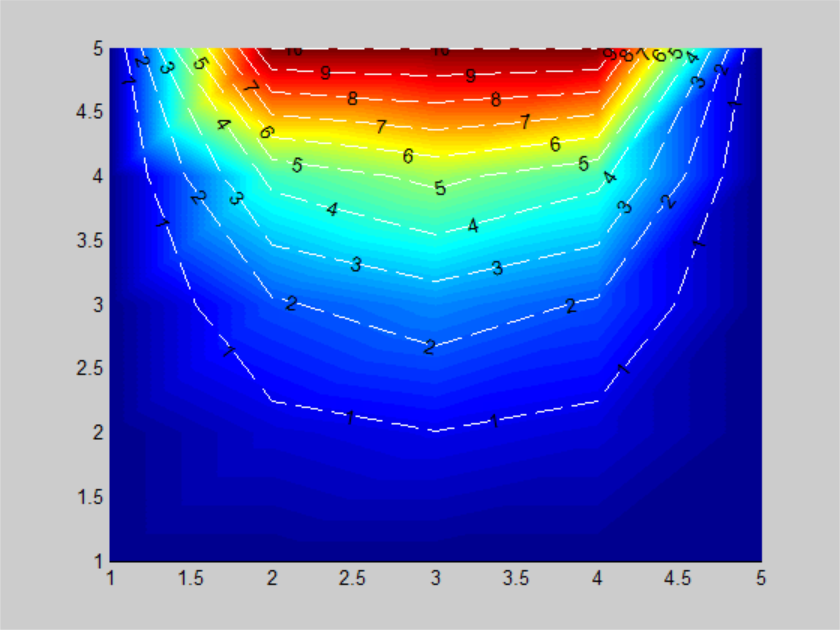


图3 全场域的电位分布图

迭代次数为8。

改变松弛因子，比较其收敛速度。

表1 α与对应的迭代次数

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| α |  |  |  |  |  |  |  |  |  |  |  |
| 迭代次数 |  |  |  |  |  |  |  |  |  |  |  |
| α |  |  |  |  |  |  |  |  |  |  |  |
| 迭代次数 |  |  |  |  |  |  |  |  |  |  |  |

