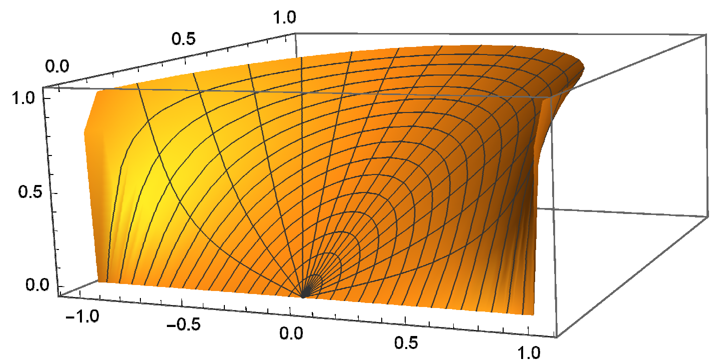
Code Part

Analytic Solution to the Dirichlet Problem

In[*]:=
```
AnalyticSolution =
  Function[{r, θ}, 2/π * ArcTan[(1 + r)/(1 - r) * Tan[θ/2]] + 2/π * ArcTan[(1 + r)/(1 - r) * Cot[θ/2]] - 1];
  纯函数              反正切               正切                   反正切              余切

ParametricPlot3D[{r * Cos[θ], r * Sin[θ], AnalyticSolution[r, θ]},
绘制三维参数图          余弦          正弦
 {r, 0, 1}, {θ, 0, π}, BoxRatios → {2.5, 2.5, 1}]
                       边界框比例
```

Out[*]=

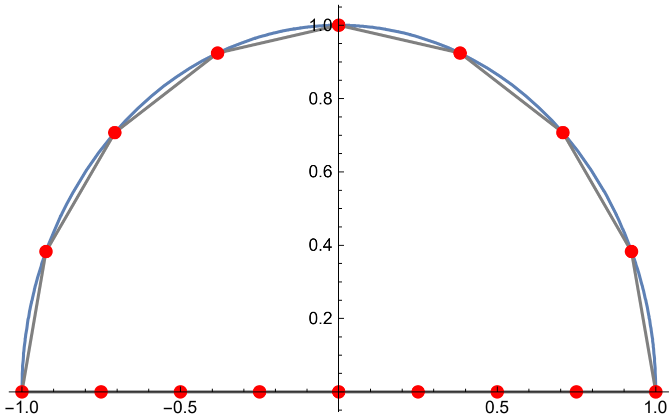


Rough Idea of Dividing the Boundary

In[*]:=
```
DivisionPointsP = {};
For[i = 0, i ≤ 8, i++,
For循环
 DivisionPointsP = Insert[DivisionPointsP,
                   插入
   N[CoordinateTransform["Polar" → "Cartesian", {1, i/8 π}], 10], -1]
   ⋯ 坐标变换
]
DivisionPointsP = Join[DivisionPointsP,
                  连接
   {{-0.75, 0 }, {-0.5, 0}, {-0.25, 0}, {0, 0}, {0.25, 0}, {0.5, 0}, {0.75, 0}, {1, 0}}];
```

In[ ]:=
```
Show[Plot[Sqrt[1 - x^2], {x, -1, 1}],
显示  绘图  平方根
 ListLinePlot[DivisionPointsP, PlotStyle → {PointSize[Large], Gray}],
 绘制点集的线条                     绘图样式      点的大小    大      灰色
 ListPlot[DivisionPointsP, PlotStyle → {PointSize[Large], Red}]]
                           绘图样式      点的大小    大      红色
```

Out[ ]=



Implementation of Boundary Element Methods

In[1]:=
```
NumberDivideP = 40;
TotalNumberPointsP = 3 * NumberDivideP;
```

In[3]:=
```
PointsOnLineP = {};
For[i = 0, i ≤ NumberDivideP - 1, i++,
For循环
  PointsOnLineP = Insert[PointsOnLineP, -1 + (2 * i)/(NumberDivideP), -1]
                  插入
 ];
PointsOnCurveP = {};
For[i = 0, i ≤ 2 * NumberDivideP, i++,
For循环
  PointsOnCurveP = Insert[PointsOnCurveP, (i)/(2 * NumberDivideP), -1]
                   插入
 ];
PointsXP = PointsOnLineP;
For[i = 1, i ≤ Length[PointsOnCurveP], i++,
For循环          长度
  PointsXP = Insert[PointsXP, N[Cos[PointsOnCurveP[[i]] * π]], -1]
             插入              ·· 余弦
 ];
PointsYP = ConstantArray[0, Length[PointsOnLineP]];
           常量数组         长度
For[i = 1, i ≤ Length[PointsOnCurveP], i++,
                长度
  PointsYP = Insert[PointsYP, N[Sin[PointsOnCurveP[[i]] * π]], -1]
             插入              ·· 正弦
```

```
 ];
SegmentsLenP = {};
For[i = 1, i ≤ TotalNumberPointsP, i++,
For循环
  SegmentsLenP = Insert[SegmentsLenP,
                 插入
    Norm[N[{PointsXP[[i + 1]] - PointsXP[[i]], PointsYP[[i + 1]] - PointsYP[[i]]}]], -1]
         数值运算
 ];
NormalPointsXP = {};
For[i = 1, i ≤ TotalNumberPointsP, i++,
For循环
  NormalPointsXP = Insert[NormalPointsXP, (PointsYP[[i + 1]] - PointsYP[[i]]) / SegmentsLenP[[i]], -1]
                   插入
 ];
NormalPointsYP = {};
For[i = 1, i ≤ TotalNumberPointsP, i++,
For循环
  NormalPointsYP = Insert[NormalPointsYP, -(PointsXP[[i + 1]] - PointsXP[[i]]) / SegmentsLenP[[i]], -1]
                   插入
 ];
```

