



FORTTRAN & MATHEMATICA

(PRACTICAL)

Subject Code: 243718

FAISAL MOHAMMAD MAINOL QUADER

Lecturer

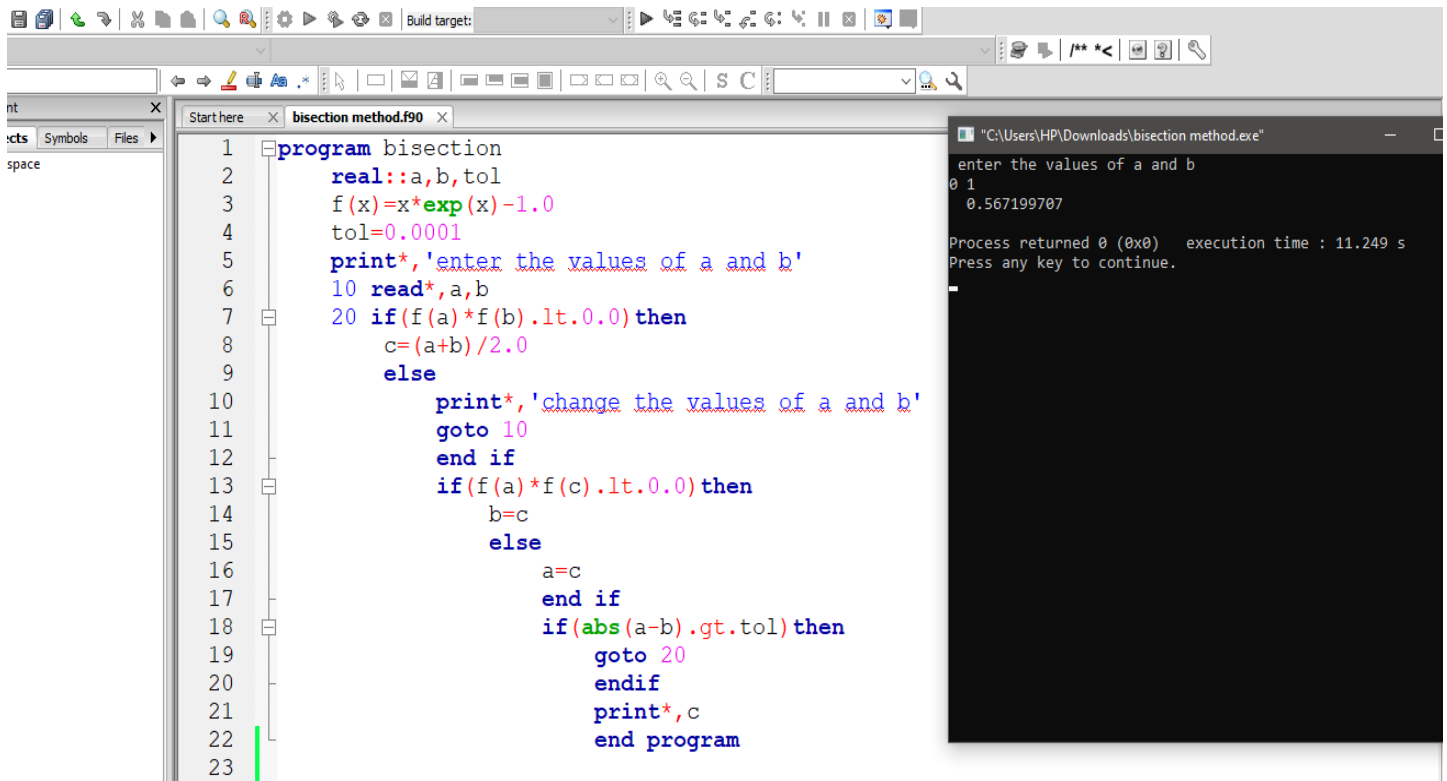
Department of Mathematics
Chittagong College, Chattogram

FORTRAN

PROBLEM-01

- Write a FORTRAN program to find a root, using bisection method, of the equation $xe^x = 1$.

Solⁿ:



The screenshot displays a Fortran IDE with a file named 'bisection method.f90'. The code implements the bisection method to find the root of the equation $xe^x = 1$. The program defines a function $f(x) = x \cdot \exp(x) - 1.0$ and a tolerance $tol = 0.0001$. It prompts the user to enter values for a and b . The bisection process involves calculating the midpoint $c = (a+b)/2.0$ and updating the interval based on the sign of the function values. The process continues until the absolute difference between a and b is greater than the tolerance. The final root value c is printed.

```
1 program bisection
2   real::a,b,tol
3   f(x)=x*exp(x)-1.0
4   tol=0.0001
5   print*, 'enter the values of a and b'
6   10 read*,a,b
7   20 if(f(a)*f(b).lt.0.0) then
8       c=(a+b)/2.0
9   else
10      print*, 'change the values of a and b'
11      goto 10
12  end if
13  if(f(a)*f(c).lt.0.0) then
14      b=c
15  else
16      a=c
17  end if
18  if(abs(a-b).gt.tol) then
19      goto 20
20  endif
21  print*,c
22  end program
23
```

The execution output shows the user entering '0 1', resulting in the root value '0.567199707'. The process returned 0 (0x0) with an execution time of 11.249 s. The user is prompted to press any key to continue.

