The Numerical Template Toolbox or *Why I Stopped Worrying & Loved Boost::Proto*

Joel Falcou

LRI, Equipe PARALL - Université Paris Sud XI

19 Fev. 2010

Context

In Scientific Computing ...

- there is **Scientific** ..

Context

In Scientific Computing ...

- there is **Scientific** ..
  - Applications are domain driven
  - Users are often programming-agnostic
  - Huge base of legacy code in fancy languages

Context

In Scientific Computing ...

- there is **Scientific** ..
  - Applications are domain driven
  - Users are often programming-agnostic
  - Huge base of legacy code in fancy languages
- and there is **Computing** ...

Context

In Scientific Computing ...

- there is **Scientific** ..
  - Applications are domain driven
  - Users are often programming-agnostic
  - Huge base of legacy code in fancy languages
- and there is **Computing** ...
  - Often implies performances ...
  - ... which implies architectures support
  - .. which require expertise

Context

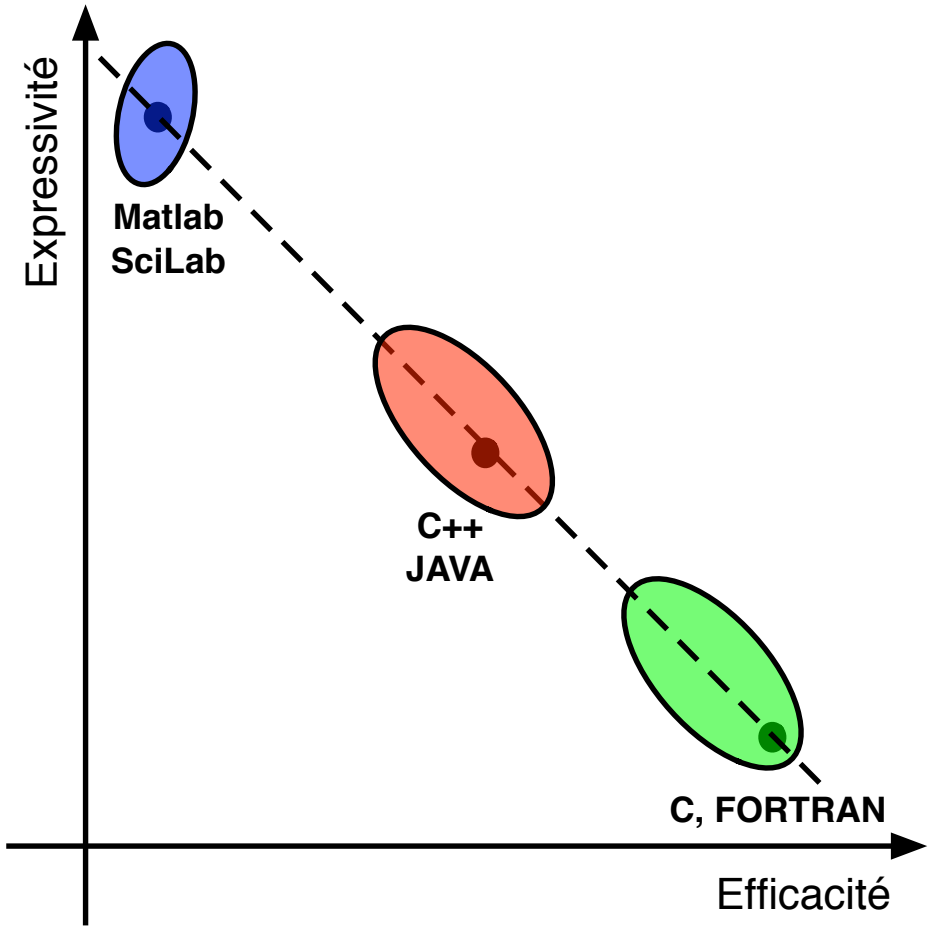
In Scientific Computing ...

- there is **Scientific** ..
  - Applications are domain driven
  - Users are often programming-agnostic
  - Huge base of legacy code in fancy languages
- and there is **Computing** ...
  - Often implies performances ...
  - ... which implies architectures support
  - .. which require expertise

The Problem

People **using** computer to do science want to do **science** first and don't care about the needed nuts and bolts and often don't understand them at all...

Fauna and Flora in S.C.



