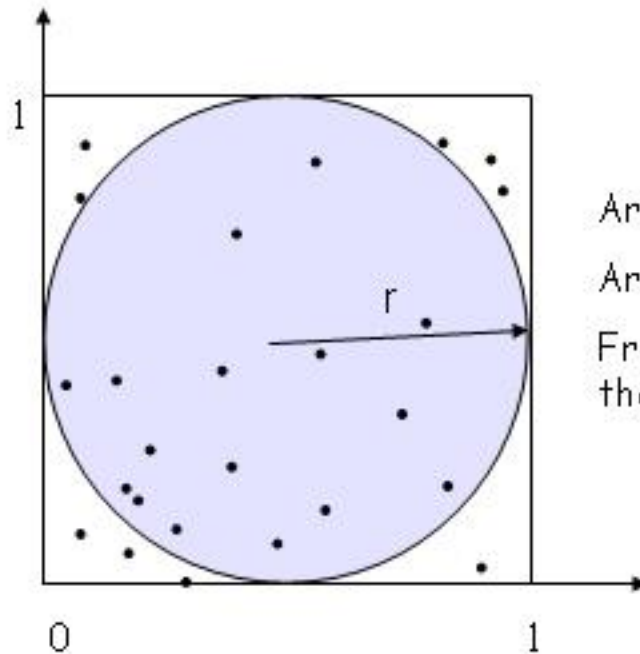


# Calculation of $\pi$ by Integration

V. Sundararajan

## Hit or Miss Method

- Based on generating random numbers in a unit length square and counting number of points that fall within largest circle inscribed in the square



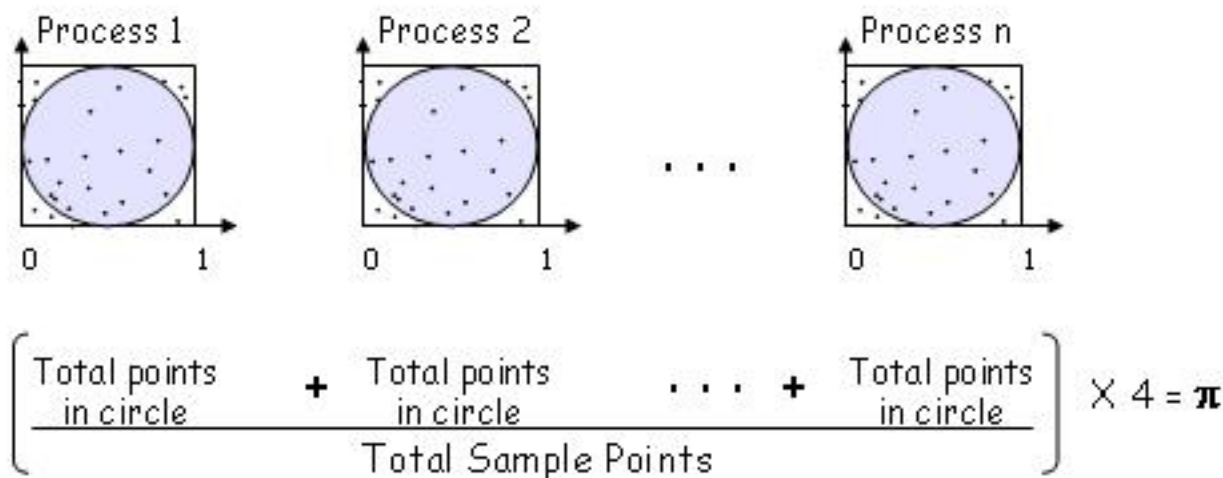
Area of the circle =  $\pi r^2 = \pi/4$

Area of the square =  $1 \times 1$

Fraction of random points  
that fall in the circle  $\sim \pi/4$

## Computing value of $\pi$ using MPI

- Assign fixed number of points to each process
- Each process generates random points and keeps track of no. of points that land in circle locally
- After all processes finish execution, their counts are collected to compute the value of  $\pi$



## Initialisation

```
! Calculation of Pi using MPI
```

```
implicit none
```

```
integer i,seed,npoints,thseed
```

```
integer IERR,NPROCS,MYID,STATUS,tag
```

```
double precision x,y,d,pi,sum
```

```
include 'mpif.h'
```

```
seed=1246
```

```
npoints=800000
```

```
sum=0
```

```
tag=99
```

```
CALL MPI_INIT(IERR)
```

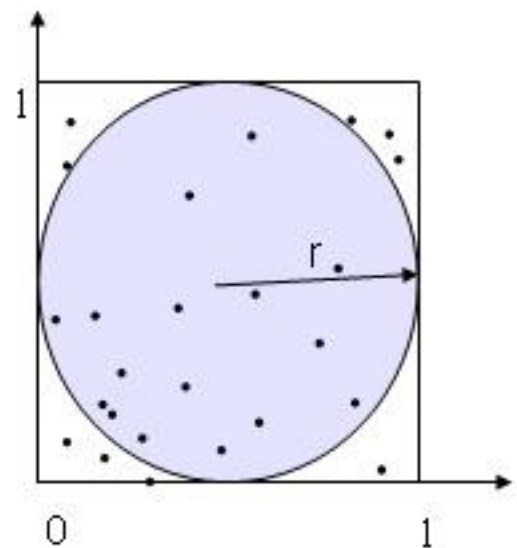
```
CALL MPI_COMM_RANK(MPI_COMM_WORLD,MYID,IERR)
```

```
CALL MPI_COMM_SIZE(MPI_COMM_WORLD,NPROCS,IERR)
```

## Division of work and Calculation

```
thseed=seed/(MYID+1)
call srand(thseed)
print *, "seed= ", thseed, "Tid= ", MYID

DO i=MYID+1, npoints, NPROCS
  x = rand(0)
  y = rand(0)
  d = (x-0.5)*(x-0.5)+(y-0.5)*(y-0.5)
  if (d.le.0.25)sum= sum+1.0
ENDDO
sum=sum*4;
```



## Collection of Results

```
If(MYID.EQ.0) then
  pi=sum
  DO i=2,NPROCS
    CALL MPI_RECV(sum,1,mpi_double_precision,i-1,tag,
&                MPI_COMM_WORLD,STATUS,IERR)
    pi=pi+sum
  ENDDO
else
  CALL MPI_SEND(sum,1,mpi_double_precision,0,tag,
&              MPI_COMM_WORLD,IERR)
endif

print *, " Sum= ", sum, " pi= ", pi/npoints
CALL MPI_FINALIZE(IERR)
end
```