PROBLEM-02

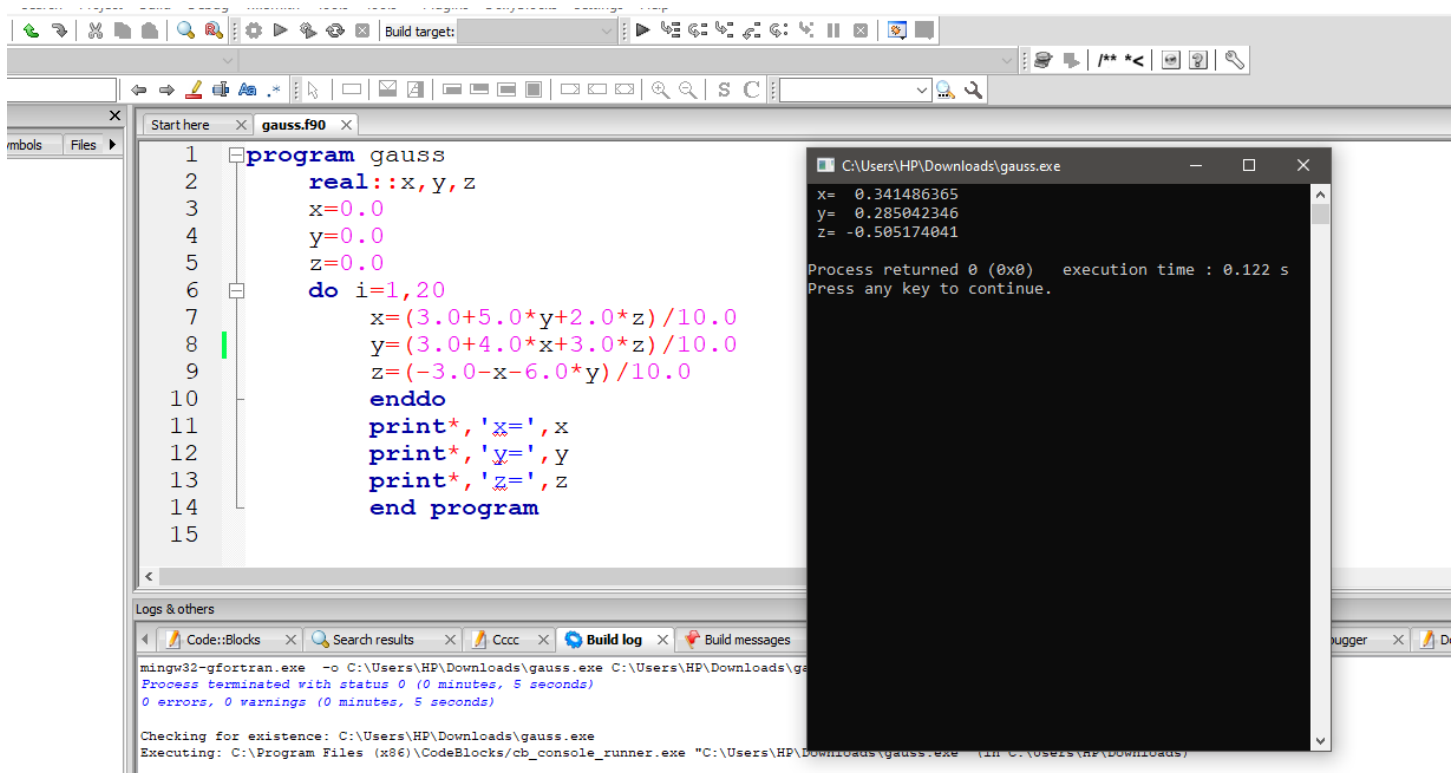
- Write a FORTRAN program to solve the following system of linear equation by Gauss-Seidel method :

$$10x - 5y - 2z = 3$$

$$4x - 10y + 3z = -3$$

$$x + 6y + 10z = -3$$

Solⁿ:



The screenshot displays the Code::Blocks IDE with a Fortran program named 'gauss.f90' and its execution output in a console window.

Fortran Program (gauss.f90):

```
1 program gauss
2   real :: x, y, z
3   x=0.0
4   y=0.0
5   z=0.0
6   do i=1,20
7     x=(3.0+5.0*y+2.0*z)/10.0
8     y=(3.0+4.0*x+3.0*z)/10.0
9     z=(-3.0-x-6.0*y)/10.0
10  enddo
11  print*, 'x=', x
12  print*, 'y=', y
13  print*, 'z=', z
14  end program
15
```

Execution Output (C:\Users\HP\Downloads\gauss.exe):

```
x= 0.341486365
y= 0.285042346
z= -0.505174041

Process returned 0 (0x0)   execution time : 0.122 s
Press any key to continue.
```

Build Log:

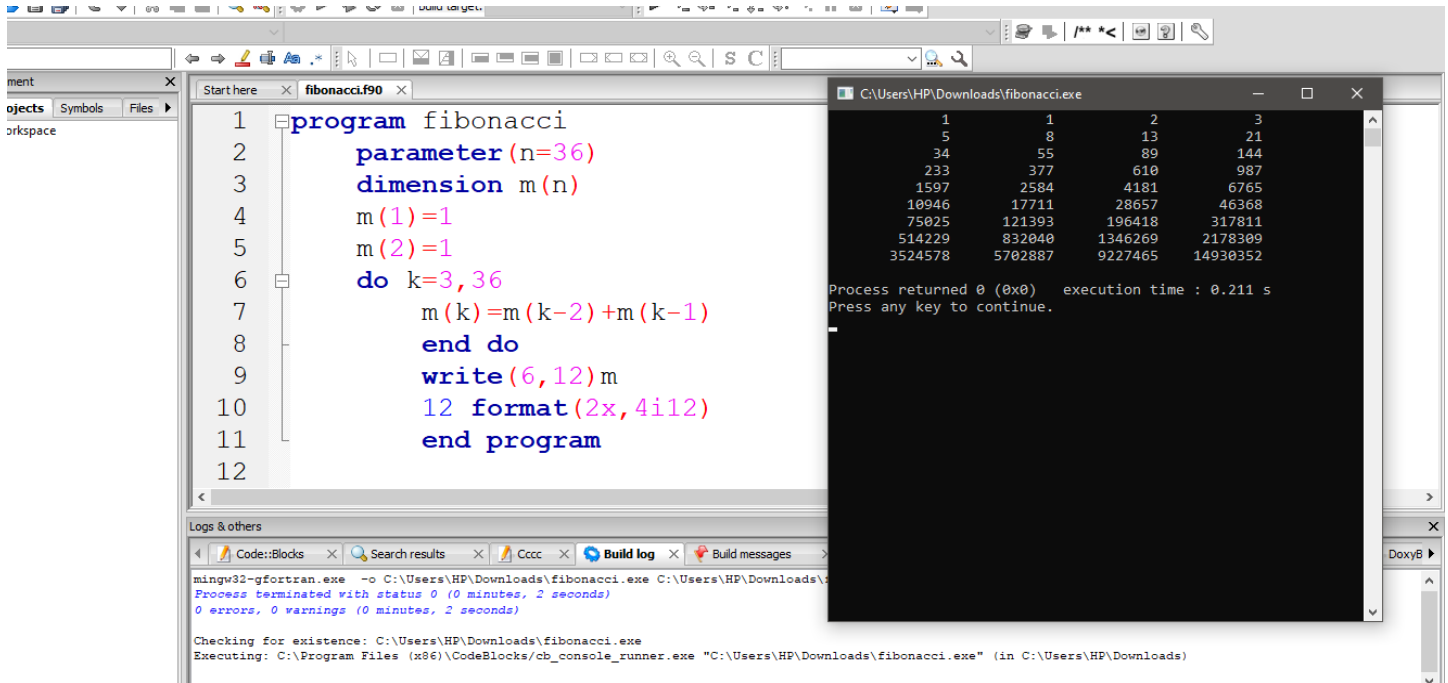
```
mingw32-gfortran.exe -o C:\Users\HP\Downloads\gauss.exe C:\Users\HP\Downloads\gauss.f90
Process terminated with status 0 (0 minutes, 5 seconds)
0 errors, 0 warnings (0 minutes, 5 seconds)

Checking for existence: C:\Users\HP\Downloads\gauss.exe
Executing: C:\Program Files (x86)\CodeBlocks\cb_console_runner.exe "C:\Users\HP\Downloads\gauss.exe" (in C:\Users\HP\Downloads\)
```

