Approximation of Pi
Compute π value using p processors.

Integration to evaluate π

Computer approximations to π by using numerical integration
Know

$$tan(45^0) = 1;$$

same as

$$tan\frac{\pi}{4} = 1;$$

So that;

$$4 * tan^{-1}1 = \pi$$

From the integral tables we can find

$$tan^{-1}x = \int \frac{1}{1+x^2}dx$$

or

$$tan^{-1}1 = \int_0^1 \frac{1}{1+x^2}dx$$

Using the mid-point rule with panels of
uniform length $h = 1/n$, for various values of n.
Evaluate the function at the midpoints of
each subinterval (x_{i-1}, x_i) $i * h - h/2$ is the midpoint.
Formula for the integral is

$$x = \sum_{i=1}^{n} f(h * (i - 1/2))$$

$$\pi = h * x$$

where

$$f(x) = \frac{4}{1+x^2}$$

Approximation of Pi
Compute π value using p processors.


```
#include "mpi.h"
#include <math.h>
int main(int argc, char *argv[])
{
int done = 0, n, myid, numprocs, i, rc;
double PI25DT = 3.14159265358979323846264643;
double mypi, pi, h, sum, x, a;
MPI_Init(&argc,&argv);
MPI_Comm_size(MPI_COMM_WORLD,&numprocs);
MPI_Comm_rank(MPI_COMM_WORLD,&myid);
while (!done)  {
  if (myid == 0) {
    printf("Enter the number of intervals: (0 quits) ");
    scanf("%d",&n);
  }
  MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);
  if (n == 0) break;
```

Approximation of Pi
Compute π value using p processors.

```
h   = 1.0 / (double) n;
 sum = 0.0;
 for (i = myid + 1; i <= n; i += numprocs) {
   x = h * ((double)i - 0.5);
   sum += 4.0 / (1.0 + x*x);
 }
 mypi = h * sum;
 MPI_Reduce(&mypi, &pi, 1, MPI_DOUBLE, MPI_SUM, 0,
            MPI_COMM_WORLD);
 if (myid == 0)
   printf("pi is approximately %.16f, Error is %.16f\n",
          pi, fabs(pi - PI25DT));
}
MPI_Finalize();
  return 0;
}
```