图4 α-迭代次数

可见，值越接近，所需迭代次数越少，即收敛速度越快。

**1.3 划分网格加倍**

则,。

计算分析类似上题，计算结果如下。

0.00 10.00 10.00 10.00 10.00 10.00 10.00 10.00 0.00

0.00 4.83 6.61 7.31 7.49 7.31 6.61 4.83 0.00

0.00 2.69 4.31 5.12 5.36 5.12 4.31 2.69 0.00

0.00 1.64 2.82 3.50 3.71 3.50 2.82 1.64 0.00

0.00 1.03 1.84 2.33 2.50 2.33 1.84 1.03 0.00

0.00 0.64 1.17 1.50 1.62 1.50 1.17 0.64 0.00

0.00 0.38 0.69 0.89 0.96 0.89 0.69 0.38 0.00

0.00 0.17 0.32 0.42 0.45 0.42 0.32 0.17 0.00

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

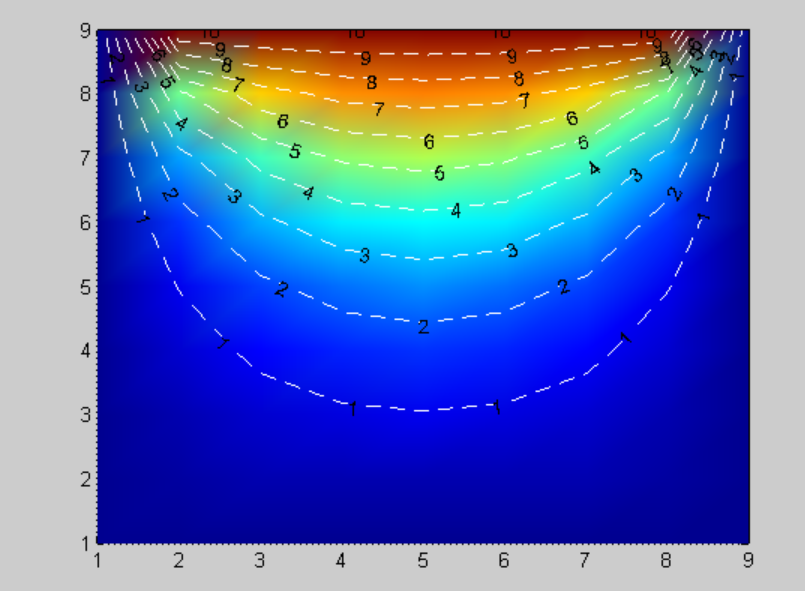


图5 全场域电位分布

迭代次数为16。

相比于网格较少的计算结果，网格加倍后，迭代次数增加，场域内电位的数值分布更加具体，分布图更加精细平滑。

2 计算长直接地金属槽中的电场分布。其长为a，宽为0.6a，侧壁与底面电位均为0，顶盖电位相对值为100V。根据场分布的对称性，以半场域为计算区域，并以将半场域由正方形网格予以精细划分。槽内电位函数满足拉普拉斯方程，计算槽内电位分布。

**2.1有限差分法解题**

对于槽内点位分布，可理想化为二维问题。选定直角坐标系，槽内电位函数满足拉普拉斯方程。

2.1.1离散化处理

选取半场域进行计算，对计算区域进行离散处理，取正方形网格离散，步长，在方向的节点数为41，在方向的节点数为25。

2.1.2 超松弛迭代法的差分方程形式

其中称为加速收敛因子或超松弛因子，它的数值决定超松弛程度，影响迭代解收敛的速度。其取值范围为。

若一矩形区域由边长为的正方形网格划分为和，且和都较大，则

所以这里可以得到。

2.1.3 给出边界条件

对于中间轴线上的点，由于电势线平行于轴，故电场方向平行于轴线，即

所以在轴线上

其他边界属于第一类边界条件，可以利用直接赋值的方式，即

2.1.4 内部节点赋初值

对于内部条件赋值，可知场内各处电位必然介于和之间，采用等差递增的方式来赋初值，即

2.1.5 检查迭代解收敛的条件

计算精度为千分之一，即对于场域内任一节点，两次相邻迭代值需满足。

源代码请见附录2。

* + 1. 计算结果

左半场域的数值结果如下。

0.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00

0.00 49.90 69.56 78.75 83.74 86.77 88.78 90.19 91.22 91.99 92.59 93.07 93.44 93.73 93.97 94.15 94.29 94.40 94.47 94.51 94.53

0.00 30.03 49.59 61.71 69.43 74.57 78.16 80.75 82.69 84.17 85.32 86.23 86.95 87.53 87.98 88.35 88.62 88.83 88.97 89.06 89.08

0.00 20.63 37.06 49.08 57.70 63.93 68.52 71.97 74.61 76.67 78.29 79.58 80.61 81.44 82.10 82.62 83.02 83.32 83.53 83.65 83.69

0.00 15.44 28.93 39.84 48.36 54.94 60.03 64.00 67.12 69.60 71.59 73.19 74.48 75.52 76.35 77.02 77.53 77.91 78.18 78.33 78.38

0.00 12.20 23.37 32.99 40.96 47.43 52.66 56.87 60.28 63.03 65.27 67.10 68.59 69.80 70.78 71.56 72.16 72.62 72.93 73.12 73.18

0.00 9.99 19.38 27.78 35.04 41.18 46.31 50.56 54.07 56.98 59.38 61.35 62.98 64.32 65.41 66.27 66.95 67.46 67.82 68.02 68.09

0.00 8.38 16.37 23.72 30.25 35.94 40.83 44.98 48.48 51.43 53.90 55.96 57.67 59.09 60.25 61.18 61.91 62.46 62.85 63.07 63.15

0.00 7.14 14.02 20.46 26.31 31.52 36.08 40.04 43.45 46.35 48.82 50.91 52.66 54.12 55.32 56.29 57.05 57.63 58.03 58.27 58.35

0.00 6.15 12.13 17.78 23.00 27.73 31.95 35.66 38.91 41.72 44.13 46.19 47.94 49.40 50.61 51.60 52.38 52.97 53.38 53.63 53.71

0.00 5.34 10.56 15.53 20.18 24.45 28.31 31.76 34.81 37.48 39.79 41.79 43.50 44.94 46.14 47.12 47.90 48.49 48.90 49.15 49.23

0.00 4.66 9.23 13.61 17.75 21.58 25.09 28.25 31.08 33.58 35.78 37.68 39.32 40.72 41.88 42.84 43.60 44.18 44.59 44.83 44.91

0.00 4.08 8.08 11.95 15.61 19.04 22.20 25.09 27.68 30.00 32.05 33.84 35.39 36.72 37.83 38.75 39.49 40.05 40.45 40.68 40.76