What do we want to do ?

Techniques

Use modern C++ idioms

- Expression Templates
- Policy based classes
- Meta-programming

What do we want to do ?

Techniques

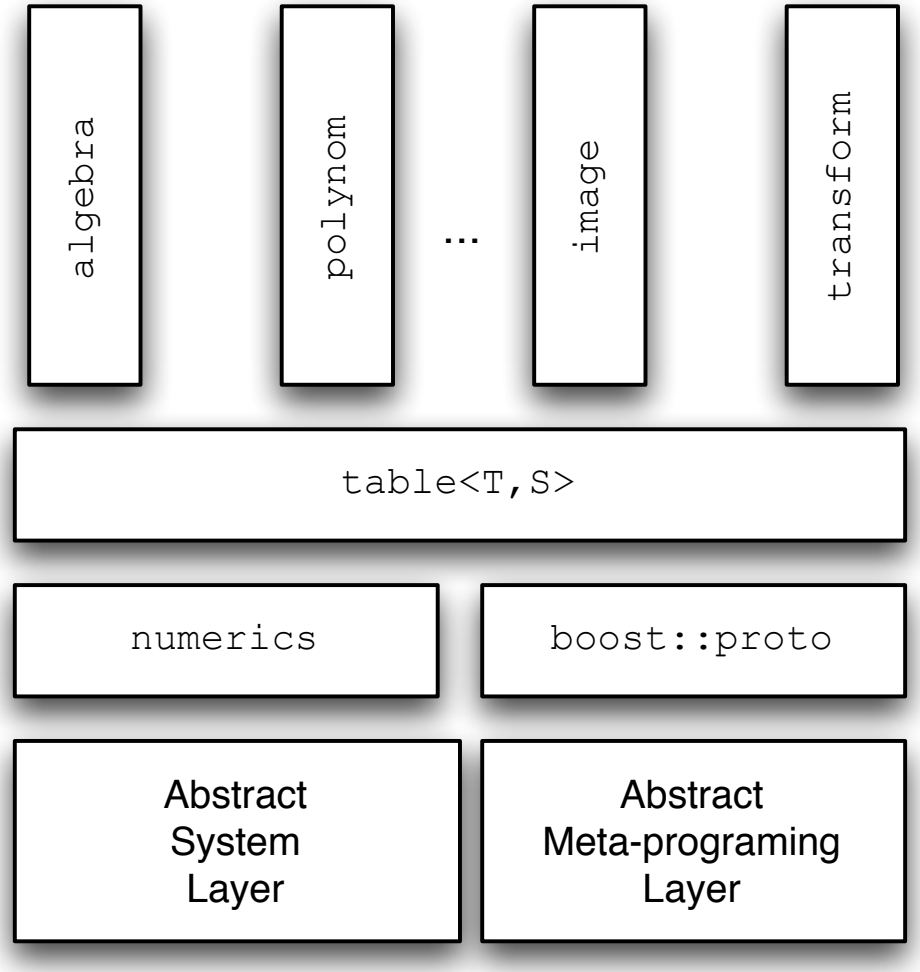
Use modern C++ idioms

- Expression Templates
- Policy based classes
- Meta-programming

Goals

- Hides all complex C++ construct away
- Provide a ”familliar” interface to users
- Let ”Power-users” be able to fine tunes

NT2 Global Layout



NT2 Outline

The User Level

- You know MATLAB, you know NT2
- Support for 280+ functions on values and containers
- Semantic-heavy interface