PROBLEM-03

- Write a FORTRAN program to find first 36 terms of Fibonacci sequence, the output should have four numbers in a line.

Solⁿ:



The screenshot displays the Code::Blocks IDE with a Fortran program named 'fibonacci.f90' and its execution output in a console window.

Source Code (fibonacci.f90):

```
1 program fibonacci
2   parameter(n=36)
3   dimension m(n)
4   m(1)=1
5   m(2)=1
6   do k=3, 36
7     m(k)=m(k-2)+m(k-1)
8   end do
9   write(6,12)m
10  12 format(2x,4i12)
11 end program
12
```

Execution Output (C:\Users\HP\Downloads\fibonacci.exe):

| | | | |
|---------|---------|---------|----------|
| 1 | 1 | 2 | 3 |
| 5 | 8 | 13 | 21 |
| 34 | 55 | 89 | 144 |
| 233 | 377 | 610 | 987 |
| 1597 | 2584 | 4181 | 6765 |
| 10946 | 17711 | 28657 | 46368 |
| 75025 | 121393 | 196418 | 317811 |
| 514229 | 832040 | 1346269 | 2178309 |
| 3524578 | 5702887 | 9227465 | 14930352 |

Process returned 0 (0x0) execution time : 0.211 s
Press any key to continue.

Build Log:

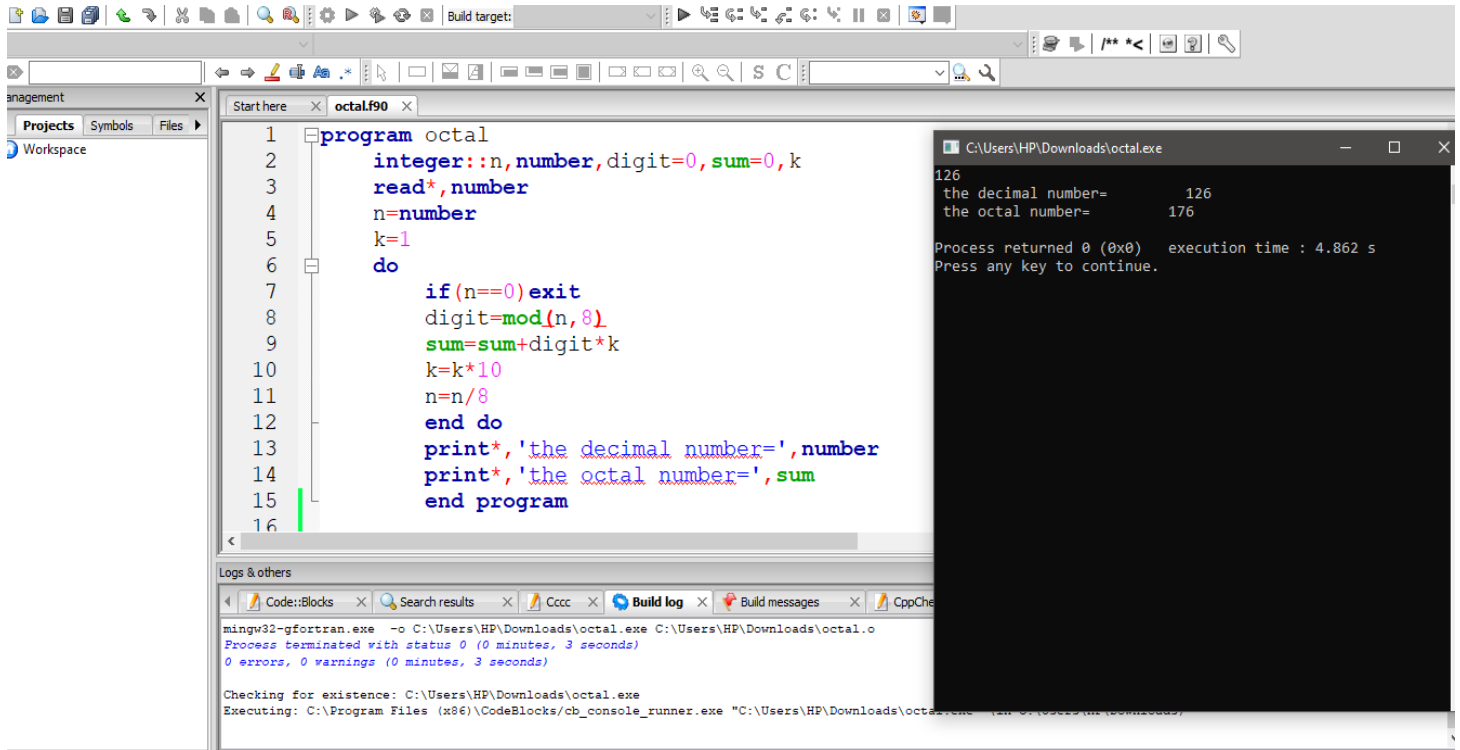
```
mingw32-gfortran.exe -o C:\Users\HP\Downloads\fibonacci.exe C:\Users\HP\Downloads\
Process terminated with status 0 (0 minutes, 2 seconds)
0 errors, 0 warnings (0 minutes, 2 seconds)

Checking for existence: C:\Users\HP\Downloads\fibonacci.exe
Executing: C:\Program Files (x86)\CodeBlocks\cb_console_runner.exe "C:\Users\HP\Downloads\fibonacci.exe" (in C:\Users\HP\Downloads)
```

PROBLEM-04

- Write a FORTRAN program to convert a decimal positive integer to octal number system.