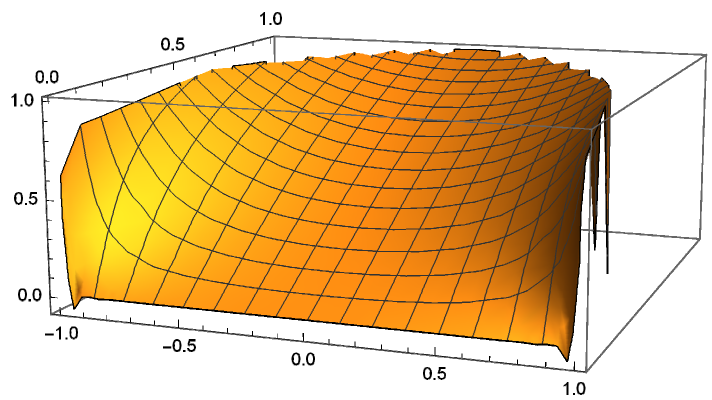
In[ ]:=
```
FP = Function[{i, k, x, y},
   FA = Function[kk, SegmentsLenP[[kk]]^2][k];
   FB = Function[{kk, xx, yy},
      (-NormalPointsYP[[kk]] * (PointsXP[[kk]] - xx) + NormalPointsXP[[kk]] *
          (PointsYP[[kk]] - yy)) * 2 * SegmentsLenP[[kk]]][k, x, y];
   FE = Function[{kk, xx, yy}, (PointsXP[[kk]] - xx)^2 + (PointsYP[[kk]] - yy)^2][k, x, y];
   Decision = 4 * FA * FE - FB^2;
   If[i == 1,

    Re[If[Decision == 0, SegmentsLenP[[k]]/(2 * π) * (Log[SegmentsLenP[[k]]] +

          (1 + FB/(2 * FA)) * Log[Abs[1 + FB/(2 * FA)]] - FB/(2 * FA) * Log[Abs[FB/(2 * FA)]] - 1),

       SegmentsLenP[[k]]/(4 * π) * (2 * (Log[SegmentsLenP[[k]]] - 1) -

          FB/(2 * FA) * Log[Abs[FE/FA]] + (1 + FB/(2 * FA)) * Log[Abs[1 + (FB + FE)/FA]] +

          Sqrt[Decision]/FA * (ArcTan[(2 * FA + FB)/Sqrt[Decision]] - ArcTan[FB/Sqrt[Decision]]))

      ]]
    , Re[If[Decision == 0, 0,

      SegmentsLenP[[k]] * ((NormalPointsXP[[k]] * (PointsXP[[k]] - x) +
            NormalPointsYP[[k]] * (PointsYP[[k]] - y)) / (π * Sqrt[Decision])) *

        (ArcTan[(2 * FA + FB)/Sqrt[Decision]] - ArcTan[FB/Sqrt[Decision]])

      ]]
    ]
   ];

MidPointsXP = PointsOnLineP + 1/(NumberDivideP * 2);
For[i = 1, i ≤ Length[PointsOnCurveP], i++,
 MidPointsXP =
```

```
  Insert[MidPointsXP, Cos[(PointsOnCurveP[[i]] + 1/(NumberDivideP * 4)) * π], -1]
 ];
MidPointsYP = ConstantArray[0, Length[PointsOnLineP]];
For[i = 1, i ≤ Length[PointsOnCurveP], i++,
 MidPointsYP =
  Insert[MidPointsYP, Sin[(PointsOnCurveP[[i]] + 1/(NumberDivideP * 4)) * π], -1]
 ];
BoundsP = {};
For[i = 1, i ≤ TotalNumberPointsP, i++,
 BoundsP =
   Insert[BoundsP, N[If[N[MidPointsXP[[i]]^2 + MidPointsYP[[i]]^2] == 1, 1, 0]], -1];
 ];
FactorXP = {};
For[i = 1, i ≤ TotalNumberPointsP, i++,
 FactorXP = Insert[FactorXP, (PointsXP[[i + 1]] - PointsXP[[i]])/2 + PointsXP[[i]], -1];
 ];
FactorYP = {};
For[i = 1, i ≤ TotalNumberPointsP, i++,
 FactorYP = Insert[FactorYP, (PointsYP[[i + 1]] - PointsYP[[i]])/2 + PointsYP[[i]], -1];
 ];
aP = {};
For[i = 1, i ≤ TotalNumberPointsP, i++,
 Temp = {};
 For[j = 1, j ≤ TotalNumberPointsP, j++,
  Temp = Insert[Temp, N[-FP[1, j, FactorXP[[i]], FactorYP[[i]]]], -1];
 ];
 aP = Insert[aP, Temp, -1];
 ];
bP = {};
```

```
For[i = 1, i ≤ TotalNumberPointsP, i++,
 Temp = {};
 For[j = 1, j ≤ TotalNumberPointsP, j++,
  Temp = Insert[Temp, N[BoundsP[[j]] * (-FP[2, j, FactorXP[[i]], FactorYP[[i]]] +
        Function[{ii, jj}, If[ii == jj, 1, 0]][i, j] / 2)], -1];
 ];
 bP = Insert[bP, Sum[Temp[[k]], {k, 1, TotalNumberPointsP}], -1];
];
zP = LinearSolve[aP, bP];
BoundsUP = {};
For[i = 1, i ≤ TotalNumberPointsP, i++,
 BoundsUP = Insert[BoundsUP, N[BoundsP[[i]]], -1];
];
BoundsNP = {};
For[i = 1, i ≤ TotalNumberPointsP, i++,
 BoundsNP = Insert[BoundsNP, zP[[i]], -1];
];
```

```
In[ ]:= BEMP = Function[{x, y}, Sum[BoundsUP[[i]] * FP[2, i, x, y] -
      BoundsNP[[i]] * FP[1, i, x, y] , {i, 1, TotalNumberPointsP}]];
```

```
In[ ]:= Plot3D[BEMP[x, y], {x, -1, 1}, {y, 0, Sqrt[1 - x^2]}]
```

Out[ ]=


Comparison Between Boundary Element Solution and Analytic Solution

```
In[ ]:= PointsPA = {{0.10, 0.20}, {0.10, 0.30}, {0.1, 0.40}, {0.50, 0.20},
    {0.50, 0.30}, {0.50, 0.40}, {0.90, 0.20}, {0.90, 0.30}, {0.90, 0.40}};
```

In[ ]:=
```
ValueP120 = {};
For[i = 1, i ≤ Length[PointsPA], i++,
 ValueP120 = Insert[ValueP120, BEMP[PointsPA[[i, 1]], PointsPA[[i, 2]]], -1]
]
ValueP120
```

Out[ ]=
```
{0.253738, 0.374381, 0.488345, 0.326665,
 0.469776, 0.595548, 0.771894, 0.895153, 0.976404}
```

In[ ]:=
```
ValueP60 = {};
For[i = 1, i ≤ Length[PointsPA], i++,
 ValueP60 = Insert[ValueP60, BEMP[PointsPA[[i, 1]], PointsPA[[i, 2]]], -1]
]
ValueP60
```

Out[ ]=
```
{0.253818, 0.374508, 0.488517, 0.326754,
 0.469958, 0.595803, 0.773119, 0.896111, 0.977072}
```