NT2 Outline

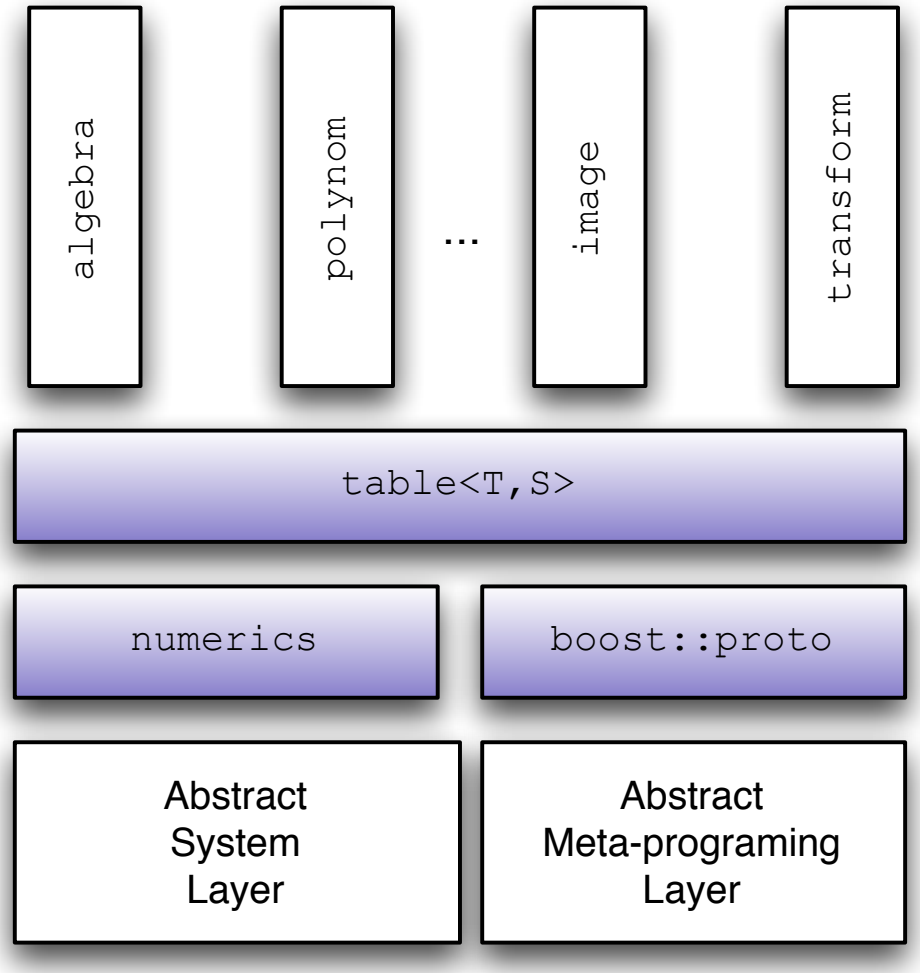
The User Level

- You know MATLAB, you know NT2
- Support for 280+ functions on values and containers
- Semantic-heavy interface

The Dev. Level

- Adding features should be easy
- Generic components built on themselves
- Simplify house-keeping

NT2 Main Interface



NT2 Main Interface

The `table` class

- Models generic, multidim. MATLAB array of values
- Notion of size and position as first class object
- Power-user can use STL or other C++ like interface

