Parade MMT: debugging activity

F. Cinquini
f.cinquini@cineca.it

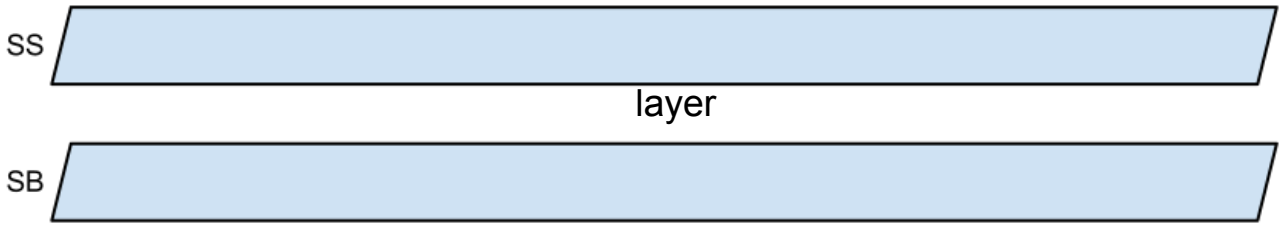
Outline

- Introduction
  - Context
- Getting started
  - Compiling the code
  - Input: Constant velocity
  - Input: Linear Velocity (map)
- Testing
  - 5 layer case: output comparison
  - Debugging: Totalview
  - Hypothesis: variable reference?
- Perspectives

Context

Parade:

- **Model Builder**
- Ray Tracer
- Output Reader



SS and SB are horizons, a layer has specific velocity attributes defined by a map or a constant value.

Tetrahedrons are build in the layer between horizons: a ray is traced from one horizon to another through the tetrahedrons.

The ray will be divided into segments with their own length and velocities depending from the input velocities.

A first bug in the model builder has been fixed and the code works for a single layer case.
(A wrong variable reference in the file mmsCurvelinearMeshedModelBuilder.cpp)

For n layer > 1, the results obtained with a constant velocity and a map of constant velocity are not consistent.

Compiling Parade

The code have been compiled on PLX machine.

Configure arguments:

./configure --with-mode=dbg --with-cxx=g++ --with-w64 --with-lib64 --with-stl --enable-shared --with-thread --with-ogl --with-x --with-python --with-pio --with-rt --with-spatialindex

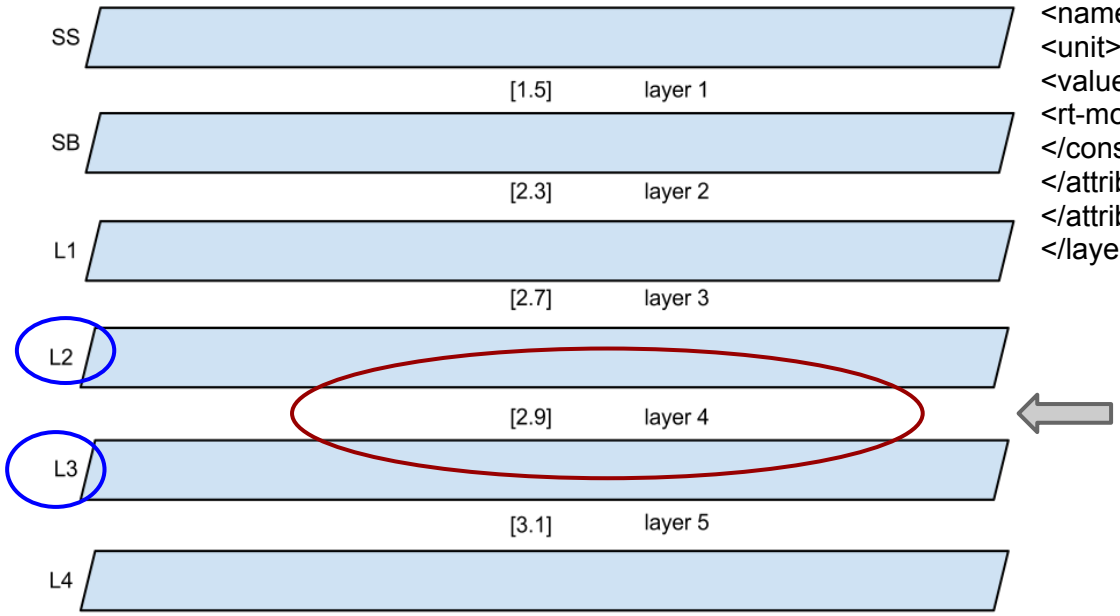
Remarks:

- OK with gnu C++ compiler v. 4.1.2 or older.
- Error with gnu C++ compiler v. 4.4.6 or newer: an include has to be modified (.
- To compile on eni0x and on PLX **"tk"** support should be disabled (tk libraries are too recent)

Input I

simple case - constant velocity:

- Define 6 horizons: *SS, SB, L1, L2, L3, L4*
- Define 5 Layers (regions between horizons)
- Each layer is defined by 2 horizons
- For each layer a **constant velocity** is given



Layer definition example:

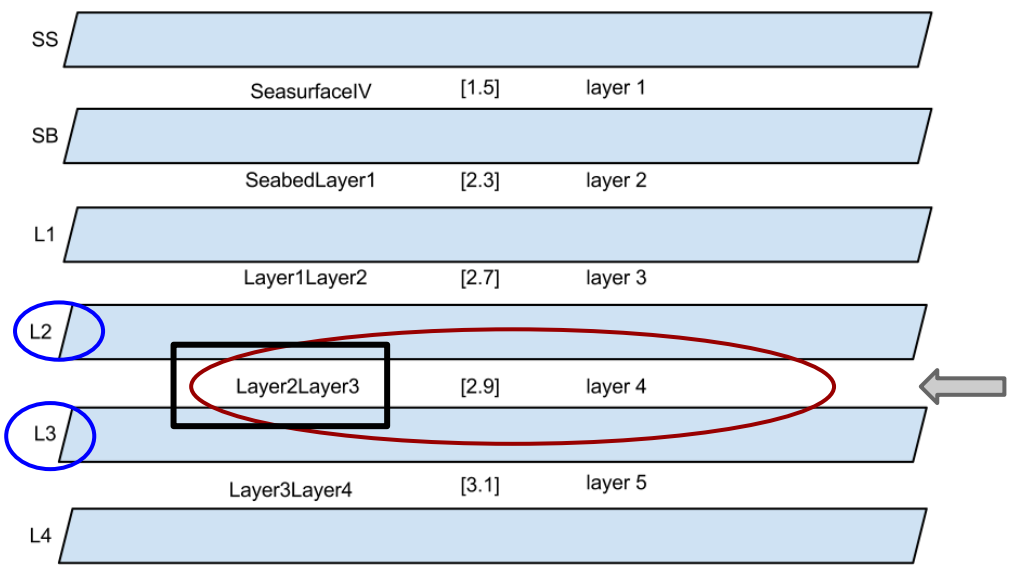
```
<layer>
<name>layer4</name>
<id>4</id>
<top-surface>L2</top-surface>
<base-surface>L3</base-surface>
<attributes>
<attribute>
<constant-attribute>
<name>P Velocity</name>
<unit>km/s</unit>
<value>2.9</value>
<rt-model-velocity-type>velocity</rt-model-velocity-type>
</constant-attribute>
</attribute>
</attributes>
</layer>
```

Constant velocity tracer
is employed by default

Input II

complex case - linear velocity:

- Define 6 horizons: *SS, SB, L1, L2, L3, L4*
- Define 5 Layers (regions between horizons)
- Each layer is defined by 2 horizons
- Define a **top surface velocity** for the top horizon of the layer and a **bottom surface velocity** for the bottom horizon of the layer



Layer definition example:

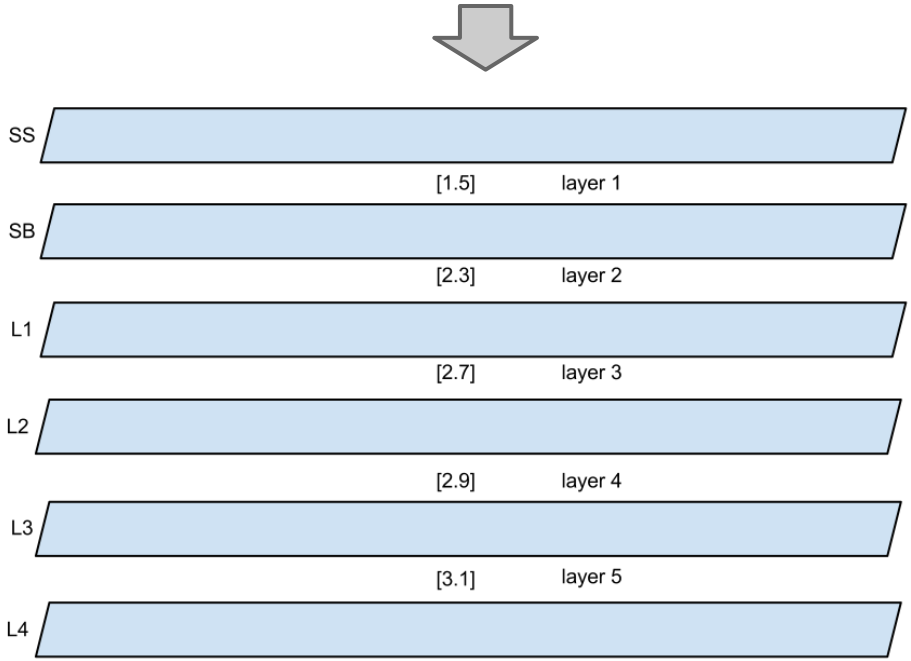
```
<layer>
<name>layer4</name>
<id>4</id>
<top-surface>L2</top-surface>
<base-surface>L3</base-surface>
<division>1</division>
<attributes>
<attribute>
<variable-attribute>
<name>P Velocity</name>
<unit>km/s</unit>
<top-surf-attr-file>layer2layer3lV.zyc</top-surf-attr-file>
<base-surf-attr-file>layer2layer3lV.zyc</base-surf-attr-file>
<rt-model-velocity-type>linear-velocity</rt-model-velocity-type>
</variable-attribute>
</attribute>
</attributes>
</layer>
```

Function: ComputeTetraVelFunction
(Recompute linear velocity function coefficients)

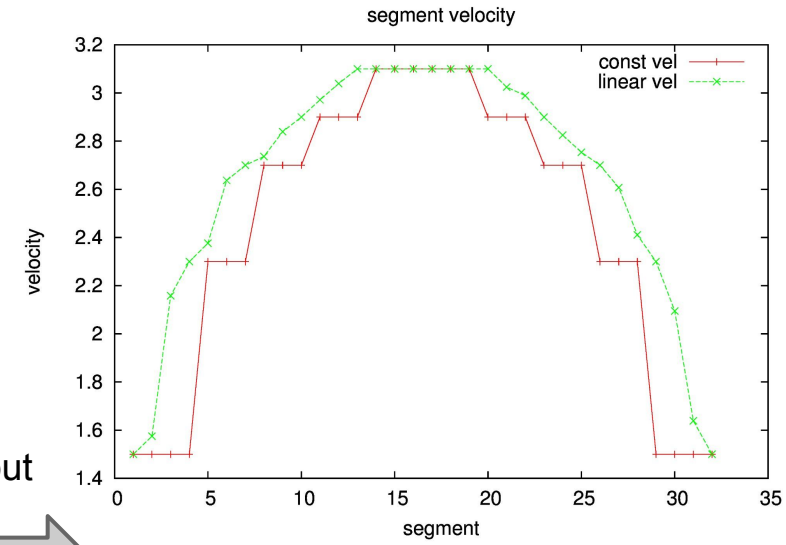
IF NO GRADIENT IS FOUND
FOR A TETRA ELEMENT
--> back to constant velocity tracer