In[ ]:=
```
PointsPAPolar = {};
For[i = 1, i ≤ Length[PointsPA], i++,
 PointsPAPolar = Insert[PointsPAPolar,
   CoordinateTransform["Cartesian" → "Polar", PointsPA[[i]] ], -1]
]
PointsPAPolar
```

Out[ ]=
```
{{0.223607, 1.10715}, {0.316228, 1.24905}, {0.412311, 1.32582},
 {0.538516, 0.380506}, {0.583095, 0.54042}, {0.640312, 0.674741},
 {0.921954, 0.218669}, {0.948683, 0.321751}, {0.984886, 0.418224}}
```

In[ ]:=
```
ValueExact = {};
For[i = 1, i ≤ Length[PointsPAPolar], i++,
 ValueExact = Insert[ValueExact,
   AnalyticSolution[PointsPAPolar[[i, 1]], PointsPAPolar[[i, 2]]], -1]
]
ValueExact
```

Out[ ]=
```
{0.253707, 0.374334, 0.488284, 0.326623,
 0.469708, 0.595458, 0.771599, 0.894863, 0.976138}
```

In[ ]:=
```
PointsPA2 = {{0.1, 0.95}, {0.1, 0.96}, {0.1, 0.97}, {0.1, 0.98}, {0.1, 0.99}}
```

Out[ ]=
```
{{0.1, 0.95}, {0.1, 0.96}, {0.1, 0.97}, {0.1, 0.98}, {0.1, 0.99}}
```

In[ ]:=
```
PointsPAPolar2 = {};
For[i = 1, i ≤ Length[PointsPA2], i++,
 PointsPAPolar2 = Insert[PointsPAPolar2,
   CoordinateTransform["Cartesian" → "Polar", PointsPA2[[i]] ], -1]
]
PointsPAPolar2
```

Out[ ]=
```
{{0.955249, 1.46592}, {0.965194, 1.467},
 {0.975141, 1.46807}, {0.985089, 1.46911}, {0.995038, 1.47013}}
```

In[ ]:=
```
ValueExact2 = {};
For[i = 1, i ≤ Length[PointsPAPolar2], i++,
 ValueExact2 = Insert[ValueExact2,
   AnalyticSolution[PointsPAPolar2[[i, 1]], PointsPAPolar2[[i, 2]]], -1]
]
ValueExact2
```

Out[ ]=
```
{0.970703, 0.97733, 0.983891, 0.990386, 0.996817}
```

In[ ]:=
```
ValueP1202 = {};
For[i = 1, i ≤ Length[PointsPA2], i++,
 ValueP1202 = Insert[ValueP1202, BEMP[PointsPA2[[i, 1]], PointsPA2[[i, 2]]], -1]
]
ValueP1202
```

Out[ ]=
```
{0.970805, 0.977433, 0.983994, 0.990491, 0.996929}
```

In[ ]:=
```
(ValueP1202 - ValueExact2 )
```

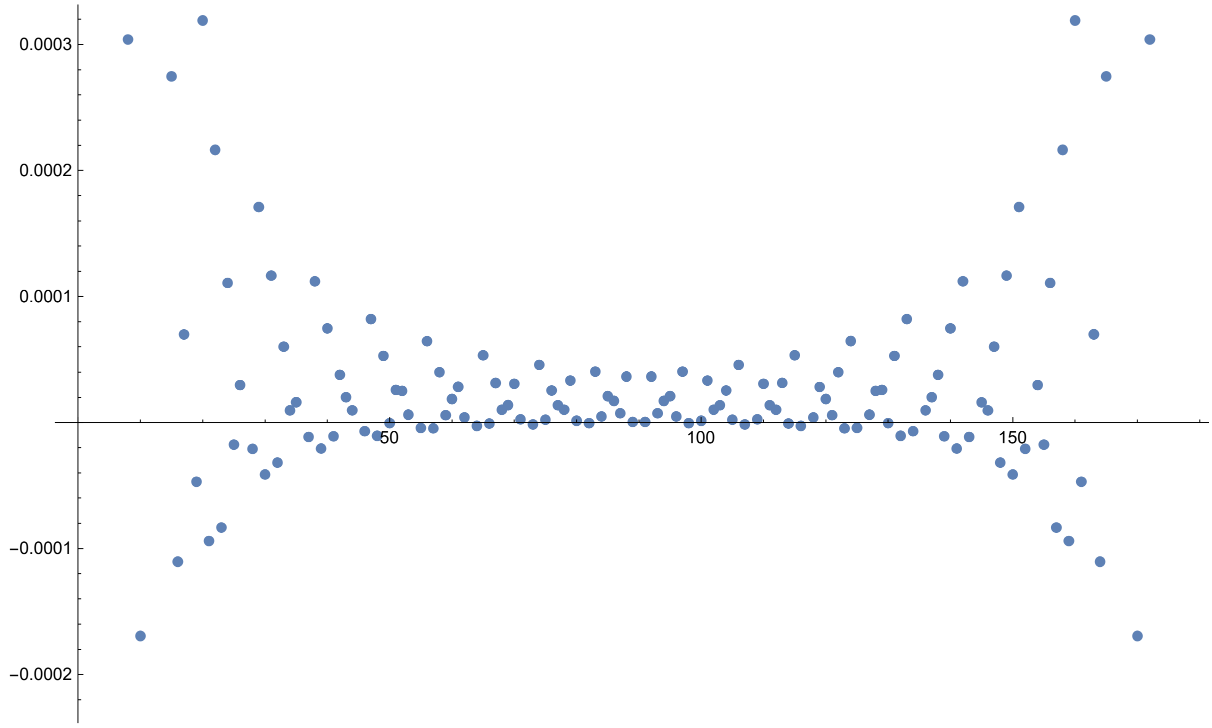
Out[ ]=
```
{0.000102288, 0.000102576, 0.000103059, 0.00010473, 0.000112379}
```

In[ ]:=
```
PointsPA3 = {};
For[i = 1, i ≤ 180, i++,
 PointsPA3 = Insert[PointsPA3,
   N[CoordinateTransform["Polar" → "Cartesian", {1, i/180 π}], 10], -1]
]
```

In[ ]:=
```
ValueP1203 = {};
For[i = 1, i ≤ Length[PointsPA3], i++,
 ValueP1203 = Insert[ValueP1203, BEMP[PointsPA3[[i, 1]], PointsPA3[[i, 2]]], -1]
]
```

In[ ]:=
```
Show[ListPlot[ValueP1203], ListPlot[ConstantArray[1, 180]]]
显示  绘制点集                绘制点集      常量数组
```

