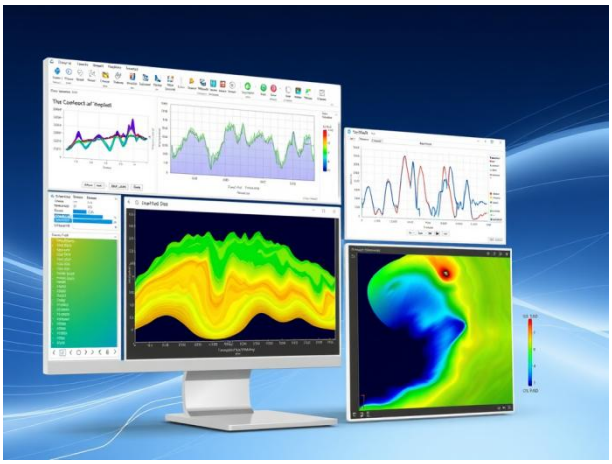
Chapter 8: Graphics

Chapter 8: Graphics in Fortran

8.1 Introduction to Graphics in Fortran

Fortran, originally designed for numerical computations, does not have built-in graphical capabilities like modern programming languages. However, graphics can be implemented in Fortran using external libraries such as:

- **PLplot** : Une bibliothèque de traçage scientifique pour Fortran.
- **DISLIN** : Une bibliothèque de haut niveau pour le traçage de données.
- **GNUPLOT** : Un outil externe qui peut être contrôlé depuis Fortran.
- **OpenGL** : Une bibliothèque puissante pour les graphiques 2D et 3D, utilisée en combinaison avec Fortran.
- **Origin** : Un logiciel de traitement de données et de traçage qui permet d'importer des données depuis Fortran pour créer des graphiques avancés.
- **Excel** : Un outil largement utilisé pour l'analyse de données qui peut être utilisé pour tracer des graphiques à partir de données générées par Fortran.
- **Tecplot** : Un logiciel de visualisation de données qui peut traiter des fichiers de sortie de Fortran pour créer des visualisations 2D et 3D complexes.



8.2 Plotting with PLplot

PLplot is one of the most common tools for plotting graphs in Fortran. It supports 2D and 3D plots.

8.2.1 Installing PLplot

To use PLplot, you need to install it on your system. You can install it using:

- **Linux (Ubuntu/Debian)**:

  sudo apt-get install plplot

- **Windows**: Install the PLplot package from the official website.

8.2.2 Basic Fortran Code for 2D Plotting with PLplot

```
PROGRAM simple_plot
  USE plplot
  IMPLICIT NONE
  REAL, DIMENSION(10) :: x, y
  INTEGER :: i
  ! Generate data
  DO i = 1, 10
    x(i) = REAL(i)
    y(i) = x(i)**2  ! y = x^2
  END DO

  ! Initialize PLplot
  CALL plinit()
  CALL plenv(0.0, 10.0, 0.0, 100.0, 0, 0)
  CALL pllab('X-axis', 'Y-axis', 'Fortran Plot Example')

  ! Plot the data
  CALL plline(10, x, y)

  ! Close PLplot
  CALL plend()
END PROGRAM simple_plot
```

Explanation of the Code:

1. plinit(): Initializes the PLplot library.
2. plenv(xmin, xmax, ymin, ymax, just, axis): Defines the plotting area.
3. pllab(xlabel, ylabel, title): Sets the labels and title.
4. plline(n, x, y): Draws a line connecting n points.
5. plend(): Ends the plot.

8.3 Plotting with DISLIN

DISLIN is another powerful library for scientific graphics. It is easier to use than PLplot.

8.3.1 Basic Fortran Code for 2D Plotting with DISLIN

```
PROGRAM dislin_plot
```

```
    IMPLICIT NONE
    INTEGER, PARAMETER :: n = 100
    REAL :: x(n), y(n)
    INTEGER :: i

    ! Generate sine wave data
    DO i = 1, n
       x(i) = 6.28 * REAL(i-1) / REAL(n-1)
       y(i) = SIN(x(i))
    END DO
    ! Initialize DISLIN
    CALL METAFL('XWIN')  ! Display on screen
    CALL DISINI()
    CALL NAME('X-axis', 'X')
    CALL NAME('Y-axis', 'Y')
    CALL LABDIG(-1, 'X')
    CALL LABDIG(-1, 'Y')
    CALL TITLIN('Sine Wave Plot', 2)
    CALL GRAF(0.0, 6.28, 0.0, 1.57, -1.0, 1.0, -1.0, 0.5)
    ! Draw curve
    CALL CURVE(x, y, n)
    ! Close DISLIN
    CALL DISFIN()
END PROGRAM dislin_plot
```

8.4 Using GNUPLOT with Fortran

GNUPLOT is an external tool that can be controlled from a Fortran program by writing data to a file and then executing GNUPLOT commands.

8.4.1 Writing Data for GNUPLOT

```
PROGRAM gnuplot_example
    IMPLICIT NONE
    INTEGER :: i
    REAL :: x, y
    OPEN(10, FILE='data.txt', STATUS='REPLACE')

    DO i = 1, 100
```

```
    x = REAL(i) / 10.0
    y = SIN(x)
    WRITE(10,*) x, y
  END DO

  CLOSE(10)

  ! Execute GNUPLOT script
  CALL SYSTEM('gnuplot -e "plot "data.txt" using 1:2 with lines"')
END PROGRAM gnuplot_example
```

This program:

a) Writes data points to a file (data.txt).

b) Calls GNUPLOT to plot the data.

8.5 3D Graphics with OpenGL in Fortran

OpenGL allows complex 3D visualizations in Fortran.

8.5.1 Basic OpenGL Code in Fortran

```
PROGRAM opengl_example
  USE opengl
  IMPLICIT NONE

  ! OpenGL initialization
  CALL glutInit()
  CALL glutCreateWindow("OpenGL Fortran")
  CALL glClearColor(0.0, 0.0, 0.0, 1.0)

  ! Display function
  CALL glutDisplayFunc(display)
  CALL glutMainLoop()

CONTAINS
  SUBROUTINE display()
    CALL glClear(GL_COLOR_BUFFER_BIT)
    CALL glBegin(GL_TRIANGLES)
    CALL glColor3f(1.0, 0.0, 0.0)
    CALL glVertex2f(0.0, 1.0)
```

```
    CALL glVertex2f(-1.0, -1.0)
    CALL glVertex2f(1.0, -1.0)
    CALL glEnd()
    CALL glFlush()
  END SUBROUTINE display
END PROGRAM opengl_example
```

This program creates a simple OpenGL window displaying a red triangle.

8.6 Choosing the Right Graphics Library

Feature	PLplot	DISLIN	GNUPLOT	OpenGL
2D Plots	□	□	□	□
3D Plots	□	□	□	□
Ease of Use	Medium	Easy	Medium	Hard
Installation	Requires setup	Requires setup	External program	Complex

Recommendation:

- Use **PLplot** or **DISLIN** for scientific plotting.
- Use **GNUPLOT** if you prefer external tools.
- Use **OpenGL** for 3D visualizations.

8.7 Conclusion

- Fortran does not have built-in graphics but supports **external libraries**.
- **PLplot and DISLIN** are great for **scientific visualization**.
- **GNUPLOT** is useful for simple **2D plotting**.
- **OpenGL** enables advanced **3D graphics**.

Would you like me to add **exercises or additional explanations** on a specific library?