PROPOSITION of PROJECTS

necessary to get credit in
the 'Introduction to Parallel Programing in 2014'

GENERALITIES

- The text may be written in Polish, or English.
- The text has to contain two parts:
  1. **Theoretical Introduction giving formulation of the problem, necessary explanations, description of obtained results**
  2. **Precise information about the computer program realizing the problem to be solved, instruction how to prepare data, how to run the program etc.**

==================================

PROJECT NR 1.
BOX SCHEME OF THE ORDER 2

Consider **the linear transport equation in 1D:**

$$(1) \qquad u_t(t,x) + \alpha u_x(t,x) = 0, \ t \in [0,T], \ x \in \Omega = [0,L]$$

and the **BOX - scheme (2)** for **(1)**

$$(2) \qquad a u_{k+1}^{n+1} + b u_k^{n+1} = b u_{k+1}^n + a u_k^n$$

on the grid

$$x_k = kh, \quad t_n = n\tau, \quad \lambda = \frac{\tau}{h}$$

$$k = 0, 1, \cdots, M, \quad n = 0, 1, \cdots, N, \quad hM = L, \quad \tau N = T$$

We introduced also the **integral version** of equation **(1)** on 'grid-box'

$$(*)\qquad \begin{array}{cccc} n+1 & & - & \\ & | & & | \\ n & & - & \\ & k & & k+1 \end{array}$$

$$(3)\qquad \int_{x_k}^{x_{k+1}} \int_{t_n}^{t_{n+1}} (u_t + \alpha u_x) dt dx = 0$$

or

$$(4)\qquad \int_{x_k}^{x_{k+1}} [u(t_{n+1}, x) - u(t_n, x)] dx +$$

$$+\alpha \int_{t_n}^{t_{n+1}} [u(t, x_{k+1}) - u(t, x_k)] dt = 0$$

The **Box-scheme (2)** was derived from the integral equation **(4)** via approximation of integrals by the 'trapezoidal rule' of approximate integration. We proved, that the Box scheme **(2)** approximates **(1)** with the order **1**.

VARIOUS QUESTIONS AND SMALL PROBLEMS CONCERNING THE BOX SCHEME

- The center of the **BOX (∗)** is a particular point. Find the order of approximation for the soluton of **(1)** in this point.

- Take the four centers of the four sides of **(∗)**, and use them to approximate the four integrals in the formula **(4)**. Instead applying the trapezoidal rule, use the **Gaussian rule of approximation of integral (with a single node)**. Write down the obtained finite difference scheme (we shall call it a **New Box Scheme**).

  Find the order of approximation of the four integrals.

  Find the order of approximation of the solution of **(1)** in these four points.

- Suppose, we want to apply the **NEW BOX SCHEME**.

  QUESTION: What is the order of approximation of the solution of **(1)** at the center of the **NEW BOX**?

A LOGISTIC PROBLEM: It seems that the most natural way to apply the **NEW BOX** is use of the grid **skew with respect to the old one.**

What to do if we dont want to go **too far from the usual rectangular domain with sides parallel to the x and t axes**?

- Suppose that all our difficulties quotted above have been overcomed in the satisfactory way.

  IMPORTANT QUESTION: HOW TO BUILD PARALLEL ALGORITHM BASED ON THE NEW BOX SCHEME, AND THEN TO BUILD THE CLUSTER PROGRAM?

=======================================

PROJECT NR 2
TWO DIMENSIONAL TRANSPORT IN RECTANGULAR DOMAIN

Consider two dimensional rectangular domain $\Omega = [0, L1] \times [0, L2]$ in R^2 and the two dimensional transport equation

$$u_t + \alpha_1 u_{x_1} + \alpha_2 u_{x_2} = 0 \tag{1}$$

where $u = u(t, x_1, x_2, \alpha_1, \alpha_2)$ is a function of five independent variables: $\underline{x} = (x_1, x_2) \in \Omega$ and $\underline{\alpha} = (\alpha_1, \alpha_2) \in \mathcal{A} = [AL, AR]$. Let us emphasize that the pair (α_1, α_2) may be interpreted as some 'elementary velocity'.
The time splitting technique will be used at each time level of the time grid:

$$\{t_n\}, \quad n = 0, 1, \cdots, NN, \quad t_{n+1} = t_n + \tau$$

$$v_t + \alpha_1 v_{x_1} = 0, \quad v(t_n, \cdot) = u(t_n, \cdot) \tag{2}$$

$$w_t + \alpha_2 w_{x_2} = 0, \quad w(t_n, \cdot) = v(t_{n+1}, \cdot) \tag{3}$$

Remember that the choice of algorithm has to depend on the sign of α_i!

DIVISION BETWEEN PROCESSORS

Assume that for fixed time level t_n our grid is as follows:

$$x_{1,k_1}, \quad x_{2,k_2}, \quad \alpha_{1,l_1}, \quad \alpha_{2,l_2}$$

$$x_{i,k_i+1} = x_{i,k_i} + h_i, \quad k_i = 0, 1, \cdots, M_i, \quad i = 1, 2$$

$$\alpha_{i,l_i+1} = \alpha_{i,l_i} + ah_i, \quad l_i = 0, 1, \cdots, MA_i, \quad i = 1, 2$$

Perhaps the simplest possibility is to divide the $\{x_1, \ x_2\}$ grid into equal strips parallel to x_2-axis, and any such strip, together with whole $\{\alpha_1, \ \alpha_2\}$ grid assigne to one processor. Now we solve the equation (2) on the entire grid, using the Schur method.