Solⁿ:



```
1 program octal
2   integer :: n, number, digit=0, sum=0, k
3   read*, number
4   n=number
5   k=1
6   do
7       if (n==0) exit
8       digit=mod(n,8)
9       sum=sum+digit*k
10      k=k*10
11      n=n/8
12  end do
13  print*, 'the decimal number=', number
14  print*, 'the octal number=', sum
15  end program
16
```

Output:

```
126
the decimal number=      126
the octal number=       176

Process returned 0 (0x0)   execution time : 4.862 s
Press any key to continue.
```

Build log:

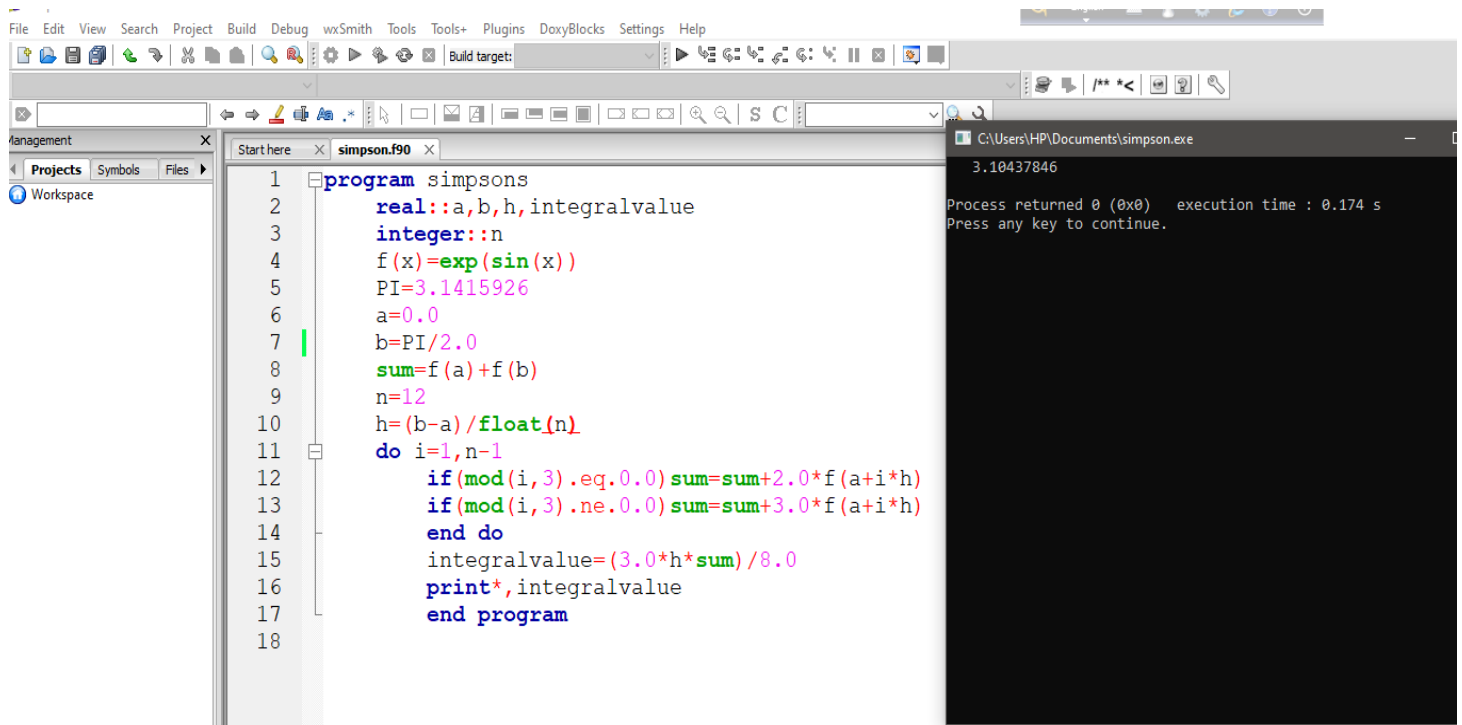
```
mingw32-gfortran.exe -o C:\Users\HP\Downloads\octal.exe C:\Users\HP\Downloads\octal.o
Process terminated with status 0 (0 minutes, 3 seconds)
0 errors, 0 warnings (0 minutes, 3 seconds)

Checking for existence: C:\Users\HP\Downloads\octal.exe
Executing: C:\Program Files (x86)\CodeBlocks\cb_console_runner.exe "C:\Users\HP\Downloads\octal.exe"
```

PROBLEM-05

- Write a FORTRAN program for Simpson's $\frac{3}{8}$ rule to evaluate the integral $\int_0^{\frac{\pi}{2}} e^{\sin x} dx$.

Solⁿ:



The image shows a screenshot of a code editor and a terminal window. The code editor displays a FORTRAN program named 'simpson.f90' which implements Simpson's 3/8 rule to evaluate the integral $\int_0^{\frac{\pi}{2}} e^{\sin x} dx$. The program defines variables for the limits of integration (a=0.0, b=PI/2.0), the number of intervals (n=12), and the step size (h=(b-a)/float(n)). It then uses a loop to calculate the sum of function values at various points, applying the 3/8 rule weights (2.0 for the first and last points, and 3.0 for the intermediate points). The final result is printed as 'integralvalue'.

