

OPEN CASCADE Multiple Variable Function

eryar@163.com

Abstract. Multiple variable function with gradient and Hessian matrix is very very import in OPEN CASCADE optimization algorithms. In order to understand these optimization algorithm better, let's study some basic knowledge about Gradient, Hessian Matrix.

Key Words. Multiple Variable Function, Gradient, Hessian Matrix, 最优化算法,

1. Introduction

当函数只有一个自变量时, 感觉相对简单, 一切还在掌握之中。如参数 u 表示的 B 样条曲线, 当参数 u 在 $[0, 1]$ 之间变化时, 就可以得到与参数对应的曲线上的点。当函数有多个自变量时, 就有点力不从心, 毕竟人们都喜欢安稳, 不喜欢太多的变化。当变化太多时, 得到的结果也更丰富多彩。如有两个参数 u, v 表示的 B 样条曲面, 参数变化的范围是一个 $u \in [0, 1], v \in [0, 1]$ 矩形空间, 与参数 u, v 对应的是一片曲面上的点。

在实际应用中, 与多元函数的一阶导数 (Gradient 梯度)、二阶导数 (Hessian Matrix) 和多元函数的极值等概念也是理解非线性最优化问题的基础知识。

OPEN CASCADE 中一些求极值和逼近等算法中就使用了非线性最优化算法, 如下类图所示:

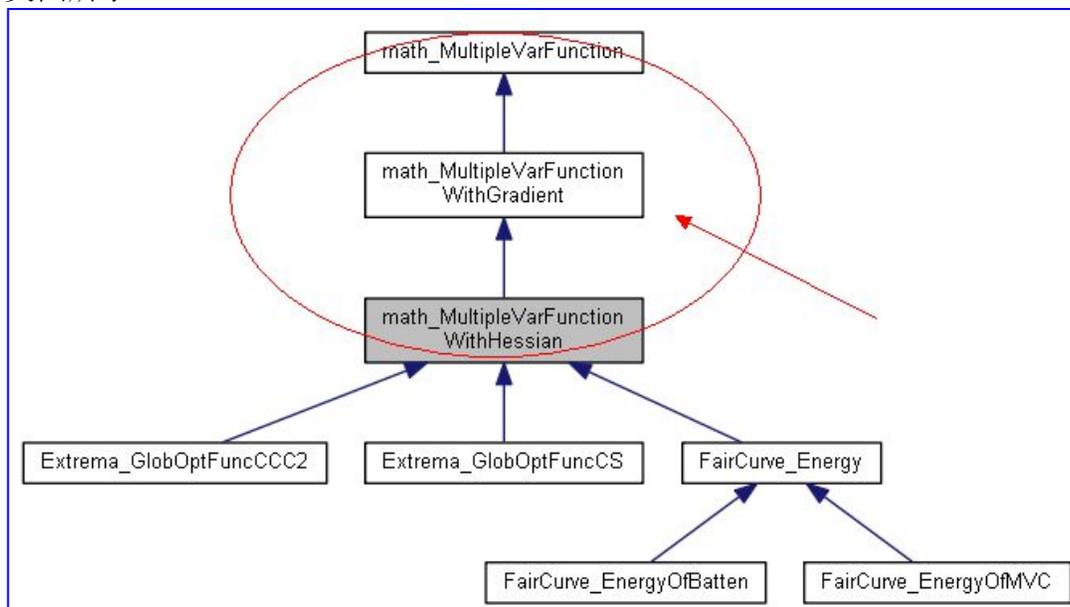


Figure 1.1 Multiple Variable Function in OPEN CASCADE

如上图所示, 具有二阶导数 (Hessian Matrix) 的多元函数应用到全局优化的求极值算法中。在用能量法实现的曲线光滑 (Fair Curve) 算法中也使用到了具有二阶导数的多元函数。为了更好地理解最优化的具体实现, 先学习下 OPEN CASCADE 中与多元函数有关概念的表达方式: 多元函数、梯度 Gradient 和 Hessian Matrix。

2. Multiple Variable Function

在《高等数学》中给出了多元函数的定义：设 D 是平面上一个点集。如果对于每个点 $P(x,y) \in D$ ，变量 z 按照一定法则总有确定的值和它对应，则称 z 是变量 x,y 的二元函数。点集 D 称为该函数的定义域， x,y 称为自变量， z 称为因变量。当自变量的个数大于 1 个时，即 $n \geq 2$ ， n 元函数统称为多元函数。