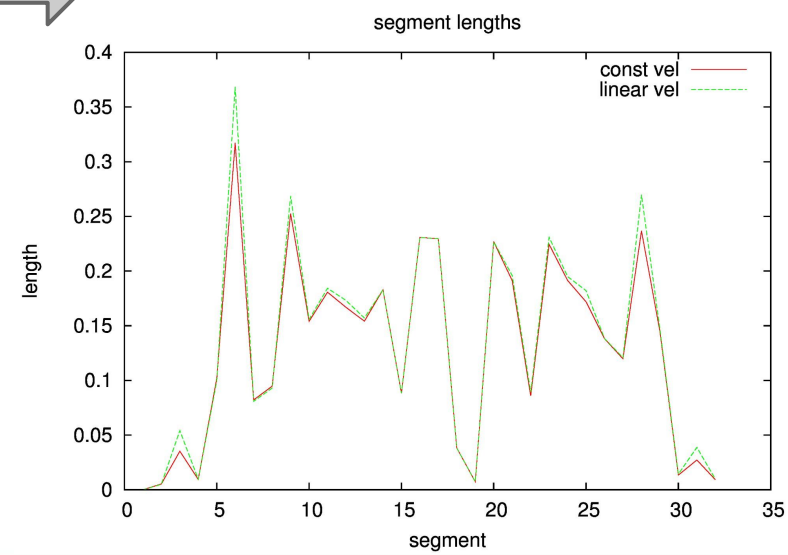
Test case: different output

Input



Output

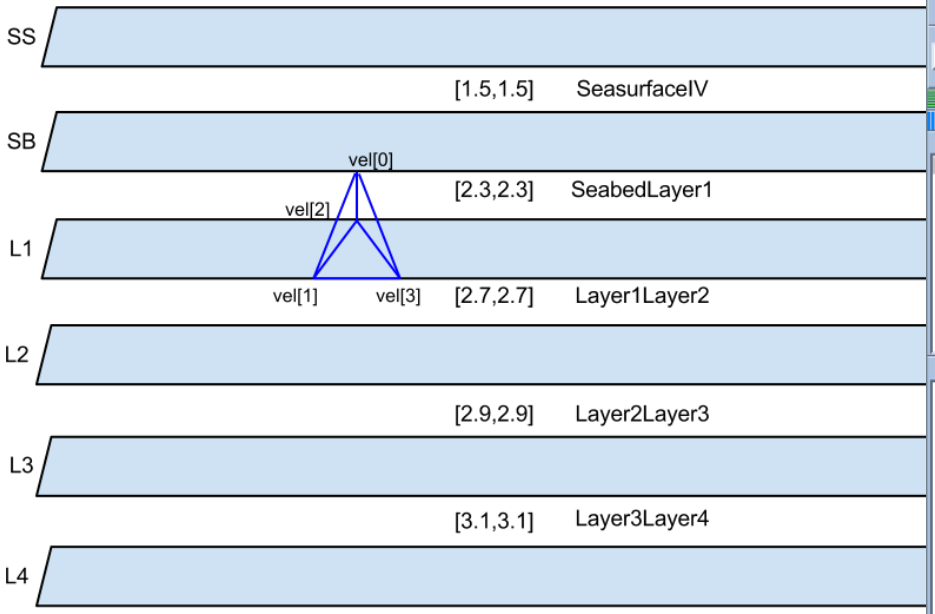


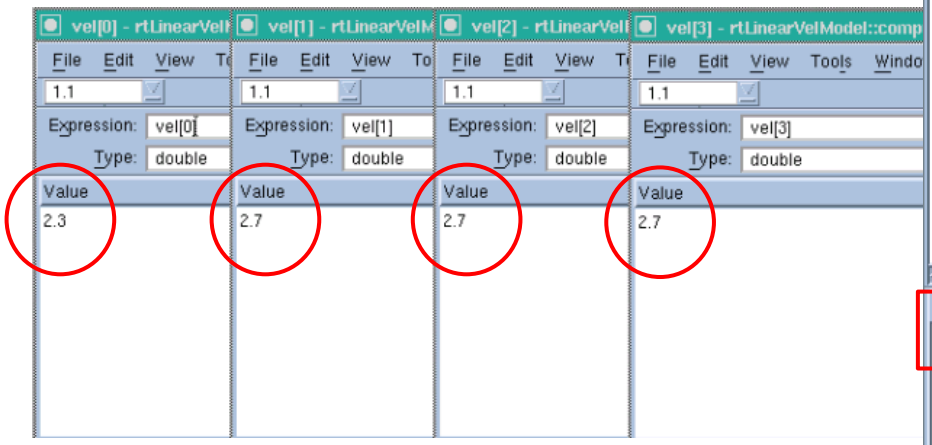


