**G-Node developer workshop on neuronal GPU-computing**
**Munich, April 12-13, 2012**

# Workshop dev project topics

## Code generation for GPUs

suggested by common interest
Followers: Giovanni Idili, Dan Goodman (?), Thomas Nowotny (?), Damien Drix (?)

## Cable equations on GPU

Suggested by Romain Brette
Followers: Remy Degenne, Giovanni Idili

## Interfacing Brian to GeNN

Suggested by Thomas Nowotny
Followers: Dan Goodman,

## Integrating NeMo and Brian

Suggested by Andreas Fidjeland
Followers: Dan Goodman

## A PyNN wrapper for GeNN

Suggested by Michael Schmuker
Followers: Thomas Nowotny, Pierre Yger

# Workshop talks

## Dan Goodman: Code generation for GPUs

In order for a neural network simulator to be flexible, it must allow users to specify arbitrary differential equations, threshold, reset and refractoriness conditions and equations. To achieve this, we generate CUDA code at runtime, and compile and run this using the PyCUDA package. The upcoming code generation framework in Brian allows for generation of Python, C++ and GPU code from the same set of defining equations and conditions. Issues I will cover include: converting between different expression syntaxes; handling loops, threads and vector

operations; handling control flow (if statements); enabling users to define extensions. This talk is from the perspective of a package author: code has to be generic and extensible. However, many of the issues will apply in more specific cases as well.

# Lightning talks

## Fabian Schoenfeld (Uni Bochum): Spatial representation in the hippocampus

There is a number of cells in the rodent hippocampus whose activity directly correlates with the animal's spatial activity: "Place Cells" which code for its actual location, "Head Direction Cells" which code for the direction the animal is looking at, and "Grid Cells" which introduce a allocentric metric. The aim of our modeling project is to simulate this activity based on realistic input and the Slowness Principle. The computation heavy training of our model is based on MDP ("Modeling toolkit for Data Processing") for Python and boosted by CUDA at its core bottleneck. In this talk I would like to introduce you to our project and present a small case study on how we fixed our primary bottleneck with CUDA and how we link CUDA to our Python implementation.

## David Higgins: OpenCL API calls

no abstract.