在 OPEN CASCADE 中与多元函数最直接的对应就是类 `Geom_Surface`，即参数表示的曲面，参数 u,v 的取值区域为多元函数的定义域，当指定 uv 时得到的曲面上的点即为多元函数值。OPEN CASCADE 的数学包中还给出了更抽象地多元函数类：`math_MultipleVarFunction`。

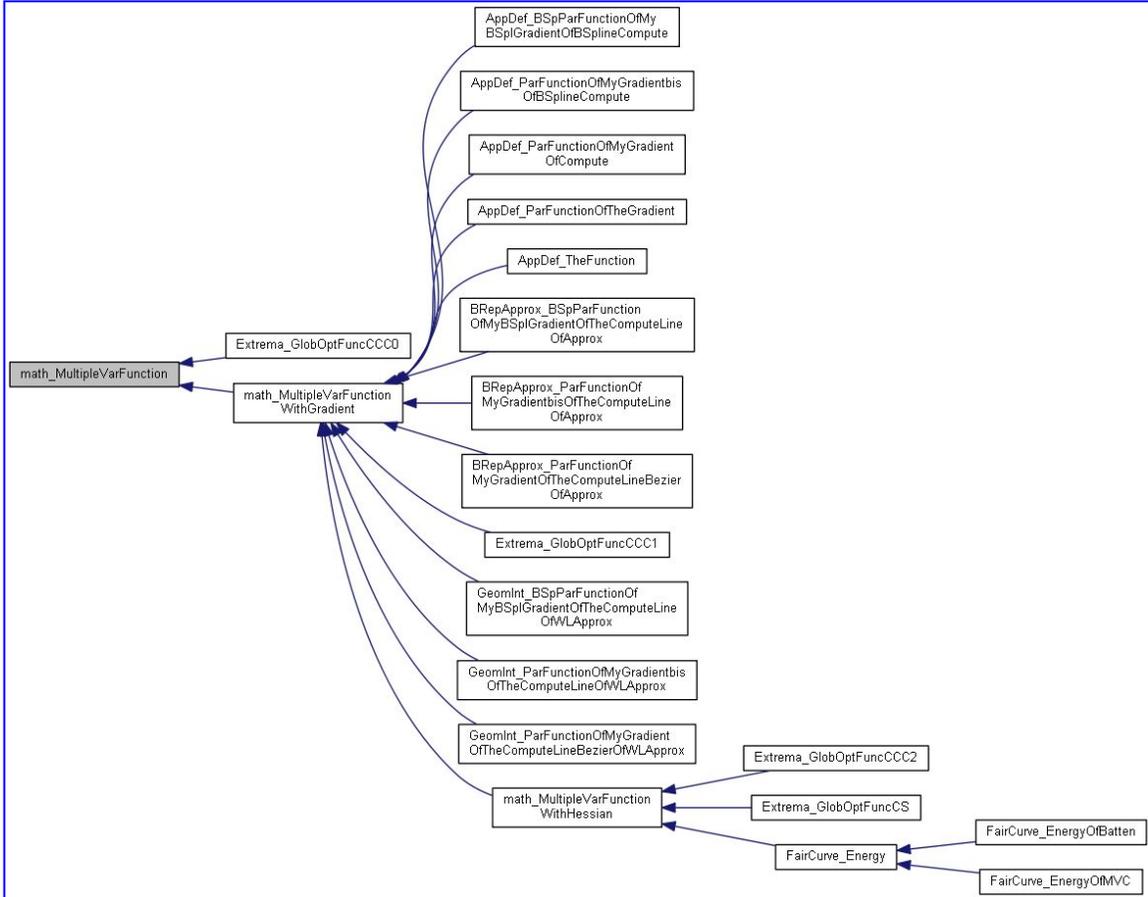


Figure 2.1 `math_MultipleVarFunction` class diagram

由上面的类图可知，类 `math_MultipleVarFunction` 是个抽象类，有两个纯虚函数：

- ❖ `NbVariables() const = 0`：自变量的个数；
- ❖ `Value(const math_Vector& X, Standard_Real& F) = 0`：计算指定自变量 X 对应的函数值 F 。自变量 X 是个向量，其中的值分别对应多个自变量；