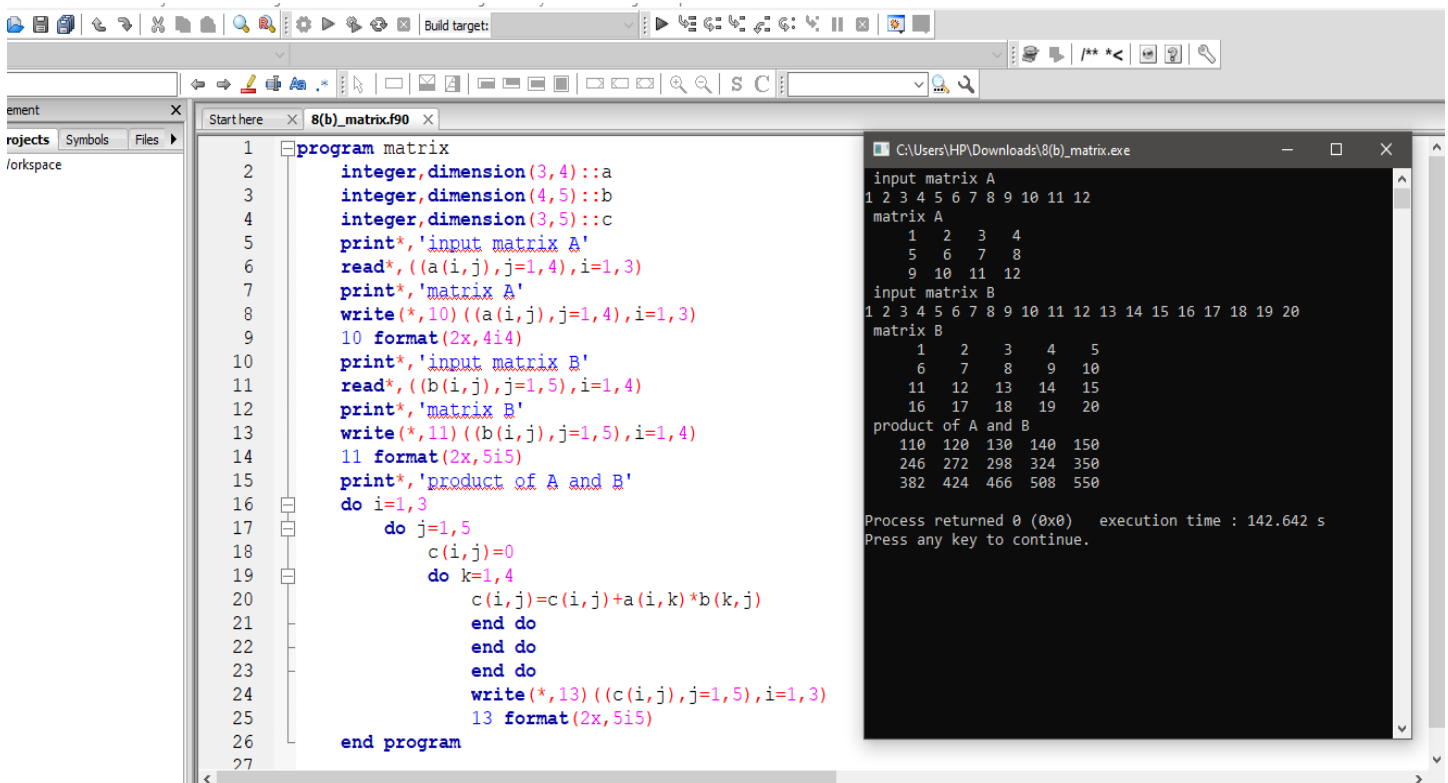
```
1 program simpsons
2   real::a,b,h,integralvalue
3   integer::n
4   f(x)=exp(sin(x))
5   PI=3.1415926
6   a=0.0
7   b=PI/2.0
8   sum=f(a)+f(b)
9   n=12
10  h=(b-a)/float(n)
11  do i=1,n-1
12    if(mod(i,3).eq.0.0) sum=sum+2.0*f(a+i*h)
13    if(mod(i,3).ne.0.0) sum=sum+3.0*f(a+i*h)
14  end do
15  integralvalue=(3.0*h*sum)/8.0
16  print*,integralvalue
17  end program
18
```

The terminal window shows the output of the program: '3.10437846'. Below the output, it displays the message 'Process returned 0 (0x0) execution time : 0.174 s' and 'Press any key to continue.'.

PROBLEM-06

- Write a FORTRAN program to compute the product of two matrices.

Solⁿ:



The screenshot displays a Fortran development environment with two windows. The left window, titled '8(b)_matrix.f90', contains the source code for a program that reads two matrices, A and B, and computes their product. The right window, titled 'C:\Users\HP\Downloads\8(b)_matrix.exe', shows the program's execution output, including the input matrices and the resulting product matrix.

```
1 program matrix
2   integer, dimension(3,4)::a
3   integer, dimension(4,5)::b
4   integer, dimension(3,5)::c
5   print*, 'input matrix A'
6   read*, ((a(i,j), j=1,4), i=1,3)
7   print*, 'matrix A'
8   write(*,10) ((a(i,j), j=1,4), i=1,3)
9   10 format(2x,4i4)
10  print*, 'input matrix B'
11  read*, ((b(i,j), j=1,5), i=1,4)
12  print*, 'matrix B'
13  write(*,11) ((b(i,j), j=1,5), i=1,4)
14  11 format(2x,5i5)
15  print*, 'product of A and B'
16  do i=1,3
17      do j=1,5
18          c(i,j)=0
19          do k=1,4
20              c(i,j)=c(i,j)+a(i,k)*b(k,j)
21          end do
22      end do
23  end do
24  write(*,13) ((c(i,j), j=1,5), i=1,3)
25  13 format(2x,5i5)
26 end program
```

Execution Output:

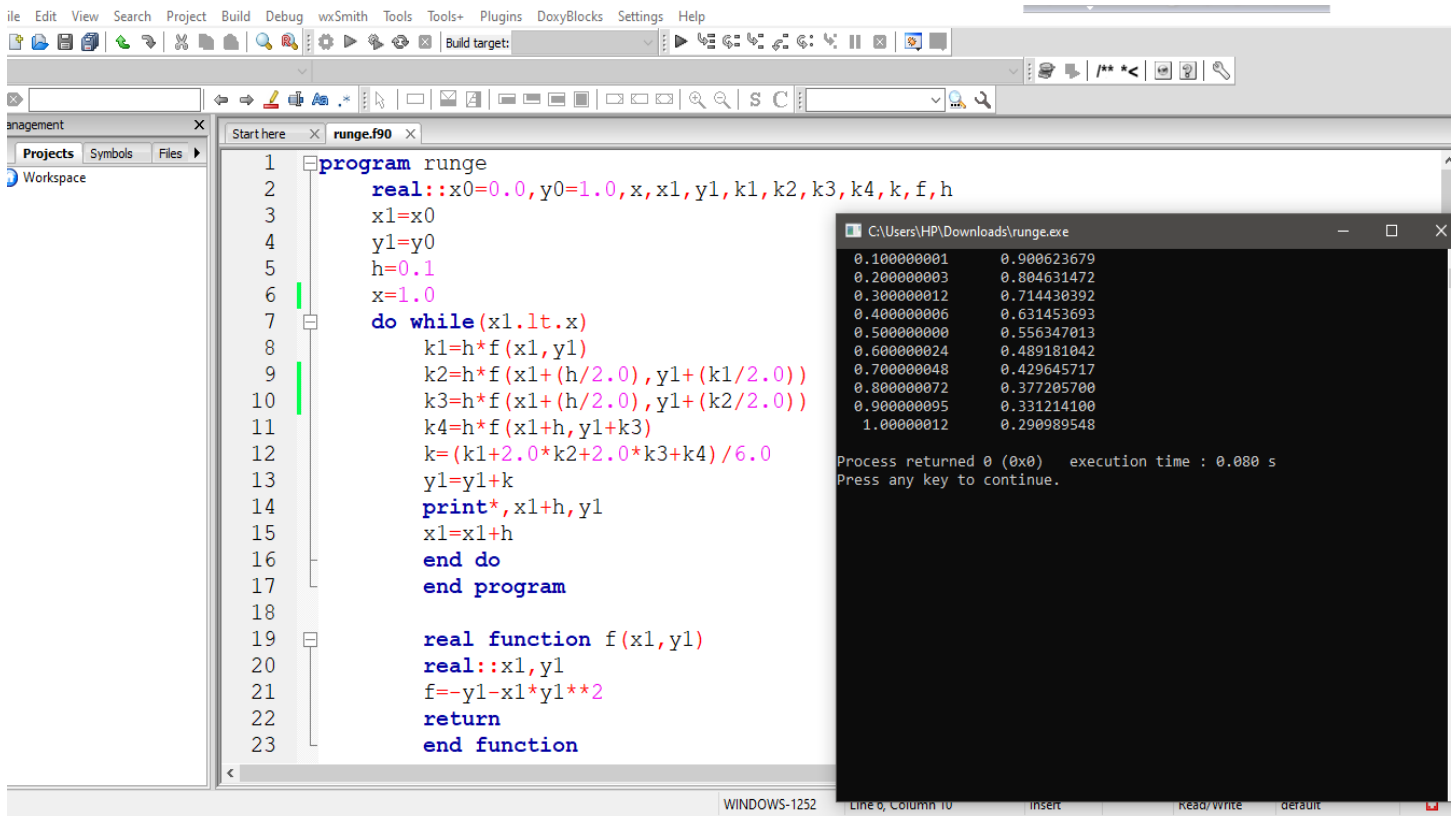
```
input matrix A
1 2 3 4 5 6 7 8 9 10 11 12
matrix A
  1  2  3  4
  5  6  7  8
  9 10 11 12
input matrix B
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
matrix B
  1  2  3  4  5
  6  7  8  9 10
 11 12 13 14 15
 16 17 18 19 20
product of A and B
110 120 130 140 150
246 272 298 324 350
382 424 466 508 550

Process returned 0 (0x0)   execution time : 142.642 s
Press any key to continue.
```


PROBLEM-07

- Write a FORTRAN program to find the solution of the differential equation $\frac{dy}{dx} + y + xy^2 = 0$, $y(0) = 1$ for $x \in [0,1]$ with $h = 0.1$ using Runge-Kutta fourth order method.

Solⁿ:



The screenshot displays a Fortran IDE with the source code for a program named 'runge.f90'. The code implements the Runge-Kutta fourth order method to solve the differential equation $\frac{dy}{dx} + y + xy^2 = 0$ with initial condition $y(0) = 1$ over the interval $x \in [0, 1]$ using a step size $h = 0.1$. The program defines a function $f(x, y)$ and a main program 'runge' that calculates the solution at each step and prints the results.

```
1 program runge
2   real::x0=0.0,y0=1.0,x,x1,y1,k1,k2,k3,k4,k,f,h
3   x1=x0
4   y1=y0
5   h=0.1
6   x=1.0
7   do while(x1.lt.x)
8     k1=h*f(x1,y1)
9     k2=h*f(x1+(h/2.0),y1+(k1/2.0))
10    k3=h*f(x1+(h/2.0),y1+(k2/2.0))
11    k4=h*f(x1+h,y1+k3)
12    k=(k1+2.0*k2+2.0*k3+k4)/6.0
13    y1=y1+k
14    print*,x1+h,y1
15    x1=x1+h
16  end do
17 end program

18
19 real function f(x1,y1)
20 real::x1,y1
21 f=-y1-x1*y1**2
22 return
23 end function
```

The execution output shows the results of the Runge-Kutta method for each step of $h = 0.1$. The output is as follows:

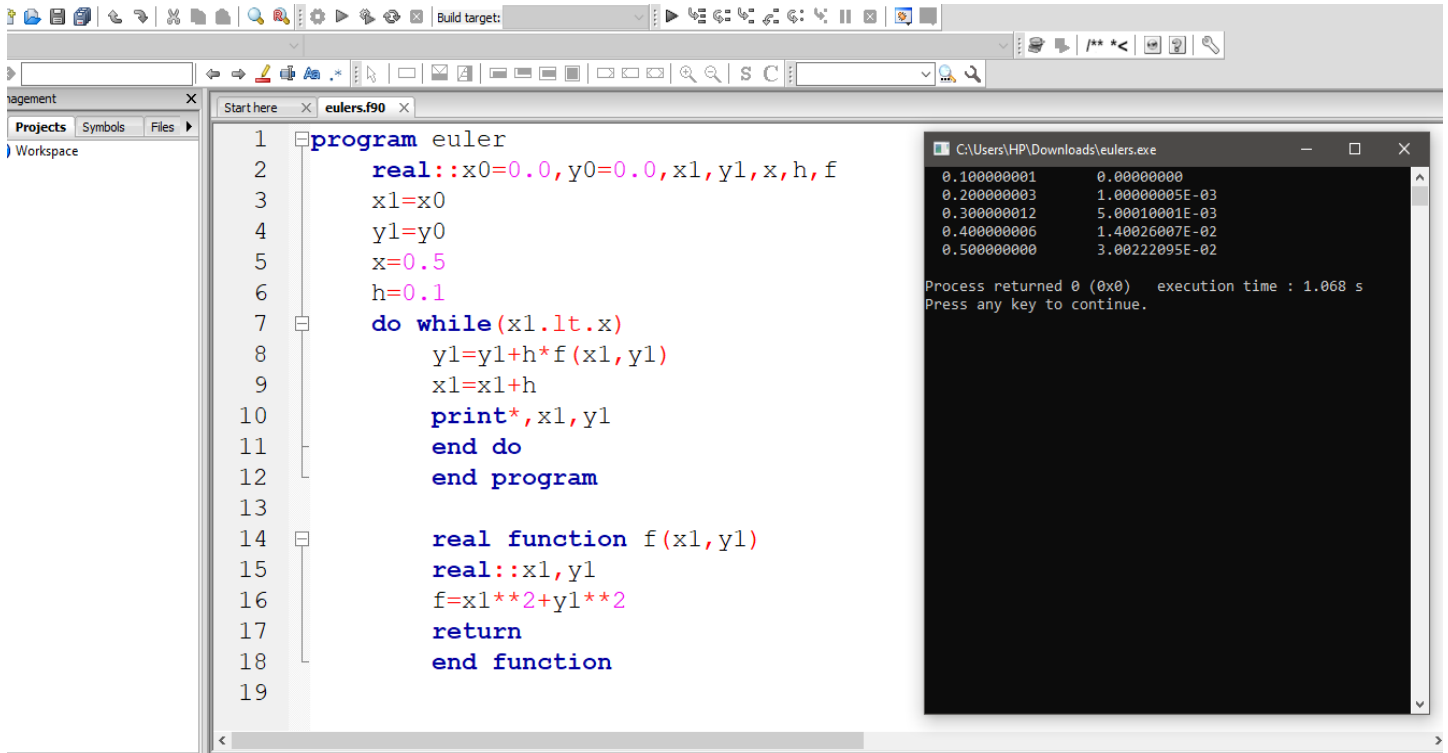
| x | y |
|------------|-------------|
| 0.10000001 | 0.900623679 |
| 0.20000003 | 0.804631472 |
| 0.30000012 | 0.714430392 |
| 0.40000006 | 0.631453693 |
| 0.50000000 | 0.556347013 |
| 0.60000024 | 0.489181042 |
| 0.70000048 | 0.429645717 |
| 0.80000072 | 0.377205700 |
| 0.90000095 | 0.331214100 |
| 1.0000012 | 0.290989548 |

Process returned 0 (0x0) execution time : 0.080 s
Press any key to continue.

PROBLEM-08

- Write a FORTRAN program to find the solution of $\frac{dy}{dx} = x^2 + y^2, y(0) = 0$ in the range $0 \leq x \leq 0.5$ with $h = 0.1$ using Euler's method.

Solⁿ:



The screenshot shows a code editor with a FORTRAN program named 'eulers.f90' and a terminal window displaying its execution results.

```
1 program euler
2   real::x0=0.0,y0=0.0,x1,y1,x,h,f
3   x1=x0
4   y1=y0
5   x=0.5
6   h=0.1
7   do while(x1.lt.x)
8     y1=y1+h*f(x1,y1)
9     x1=x1+h
10    print*,x1,y1
11  end do
12 end program

13
14 real function f(x1,y1)
15 real::x1,y1
16 f=x1**2+y1**2
17 return
18 end function
19
```

The terminal window, titled 'C:\Users\HP\Downloads\Eulers.exe', shows the following output:

| x | y |
|-------------|----------------|
| 0.100000001 | 0.00000000 |
| 0.200000003 | 1.00000005E-03 |
| 0.300000012 | 5.00010001E-03 |
| 0.400000006 | 1.40026007E-02 |
| 0.500000000 | 3.00222095E-02 |

Process returned 0 (0x0) execution time : 1.068 s
Press any key to continue.

MATHEMATICA

PROBLEM-01

➤ Show that, $f(x) = \begin{cases} x \cos\left(\frac{1}{x}\right) & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases}$ is continuous at $x = 0$ but not differentiable.

Solⁿ:

```
File Edit Insert Format Cell Graphics Evaluation Palettes Window Help

In[10]:= f[x_] := { x Cos[1/x] x != 0
                  0      x == 0

construct pure function

In[3]:= lhs = Limit[f[x], x -> 0, Direction -> "FromBelow"]

In[4]:= rhs = Limit[f[x], x -> 0, Direction -> "FromAbove"]

In[5]:= fv = f[0]

In[6]:= lhs == rhs == fv

f(x) is continuous at x=0

In[7]:= lhs = Limit[f[0+h] - f[0], h -> 0, Direction -> "FromBelow"]

In[8]:= rhs = Limit[f[0+h] - f[0], h -> 0, Direction -> "FromAbove"]

In[9]:= diff = Limit[f[0+h] - f[0], h -> 0]

f(x) is not differentiable at x=0
```

PROBLEM-02

➤ Given $f(x, y) = \begin{cases} \frac{xy(x^2 - y^2)}{x^2 + y^2} & \text{if } (x, y) \neq (0, 0) \\ 0 & \text{if } (x, y) = (0, 0) \end{cases}$ show that, $f_{xy}(0, 0) \neq f_{yx}(0, 0)$.

Solⁿ:

Edit Insert Format Cell Graphics Evaluation Palettes Window Help

$$\text{In[1]:= } f[x_, y_] := \text{Piecewise}\left[\left\{\left\{\frac{x * y * (x^2 - y^2)}{x^2 + y^2}, x \neq 0 \mid \mid y \neq 0\right\}, \{0, x == 0 \&\& y == 0\}\right\}\right]$$

$$\text{In[2]:= } fx[x_, y_] := \text{Limit}\left[\frac{f[x + h, y] - f[x, y]}{h}, h \rightarrow 0\right]$$

$$\text{In[3]:= } fy[x_, y_] := \text{Limit}\left[\frac{f[x, y + k] - f[x, y]}{k}, k \rightarrow 0\right]$$

$$\text{In[4]:= } fxy = \text{Limit}\left[\frac{fx[0, k] - fx[0, 0]}{k}, k \rightarrow 0\right]$$

$$\text{In[5]:= } fyx = \text{Limit}\left[\frac{fy[h, 0] - fy[0, 0]}{h}, h \rightarrow 0\right]$$

$$\text{In[6]:= } fxy == fyx$$

PROBLEM-03

- Verify Roll's theorem with graph for the function $f(x) = (x - 2)(x - 3)(x - 4)$ in the interval $(2,3)$.

