CUDA Programming on NVIDIA GPUs
Mike Giles

Practical 3: Finite Difference notes

This practical uses finite difference methods to approximate the solution of the Laplace PDE

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = 0$$

on the unit cube $0 \leq x, y, z, \leq 1$, subject to specified values for $u(x, y, z)$ on the boundary.

Using a uniform grid with spacing Δ in each direction, we define $u_{i,j,k}$ to be an approximation to $u(i\Delta, j\Delta, k\Delta)$. We then have

$$\begin{aligned}
\frac{\partial^2 u}{\partial x^2} &\approx \Delta^{-2} \left(u_{i+1,j,k} - 2u_{i,j,k} + u_{i+1,j,k}\right) \\
\frac{\partial^2 u}{\partial y^2} &\approx \Delta^{-2} \left(u_{i,j+1,k} - 2u_{i,j,k} + u_{i,j+1,k}\right) \\
\frac{\partial^2 u}{\partial z^2} &\approx \Delta^{-2} \left(u_{i,j,k+1} - 2u_{i,j,k} + u_{i,j,k+1}\right)
\end{aligned}$$

and using these approximations in the Laplace PDE gives

$$\begin{aligned}
\Delta^{-2} \left(u_{i+1,j,k} - 2u_{i,j,k} + u_{i+1,j,k}\right) &+ \\
\Delta^{-2} \left(u_{i,j+1,k} - 2u_{i,j,k} + u_{i,j+1,k}\right) &+ \\
\Delta^{-2} \left(u_{i,j,k+1} - 2u_{i,j,k} + u_{i,j,k+1}\right) &= 0
\end{aligned}$$

which can be re-arranged to give

$$u_{i,j,k} = \frac{1}{6} \left(u_{i+1,j,k} + u_{i+1,j,k} + u_{i,j+1,k} + u_{i,j+1,k} + u_{i,j,k+1} + u_{i,j,k+1}\right).$$

To solve this linear system of equations, given specified boundary conditions, we use the Jacobi iteration

$$u^{(n+1)}_{i,j,k} = \frac{1}{6} \left(u^{(n)}_{i+1,j,k} + u^{(n)}_{i+1,j,k} + u^{(n)}_{i,j+1,k} + u^{(n)}_{i,j+1,k} + u^{(n)}_{i,j,k+1} + u^{(n)}_{i,j,k+1}\right).$$

It can be proved that this converges to the solution of the finite difference equations.

Note: there are other much better iterative methods (conjugate gradient, multigrid) which should be used for real applications but they are more complicated – that's why we are using Jacobi iteration in this practical.