下面给出一个多元函数的具体应用，即计算二重积分。题目来源为《高等数学》教材习题 9-2 (1)：

$$\iint_D (x^2 + y^2) d\sigma = \frac{8}{3} \quad D \in (|x| \leq 1, |y| \leq 1)$$

在 OPEN CASCADE 中计算上述二重积分代码如下所示:

```
/*
 *   Copyright (c) 2015 Shing Liu All Rights Reserved.
 *
 *       File : main.cpp
 *       Author : Shing Liu(eryar@163.com)
 *       Date : 2015-11-28 21:00
 *       Version : OpenCASCADE6.9.0
 *
 *   Description : Test Gauss Multiple integration.
 */

#define WNT
#include <math_MultipleVarFunction.hxx>
#include <math_GaussMultipleIntegration.hxx>

#pragma comment(lib, "TKernel.lib")
#pragma comment(lib, "TKMath.lib")

class math_TestFunction : public math_MultipleVarFunction
{
public:
    virtual Standard_Integer NbVariables() const
    {
        return 2;
    }

    virtual Standard_Boolean Value(const math_Vector& X, Standard_Real& F)
    {
        F = X(1) * X(1) + X(2) * X(2);

        return Standard_True;
    }
};

void testMultipleIntegration(void)
{
    math_Vector aLower(1, 2);
    math_Vector aUpper(1, 2);
    math_IntegerVector aOrder(1, 2, 10);
}
```

```

aLower(1) = -1.0;
aLower(2) = -1.0;

aUpper(1) = 1.0;
aUpper(2) = 1.0;

math_TestFunction aFunction;
math_GaussMultipleIntegration aIntegrator(aFunction, aLower, aUpper,
aOrder);

if (aIntegrator.IsDone())
{
    std::cout << aIntegrator;
}
}

int main(int argc, char* argv[])
{
    testMultipleIntegration();

    return 0;
}

```

通过从抽象类 `math_MultipleVarFunction` 派生出一个具体的多元函数类，在虚函数 `Value()` 计算指定向量 `X` 对应的函数值，通过函数 `NbVariables()` 确定自变量的个数。为什么可以对类 `math_Vector` 的对象用括号直接进行取值和赋值呢？因为在这个类中重载了括号运算符。计算结果如下图所示：

```

C:\Windows\system32\cmd.exe
math_GaussMultipleIntegration Status = Done
Integration value = 2.66667
Press any key to continue . . .

```

Figure 2.2 Integration Value

因为使用 `math_GaussMultipleIntegration` 时需要指定积分区域，且只能为常数，所以如下所示的积分区域中包含变量的多重积分就不能计算：

$$\int_0^2 dy \int_{y^2}^{2y} dx$$

OPEN CASCADE 中与多元函数对应的一元函数的积分计算参考：

OPEN CASCADE Gauss Integration:

<http://www.cppblog.com/eryar/archive/2014/09/11/208275.html>

3. Gradient

在二元函数情形，设计函数 $z=f(x,y)$ 在平面区域 D 内具有一阶连续偏导数，则对于每一点 $P(x,y) \in D$ 都可以定出一个向量：

$$\text{grad}f(x, y) = \frac{\partial f}{\partial x} i + \frac{\partial f}{\partial y} j$$

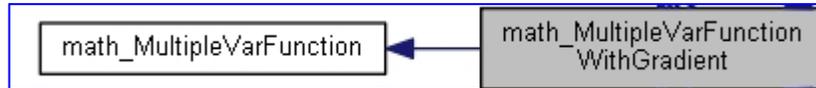
这向量称为函数 $z=f(x,y)$ 在点 $P(x,y)$ 的梯度(gradient)，记作 $\text{grad}f(x,y)$ 。对于多元函数 $u=f(X)$ ， $X=(x_1, x_2, \dots, x_n)^T$ ，有如下定义：

设 $u=f(X)$ ， $X \in S$ ，若在点 $x_0=(x_1(0), x_2(0), \dots, x_n(0))^T$ 处对于自变量 $X=(x_1, x_2, \dots, x_n)^T$ 的各个分量的偏导数 $\frac{\partial f(x_0)}{\partial x_i}$ ($i=1, 2, \dots, n$) 都存在，则称函数 $u=f(X)$ 在点 x_0 处一阶可导，并称向量 $\nabla f(x_0) = (\frac{\partial f(x_0)}{\partial x_1}, \frac{\partial f(x_0)}{\partial x_2}, \dots, \frac{\partial f(x_0)}{\partial x_n})^T$ 是 $u=f(X)$ 在点 x_0 处的梯度 Gradient 或一阶导数。