0.00 3.57 7.07 10.47 13.72 16.77 19.60 22.20 24.57 26.69 28.58 30.24 31.69 32.93 33.98 34.85 35.55 36.08 36.46 36.68 36.76

0.00 3.11 6.18 9.16 12.01 14.71 17.23 19.56 21.69 23.61 25.33 26.86 28.19 29.34 30.31 31.12 31.78 32.28 32.63 32.84 32.91

0.00 2.70 5.37 7.96 10.46 12.83 15.05 17.12 19.01 20.74 22.29 23.66 24.87 25.92 26.81 27.55 28.15 28.61 28.94 29.13 29.20

0.00 2.33 4.63 6.87 9.04 11.10 13.04 14.85 16.51 18.04 19.41 20.63 21.72 22.66 23.46 24.13 24.67 25.08 25.38 25.56 25.61

0.00 1.98 3.95 5.86 7.71 9.48 11.15 12.71 14.16 15.48 16.68 17.75 18.70 19.53 20.24 20.83 21.31 21.68 21.94 22.10 22.15

0.00 1.66 3.31 4.92 6.47 7.96 9.38 10.70 11.93 13.05 14.08 14.99 15.81 16.52 17.13 17.64 18.06 18.38 18.61 18.74 18.79

0.00 1.36 2.71 4.03 5.30 6.53 7.69 8.78 9.79 10.73 11.58 12.34 13.02 13.62 14.13 14.56 14.90 15.17 15.36 15.48 15.52

0.00 1.07 2.13 3.17 4.18 5.15 6.07 6.93 7.74 8.48 9.16 9.77 10.32 10.79 11.20 11.55 11.83 12.04 12.20 12.29 12.32

0.00 0.80 1.58 2.35 3.10 3.82 4.51 5.15 5.75 6.31 6.81 7.27 7.68 8.04 8.35 8.60 8.81 8.98 9.09 9.16 9.19

0.00 0.53 1.05 1.56 2.05 2.53 2.98 3.41 3.81 4.18 4.52 4.82 5.09 5.33 5.54 5.71 5.85 5.96 6.04 6.08 6.10

0.00 0.26 0.52 0.77 1.02 1.26 1.48 1.70 1.90 2.08 2.25 2.40 2.54 2.66 2.76 2.85 2.92 2.97 3.01 3.03 3.04

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

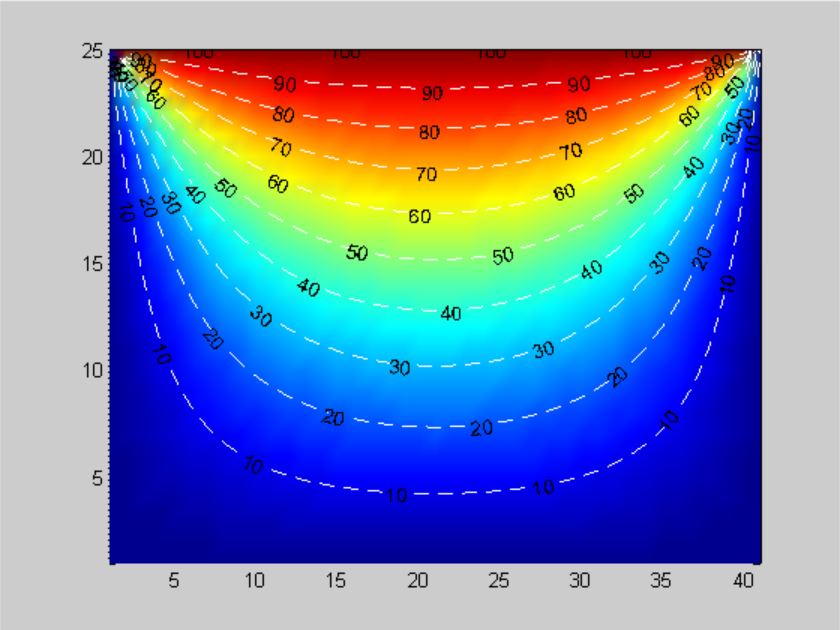


图6 全场域的电位分布

**2.2 若顶盖电位变为**，即

分析与上题类似，源代码请见附录3。对于左半场域，计算结果如下

0.00 7.85 15.64 23.34 30.90 38.27 45.40 52.25 58.78 64.94 70.71 76.04 80.90 85.26 89.10 92.39 95.11 97.24 98.77 99.69 100.00

