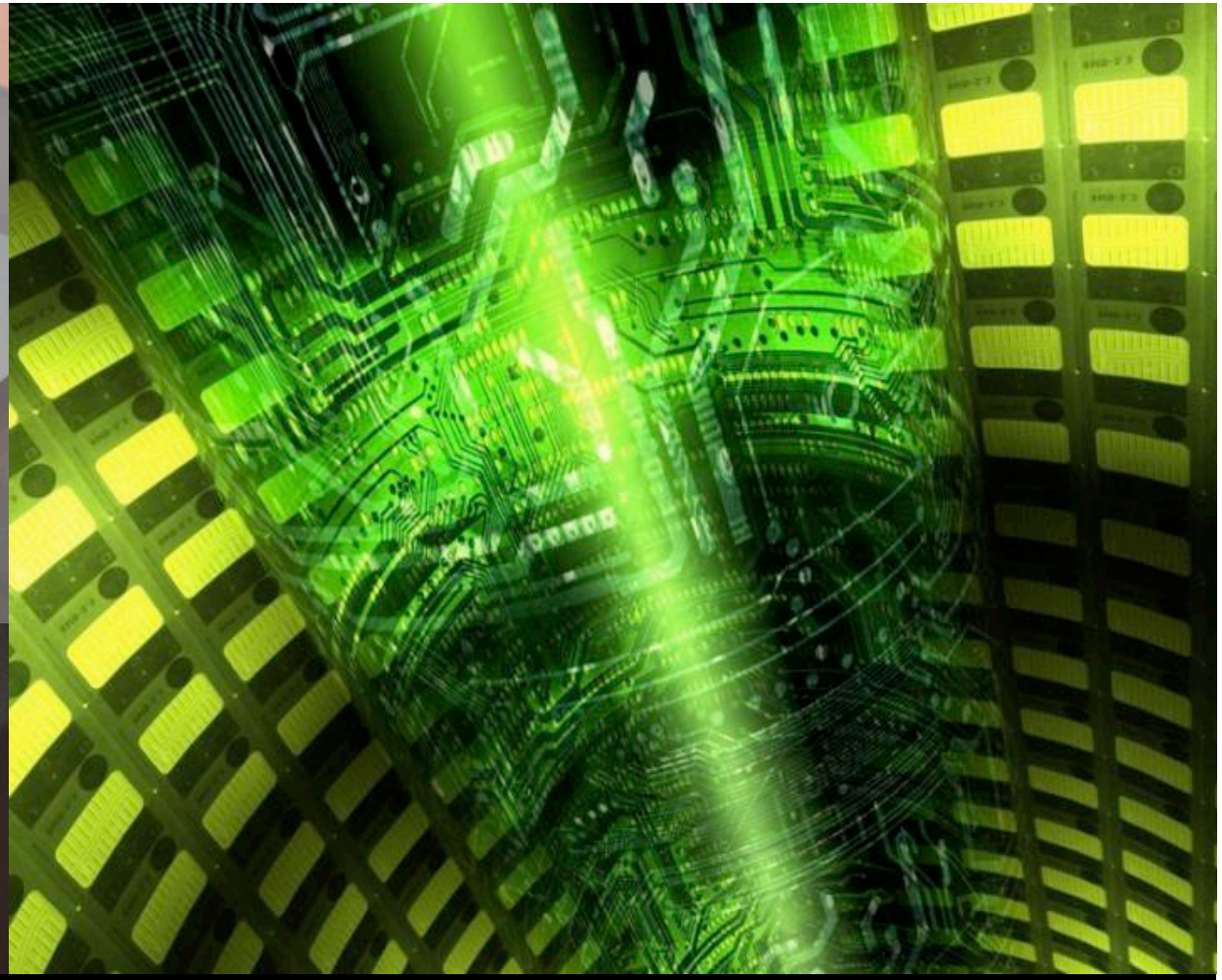


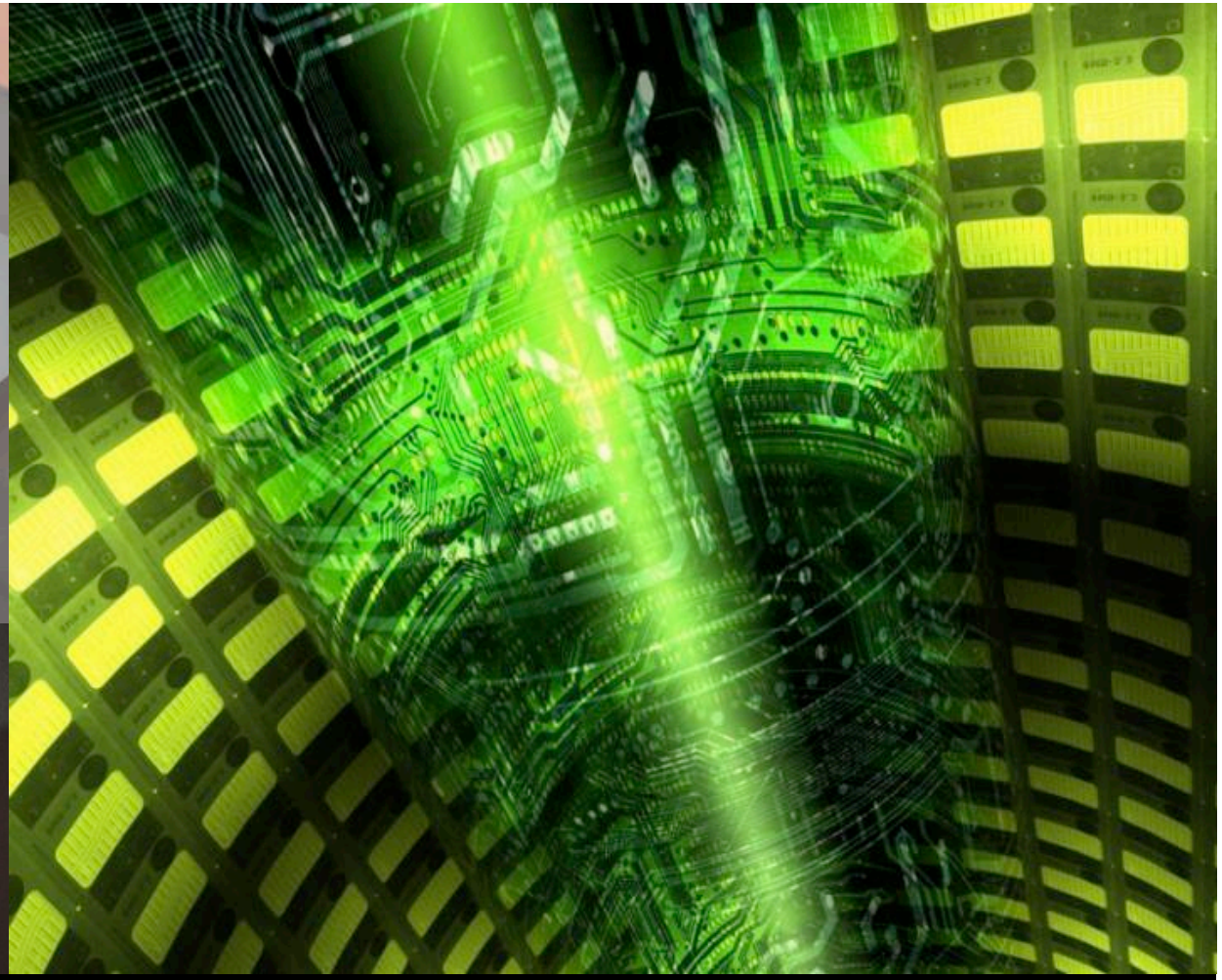
Introduction

Scilab on steroïds

- Objective :
  - Provide an easy way to go from a scripted prototype in Scilab to a full-fledged application
  - In fact an application taking advantage of GPU acceleration

Several steps

- Integrating GPU-enabled libraries
- Introducing a pragma-based typing system
- Enabling the generation of an autonomous application on GPU
- Extending the typing system with automated type inferences



HPC Project - Corporate Presentation

How to go from Scilab…

- An interpreted
- Dynamical environment
- Ideal for prototyping

… to a full application

- Typed variable
- To enable a compilation
- For performance
- To directly go to production

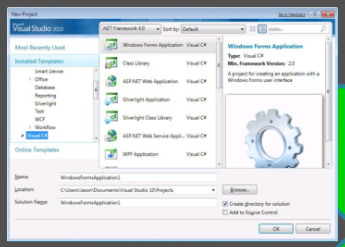
Producing application from Scilab scripts

- If you write a less dynamic code
- Due to an internal representation of the code
- A code C can be generated from a Scilab script
- So you could get an autonomous application
- From there, you could get a GPU-accelerated application

Par4All source-to-source compiler

- What is it ?

A usual C code



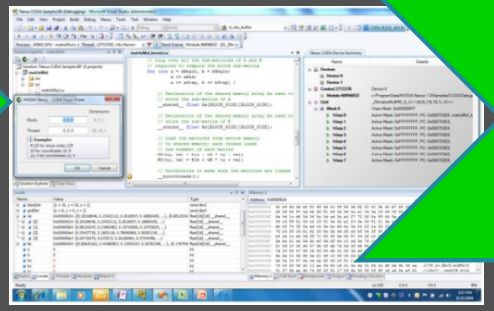
Benefit :
only one code

source to source Compiler

Why it is not working?