Out[ ]=

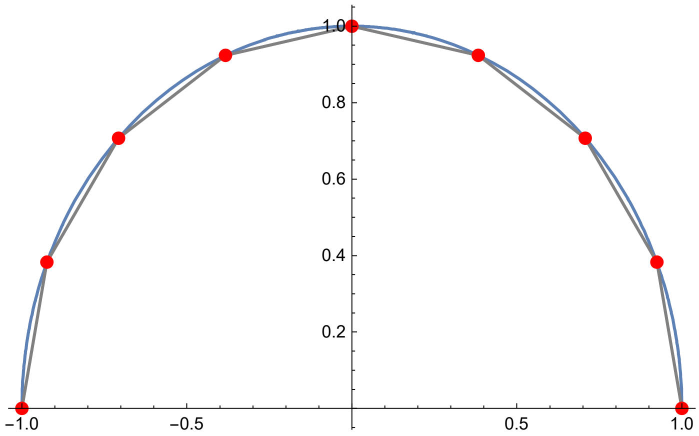


Rough Idea of Dividing Boundary When Applying BEM Using Green Function

In[ ]:=
```
DivisionPointsG = {};
For[i = 0, i ≤ 8, i++,
For循环

 DivisionPointsG = Insert[DivisionPointsG,
                   插入

   N[CoordinateTransform["Polar" → "Cartesian", {1, i/8 π}], 10], -1]
   …坐标变换

]
```

In[*]:=
```
Show[Plot[Sqrt[1 - x^2], {x, -1, 1}],
显示  绘图  平方根
 ListLinePlot[DivisionPointsG, PlotStyle → {PointSize[Large], Gray}],
 绘制点集的线条  绘图样式  点的大小  大  灰色
 ListPlot[DivisionPointsG, PlotStyle → {PointSize[Large], Red}]]
 绘图样式  点的大小  大  红色
```

Out[*]=



Implementation of BEM Using Green Function

In[*]:=
```
NumberDivideG = 60;
TotalNumberPointsG = 2 * NumberDivideG;
```

```
PointsOnCurveG = {};
For[i = 0, i ≤ 2 * NumberDivideG, i++,
  PointsOnCurveG = Insert[PointsOnCurveG, i/(2 * NumberDivideG), -1]
 ];
PointsXG = {};
For[i = 1, i ≤ Length[PointsOnCurveG], i++,
  PointsXG = Insert[PointsXG, N[Cos[PointsOnCurveG[[i]] * π]], -1]
 ];
PointsYG = {};
For[i = 1, i ≤ Length[PointsOnCurveG], i++,
  PointsYG = Insert[PointsYG, N[Sin[PointsOnCurveG[[i]] * π]], -1]
 ];
SegmentsLenG = {};
For[i = 1, i ≤ TotalNumberPointsG, i++,
  SegmentsLenG = Insert[SegmentsLenG,
    Norm[N[{PointsXG[[i + 1]] - PointsXG[[i]], PointsYG[[i + 1]] - PointsYG[[i]]}]], -1]
 ];
NormalPointsXG = {};
For[i = 1, i ≤ TotalNumberPointsG, i++,
  NormalPointsXG = Insert[NormalPointsXG, (PointsYG[[i + 1]] - PointsYG[[i]])/SegmentsLenG[[i]], -1]
 ];
NormalPointsYG = {};
For[i = 1, i ≤ TotalNumberPointsG, i++,
  NormalPointsYG = Insert[NormalPointsYG, -(PointsXG[[i + 1]] - PointsXG[[i]])/SegmentsLenG[[i]], -1]
 ];
```

```
FG = Function[{i, k, x, y},
     纯函数
  FA = Function[kk, SegmentsLenG[[kk]]^2][k];
       纯函数
  FB = Function[{kk, xx, yy},
       纯函数
     (-NormalPointsYG[[kk]] * (PointsXG[[kk]] - xx) + NormalPointsXG[[kk]] *
         (PointsYG[[kk]] - yy)) * 2 * SegmentsLenG[[kk]]][k, x, y];
  FE = Function[{kk, xx, yy}, (PointsXG[[kk]] - xx)^2 + (PointsYG[[kk]] - yy)^2][k, x, y];
       纯函数
  Decision = 4 * FA * FE - FB^2;
  If[i == 1,
  如果
   Re[If[Decision == 0, SegmentsLenG[[k]]/(2 * π) * (Log[SegmentsLenG[[k]]] +
   …  如果                                            对数
       (1 + FB/(2 * FA)) * Log[Abs[1 + FB/(2 * FA)]] - FB/(2 * FA) * Log[Abs[FB/(2 * FA)]] - 1),
                           对数 绝对值                                 对数 绝对值
     SegmentsLenG[[k]]/(4 * π) * (2 * (Log[SegmentsLenG[[k]]] - 1) -
                                       对数
       FB/(2 * FA) * Log[Abs[FE/FA]] + (1 + FB/(2 * FA)) * Log[Abs[1 + (FB + FE)/FA]] +
                     对数 绝对值                          对数 绝对值
       Sqrt[Decision]/FA * (ArcTan[(2 * FA + FB)/Sqrt[Decision ]] - ArcTan[FB/Sqrt[Decision ]]))
                            反正切                                   反正切
    ]]
   , Re[If[Decision == 0, 0,
     …  如果
     SegmentsLenG[[k]] * ((NormalPointsXG[[k]] * (PointsXG[[k]] - x) +
          NormalPointsYG[[k]] * (PointsYG[[k]] - y)) / (π * Sqrt[Decision ])) *
                                                              平方根
      (ArcTan[(2 * FA + FB)/Sqrt[Decision ]] - ArcTan[FB/Sqrt[Decision ]])
       反正切                                   反正切
    ]]
  ]
 ];
MidPointsXG = {};
For[i = 1, i ≤ Length[PointsOnCurveG], i++,
For循环       长度
  MidPointsXG =
```