0.00 7.22 14.40 21.50 28.45 35.24 41.80 48.11 54.12 59.80 65.11 70.02 74.49 78.51 82.04 85.07 87.57 89.53 90.95 91.79 92.08

0.00 6.65 13.25 19.78 26.18 32.42 38.46 44.27 49.80 55.02 59.91 64.43 68.54 72.24 75.49 78.28 80.58 82.38 83.68 84.46 84.73

0.00 6.11 12.19 18.18 24.07 29.81 35.36 40.70 45.79 50.59 55.08 59.23 63.02 66.42 69.40 71.96 74.08 75.74 76.93 77.65 77.89

0.00 5.61 11.19 16.70 22.11 27.38 32.48 37.38 42.05 46.46 50.59 54.40 57.88 61.00 63.75 66.10 68.04 69.57 70.66 71.32 71.54

0.00 5.15 10.27 15.32 20.28 25.12 29.80 34.29 38.58 42.63 46.41 49.91 53.10 55.96 58.48 60.64 62.42 63.82 64.82 65.43 65.63

0.00 4.72 9.41 14.04 18.58 23.01 27.30 31.42 35.34 39.05 42.52 45.72 48.64 51.27 53.57 55.55 57.18 58.47 59.39 59.94 60.13

0.00 4.31 8.60 12.84 16.99 21.05 24.97 28.73 32.32 35.72 38.89 41.82 44.49 46.89 49.00 50.81 52.30 53.47 54.32 54.82 54.99

0.00 3.94 7.85 11.72 15.51 19.21 22.79 26.23 29.51 32.60 35.50 38.17 40.61 42.80 44.73 46.38 47.74 48.81 49.58 50.04 50.20

0.00 3.59 7.15 10.67 14.13 17.49 20.75 23.88 26.87 29.69 32.32 34.76 36.98 38.98 40.73 42.23 43.47 44.45 45.15 45.57 45.71

0.00 3.26 6.49 9.69 12.83 15.88 18.84 21.69 24.40 26.96 29.35 31.56 33.58 35.39 36.98 38.35 39.48 40.36 41.00 41.38 41.51

0.00 2.95 5.88 8.77 11.61 14.37 17.05 19.62 22.08 24.39 26.56 28.56 30.39 32.02 33.47 34.70 35.72 36.52 37.10 37.44 37.56

0.00 2.66 5.29 7.90 10.46 12.95 15.36 17.68 19.89 21.98 23.93 25.73 27.38 28.86 30.15 31.27 32.19 32.91 33.43 33.74 33.84

0.00 2.38 4.75 7.08 9.37 11.61 13.77 15.85 17.83 19.70 21.45 23.07 24.54 25.86 27.03 28.02 28.85 29.50 29.96 30.24 30.33

0.00 2.12 4.23 6.31 8.35 10.34 12.26 14.11 15.88 17.54 19.10 20.54 21.85 23.03 24.07 24.96 25.69 26.27 26.68 26.93 27.01

0.00 1.87 3.73 5.57 7.37 9.13 10.83 12.47 14.02 15.49 16.87 18.14 19.30 20.34 21.26 22.04 22.69 23.20 23.56 23.78 23.86

0.00 1.64 3.26 4.87 6.44 7.98 9.47 10.89 12.26 13.54 14.74 15.85 16.87 17.78 18.58 19.26 19.83 20.27 20.59 20.79 20.85

0.00 1.41 2.81 4.20 5.55 6.88 8.16 9.39 10.56 11.67 12.71 13.67 14.54 15.32 16.01 16.60 17.09 17.47 17.75 17.92 17.97

0.00 1.19 2.38 3.55 4.70 5.82 6.90 7.94 8.94 9.87 10.75 11.56 12.30 12.96 13.55 14.05 14.46 14.78 15.02 15.16 15.20

0.00 0.98 1.96 2.92 3.87 4.79 5.69 6.55 7.36 8.14 8.86 9.53 10.14 10.68 11.16 11.58 11.92 12.18 12.37 12.49 12.53

0.00 0.78 1.55 2.32 3.07 3.80 4.51 5.19 5.84 6.45 7.02 7.55 8.03 8.47 8.85 9.18 9.45 9.66 9.81 9.90 9.93

0.00 0.58 1.16 1.73 2.29 2.83 3.36 3.86 4.35 4.80 5.23 5.62 5.98 6.31 6.59 6.83 7.03 7.19 7.30 7.37 7.40