NT2 Main Interface

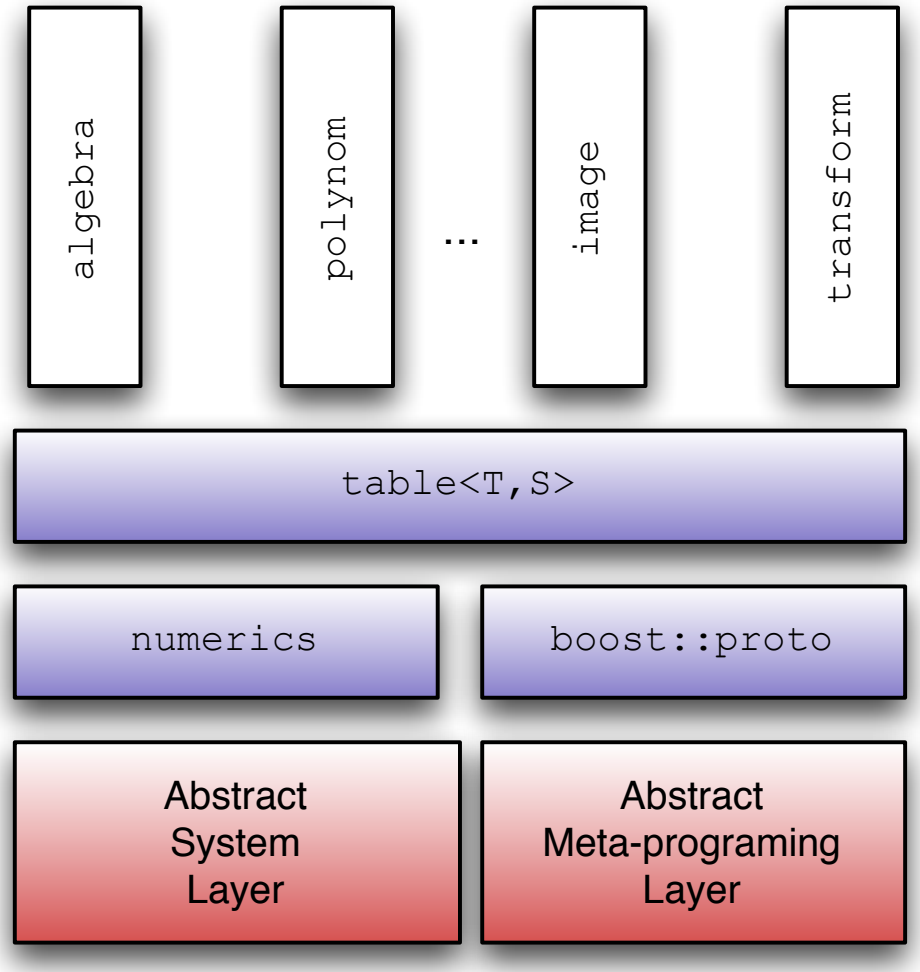
The `table` class

- Models generic, multidim. MATLAB array of values
- Notion of size and position as first class object
- Power-user can use STL or other C++ like interface

Relation to `proto`

- `proto` grammars + SFINAE simplify error messages
- All containers generates a single familly of generic AST
- All operations are done recursively with `proto` transforms
- Dev. can add any number of AST transformation steps

NT2 Internal Structure



NT2 Implementation

Support ILP/TLP

- Functions are tag-dispatched generic P.F.O.
- New architecture specific code is tagged and automatically recognized
- Dev. can specialize variation point in the A.S.L.

NT2 Implementation

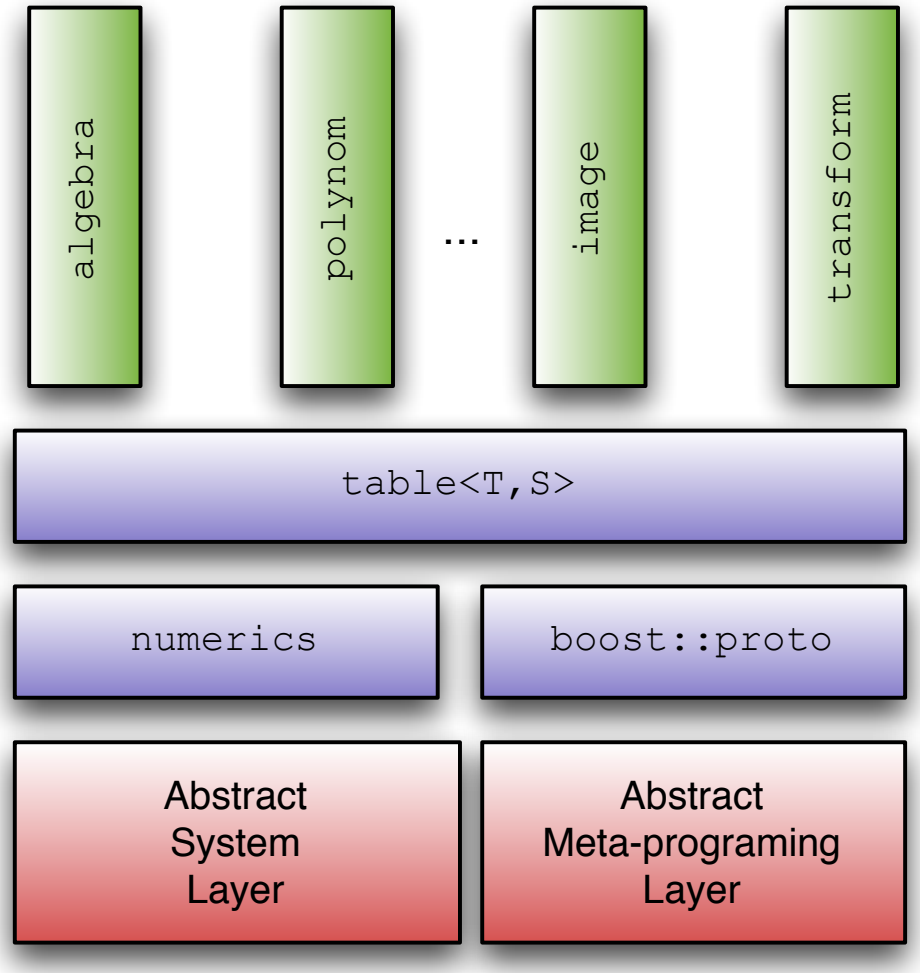
Support ILP/TLP

- Functions are tag-dispatched generic P.F.O.
- New architecture specific code is tagged and automatically recognized
- Dev. can specialize variation point in the A.S.L.