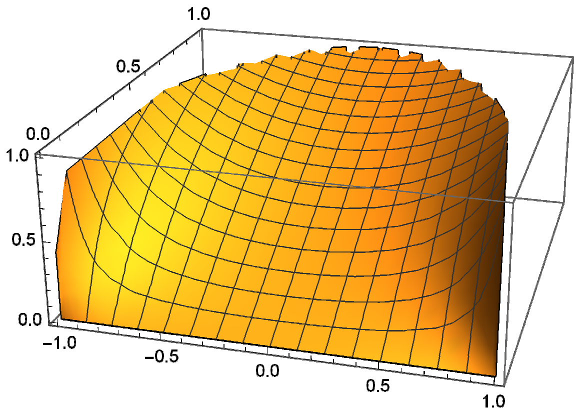
```
  Insert[MidPointsXG, Cos[(PointsOnCurveG[[i]] + 1/(NumberDivideG * 4)) * π], -1]
 ];
MidPointsYG = {};
For[i = 1, i ≤ Length[PointsOnCurveG], i++,
 MidPointsYG =
  Insert[MidPointsYG, Sin[(PointsOnCurveG[[i]] + 1/(NumberDivideG * 4)) * π], -1]
 ];
BoundsG = {};
For[i = 1, i ≤ TotalNumberPointsG, i++,
 BoundsG = Insert[BoundsG, 1, -1];
 ];
FactorXG = {};
For[i = 1, i ≤ TotalNumberPointsG, i++,
 FactorXG = Insert[FactorXG, (PointsXG[[i + 1]] - PointsXG[[i]])/2 + PointsXG[[i]], -1];
 ];
FactorYG = {};
For[i = 1, i ≤ TotalNumberPointsG, i++,
 FactorYG = Insert[FactorYG, (PointsYG[[i + 1]] - PointsYG[[i]])/2 + PointsYG[[i]], -1];
 ];
aG = {};
For[i = 1, i ≤ TotalNumberPointsG, i++,
 Temp = {};
 For[j = 1, j ≤ TotalNumberPointsG, j++,
  Temp = Insert[Temp, N[- (FG[1, j, FactorXG[[i]], FactorYG[[i]]] -
        FG[1, j, FactorXG[[i]], -FactorYG[[i]]])], -1];
 ];
 aG = Insert[aG, Temp, -1];
 ];
bG = {};
For[i = 1, i ≤ TotalNumberPointsG, i++,
```

```
For循环
    Temp = {};
    For[j = 1, j ≤ TotalNumberPointsG, j++,
    For循环
     Temp = Insert[Temp, N[BoundsG[[j]] *
            插入          数值运算
          (-FG[2, j, FactorXG[[i]], FactorYG[[i]]] + FG[2, j, FactorXG[[i]],
             -FactorYG[[i]]] + Function[{ii, jj}, If[ii == jj, 1, 0]][i, j] / 2)], -1];
                               纯函数               如果
    ];
    bG = Insert[bG, Sum[Temp[[k]], {k, 1, TotalNumberPointsG}], -1];
         插入       求和
   ];
  zG = LinearSolve[aG, bG];
       线性求解
  BoundsUG = {};
  For[i = 1, i ≤ TotalNumberPointsG, i++,
  For循环
    BoundsUG = Insert[BoundsUG, N[BoundsG[[i]]], -1];
               插入             数值运算
   ];
  BoundsNG = {};
  For[i = 1, i ≤ TotalNumberPointsG, i++,
  For循环
    BoundsNG = Insert[BoundsNG, zG[[i]], -1];
               插入
   ];
```

```
In[ ]:= BEMG = Function[{x, y}, Sum[BoundsUG[[i]] * (FG[2, i, x, y] - FG[2, i, x, -y]) -
               纯函数              求和
          BoundsNG[[i]] * (FG[1, i, x, y] - FG[1, i, x, -y]) , {i, 1, TotalNumberPointsG}]];
```

```
In[ ]:= Plot3D[BEMG[x, y], {x, -1, 1}, {y, 0, Sqrt[1 - x^2]}]
        绘制三维图形                                  平方根
```

Out[ ]=



Comparison Between BES Using Green Function and Analytic Solution

```
In[ ]:= PointsGA = {{0.10, 0.20}, {0.10, 0.30}, {0.1, 0.40}, {0.50, 0.20},
          {0.50, 0.30}, {0.50, 0.40}, {0.90, 0.20}, {0.90, 0.30}, {0.90, 0.40}};
```

In[ ]:=
```
ValueG60 = {};
For[i = 1, i ≤ Length[PointsGA], i++,
 ValueG60 = Insert[ValueG60, BEMG[PointsGA[[i, 1]], PointsGA[[i, 2]]], -1]
]
ValueG60
```

Out[ ]=
```
{0.253793, 0.374455, 0.488432, 0.326824,
 0.469954, 0.595719, 0.772824, 0.895567, 0.976611}
```

In[ ]:=
```
ValueG120 = {};
For[i = 1, i ≤ Length[PointsGA], i++,
 ValueG120 = Insert[ValueG120, BEMG[PointsGA[[i, 1]], PointsGA[[i, 2]]], -1]
]
ValueG120
```

Out[ ]=
```
{0.253729, 0.374364, 0.488321, 0.326673,
 0.469769, 0.595523, 0.771899, 0.895035, 0.976256}
```

In[ ]:=
```
PointsGA2 = {{0.10, 0.95}, {0.10, 0.96}, {0.10, 0.97}, {0.10, 0.98}, {0.10, 0.99}}
```

Out[ ]=
```
{{0.1, 0.95}, {0.1, 0.96}, {0.1, 0.97}, {0.1, 0.98}, {0.1, 0.99}}
```

In[ ]:=
```
ValueG802 = {};
For[i = 1, i ≤ Length[PointsGA2], i++,
 ValueG802 = Insert[ValueG802, BEMG[PointsGA2[[i, 1]], PointsGA2[[i, 2]]], -1]
]
ValueG802
```

Out[ ]=
```
{0.970807, 0.977434, 0.983995, 0.990492, 0.99693}
```

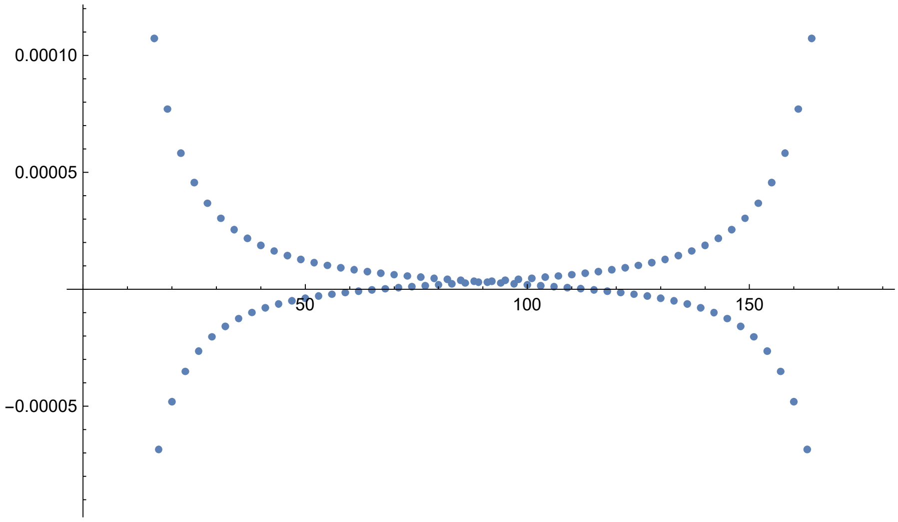
In[ ]:=
```
ValueG802 - ValueExact2
```

Out[ ]=
```
{0.000104368, 0.000104222, 0.000104274, 0.000105507, 0.000112669}
```

