Sorbonne Université
Calcul haute performance : programmation et algorithmique avancées

Evaluated training session 24th October 2024
Simulation of explicit and implicit schemes on GPU

Students must submit a single .cu file that contains the solution. At the top of the source file, they must specify the family names and the first names of the pair. At the end of the session, they have to send the code to the email address of the instructor (L. ABBAS TURKI). It is necessary to name the source file that contains the solution using the syntax : Name1Name2.cu if the job is done in pairs or Name1Name2Name3.cu if the work is done in trinomials.

We are interested in parabolic Partial Differential Equation (PDE) simulation using several schemes. We thus compute $F(t,x) : [0,T] \times \mathbb{R}^+ \to \mathbb{R}^+$ solution of the backward PDE

$$\frac{\partial F}{\partial t}(t,x) + rx\frac{\partial F}{\partial x}(t,x) + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 F}{\partial x^2}(t,x) = rF(t,x), \quad F(T,x) = f(x). \tag{1}$$

Using the change of variable $u(t,x) = e^{r(T-t)}F(t,e^x)$, it will suffice to solve the equivalent PDE

$$\frac{1}{2}\sigma^2 \frac{\partial^2 u}{\partial x^2}(t,x) + \mu\frac{\partial u}{\partial x}(t,x) = -\frac{\partial u}{\partial t}(t,x), \quad \mu = r - \frac{\sigma^2}{2}, \quad u(T,x) = f(e^x) \tag{2}$$

In section 1, students have to write the syntax of `PDE_diff_k1` and `PDE_diff_k2` which solve (2) explicitly for several values of σ. In section 2, students have to write the syntax of `PDE_diff_k3` for implicit resolution and that of `PDE_diff_k4` for a resolution using the Crank-Nicolson scheme.

In all sections, the final value $f(e^x) = \max(K - e^x, 0)$. In file `PDE.cu`, the values at the boundaries `pmin` and `pmax`, associated with f, are also given. The values of the parameters are : $T = 1$, $K = 100$, $S_0 = 100$, $r = 0.1$ and we will launch several simulations parameterized by σ which takes its values in $[0.1, 0.5]$. Discretization uses 256 cells for variable x and 64 cells for σ. Furthermore, we discretize $[0,T]$ with 10000 time steps.

1 Explicit scheme for PDE simulation

Considering an explicit Euler scheme centered in space, the simulation of u is done using :

$$\begin{gathered} u_{i,j} = p_u u_{i+1,j+1} + p_m u_{i+1,j} + p_d u_{i+1,j-1}, \quad u_{i,j} = u(t_i, x_j), \quad u(T,x) = f(e^x), \\ p_u = \frac{\sigma^2 \Delta t}{2\Delta x^2} + \frac{\mu \Delta t}{2\Delta x}, \quad p_m = 1 - \frac{\sigma^2 \Delta t}{\Delta x^2}, \quad p_d = \frac{\sigma^2 \Delta t}{2\Delta x^2} - \frac{\mu \Delta t}{2\Delta x}. \end{gathered} \tag{3}$$

1. Keeping the time loop outside, define the syntax of `PDE_diff_k1` for the simulation of (3) which correctly approximates the value of $F(0,x)$ when x is included in the interval $[0.7 * S_0, 1.3 * S_0]$.
2. Now introducing the time loop in the kernel, define the syntax of `PDE_diff_k2` for the simulation of (3) which approximates the value of $F(0,x)$ when x is included in the interval $[0.7 * S_0, 1.3 * S_0]$. In this code, use dynamic shared memory allocation.
3. Compare solution 1 and 2 and debug the program using the analytical solution $F(t,x)$ of the PDE (1) given by the closed form expression

$$F(t,x) = N(-d_2)Ke^{-r(T-t)} - N(-d_1)x, \quad N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-y^2/2} dy. \tag{4}$$

with $d_1 = \dfrac{\ln(x/K) + (r + \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}$ and $d_2 = d_1 - \sigma\sqrt{T-t}$, N is the cumulative distribution function of Gaussian random variable. N is `NP` given in the code.

2 Implicit and Crank-Nicolson scheme for PDE simulation

Considering an implicit Euler scheme centered in space, the simulation of u is done using :

$$\begin{gathered} u_{i+1,j} = q_u u_{i,j+1} + q_m u_{i,j} + q_d u_{i,j-1}, \quad u_{i,j} = u(t_i, x_j), \quad u(T,x) = f(e^x), \quad \mu = r - \tfrac{\sigma^2}{2} \\ q_u = -\frac{\sigma^2 \Delta t}{2\Delta x^2} - \frac{\mu \Delta t}{2\Delta x}, \quad q_m = 1 + \frac{\sigma^2 \Delta t}{\Delta x^2}, \quad q_d = -\frac{\sigma^2 \Delta t}{2\Delta x^2} + \frac{\mu \Delta t}{2\Delta x} \end{gathered} \tag{5}$$

4. By putting the time loop into the kernel, define the syntax of `PDE_diff_k3` for the simulation of (5) which approximates the value of $F(0,x)$ when x is included in the interval $[0.7 * S_0, 1.3 * S_0]$. In this code, use the dynamic allocation of shared memory.

With a Crank-Nicolson Scheme, we find the following induction

$$\begin{gathered} q_u u_{i,j+1} + q_m u_{i,j} + q_d u_{i,j-1} = p_u u_{i+1,j+1} + p_m u_{i+1,j} + p_d u_{i+1,j-1}, \quad \mu = r - \tfrac{\sigma^2}{2} \\ p_u = \frac{\sigma^2 \Delta t}{4\Delta x^2} + \frac{\mu \Delta t}{4\Delta x}, \quad p_m = 1 - \frac{\sigma^2 \Delta t}{2\Delta x^2}, \quad p_d = \frac{\sigma^2 \Delta t}{4\Delta x^2} - \frac{\mu \Delta t}{4\Delta x} \\ q_u = -\frac{\sigma^2 \Delta t}{4\Delta x^2} - \frac{\mu \Delta t}{4\Delta x}, \quad q_m = 1 + \frac{\sigma^2 \Delta t}{2\Delta x^2}, \quad q_d = -\frac{\sigma^2 \Delta t}{4\Delta x^2} + \frac{\mu \Delta t}{4\Delta x} \end{gathered} \tag{6}$$

5. By putting the time loop into the kernel, define the syntax of `PDE_diff_k4` for the simulation of (6) which approximates the value of $F(0,x)$ when x is included in the interval $[0.7 * S_0, 1.3 * S_0]$. In this code, use the dynamic allocation of shared memory.

Regarding the resolution of systems of sizes n, one system per block, we use the parallel cyclic reduction for each system in each block. This routine is also given. This method consists of solving a system $TX = Y$ where

$$T = \begin{pmatrix} d_1 & c_1 & & & & \\ a_2 & d_2 & c_2 & & 0 & \\ & a_3 & d_3 & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & \ddots & \ddots & c_{n-1} \\ & & & & a_n & d_n \end{pmatrix}$$ and n is a power of 2 like $n = 256, 512, 1024, \ldots$

starting with a system of n equations with n unknowns to two systems of $n/2$ equations with $n/2$ unknowns, then to 4 systems of $n/4$ equations to $n/4$ unknowns, ..., up to $n/2$ systems of 2 equations with 2 unknowns that can be easily solved.