0.00 0.38 0.77 1.15 1.52 1.88 2.23 2.56 2.88 3.19 3.47 3.73 3.97 4.18 4.37 4.53 4.67 4.77 4.84 4.89 4.91

0.00 0.19 0.38 0.57 0.76 0.94 1.11 1.28 1.44 1.59 1.73 1.86 1.98 2.08 2.18 2.26 2.33 2.38 2.41 2.44 2.45

0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

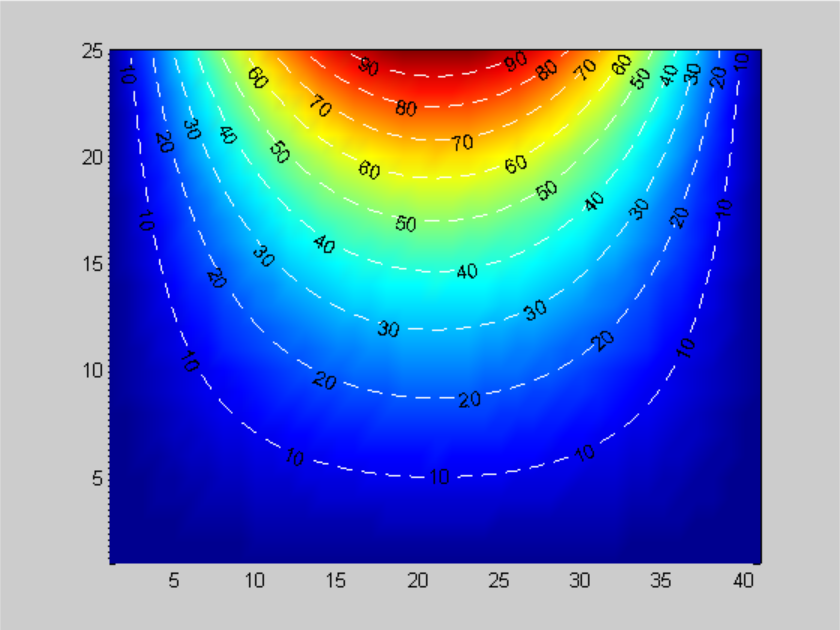


图7 全场域的电位分布

**2.3计算电机电枢槽气隙磁位分布**。忽略曲度效应，采用直角坐标，计算二维恒定场。令定子和转子磁铁中，是对称线，考虑对称性有

求内部边界上全部节点磁位，计算精度取。

2.3.1取半场域来计算。

对计算区域进行离散处理，取正方形网格离散，步长，在方向的节点数为12，方向的节点数都为75。

2.3.2超松弛迭代法的差分方程形式

其中称为加速收敛因子或超松弛因子，它的数值决定超松弛程度，影响迭代解收敛的速度。其取值范围为。这里取。

2.3.3给出边界条件

对于边，

所以

同理对于边，

其他边界属于第一类边界条件，可以利用直接赋值的方式，即

2.3.4 内部节点赋初值

对于内部条件赋值，可知场内各处电位必然介于和之间，采用等差递增的方式来赋初值，即

2.3.5 检查迭代解收敛的条件

计算精度为千分之一，即对于场域内任一节点，两次相邻迭代值需满足。

源代码请见附录4。

2.3.6 对于左半场域，计算结果如下

100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00

87.81 87.83 87.88 87.98 88.13 88.31 88.53 88.76 88.98 89.17 89.29 89.33

75.58 75.61 75.73 75.92 76.21 76.58 77.04 77.53 78.00 78.39 78.65 78.74

63.27 63.32 63.49 63.77 64.20 64.78 65.51 66.32 67.10 67.75 68.18 68.32

50.86 50.92 51.12 51.48 52.04 52.84 53.89 55.13 56.34 57.34 57.98 58.21

38.32 38.38 38.60 38.99 39.64 40.64 42.09 43.96 45.80 47.28 48.21 48.53

25.64 25.70 25.89 26.25 26.88 27.98 29.86 32.82 35.63 37.76 39.06 39.49

12.86 12.89 13.00 13.23 13.67 14.55 16.56 21.83 26.14 29.08 30.76 31.31