In[ ]:=
```
ValueG1203 = {};
For[i = 1, i ≤ Length[PointsPA3], i++,
 ValueG1203 = Insert[ValueG1203, BEMG[PointsPA3[[i, 1]], PointsPA3[[i, 2]]], -1]
]
```

In[ ]:=
```
Show[ListPlot[ValueG1203]]
显示  绘制点集
```

Out[ ]=



Comparison Between BES and BES Using Green Function

In[ ]:=
```
PointsGP = {{0.10, 0.85}, {0.10, 0.88}, {0.10, 0.90}, {0.10, 0.93}, {0.10, 0.95}};
```

In[ ]:=
```
ValueG1204 = {};
For[i = 1, i ≤ Length[PointsGP], i++,
For循环          长度
 ValueG1204 = Insert[ValueG1204, BEMG[PointsGP[[i, 1]], PointsGP[[i, 2]]], -1];
                插入
]
ValueG1204
```

Out[ ]= {0.900688, 0.922447, 0.936596, 0.957293, 0.970749}

In[ ]:=
```
ValueP1204 = {};
For[i = 1, i ≤ Length[PointsGP], i++,
For循环          长度
 ValueP1204 = Insert[ValueP1204, BEMP[PointsGP[[i, 1]], PointsGP[[i, 2]]], -1];
                插入
]
ValueP1204
```

Out[ ]= {0.90074, 0.922501, 0.93665, 0.957348, 0.970805}

In[ ]:=
```
PointsGPPolar = {};
For[i = 1, i ≤ Length[PointsGP], i++,
For循环          长度
 PointsGPPolar = Insert[PointsGPPolar,
                  插入
   CoordinateTransform["Cartesian" → "Polar", PointsGP[[i]]], -1]
   坐标变换
]
PointsGPPolar
```

Out[ ]= {{0.855862, 1.45369}, {0.885664, 1.45765},
 {0.905539, 1.46014}, {0.935361, 1.46368}, {0.955249, 1.46592}}

In[ ]:=
```
ValueExactGP = {};
For[i = 1, i ≤ Length[PointsGPPolar], i++,
└For循环      └长度
 ValueExactGP = Insert[ValueExactGP,
                └插入
    AnalyticSolution[PointsGPPolar[[i, 1]], PointsGPPolar[[i, 2]]], -1];
]
ValueExactGP
```

Out[ ]= {0.900641, 0.922401, 0.936549, 0.957247, 0.970703}

In[ ]:=
```
ErrorG120 = ValueG1204 - ValueExactGP
```

Out[ ]= {0.0000468309, 0.0000467238, 0.0000466232, 0.0000464328, 0.0000462819}

In[ ]:=
```
ErrorP120 = ValueP1204 - ValueExactGP
```

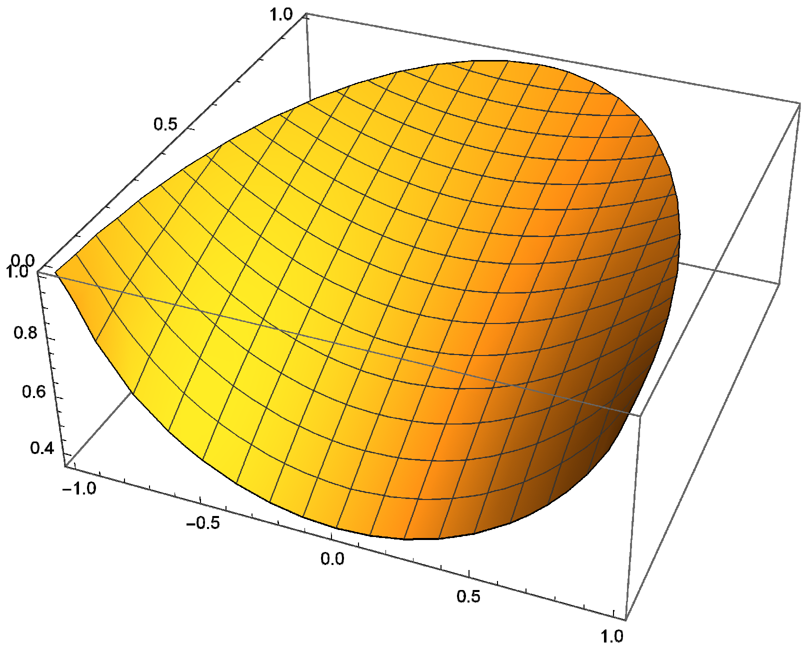
Out[ ]= {0.0000989915, 0.00010014, 0.000100826, 0.000101743, 0.000102288}

Approximate Real Solution to Mixed Boundary Value Problem

In[ ]:=
```
AnalyticSolution2 = NDSolveValue[{-∇²{x,y} u[x, y] == NeumannValue[-1, y == 0],
                    └数值解的值                            └诺伊曼边值
   DirichletCondition[u[x, y] == 1, x^2 + y^2 == 1]},
   └狄里克雷条件
  u, {x, y} ∈ Disk[{0, 0}, 1, {0, π}]]
                └圆盘
```

Out[ ]= InterpolatingFunction[Domain: {{-1., 1.}, {0., 1.}} Output: scalar]

In[ ]:=
```
Plot3D[AnalyticSolution2[x, y], {x, y} ∈ Disk[{0, 0}, 1, {0, π}]]
└绘制三维图形                                 └圆盘
```

