Google Summer of Code 2014

Automatic Differentiation library using Cling

https://github.com/vgvassilev/clad

Mentors: Lorenzo Moneta
Vassil Vassilev

Student: Martin Vassilev

About me

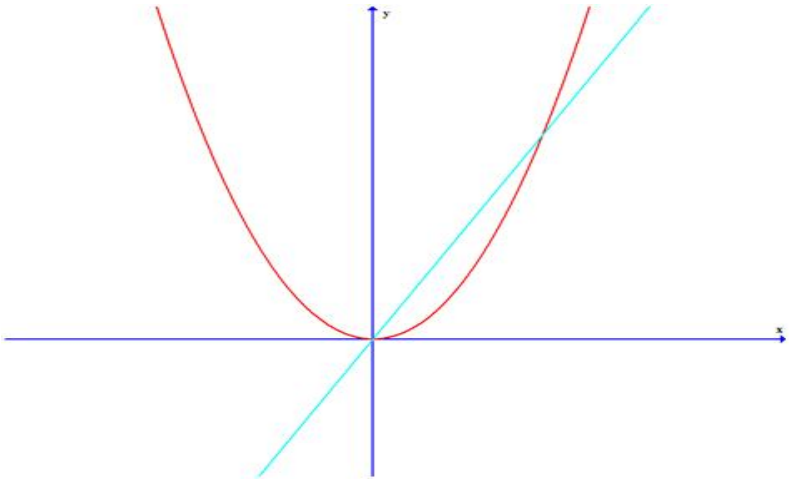
- Personal details
  - Software Technologies in University of Plovdiv

Open source software

- Participating in several open source projects
  - SolidReflector
    - Interactive .NET decompiler
  - SolidOpt
    - Framework for automatic optimizations
  - SolidV
    - Library for drawing 2D graphics

CLAD

- Motivation for participating
- What is the Clad project
  - Clang based c++ plugin
  - Provides a tool for applying automatic differentiation



Status of CLAD

- Deriving method functions from templated classes
- Add support for floating point literals
- Add tests and demos, code cleanup
- Improve the build system
- Implement cloners
- Derived functions are valid executable c++ code

Example

- Function calculating the normal vector of a point in a sphere

```
// Function descibing a sphere
float sphere_implicit_func(float x, float y, float z, float xc, float yc, float zc, float r) {
  return (x-xc)*(x-xc) + (y-yc)*(y-yc) + (z-zc)*(z-zc) - r*r;
}

int main() {
  // Differentiate implicit sphere function. Clad will produce the three partial derivatives.
  auto sphere_implicit_func_dx = clad::differentiate(sphere_implicit_func, 1);
  auto sphere_implicit_func_dy = clad::differentiate(sphere_implicit_func, 2);
  auto sphere_implicit_func_dz = clad::differentiate(sphere_implicit_func, 3);

  // Point P=(x,y,z) on surface
  float x = 5.0f;
  float y = 0;
  float z = 0;

  // Calculate normal vector in the point P
  float Nx = sphere_implicit_func_dx.execute(x, y, z, 0, 0, 0, 5.0f);
  float Ny = sphere_implicit_func_dy.execute(x, y, z, 0, 0, 0, 5.0f);
  float Nz = sphere_implicit_func_dz.execute(x, y, z, 0, 0, 0, 5.0f);

  printf("Result is N=(%f,%f,%f)\n", Nx, Ny, Nz);

  return 0;
}
```

Google Summer of Code

Thank you