

②放物型 : $B^2 - 4AC = 0$

[例] $\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2} + f(x, t)$ (拡散方程式など)

③双曲型 : $B^2 - 4AC > 0$

[例] $\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} + f(x, t)$ (波動伝播など)

(3)楕円型のプログラム例

■考え方

楕円型の偏微分方程式には時間の項がありません。すなわち、楕円型の偏微分方程式を解くことは、定常状態における状態を求めることとなります。

■シートの定義

ラプラス方程式とポアソン方程式の左辺は同じであり、右辺が 0 となるか、ある関数値となるかだけが異なっています。したがって両方程式の左辺に関する処理は同じです。そこで両処理を共通化することにします。

まず、以下のような Excel シートを用意します。

	A	B	C	D	E	F
1	Xメッシュ数	Yメッシュ数	収斂回数	誤差範囲	途中計算回数	誤差
2	100	100	1000	0.01		
3						
4		ラプラスの方程式				
5						
6						
7		分布表示				
8						
9						
10		ポアソンの方程式				
11						

■結果の色マップ表示

微小格子における関数値を視覚的に観察するために Excel シートのセルを微小格子として関数値をセル色で表現するプログラムを作成します。このために、まず「分布」シートのセルの行高さと桁幅を小さくかつ正方形にします。そして、「分布表示」ボタンに、表 8-7 のようなプログラムを割り当てます。表 8-7 のプログラムでは「結果」シートのデータを「分布」シートに色マップとして表示します。

表 8-7 色マップ表示

```
Sub ボタン2_Click()
Dim CD(10) As Integer
CD(0) = 5: CD(1) = 41: CD(2) = 33: CD(3) = 34: CD(4) = 36
CD(5) = 6: CD(6) = 44: CD(7) = 45: CD(8) = 46: CD(9) = 3
Worksheets("分布").Select
MY = KY + 1
With Worksheets("結果")
For j = 1 To KY - 1
For i = 1 To KX - 1
ID = 9 * (Val(.Cells(KY - j, i) - UMin) / UDX)
If ID > 9 Then ID = 9
Worksheets("分布").Cells(j, i).Select
Selection.Interior.ColorIndex = CD(ID)
Selection.Interior.Pattern = xlSolid
Next
Next
End With
End Sub
```

■VBAによるプログラム

表 8-8 にプログラム例を示します。プログラム中では、ID1 が前状態、ID2 が現状態のデータを示す添え字です。ID1 と ID2 は 1 回の計算ごとに 0 と 1 を切り替えています。なお、プログラムが動いていることを確認するために、50 回ごとに途中経過の誤差を表示しています。

境界条件では、左側と下部の値を 0、上部と右側を固定的な値としています。温度分布で考えると、左側と下部を冷却し、上部と右側を熱しているときの定常状態の分布を求めます。

表 8-8 楕円型偏微分方程式のプログラム (その1)

```

Private ボアソン As Boolean
Private KX As Integer: Private KY As Integer: Private NN As Integer
Private EPS As Double: Private U() As Double: Private ID1 As Integer
Private ID2 As Integer: Private F() As Double: Private UMax As Double
Private UMin As Double: Private UDX As Double
Private Sub 初期条件(KX, KY, MX, MY, DX, DY)
For j = 0 To KY
For i = 0 To KX
U(0, i, j) = 0: U(1, i, j) = 0: F(i, j) = 0
If ボアソン Then ' ボアソンの方程式のとき右辺設定
F(i, j) = 80 * (DX * (MX - i) + DY * (MY - j))
End If
Next
Next
End Sub
Private Sub 境界条件(KX, KY, DX, DY)
For k = 0 To 1
For i = 0 To KX: U(k, i, 0) = 0: U(k, i, KY) = 16: Next
For j = 0 To KY: U(k, 0, j) = 0: U(k, KX, j) = 8: Next
Next
End Sub
Private Sub 計算(KX, KY, NN, EPS)
Dim MX As Integer: MX = KX + 1: Dim MY As Integer: MY = KY + 1
Dim DX As Double: DX = 1# / KX: Dim DY As Double: DY = 1# / KY
ReDim U(1, KX, KY), F(KX, KY)
Dim F1 As Double: F1 = 1 / (DX * DX): Dim F2 As Double: F2 = 1 / (DY * DY)
Dim F3 As Double: F3 = 0.5 / (F1 + F2)
初期条件 KX, KY, MX, MY, DX, DY: 境界条件 KX, KY, DX, DY
収斂計算
ID1 = 1: ID2 = 0
For N = 0 To NN - 1
ID = ID1: ID1 = ID2: ID2 = ID: ER = 0
For j = 1 To KY - 1
For i = 1 To KX - 1
U(ID2, i, j) = F3 * (F1 * (U(ID1, i + 1, j) + U(ID1, i, j + 1)) +
F2 * (U(ID1, i - 1, j) + U(ID1, i, j - 1))) + F(i, j)
If Abs(U(ID2, i, j)) > EPS Then
E = Abs((U(ID2, i, j) - U(ID1, i, j)) / U(ID2, i, j))
If E > ER Then ER = E
End If
Next
Next
If ER < EPS Then Exit For
If (N Mod 50) = 0 Then ' 計算途中経過表示
With Worksheets("データ")
.Cells(2, 5) = N
.Cells(2, 6) = Format(ER, "#0.000000")
End With
Application.ScreenUpdating = True
Application.ScreenUpdating = False
End If
Next
Application.ScreenUpdating = True
End Sub

```

表 8-8 楕円型偏微分方程式のプログラム (その2)

```

Sub データ設定()
With Worksheets("データ")
KX = Val(.Cells(2, 1))
KY = Val(.Cells(2, 2))
NN = Val(.Cells(2, 3))
EPS = Val(.Cells(2, 4))
End With
ボアソン = False
End Sub
Sub 結果設定()
With Worksheets("結果")
UMax = U(ID2, 1, 1): UMin = U(ID2, 1, 1)
For j = 1 To KY - 1
For i = 1 To KX - 1
.Cells(j, i) = U(ID2, i, j)
If UMax < U(ID2, i, j) Then UMax = U(ID2, i, j)
If UMin > U(ID2, i, j) Then UMin = U(ID2, i, j)
Next
Next
UDX = UMax - UMin
End With
End Sub
Sub ボタン1_Click()
' ラプラスの方程式のための Click イベントハンドラ
データ設定
ボアソン = False
計算 KX, KY, NN, EPS
結果設定
End Sub
Sub ボタン3_Click()
' ボアソンの方程式のための Click イベントハンドラ
データ設定
ボアソン = True
計算 KX, KY, NN, EPS
結果設定
End Sub

```

■ 結果例 (なお、以下の色マップはプログラムと異なり、白黒5段階に表示しなおしてある)

ラプラスの方程式の結果
(Umax=15.35, Umin=0.00)



ポアソンの方程式の結果
(Umax=14.93, Umin=0.03)



図 8-20 楕円型偏微分方程式プログラム実行結果 (分布)