Out[ ]=


Implementation of BEM to Solve Mixed Boundary Value Problem

In[ ]:=
```
NumberDivideM = 40;
TotalNumberPointsM = 3 * NumberDivideM;

PointsOnLineM = {};
For[i = 0, i ≤ NumberDivideM - 1, i++,
For循环

  PointsOnLineM = Insert[PointsOnLineM, -1 + (2 * i)/(NumberDivideM), -1]
                  插入
 ];
PointsOnCurveM = {};
For[i = 0, i ≤ 2 * NumberDivideM, i++,
For循环

  PointsOnCurveM = Insert[PointsOnCurveM, (i)/(2 * NumberDivideM), -1]
                   插入
 ];
PointsXM = PointsOnLineM;
For[i = 1, i ≤ Length[PointsOnCurveM], i++,
For循环      长度
  PointsXM = Insert[PointsXM, N[Cos[PointsOnCurveM[[i]] * π]], -1]
             插入               ·· 余弦
 ];
PointsYM = ConstantArray[0, Length[PointsOnLineM]];
           常量数组          长度
For[i = 1, i ≤ Length[PointsOnCurveM], i++,
             长度
  PointsYM = Insert[PointsYM, N[Sin[PointsOnCurveM[[i]] * π]], -1]
             插入               ·· 正弦
 ];
SegmentsLenM = {};
For[i = 1, i ≤ TotalNumberPointsM, i++,
For循环
  SegmentsLenM = Insert[SegmentsLenM,
                 插入
    Norm[N[{PointsXM[[i + 1]] - PointsXM[[i]], PointsYM[[i + 1]] - PointsYM[[i]]}]], -1]
         数值运算
 ];
NormalPointsXM = {};
For[i = 1, i ≤ TotalNumberPointsM, i++,
For循环

  NormalPointsXM = Insert[NormalPointsXM, (PointsYM[[i + 1]] - PointsYM[[i]])/(SegmentsLenM[[i]]), -1]
                   插入
 ];
NormalPointsYM = {};
```

```
For[i = 1, i ≤ TotalNumberPointsM, i++,
For循环
  NormalPointsYM = Insert[NormalPointsYM, -(PointsXM[[i + 1]] - PointsXM[[i]])/SegmentsLenM[[i]], -1]
                   插入
 ];
```

```
In[]:= FM = Function[{i, k, x, y},

   FA = Function[kk, SegmentsLenM[[kk]]^2][k];

   FB = Function[{kk, xx, yy},
      (-NormalPointsYM[[kk]] * (PointsXM[[kk]] - xx) + NormalPointsXM[[kk]] *
         (PointsYM[[kk]] - yy)) * 2 * SegmentsLenM[[kk]]][k, x, y];
   FE = Function[{kk, xx, yy}, (PointsXM[[kk]] - xx)^2 + (PointsYM[[kk]] - yy)^2][k, x, y];

   Decision = 4 * FA * FE - FB^2;

   If[i == 1,

    Re[If[Decision == 0, SegmentsLenM[[k]]/(2 * π) * (Log[SegmentsLenM[[k]]] +

          (1 + FB/(2 * FA)) * Log[Abs[1 + FB/(2 * FA)]] - FB/(2 * FA) * Log[Abs[FB/(2 * FA)]] - 1),

        SegmentsLenM[[k]]/(4 * π) * (2 * (Log[SegmentsLenM[[k]]] - 1) -

          FB/(2 * FA) * Log[Abs[FE/FA]] + (1 + FB/(2 * FA)) * Log[Abs[1 + (FB + FE)/FA]] +

          Sqrt[Decision]/FA * (ArcTan[(2 * FA + FB)/Sqrt[Decision ]] - ArcTan[FB/Sqrt[Decision ]]))

     ]]
    , Re[If[Decision == 0, 0,

      SegmentsLenM[[k]] * ((NormalPointsXM[[k]] * (PointsXM[[k]] - x) +
           NormalPointsYM[[k]] * (PointsYM[[k]] - y)) / (π * Sqrt[Decision ])) *

       (ArcTan[(2 * FA + FB)/Sqrt[Decision ]] - ArcTan[FB/Sqrt[Decision ]])

     ]]
    ]
   ];

MidPointsXM = PointsOnLineM + 1/(NumberDivideM * 2);

For[i = 1, i ≤ Length[PointsOnCurveM], i++,

  MidPointsXM =
```

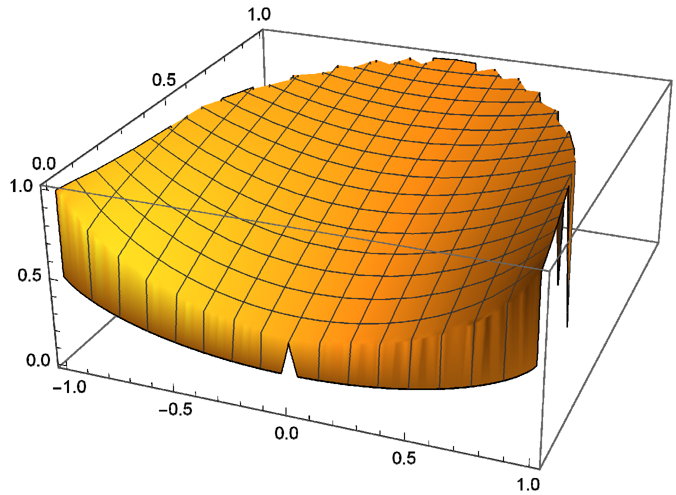
```
    Insert[MidPointsXM, Cos[(PointsOnCurveM[[i]] + 1/(NumberDivideM * 4)) * π], -1]
    插入                 余弦
  ];
MidPointsYM = ConstantArray[0, Length[PointsOnLineM]];
              常量数组          长度
For[i = 1, i ≤ Length[PointsOnCurveM], i++,
For循环          长度
   MidPointsYM =
    Insert[MidPointsYM, Sin[(PointsOnCurveM[[i]] + 1/(NumberDivideM * 4)) * π], -1]
    插入                 正弦
  ];
BoundsM = {};
For[i = 1, i ≤ NumberDivideM, i++,
For循环
   BoundsM = Insert[BoundsM, -1, -1];
             插入
  ];
For[i = 1, i ≤ 2 * NumberDivideM, i++,
For循环
   BoundsM = Insert[BoundsM, 1, -1];
             插入
  ];
FactorXM = {};
For[i = 1, i ≤ TotalNumberPointsM, i++,
For循环
   FactorXM = Insert[FactorXM, (PointsXM[[i + 1]] - PointsXM[[i]])/2 + PointsXM[[i]], -1];
              插入
  ];
FactorYM = {};
For[i = 1, i ≤ TotalNumberPointsM, i++,
For循环
   FactorYM = Insert[FactorYM, (PointsYM[[i + 1]] - PointsYM[[i]])/2 + PointsYM[[i]], -1];
              插入
  ];
aM = {};
For[i = 1, i ≤ TotalNumberPointsM, i++,
For循环
   Temp = {};
   For[j = 1, j ≤ NumberDivideM, j++,
   For循环
    Temp = Insert[Temp, N[FM[2, j, FactorXM[[i]], FactorYM[[i]]] -
           插入         数值运算
         Function[{ii, jj}, If[ii == jj, 1, 0]][i, j] / 2], -1];
                            如果
   ];
```

```
    For[j = NumberDivideM + 1, j ≤ NumberDivideM * 3, j++,
    For循环
     Temp = Insert[Temp, N[-FM[1, j, FactorXM[[i]], FactorYM[[i]]]], -1];
            插入          数值运算
    ];
    aM = Insert[aM, Temp, -1];
         插入
   ];
  bM = {};
  For[i = 1, i ≤ TotalNumberPointsM, i++,
  For循环
    Temp = {};
    For[j = 1, j ≤ NumberDivideM, j++,
    For循环
     Temp = Insert[Temp, N[BoundsM[[j]] * FM[1, j, FactorXM[[i]], FactorYM[[i]]]], -1];
            插入          数值运算
    ];
    For[j = NumberDivideM + 1, j ≤ NumberDivideM * 3, j++,
    For循环
     Temp = Insert[Temp, N[BoundsM[[j]] * (-FM[2, j, FactorXM[[i]], FactorYM[[i]]] +
            插入          数值运算
             Function[{ii, jj}, If[ii == jj, 1, 0]][i, j] / 2)], -1];
                                如果
    ];
    bM = Insert[bM, Sum[Temp[[k]], {k, 1, TotalNumberPointsM}], -1];
         插入        求和
   ];
  zM = LinearSolve[aM, bM];
       线性求解
  BoundsUM = {};
  For[i = 1, i ≤ NumberDivideM, i++,
  For循环
    BoundsUM = Insert[BoundsUM, zM[[i]], -1];
               插入
   ];
  For[i = NumberDivideM + 1, i ≤ NumberDivideM * 3, i++,
  For循环
    BoundsUM = Insert[BoundsUM, N[BoundsM[[i]]], -1];
               插入              数值运算
   ];
  BoundsNM = {};
  For[i = 1, i ≤ NumberDivideM, i++,
  For循环
    BoundsNM = Insert[BoundsNM, N[BoundsM[[i]]], -1];
               插入              数值运算
   ];
  For[i = NumberDivideM + 1, i ≤ NumberDivideM * 3, i++,
  For循环
    BoundsNM = Insert[BoundsNM, zM[[i]], -1];
               插入
   ];
```