0.00 0.00 0.00 0.00 0.00 0.00 0.00 11.78 18.02 21.65 23.60 24.22

0.00 7.29 12.51 15.88 17.78 18.39

0.00 4.88 8.83 11.61 13.23 13.77

0.00 3.41 6.32 8.48 9.78 10.22

0.00 2.43 4.57 6.19 7.20 7.54

0.00 1.76 3.32 4.53 5.29 5.55

0.00 1.28 2.42 3.32 3.88 4.08

0.00 0.93 1.77 2.43 2.85 2.99

0.00 0.68 1.29 1.78 2.09 2.19

0.00 0.50 0.95 1.30 1.53 1.61

0.00 0.36 0.69 0.95 1.12 1.18

0.00 0.27 0.51 0.70 0.82 0.86

0.00 0.20 0.37 0.51 0.60 0.63

0.00 0.14 0.27 0.38 0.44 0.46

0.00 0.11 0.20 0.28 0.32 0.34

0.00 0.08 0.15 0.20 0.24 0.25

0.00 0.06 0.11 0.15 0.17 0.18

0.00 0.04 0.08 0.11 0.13 0.14

0.00 0.03 0.06 0.08 0.09 0.10

0.00 0.02 0.04 0.06 0.07 0.07

0.00 0.02 0.03 0.04 0.05 0.05

0.00 0.01 0.02 0.03 0.04 0.04

0.00 0.01 0.02 0.02 0.03 0.03

0.00 0.01 0.01 0.02 0.02 0.02

0.00 0.01 0.01 0.01 0.02 0.02

0.00 0.00 0.01 0.01 0.01 0.01

0.00 0.00 0.01 0.01 0.01 0.01

0.00 0.00 0.00 0.01 0.01 0.01

0.00 0.00 0.00 0.00 0.01 0.01

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00 0.00

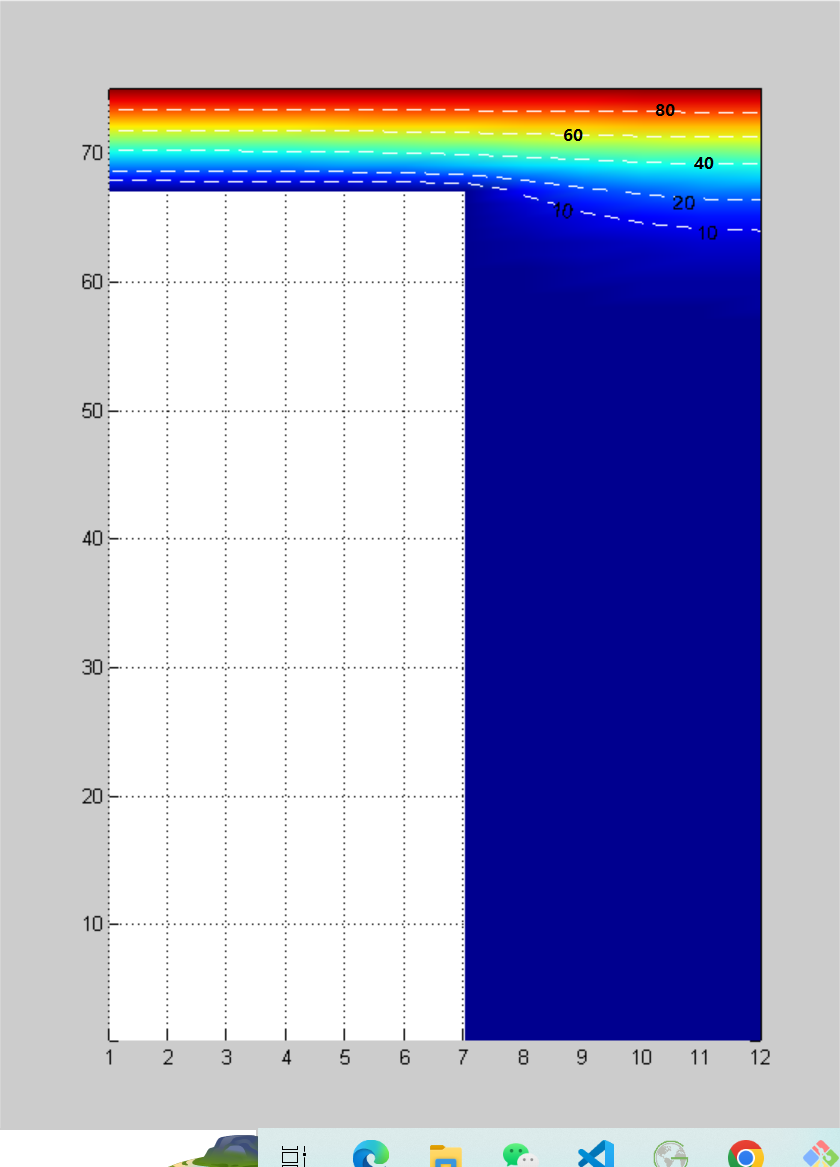
0.00 0.00 0.00 0.00 0.00 0.00 

图8 左半场域磁位分布

**附录1**

#include <iostream>

#include<cmath>

using namespace std;

int main(){

    int n=4,m=4;

    float a[m+1][n+1];

    int i,j,flag;

    float a1;

    float alpha;

