UNIVERSITÄT BASEL

Prof. Dr. Helmar Burkhart
Prof. Dr. Florina Ciorba
Computer Science

Spiegelgasse 1
CH-4051 Basel

Danilo Guerrera
Antonio Maffia
Ali Omar

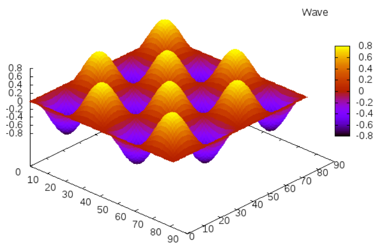
http://informatik.unibas.ch

High-Performance Computing (CS311) **Spring Semester 2016**

High-Performance Computing Project: "Run, Stencil, Run!"

During the High-Performance Computing course, we will solve the classical wave equation numerically using the different programming models which will be introduced.
We consider the classical wave equation with Dirichlet boundary conditions,

$$\begin{aligned} \frac{\partial^2 u}{\partial t^2} - c^2 \nabla^2 u &= 0 \quad \text{in } \Omega, \\ u &\equiv 0 \quad \text{on } \partial\Omega, \end{aligned}$$



and initial condition

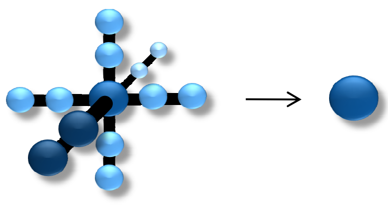
$$u(x, y, z, 0) = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z)$$

on $\Omega = [-1, 1]^3$, using an explicit finite difference method both in space and time. For the discretization in time we use a second-order leap frog scheme with time step Δt. For the discretization in space, we choose a fourth-order discretization of the Laplacian on the structured uniform grid Ω_h with step size h:

$$u_{ijk}^{n+1} - 2u_{ijk}^{n} + u_{ijk}^{n-1} - \left(\Delta t^2\right) c^2 \nabla_h^2 u_{ijk}^{n} = 0,$$

where

$$\begin{aligned} h^2 \nabla_h^2 u_{ijk} = \; & \tfrac{-15}{2} \quad u_{ijk} + \\ & \tfrac{-1}{12} \quad (u_{i-2,j,k} + u_{i,j-2,k} + u_{i,j,k-2}) + \\ & \tfrac{4}{3} \quad (u_{i-1,j,k} + u_{i,j-1,k} + u_{i,j,k-1}) + \\ & \tfrac{4}{3} \quad (u_{i+1,j,k} + u_{i,j+1,k} + u_{i,j,k+1}) + \\ & \tfrac{-1}{12} \quad (u_{i+2,j,k} + u_{i,j+2,k} + u_{i,j,k+2}) \end{aligned}$$



is the actual spatial stencil visualized above. The symbol ∇_h^2 denotes the discrete version of the Laplacian. The pseudo-code for the sequential algorithm is described by Algorithm **??**. It implements a Jacobi iteration in which 3 grids are used, u^-, which holds the values of the previous time step, u^0, which holds the values of the current time step, and u^+, which receives the values of the computations depending on u^- and u^0 (line 3).

Algorithm 1 Pseudo-code for the stencil Jacobi iteration

Require: Arrays u^{-}, u^{0}; Stencil operator ∇_h^2

Ensure: Array u^{+}

for $t \leftarrow 1 \ldots t_{\text{max}}$ **do** ▷ Iterate over the time domain

    for $(i, j, k) \in \Omega_h$ **do** ▷ Iterate over the discrete domain, the index set Ω_h

        $u^{+}_{ijk} \leftarrow 2u^{0}_{ijk} - u^{-}_{ijk} + \frac{\Delta t^2}{h^2} c^2 \nabla_h^2 u^{0}_{ijk}$ ▷ Compute the next time step at domain index (i, j, k)

    end for

    tmp $\leftarrow u^{-};\ u^{-} \leftarrow u^{0};\ u^{0} \leftarrow u^{+};\ u^{+} \leftarrow$ tmp ▷ Rotate pointers for the 3 arrays u^{-}, u^{0}, u^{+}

end for