Program Ordinary Differential Equation 2 - ODE2

Dim numnp As Integer, numel As Integer, n As Integer
Dim l As Integer, ll As Integer
Dim i As Integer, j As Integer, k As Integer, ii As Integer
Dim nqpts As Integer, ib As Integer, Lu As Integer, iprint As Integer
Dim k1 As Integer, L0 As Integer, L1 As Integer
Dim s1(2, 2) As Double, s2(2, 2) As Double, s3(2, 2) As Double
Dim sq(2) As Double, dndx(2) As Double
Dim gpts(6) As Double, gwts(6) As Double
Dim diff As Double, dx As Double, dq As Double, du As Double
Dim rl As Double, xi As Double, dxds As Double, dsdx As Double
Dim xl As Double, xg As Double, s As Double, sw As Double

Gaussian Quadrature Points and Weights

gpts(1) = 0.93246951: gpts(2) = 0.66120938: gpts(3) = 0.23861018
gpts(4) = -gpts(3): gpts(5) = -gpts(2): gpts(6) = -gpts(1)
gwts(1) = 0.17132449: gwts(2) = 0.36076157: gwts(3) = 0.46791393
gwts(4) = gwts(3): gwts(5) = gwts(2): gwts(6) = gwts(1)
Program Ordinary Differential Equation 2 - ODE2

Program Ordinary Differential Equation 2 - ODE2
Program Ordinary Differential Equation 2 - ODE2

Public Function sn(i, x) As Double
If i = 1 Then
    sn = *********** BLEEP ***********
Else
    sn = 0.5 + x + 0.5
End If
End Function

Public Function dsn(i, x) As Double
If i = 1 Then
    dsn = *********** BLEEP ***********
Else
    dsn = 0.5
End If
End Function

Public Function a(x) As Double
a = x ^ 2
End Function

Public Function b(x) As Double
b = 0
End Function

Public Function c(x) As Double
c = -1
End Function

Public Function d(x) As Double
d = 0
End Function

Public Function an(i, x) As Double
If i = 1 Then
    an = *********** BLEEP ***********
Else
    an = 0.5 + x + 0.5
End If
End Function

Public Function den(i, x) As Double
If i = 1 Then
    den = *********** BLEEP ***********
Else
    den = 0.5
End If
End Function

General function for:
d(A du/dx)/dx + B du/dx + C U + D = 0

N' - linear interpolation functions -1 < x < 1
N - linear interpolation functions -1 < x < 1
Program *Ordinary Differential Equation 2* - ODE2

Typical input file format:

```
numnp - Number of nodes in mesh
numel = numnp - 1
```

```
1  1  1.0  1.0  0.0
2  0  1.25  0.0  0.0
3  0  1.50  0.0  0.0
4  0  1.75  0.0  0.0
5  1  2.0  1.0  0.0
```

npbc – node point boundary condition
npbc(i) = 0 → u(i) is unknown
npbc(i) = 1 → u(i) is known

Node number – must be consecutively numbered

```
xord(i) – x coordinate of node point
u(i) values
q(i) values
```

Program *Ordinary Differential Equation 2* - ODE2

Typical input file format:

```
numnp - Number of nodes in mesh
numel = numnp - 1
```

```
9  1  0.0  0.0  0.0
2  0  15.0  0.0  0.0
9  0  120.0  0.0  0.0
```

Missing values of u(i) and q(i) are automatically generated in a manner identical to that used in the node and xord(i) generation

Missing nodes are automatically generated

In this case, nodes 2 – 8 are generated at an interval of 15.0 → (120-15)/(9-2)
**Program Ordinary Differential Equation 2 - ODE2**

Typical ODE2 VB interface output

- Location of input and output files
- Number of nodes and elements
- Problem has been solved
- Plot of solution
End of
ODE2 Program