Second step is to solve the equation (3) applying the BOX (without the Schur system!) but in parallel in each processor.

All results for fixed time level t_n are stored as four dimensional array, say $u(k_1, k_2, l_1, l_2)$.

MEDIUM VALUE

At each complete time step (remember that time splitting is used!) we can compute some kind of **medium value** of $u(x_1, x_2, l_1, l_2)$ with respect to α_1 and α_2. The simplest, but not the most interesting medium value is the integral

$$F(x_1, x_2) = \int_{\mathcal{A}} u(x_1, x_2 \alpha_1, \alpha_2) d\alpha_1 d\alpha_2$$

There is a lot of possibilities to define various medium values. Some kind of medium values may contain **an active mixing mechanism.**

The power necessary to run this model is supplied by initial conditions at $t = t_0 = 0$, and then, at each time step, by boundary conditions of Dirichlet type, which may depend on time t and on (α_1, α_2).

===

PROJECT NR 3
LINEAR EQUATION OF CONVECTION-DIFFUSION

This equation is of the followig form:

$$(1) \qquad u_t + \underline{\alpha}\nabla u - \nu\Delta u = 0,$$

where the diffusion coefficient ν is positive. In two dimensional case we have:

$$\nabla u = [u_{x_1}, u_{x_2}], \quad \Delta u = u_{x_1,x_1} + u_{x_2,x_2}.$$

Let us apply to (1) the **time splitting method** at the time grid

$$\{t_n\}, \quad t_{n+1} = t_n + \tau.$$

At any time step we get the following set of simpler problems:

$$(2) \qquad u_{1,t} + \alpha_1 u_{1,x_1} = 0, \quad u_1(t_n,\cdot) = u(t_n,\cdot)$$

$$(3) \qquad u_{2,t} + \alpha_2 u_{2,x_2} = 0, \quad u_2(t_n,\cdot) = u_1(t_{n+1},\cdot)$$

(do not forget that the choice of algorithm for above problems depends on the signs of α's!)

$$(4) \qquad u_{3,t} - \nu u_{3,x_1,x_1} = 0, \quad u_3(t_n,\cdot) = u_2(t_{n+1},\cdot),$$

$$(5) \qquad u_{4,t} - \nu u_{4,x_2,x_2} = 0, \quad u_4(t_n,\cdot) = u_3(t_{n+1},\cdot).$$

In the equations **(2)** and **(3)** we meet the transport problems (see PROJECT Nr 2), while for **(4)** and **(5)** we have to solve the so called **HEAT EQUATIONS**. Applying the same strategy of division between the processors as in the **PROJECT Nr 2**, we use the algorithms with Schur systems for equations with even numbers, and the algorithm without Schur systems for odd numbers equations. For transport equations we use either the two-diagonal systems with the Schur equations or the BOX scheme, while for Heat Equation we use tree-diagonal system with Schur matrix or without Schur matrix, always with the boundary conditions at both sides of the interval. If we consider the α's as independent variables, then we may follow the entire way described in the **PROJECT Nr 2.**

====================================

PROJECT Nr. 4
TIME SPLITTING OF THE HIGH ORDER

In the paper:

"OPTIMIZED HIGH-ORDER SPLITTING METHODS FOR SOME CLASSES OF PARABOLIC EQUATIONS"
(by S.BLANES, F.CASAS, P.CHARTIER AND A.MURUA, Mathematics of Computation, vol., 82 nr 283, pp. 1559-1576)

some information on **Time splitting Algorithms of high order** are given. I have a copy of this paper, and I can give it to the person interested in this subject.

Informations given in this paper can be applied in our work.

We can try to apply it for approximate solution of the equation of transport, (even if it is equation of the hyperbolic type), to parabolic heat equation or to the equation of convection-diffusion which is of mixed type. As in the PROJECTS 2 and 3, the splitting should concern the dimension of the problem: we replace the two dimensional problem by one dimensional problems solved at two steps, but now with the new time spplitting algorithm. The work proposed here should try to answer the question:

This higher order splitting is better in the computational work than the simple one used in our projects?

==

PROJECT Nr 5
QUASI SCHOCK WAVE IN 2D

This project is in fact a simplified version of the **PROJECT Nr 2**, but here the goal is different.

It is known that such effects as **Shock Waves** can be realized in conservation laws equations (the transport equation is the equation of this kind) when the noncontinuity occurs in the initial condition. Let us try to get something like **Shock Waves** in two dimensional transport, but under rather special conditions.

The description of what has to be done, can be found in the **PROJECT Nr 2**. The initial condition can be a **two-dimensional step-function** (it is very easy!). Simplification is such that $\boldsymbol{\alpha}$'s are no more treated as the independent variables, but now the chosen $\boldsymbol{\alpha_1, \cdots, \alpha_4}$ are the data of the program. We know, that the $\boldsymbol{\alpha}$'s are interpreted as coordinates of velocity.

The computational experiment is following: we push with (very) high velocity (high $\boldsymbol{\alpha}$'s) our travelling object through our two dimensional domain. The power is given by Dirichlet boundary values which may grow with the time. Of course, we observe the results.