Extending NT2

- Compatible with `Boost` : serialization, fusion, MPL, MPI, etc ...
- New subsystem can be added through a set of extensions points
- Dev. can specialize variation point in the A.M.L.

NT2 Toolboxes



NT2 Toolboxes

Objectives

- Increase expressivity by specifying **semantic heavy** entities
- Semantic is usable all around AST transforms
- High-level informations grants better optimization

NT2 Toolboxes

Objectives

- Increase expressivity by specifying **semantic heavy** entities
- Semantic is usable all around AST transforms
- High-level informations grants better optimization

Examples

- `algebra` : matrix, vector, covector
- `polynoms` : root computing, polynom products
- `geometry` : shape and rigid transform

Sample Code

MATLAB to NT2

```
R = I(:,:,1);
G = I(:,:,2);
B = I(:,:,3);

Y = min(bitshift(abs(2104*R+4130*G+
                     802*B+135168),-13),235);
U = min(bitshift(abs(-1214*R-2384*G+
                      3598*B+1052672),-13),240);
V = min(bitshift(abs(3598*R-3013*G-
                     585*B+1052672),-13),240);
```

Sample Code

MATLAB to NT2

```
table<> R = I(_,_,1);
table<> G = I(_,_,2);
table<> B = I(_,_,3);

Y = min(bitshift(abs(2104*R+4130*G+
                     802*B+135168),-13),235);
U = min(bitshift(abs(-1214*R-2384*G+
                      3598*B+1052672),-13),240);
V = min(bitshift(abs(3598*R-3013*G-
                     585*B+1052672),-13),240);
```

Sample Code

MATLAB to NT2

```
table<settings(shallow)> R = I(_,_,1);
table<settings(shallow)> G = I(_,_,2);
table<settings(shallow)> B = I(_,_,3);

Y = min(bitshift(abs(2104*R+4130*G+
                     802*B+135168),-13),235);
U = min(bitshift(abs(-1214*R-2384*G+
                      3598*B+1052672),-13),240);
V = min(bitshift(abs(3598*R-3013*G-
                     585*B+1052672),-13),240);
```

Sample Code

MATLAB to NT2

```
table<uint32_t,settings(shallow)> R = I(_,_,1);
table<uint32_t,settings(shallow)> G = I(_,_,2);
table<uint32_t,settings(shallow)> B = I(_,_,3);;

Y = min(bitshift(abs(2104*R+4130*G+
                     802*B+135168),-13),235);
U = min(bitshift(abs(-1214*R-2384*G+
                      3598*B+1052672),-13),240);
V = min(bitshift(abs(3598*R-3013*G-
                     585*B+1052672),-13),240);
```

Let's round this up !

Computing for Scientist

- Contrary to other array/algebra library, NT2 choose to look strange for C++ users and easy for MATLAB users.
- We rely heavily on `Boost` as it simplify and streamline, platform support and modularization.
- `proto` helps us writing our code as real EBNF and semantic actions, just using templates
- Long list of existing applications running with NT2 : autonomous vehicles and drone, hand held device IP, etc...

Current and Future Works

What's we're cooking at the moment

- GPU support : ITOC project (CEA/LIX/LRI)
- Sparse matrix support (EDF R&D)
- Cell support : OMTE Digiteo project

Current and Future Works

What's we're cooking at the moment

- GPU support : ITOC project (CEA/LIX/LRI)
- Sparse matrix support (EDF R&D)
- Cell support : OMTE Digiteo project

Expected to start (soon)

- Unify optimization thanks to **polyhedral model**
- Export parts of NT2 internals as Boost library
- MPI support : PhD starting in 2010
- AVX prototype thanks to Intel simulator

Thanks for your attention