    FILE \*fp;

    int count=0;

alpha=2/(1+sin(3.141592653/n));

//initial

    for(i=0;i<=m;i++){

        a[i][0]=0;

        a[i][n]=0;

    }

    for(j=1;j<n;j++){

        a[0][j]=10;

    }

    for(j=1;j<=n;j++){

        a[m][j]=0;

    }

    for(i=1;i<m;i++){

        for(j=1;j<n;j++){

            a[i][j]=10.0-10.0/m\*i;

        }

    }

    //iteration

    do{

        flag=0;

        for(i=1;i<m;i++){

            for(j=1;j<n;j++){

                a1=a[i][j];

                a[i][j]=a[i][j]+alpha/4\*(a[i][j-1]+a[i-1][j]+

                a[i][j+1]+a[i+1][j]-4\*a[i][j]);

                if(fabs(a1-a[i][j])>0.001)flag++;

            }

        }

        count++;

    }while(flag);

    //output

    for(i=0;i<=m;i++){

        for(j=0;j<=n;j++){

            printf("%8.4f",a[i][j]);

        }

        printf("\n");

    }

    //file

    if((fp=fopen("kk.txt","w+"))==NULL){

        printf("\nCannot open this file. \n");

        exit(0);

    }

    fputs("\n\n\n",fp);

    for(i=0;i<=m;i++){

        for(j=0;j<=n;j++){

            fprintf(fp,"%6.2f ",a[i][j]);

        }

        fputs("\n",fp);

    }

    fclose(fp);

}

**附录2**

#include <iostream>

#include<cmath>

using namespace std;

int main(){

    int n=20,m=24;

    float a[m+1][n+1];

    int i,j,flag;

    float a1;

    float alpha;

    FILE \*fp;

    int count=0;

    alpha=(2-3.141592653\*sqrt(2\*(m\*m+4\*n\*n))/(m\*2\*n));

    //initial

    for(i=0;i<=m;i++){

        a[i][0]=0;

    }

    for(j=1;j<=n;j++){

        a[0][j]=100;

    }

    for(j=1;j<=n;j++){

        a[m][j]=0;

    }

    for(i=1;i<=m;i++){

        for(j=1;j<=n;j++){

            a[i][j]=100.0-100.0/m\*i;

        }

    }

    //iteration

    do{

        flag=0;

        for(i=1;i<m;i++){

            for(j=1;j<n;j++){

                a1=a[i][j];

                a[i][j]=a[i][j]+alpha/4\*(a[i][j-1]+a[i-1][j]+

                a[i][j+1]+a[i+1][j]-4\*a[i][j]);

                if(fabs(a1-a[i][j])>0.001)flag++;

            }

                a1=a[i][n];

                a[i][n]=a[i][n]+alpha/4\*(2\*a[i][n-1]+a[i-1][n]+

                +a[i+1][n]-4\*a[i][n]);

                if(fabs(a1-a[i][n])>0.001)flag++;

        }

        count++;

    }while(flag);

    //output

    for(i=0;i<=m;i++){

        for(j=0;j<=n;j++){

            printf("%8.4f",a[i][j]);

        }

        printf("\n");

    }

    //file

    if((fp=fopen("kk2.txt","w+"))==NULL){

        printf("\nCannot open this file. \n");

        exit(0);

    }

    fputs("\n\n\n",fp);

    for(i=0;i<=m;i++){

        for(j=0;j<=n;j++){

            fprintf(fp,"%6.2f ",a[i][j]);

        }

        fputs("\n",fp);

    }

    fclose(fp);

}

**附录3**

#include <iostream>

#include<cmath>

using namespace std;

int main(){

    int n=40,m=24;

    float a[m+1][n+1];

    int i,j,flag;

    float a1;

    float alpha;

    FILE \*fp;

    int count=0;

    alpha=(2-3.141592653\*sqrt(2\*(m\*m+n\*n))/(m\*n));

    //initial

    for(i=0;i<=m;i++){

        a[i][0]=0;

        a[i][n]=0;

