Comparison of Parallel Simulation Techniques

EUROSIM - Simulation News Europe features a series on comparisons of simulation software. Simulation languages are compared in terms of their features for modelling and experimentation using simple and easily comprehensible models drawn from a number of different application areas. This series on simulation software comparisons will be continued.

This issue introduces a new type of comparison dealing with the benefits of distributed and parallel computation for simulation tasks.

Three test examples have been chosen to investigate the types of parallelisation techniques best suited to particular types of simulation tasks.

Each test example should be first solved in a serial fashion to provide a reference for the investigation of speed-up factors. The examples should then be tested using the parallel facilities (software and hardware) available. Performance should be assessed in terms of a numerical value found by dividing the time for serial solution by the time for the parallel solution (speed-up factor *f*). Wherever appropriate, serial solutions should be based on the same environment. Measurements of time should be in terms of the total elapsed time for running the task. Information must be provided about the method of parallelisation or distribution of subtasks. If of interest, more than one solution for a particular test example may be offered. Furthermore, a rough indication should be provided for the program preparation time, especially for the parallel solution.

This new type of comparison addresses users of all types of parallel and distributed facilities. The spectrum may range from simulation languages, via general purpose programming languages, to special parallel languages and from networks of workstations, via special parallel computers, to very high performance computers.

The objective is to make comparisons of different types of problems and of methods for the parallelisation of simulation tasks. It is not intended that this should involve direct comparisons of the (hardware) performance of parallel facilities.

Solutions for publication in EUROSIM- Simulation News Europe should not be more than one and a half page in length (see sample solution on page 24). Opportunities for the publication of more extended discussions will be provided at the forthcoming EUROSIM Congress in Vienna where it is expected that there will be a special session on these comparisons of parallel techniques. Further details on the EUROSIM Congress may be found on page 23.

The first test example is a **Monte Carlo study**. A damped second order mass-spring system is described by the equation

$$m\ddot{x}(t) + kx(t) + d\dot{x}(t) = 0$$

$$\dot{x}(0) = 0\ ,\ x(0) = 0.1\ ,\ k = 9000\ ,\ m = 450$$

where the damping factor d should be chosen as a random quantity uniformly distributed in the interval [800, 1200].

The task is to perform 1000 simulation runs and to calculate and store the average responses over the time interval [0,2.] for the motion *x(t)* for subsequent plotting.

Figure 1 shows some simulation runs (using ACSL with RK4-algorithm with stepsize 0.001), figure 2 gives an example of the average response (calculated under PVM in the sample solution).

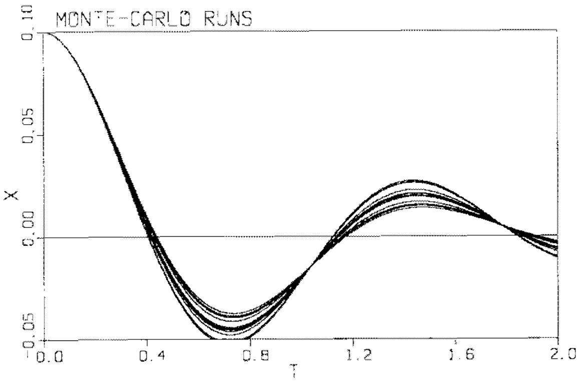

Figure 1

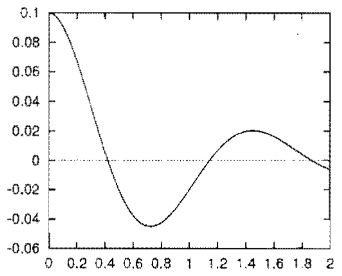

Figure 2

The second example is concerned with **coupled predator-prey population** models. Five predator-prey populations *(v1, v2)*, *(w1, w2)*, *(x1, x2)*, *(y1,y2)* and *(z1, z2)* are interacting. The model equations are:

$$\dot{v}_1 = a_v v_1 - b_v v_1 v_2 - c_v {v_1}^2$$
$$\dot{v}_2 = -\,d_v v_2 + e_v v_1 v_2 - f_v {v_2}^2 + r_v$$
$$r_v = v_2\,(g_v w_1 + h_v x_1 + j_v y_1 + k_v z_1)$$

$a_v = 2,\ b_v = 0.5,\ c_v = 0.01,\ d_v = 0.2,\ e_v = 0.4,$
$f_v = 0.02,\ g_v = 0.01,\ h_v = 0.02,\ j_v = 0.01,\ k_v = 0.03$

$$\dot{w}_1 = a_w w_1 - b_w w_1 w_2 - c_w w_1^2 + r_w$$
$$r_w = w_1(-g_w v_2 + h_w x_2)$$
$$\dot{w}_2 = -d_w w_2 + e_w w_1 w_2 - f_w w_2^2$$
$a_w = 1$, $b_w = 0.5$, $c_w = 0.02$, $d_w = 0.1$, $e_w = 0.4$, $f_w = 0.04$, $g_w = 0.02$, $h_w = 0.04$

$$\dot{x}_1 = a_x x_1 - b_x x_1 x_2 - c_x x_1^2 + r_x$$
$$r_x = -g_x x_1 v_2$$
$$\dot{x}_2 = -d_x x_2 + e_x x_1 x_2 - f_x x_2^2 + s_x$$
$$s_x = -h_x x_2 w_1$$
$a_x = 3$, $b_x = 0.9$, $c_x = 0.02$, $d_x = 0.2$, $e_x = 0.2$, $f_x = 0.04$, $g_x = 0.025$, $h_x = 0.1$