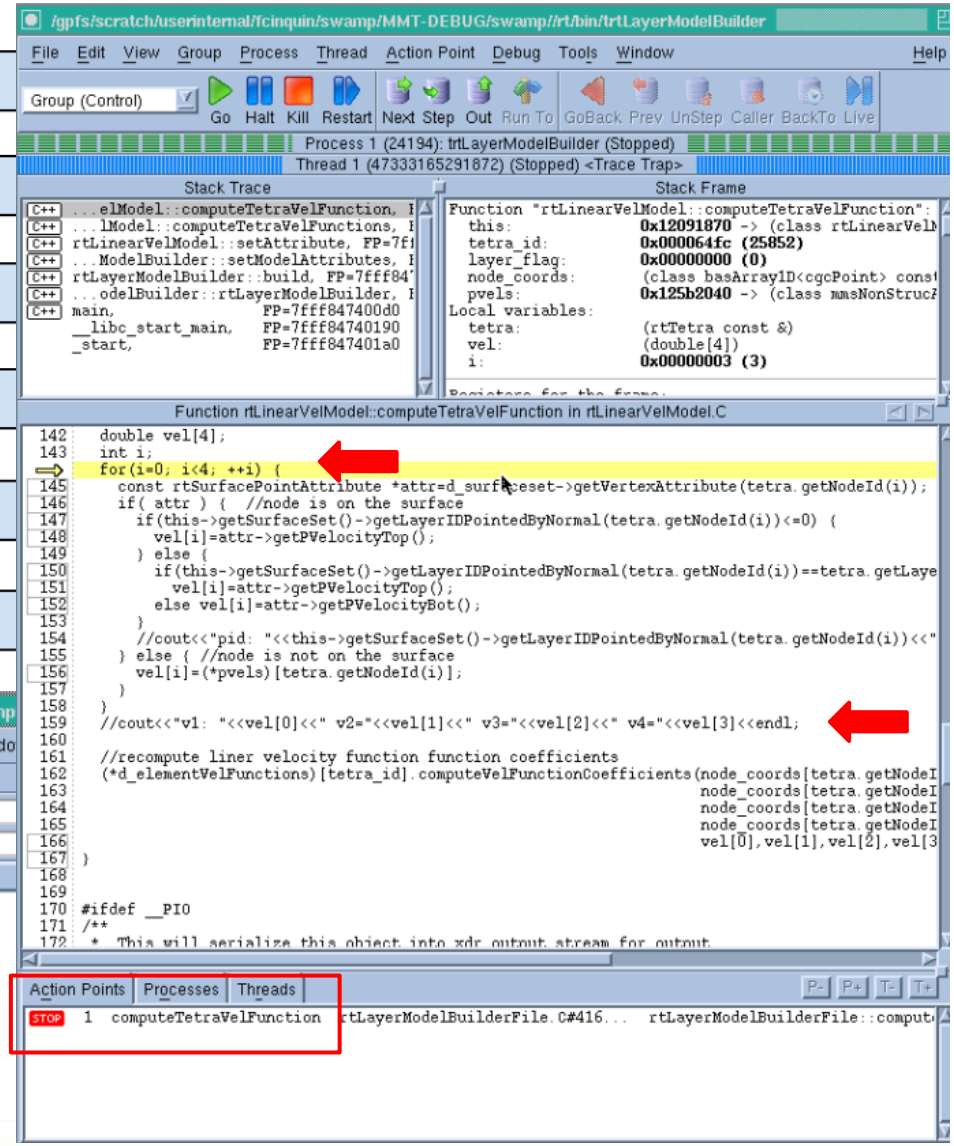
Linear velocities case:
Function: ComputeTetraVelFunction
(Recompute linear velocity function coefficients)

