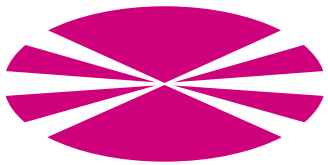
Concurrency and Parallelism. Block II Parallelism

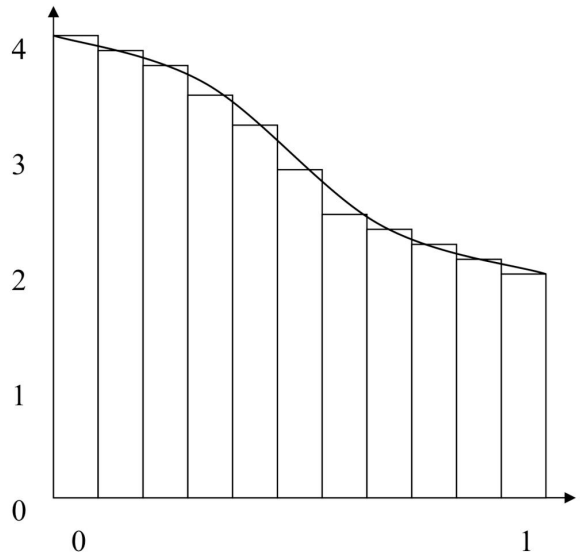
Assignment 1: estimation of PI by the integration method

Spring 2024



Estimation of PI by the integration method

- Approximate the value of PI by means of the integration of $4/(1+x^2)$ in the interval $[0,1]$.
- The interval is subdivided in N subintervals of length $1/N$.
- For each subinterval, the area of the rectangle whose height is the value of $4/(1+x^2)$ in its middle point is computed.
- The addition of the areas of the N rectangles approximates the area under the curve.
- The greater N, the more accurate is the approximation of PI



Estimation of PI by the integration method

Sequential code

```
int main(int argc, char *argv[]) {
  int i, done = 0, n;
  double PI25DT = 3.14159265358979323846264643;
  double pi, h, sum, x;

  while (!done) {
    printf("Enter the number of intervals: (0 quits) \n");
    scanf("%d",&n);
    if (n == 0) break;

    h   = 1.0 / (double) n;
    sum = 0.0;
    for (i = 1; i <= n; i++) {
      x = h * ((double)i - 0.5);
      sum += 4.0 / (1.0 + x*x);
    }
    pi = h * sum;

    printf("pi is approx. %.16f, Error: %.16f\n", pi, fabs(pi - PI25DT));
  }
}
```

Estimation of PI by the integration method

Parallelization

- SPMD implementation
- I/O (scanf/printf) is made by process 0
- Distribute n to all the processes (with Send/Recv)
- Divide the workload of the for loop with "step" i+=numprocs instead of i++
- Gather the estimation of PI in each process (with Send/Recv)

Conditions of the assignment

- Value: 0.25
- Deadline: April 22nd
- Must be done in couples and defended in the laboratory class