    }

    for(j=1;j<n;j++){

        a[0][j]=100\*sin(j\*3.141592653/40);

    }

    for(j=1;j<=n;j++){

        a[m][j]=0;

    }

    for(i=1;i<m;i++){

        for(j=1;j<n;j++){

            a[i][j]=100.0-100.0/m\*i;

        }

    }

    //iteration

    do{

        flag=0;

        for(i=1;i<m;i++){

            for(j=1;j<n;j++){

                a1=a[i][j];

                a[i][j]=a[i][j]+alpha/4\*(a[i][j-1]+a[i-1][j]+

                a[i][j+1]+a[i+1][j]-4\*a[i][j]);

                if(fabs(a1-a[i][j])>0.001)flag++;

            }

        }

        count++;

    }while(flag);

    //output

    for(i=0;i<=m;i++){

        for(j=0;j<=n;j++){

            printf("%8.4f",a[i][j]);

        }

        printf("\n");

    }

    //file

    if((fp=fopen("kk3.txt","w+"))==NULL){

        printf("\nCannot open this file. \n");

        exit(0);

    }

    fputs("\n\n\n",fp);

    for(i=0;i<=m;i++){

        for(j=0;j<=n;j++){

            fprintf(fp,"%6.2f ",a[i][j]);

        }

        fputs("\n",fp);

    }

    fclose(fp);

}

**附录4**

#include <iostream>

#include<cmath>

using namespace std;

int main(){

    int n=11,m=74;

    float a[m+1][n+1];

    int i,j,flag;

    float a1;

    float alpha;

    FILE \*fp;

    int count=0;

    alpha=1.5;

    //initial

    for(i=8;i<=74;i++){

        a[i][6]=0;

    }

    for(j=0;j<=n;j++){

        a[0][j]=100;

    }

    for(j=1;j<=n;j++){

        a[m][j]=0;

    }

    for(i=1;i<=8;i++){

        for(j=0;j<=6;j++){

            a[i][j]=100.0-100.0/8\*i;

        }

    }

    for(i=1;i<=74;i++){

        for(j=7;j<=11;j++){

            a[i][j]=100.0-100.0/74\*i;

        }

    }

    //iteration

    do{

        flag=0;

        for(i=1;i<8;i++){

            for(j=1;j<=6;j++){

                a1=a[i][j];

                a[i][j]=a[i][j]+alpha/4\*(a[i][j-1]+a[i-1][j]+

                a[i][j+1]+a[i+1][j]-4\*a[i][j]);

                if(fabs(a1-a[i][j])>0.001)flag++;

            }

                a1=a[i][0];

                a[i][0]=a[i][0]+alpha/4\*(2\*a[i][1]+a[i-1][0]+

                +a[i+1][0]-4\*a[i][0]);

                if(fabs(a1-a[i][0])>0.001)flag++;

        }

        for(i=1;i<74;i++){

            for(j=7;j<11;j++){

                a1=a[i][j];

                a[i][j]=a[i][j]+alpha/4\*(a[i][j-1]+a[i-1][j]+

                a[i][j+1]+a[i+1][j]-4\*a[i][j]);

                if(fabs(a1-a[i][j])>0.001)flag++;

            }

                a1=a[i][11];

                a[i][11]=a[i][11]+alpha/4\*(2\*a[i][10]+a[i-1][11]+

                +a[i+1][11]-4\*a[i][11]);

                if(fabs(a1-a[i][11])>0.001)flag++;

        }

        count++;

    }while(flag);

    //output

    for(i=0;i<=8;i++){

        for(j=0;j<=n;j++){

            printf("%8.4f",a[i][j]);

        }

        printf("\n");

    }

    for(i=9;i<=74;i++){

        for(j=6;j<=n;j++){

            printf("%8.4f",a[i][j]);

        }

        printf("\n");

    }

    //file

    if((fp=fopen("kk4.txt","w+"))==NULL){

        printf("\nCannot open this file. \n");

        exit(0);

    }

    fputs("\n\n\n",fp);

    for(i=0;i<=8;i++){

        for(j=0;j<=n;j++){

            fprintf(fp,"%6.2f ",a[i][j]);

        }

        fputs("\n",fp);

    }

     for(i=9;i<=74;i++){

        for(j=6;j<=n;j++){

            fprintf(fp,"%6.2f ",a[i][j]);

        }

        fputs("\n",fp);

    }

    fclose(fp);

}