A GRADIENT IS FOUND FOR TETRA ELEMENTS

Debugging: Totalview

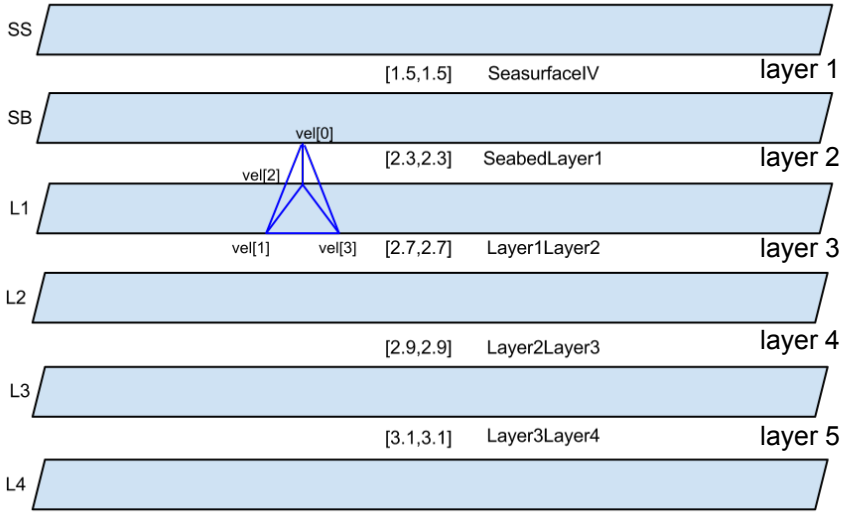






Variables storage

```
<layer>
<name>layer2</name>
<id>2</id>
<top-surface>SB</top-surface>
<base-surface>L1</base-surface>
<division>1</division>
<attributes>
<attribute>
<variable-attribute>
<name>P Velocity</name>
<unit>km/s</unit>
<top-surf-attr-file>Seabedlayer1lV.zyc</top-surf-attr-file>
<base-surf-attr-file>Seabedlayer1lV.zyc</base-surf-attr-file>
<rt-model-velocity-type>linear-velocity</rt-model-velocity-type>
</variable-attribute>
</attribute>
</attributes>
</layer>
<layer>
<name>layer3</name>
<id>3</id>
<top-surface>L1</top-surface>
<base-surface>L2</base-surface>
<division>1</division>
<attributes>
<attribute>
<variable-attribute>
<name>P Velocity</name>
<unit>km/s</unit>
<top-surf-attr-file>layer1layer2lV.zyc</top-surf-attr-file>
<base-surf-attr-file>layer1layer2lV.zyc</base-surf-attr-file>
<rt-model-velocity-type>linear-velocity</rt-model-velocity-type>
</variable-attribute>
</attribute>
</attributes>
</layer>
```



Horizon L1 keeps the last velocity defined (as top surface of layer 3).

The tetrahedron vertices assume different values along z --> a gradient is found

Output velocities are distributed as a gradient

Perspectives

- Hands on the code:
  - **check the** first bug-fix
  - **verify analogies** with other parts of the source code
    (same variable references to be modified?)

A step back: bug report

mmsCurvelinearMeshedLayerModelBuilder.C have been modified.

```
basArray1DNumeric& samples=attr->getAttributeSample(ix,iy,nz1);
            samples[1]=s1[2];

            samples=attr->getAttributeSample(ix,iy,nz2);
            mesh.getCoordinates(ix,iy,nz2,p);
            samples[0]=s2[2];
```

```
basArray1DNumeric& samples1=attr->getAttributeSample(ix,iy,nz1);
            samples1[1]=s1[2];

basArray1DNumeric& samples2=attr->getAttributeSample(ix,iy,nz2);
            samples2[0]=s2[2];
```

Is it possible that the same variable is defined somewhere else in the code?