Modelling and implementation of algorithms in applied mathematics using MPI

Curitiba, March, 2011

1. In the last exercise we wrote a parallel C-program, which computes the approximate integral $\int_a^b f(x)dx$ of a function

$$f : [a, b] - - > \mathbb{R}$$

using the following formula:

$$\int_a^b f(x)dx \approx \frac{b-a}{n} \left(\frac{1}{2} f(a) + \frac{1}{2} f(b) + \sum_{i=1}^{n-1} f\left(a + \frac{i(b-a)}{n} \right) \right).$$

   (a) Modify the program in such a way that you ask at process 0 for a,b and n. Broadcast this information to the other processes.

   (b) Use a reduction operator for gathering the local contributions.

2. Write a program for the parallel computation of a scalar product. Process 0 should read the data.

3. Implement the following sequential algorithm for the solution of the linear system arising for Finite Differences. Compare the number of needed iterations steps for Jacobis and Gauss-Seidel method. Use the residual as a stopping criterion. The right hand side is given by $f = 1$.

4. Write a parallel version of Jacobis method. Assume that the number of processes divides $N - 1$.

Algorithm 1: **Jacobis method (Poisson Problem)**

Choose initial vector $u^0 \in \mathbb{R}^n$
For $k = 1, 2, \ldots$
For $j = 1, 2, \ldots, (N-1)$
For $i = 1, 2, \ldots, (N-1)$
$u_{ij}^k = \frac{1}{4}\left(u_{i,j-1}^{k-1} + u_{i-1,j}^{k-1} + u_{i,j+1}^{k-1} + u_{i+1,j}^{k-1} + h^2 f_{ij}\right)$
end i
end j
- ————————————————————————–

Algorithm 2: **Gaus-Seidel method (Poisson Problem)**

Choose initial vector $u^0 \in \mathbb{R}^n$
For $k = 1, 2, \ldots$
For $j = 1, 2, \ldots, (N-1)$
For $i = 1, 2, \ldots, (N-1)$
$u_{ij}^k = \frac{1}{4}\left(u_{i,j-1}^{k} + u_{i-1,j}^{k} + u_{i,j+1}^{k-1} + u_{i+1,j}^{k-1} + h^2 f_{ij}\right)$
end i
end j
- ————————————————————————–