Solⁿ:

File Edit Insert Format Cell Graphics Evaluation Palettes Window Help

```
In[3]:= f[x_] = Expand[(x - 2) (x - 3) (x - 4)]
```

Since the function is a polynomial so the function is continuous and differentiable

```
In[4]:= a = 2; b = 3;
```

```
In[5]:= f[a] == f[b]
```

```
In[6]:= Plot[f[x], {x, 2, 3}]
```

```
In[7]:= NSolve[f'[c] == 0, c]
```

⏏

PROBLEM-04

- Find the intervals where the function $y = x^4 + 2x^3 - 3x^2 - 4x + 4$ is increasing and decreasing. Also find the maximum and minimum values of y .

Solⁿ:

File Edit Insert Format Cell Graphics Evaluation Palettes Window Help

```
In[1]:= f[x_] := x^4 + 2 x^3 - 3 x^2 - 4 x + 4
```

```
In[2]:= f'[x]
```

```
In[3]:= Solve[f'[x] == 0, x]
```

```
In[4]:= NumberLinePlot[{Interval[{-∞, -2}], Interval[{ -2, -1/2 }], Interval[{ -1/2, 1 }], Interval[{1, ∞}]}]
```

```
In[5]:= Plot[f[x], {x, -3, 2}, AxesOrigin -> {0, 0}]
```

```
In[6]:= f'[x] /. x -> -3
```

```
In[7]:= deint = Interval[{-∞, -2}]
```

```
In[8]:= f'[x] /. x -> -1
```

```
In[9]:= inint = Interval[{ -2, -1/2 }]
```

```
In[10]:= f'[x] /. x -> 0
```

```
In[11]:= deint = Interval[{ -1/2, 1 }]
```

```
In[12]:= f'[x] /. x -> 2
```

```
In[13]:= inint = Interval[{1, ∞}]
```

```
In[14]:= f''[x]
```

```
In[15]:= f''[x] /. x -> -2
```

```
In[16]:= min = f[x] /. x -> -2
```

```
In[17]:= f''[x] /. x -> -1/2
```

```
In[18]:= max = f[x] /. x -> -1/2
```

```
In[19]:= f''[x] /. x -> 1
```

```
In[20]:= min = f[x] /. x -> 1
```

PROBLEM-05

- Find the equation of a sphere which passes through the points $(1, 2, 0)$, $(4, 2, -3)$, $(1, 5, -3)$ and touches the plane $x + 2y + 2z = 8$. Draw the graph.

Solⁿ:

File Edit Insert Format Cell Graphics Evaluation Palettes Window Help

```
In[1]:= sphere = x^2 + y^2 + z^2 + 2*u*x + 2*v*y + 2*w*z + d == 0
```

```
In[2]:= eqn1 = sphere /. {x -> 1, y -> 2, z -> 0}
```

```
In[3]:= eqn2 = sphere /. {x -> 4, y -> 2, z -> -3}
```

```
In[4]:= eqn3 = sphere /. {x -> 1, y -> 5, z -> -3}
```

```
In[5]:= eqn4 = 
$$\frac{-u - 2v - 2w + 8}{\sqrt{1 + 4 + 4}} = \sqrt{u^2 + v^2 + w^2 - d}$$

```

```
In[6]:= s1n = Solve[eqn1 && eqn2 && eqn3 && eqn4, {u, v, w, d}]
```

```
In[7]:= a1 = sphere /. s1n[[1]]
```

```
In[8]:= a2 = sphere /. s1n[[2]]
```

```
In[9]:= p1 = RegionPlot3D[
$$191 - 18\sqrt{105} + 2 \times (-32 + 3\sqrt{105})x + x^2 + 6 \times (-11 + \sqrt{105})y + y^2 + 2 \times (-28 + 3\sqrt{105})z + z^2 \leq 0,$$
  
{x, -5, 5}, {y, -5, 5}, {z, -8, 2}]
```

```
In[10]:= p2 = ContourPlot3D[x + 2y + 2z == 8, {x, -5, 5}, {y, -5, 5}, {z, -5, 5}, Mesh -> None,  
ContourStyle -> {Opacity[0.3], Lighter[Purple, 0.1]}]
```

```
In[11]:= Show[p1, p2]
```

+

PROBLEM-06

- Show that, $f(z) = f(x) = \begin{cases} \frac{x^3 - 3xy^2 + i(y^3 - 3x^2y)}{x^2 + y^2}, & z \neq 0 \\ 0, & z = 0 \end{cases}$ is continuous at $z=0$ and the Cauchy-Riemann equations are satisfied but not differentiable at $z=0$.

Solⁿ:

| File | Edit | Insert | Format | Cell | Graphics | Evaluation | Pa |
|------|------|--------|--------|------|----------|------------|----|
|------|------|--------|--------|------|----------|------------|----|

```

f[z_] := { (x^3 - 3*x*y^2 + I*(y^3 - 3*x^2*y))/(x^2 + y^2), z != 0
           0, z == 0

In[8]:= u[x_, y_] = (x^3 - 3*x*y^2)/(x^2 + y^2)

In[9]:= v[x_, y_] = (y^3 - 3*x^2*y)/(x^2 + y^2)

In[10]:= mux = Limit[u[x, y] /. y -> m*x, x -> 0]
In[11]:= mvx = Limit[v[x, y] /. y -> m*x, x -> 0]
In[12]:= mux + I*mvx
In[13]:= f[0]
In[14]:= mux + I*mvx == f[0]

f(z) is continuous at z=0

In[15]:= u0 = u[x, 0] /. x -> 0
In[16]:= v0 = v[x, 0] /. x -> 0
In[17]:= ux = Limit[(u[0 + h, 0] - u0) / h, h -> 0]
In[18]:= uy = Limit[(u[0, 0 + k] - u0) / k, k -> 0]
In[19]:= vx = Limit[(v[0 + h, 0] - v0) / h, h -> 0]
In[20]:= vy = Limit[(v[0, 0 + k] - v0) / k, k -> 0]
In[21]:= ux == vy && vx == -uy

f(z) satisfies Cauchy-Riemann equation.

In[22]:= Dfx = Limit[(u[x, y] + I*v[x, y] - 0)/(x + I*y) /. y -> x, x -> 0]
In[23]:= Df0 = Limit[(u[x, y] + I*v[x, y] - 0)/(x + I*y) /. y -> 0, x -> 0]
In[24]:= Dfx == Df0

f(z) is not differentiable at z=0.

```

THE END