$$\dot{y}_1 = a_y y_1 - b_y y_1 y_2 - c_y y_1^2 + r_y$$
$$r_y = y_1(-g_y v_2 + h_y z_2)$$
$$\dot{y}_2 = -d_y y_2 + e_y y_1 y_2 - f_y y_2^2$$
$a_y = 1$, $b_y = 0.8$, $c_y = 0.04$, $d_y = 0.2$, $e_y = 0.6$, $f_y = 0.07$, $g_y = 0.03$, $h_y = 0.025$

$$\dot{z}_1 = a_z z_1 - b_z z_1 z_2 - c_z z_1^2 + r_z$$
$$r_z = -g_z z_1 v_2$$
$$\dot{z}_2 = -d_z z_2 + e_z z_1 z_2 - f_z z_2^2 + s_z$$
$$s_z = -h_z z_2 y_1$$
$a_z = 3$, $b_z = 0.7$, $c_z = 0.02$, $d_z = 0.5$, $e_z = 0.3$, $f_z = 0.04$, $g_z = 0.02$, $h_z = 0.04$

All initial populations are normalized to 1.

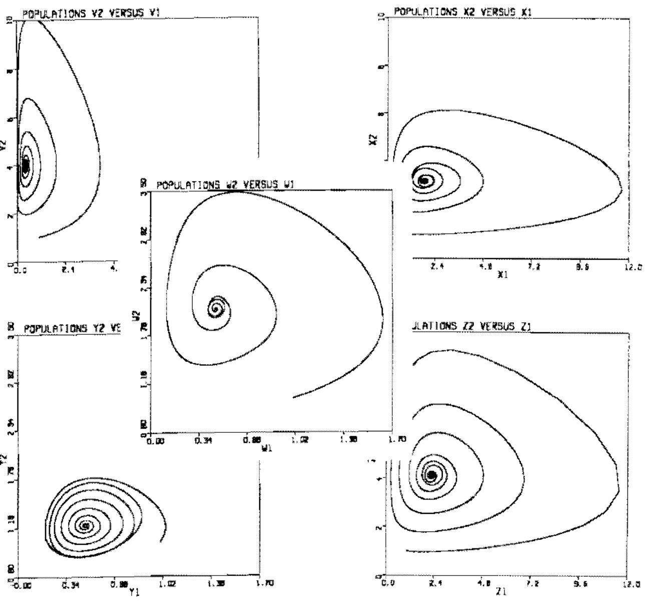

Figure 3

The task is to solve the system within the time interval [0, 100] in a serial fashion and in an appropriate parallel fashion and to provide the terminal values of each population. Solutions obtained using ACSL in double precison are shown in fig. 3, the terminal values at t=100 (stepsize 0.01, RK4-algorithm) and the equilibrium solutions are shown in the following table.

Population	t=100	Equilibrium	Difference
v_1	0.40494874	0.41235712	0.007408
v_2	3.9389271	3.992722	0.053795
w_1	0.45572852	0.45765668	0.001928
w_2	2.06389283	2.07652643	0.012634
x_1	1.86489724	1.86502448	0.000127
x_2	3.18278282	3.18097946	0.001803
y_1	0.4677289	0.47392759	0.006199
y_2	1.20213106	1.20497504	0.002844
z_1	2.27700122	2.2773994	0.000398
z_2	4.10808012	4.10656804	0.001512

It is expected that with this example little or no improvement may be found through parallelisation. Negative results are of considerable interest and should not be discarded.

The third example is based on a second order **partial differential equation** describing a swinging rope with length L fixed at one end and forced at the other.

$$u_{xx}(t,x) = a\, u_{tt}(x,t)$$

$$u(0,t) = 0,\ u(l,t) = b\, e^{-dt} \sin\omega t,\ u(x,0) = u_x(x,0) = 0$$

Discretisation by the method of lines by dividing the length into N equidistant intervals and replacing the differential quotient $u_{xx}(t,x)$ by a central difference quotient results in a set of weakly coupled equations:

$$k^2 a\, \ddot{u}_i(t) = u_{i-1}(t) - 2u_i(t) + u_{i+1}(t)\ ,\ i = 1, \dots, N-1;$$

$$u_i(0) = u_i(0) = 0,\ u_0(t) = u(0,t) = 0,$$

$$u_N(t) = u(L,t) = b\, e^{-dt} \sin\omega t$$

$$L = 10,\ a = 2,\ b = 1,\ d = 0.2,\ \omega = 1,\ k = L/N$$

The task is to solve the system of equations with a discretisation N = 800 or more lines within the time horizon [0, 30] in a serial and in an appropriate parallel fashion. As result the lines at $x=9L/10$, $x=3L/4$, $x=L/2$, $x=L/4$ and $x=L/10$ should be stored for subsequent plotting. Figure 4 shows results for these lines calculated with ACSL with double accuracy (integration stepsize 0.005, RK4-algorithm).

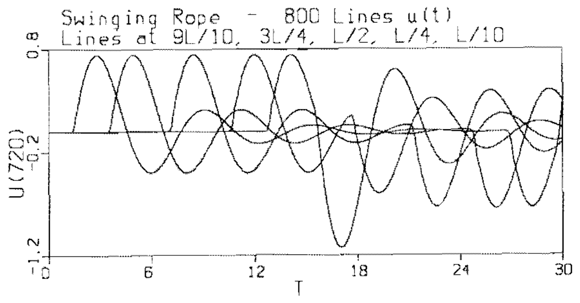

Figure 4