Some feedback on code quality

a C + Cuda code



A still-readable code

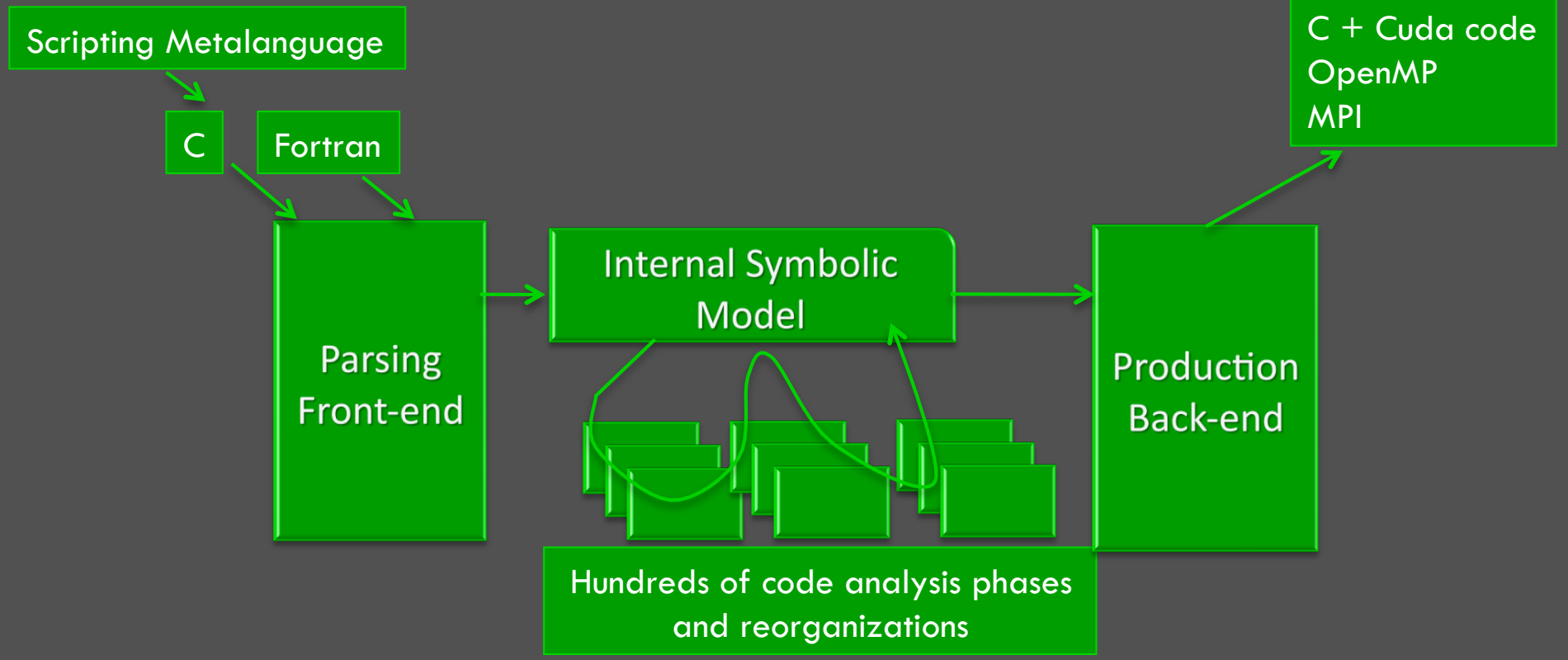
You can work on it:

Additional Tools

Back-end compiler

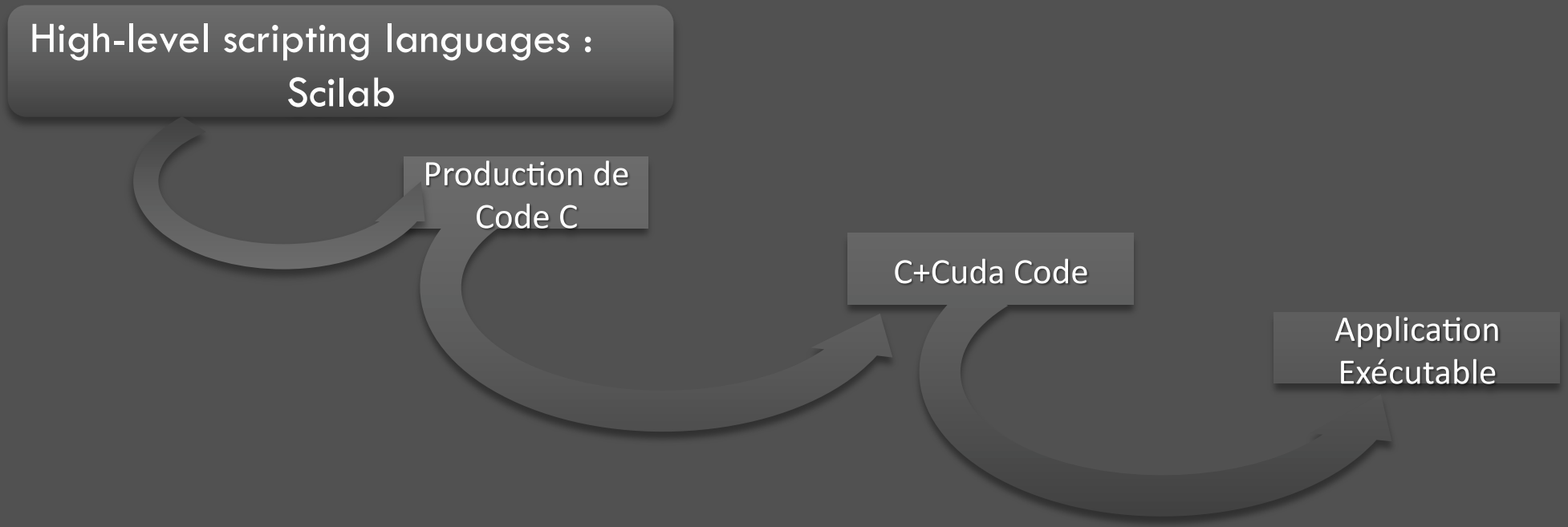
Executable Application

Par4All – the compiling flow



Scilab to GPU

- A natural extension

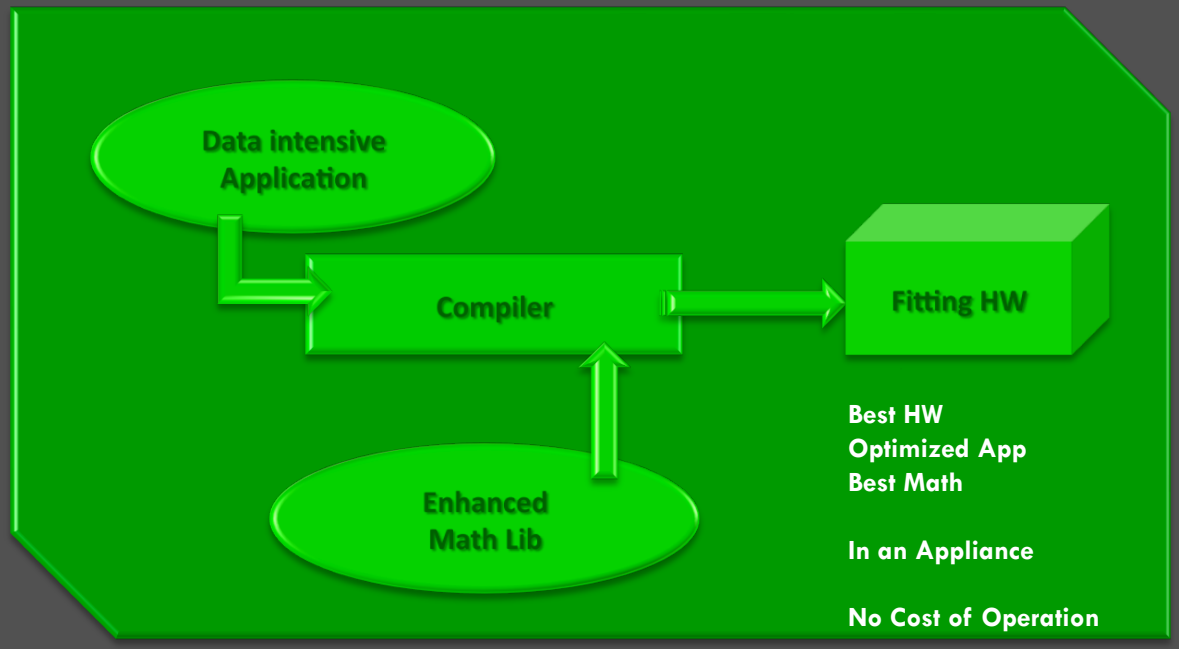


Generating type by inference

- To ease development
- Provide an environment able to produce type
  - Type generation during script writing by inference
  - Something that is provided in some strongly typed formal langages
- Of course, it is currently possible to write a script and no coherent typing can be generated
  - This will result in some limitation on “acceptable” scripts

Appliances Wild Systems

- A packaged system for immediate use
  - High-end Configuration : dual-socket X5570 & GPU Tesla



Thank you

Any question ?