In[]:=
```
BEMM = Function[{x, y}, Sum[BoundsUM[[i]] * FM[2, i, x, y] -
              纯函数                求和
       BoundsNM[[i]] * FM[1, i, x, y] , {i, 1, TotalNumberPointsM}]];
```

In[]:=
```
Plot3D[BEMM[x, y], {x, -1, 1}, {y, 0, Sqrt[1 - x^2]}]
绘制三维图形                                平方根
```

Out[]=



Comparison Between Boundary Element Solution and Approximate Real Solution

In[]:=
```
PointsMA = {{0.10, 0.20}, {0.10, 0.30}, {0.1, 0.40}, {0.50, 0.20},
   {0.50, 0.30}, {0.50, 0.40}, {0.90, 0.20}, {0.90, 0.30}, {0.90, 0.40}};
```

In[]:=
```
ValueM120 = {};
For[i = 1, i ≤ Length[PointsMA], i++,
For循环          长度
 ValueM120 = Insert[ValueM120, BEMM[PointsMA[[i, 1]], PointsMA[[i, 2]]], -1]
                插入
]
ValueM120
```

Out[]=
```
{0.550764, 0.629848, 0.701225, 0.652429,
 0.724614, 0.787831, 0.924011, 0.959372, 0.989907}
```

In[]:=
```
ValueM60 = {};
For[i = 1, i ≤ Length[PointsMA], i++,
For循环          长度
 ValueM60 = Insert[ValueM60, BEMM[PointsMA[[i, 1]], PointsMA[[i, 2]]], -1]
               插入
]
ValueM60
```

Out[]=
```
{0.551203, 0.630278, 0.701632, 0.653107,
 0.725203, 0.788329, 0.924841, 0.959895, 0.990241}
```

In[ ]:=
```
ValueExactMA = {};
For[i = 1, i ≤ Length[PointsMA], i++,
 ValueExactMA =
  Insert[ValueExactMA, AnalyticSolution2[PointsMA[[i, 1]], PointsMA[[i, 2]]], -1]
]
ValueExactMA
```

Out[ ]=
```
{0.551841, 0.630815, 0.702062, 0.652624,
 0.724802, 0.788019, 0.923646, 0.959226, 0.989775}
```

In[ ]:=
```
PointsMA2 = {{0.10, 0.95}, {0.10, 0.96}, {0.10, 0.97}, {0.0, 0.98}, {0, 0.99}};
```

In[ ]:=
```
ValueM1202 = {};
For[i = 1, i ≤ Length[PointsMA2], i++,
 ValueM1202 = Insert[ValueM1202, BEMM[PointsMA2[[i, 1]], PointsMA2[[i, 2]]], -1]
]
ValueM1202
```

Out[ ]=
```
{0.983365, 0.987142, 0.99088, 0.992721, 0.996405}
```

In[ ]:=
```
ValueExactMA2 = {};
For[i = 1, i ≤ Length[PointsMA2], i++,
 ValueExactMA2 =
  Insert[ValueExactMA2, AnalyticSolution2[PointsMA2[[i, 1]], PointsMA2[[i, 2]]], -1]
]
ValueExactMA2
```

Out[ ]=
```
{0.983369, 0.987134, 0.99086, 0.992695, 0.996367}
```

In[ ]:=
```
ValueM1202 - ValueExactMA2
```

Out[ ]= $\{-3.74495\times10^{-6}, 7.97227\times10^{-6}, 0.0000207005, 0.0000256709, 0.0000387372\}$

In[ ]:=
```
PointsMA3 = {};
For[i = 1, i ≤ 180, i++,
 PointsMA3 = Insert[PointsMA3,
   N[CoordinateTransform["Polar" → "Cartesian", {1, i/180 π}], 10], -1]
]
```

In[◦]:=
```
ValueM1203 = {};
For[i = 1, i ≤ Length[PointsMA3], i++,
 ValueM1203 = Insert[ValueM1203, BEMM[PointsMA3[[i, 1]], PointsMA3[[i, 2]]], -1]
]
```
For循环 长度 插入

In[◦]:=
```
Show[ListPlot[ValueM1203]]
```
显示 绘制点集

Out[◦]=