无约束优化最优性的一阶必要条件是：若 x_0 是无约束优化问题的局部最优值点，则 $\nabla f(x_0) = 0$ 。

由数学分析可知， $\nabla f(x)$ 的方向为 $f(X)$ 的等值面（等值线）的法线在点 x_0 处的方向，即沿梯度方向函数值变化最快。目标函数梯度为零的点称为无约束优化问题的稳定点。稳定点可能是目标函数的极大值点，也可能是极小值点，甚至二者都不是。最后一种情况对应的点称为函数的鞍点，即在从该点出发的一个方向上是函数的极大值，而在另一个方向上是极小值点。

对无约束优化问题，目标函数在最优值点的任意方向上的导数都为零，即目标函数在最优值点的切平面是水平的。不过无约束优化问题的局部最大值点和鞍点也满足上述条件。因此，要确认一个稳定点是否为最优值点，还需要考虑该点的二阶最优性条件。



在 OPEN CASCADE 中与此对应的类是 `math_MultipleVarFunctionWithGradient`，即具有梯度的多元函数。这个类也是抽象类，不能直接实例化，若要使用需要派生出新类，并实现以下几个纯虚函数：

- ❖ `NbVariables() const = 0`：多元函数自变量个数；
- ❖ `Value(const math_Vector& X, Standard_Real& F) = 0`：计算多元函数在指定变量 X 处对应的函数值 F ，通过引用传出；
- ❖ `Gradient(const math_Vector& X, math_Vector& G) = 0`：计算多元函数在指定变量 X 处的梯度值，通过引用传出；
- ❖ `Values(const math_Vector& X, Standard_Real& F, math_Vector& G) = 0`：计算多元函数在指定变量 X 处的函数值 F 和梯度值 G ；

4. Hessian Matrix

多元函数的二阶导数和 Hessian Matrix 的定义如下：设 $u=f(X)$, $x_0 \in S$, 若 f 在点 $x_0 \in S$ 处对于自变量 $x \in S$ 的各分量的二阶偏导数 $\frac{\partial^2 f(x_0)}{\partial x_i \partial x_j}$ ($i, j = 1, 2, \dots, n$) 都存在, 则称函数 $f(X)$ 在点 x_0 处二阶可导, 且称矩阵

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x_0)}{\partial x_1^2} & \frac{\partial^2 f(x_0)}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f(x_0)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x_0)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x_0)}{\partial x_2^2} & \dots & \frac{\partial^2 f(x_0)}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x_0)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x_0)}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f(x_0)}{\partial x_n^2} \end{bmatrix}$$

为 $f(X)$ 在点 x_0 处的二阶导数或 Hessian Matrix, Hessian Matrix 有时也记作 $H(x_0)$ 。

无约束最优化最优性条件的二阶必要条件为：设 $f(X)$ 在点 $x_0 \in S$ 处二次可微, 若 x_0 最 $f(X)$ 的局部极小点, 则 $\nabla f(x_0) = 0$ 且 $\nabla^2 f(x_0)$ 半正定。

无约束最优化最优性条件的二阶充分条件为：设 $f(X)$ 在点 $x_0 \in S$ 处二次可微, 若 $\nabla f(x_0) = 0$ 且 $\nabla^2 f(x_0)$ 正定, 则 x_0 是函数 $f(X)$ 的严格局部极小点。

二次型是 $X=(x_1, x_2, \dots, x_n)^T$ 的二次齐次函数, 它在研究非线性最优化问题中具有重要作用。通过线性代数中正定二次型相关定理, 可以得出 Hessian Matrix 正定的判断条件。相对于高等数学中关于二元函数极值充分条件的定理, 使用 Hessian Matrix 更具一般性, 适用于任意多元函数。终于知道正定二次型的一个应用, 将原来在线性代数课本中突然冒出这么个抽象概念, 能与实际应用有些关联。看来线性代数必需在实际应用中才能有所理解, 如关于现代控制工程中的微分方程组的求解, 会用到特征值的理论; 无约束最优化问题中极值充分条件的说明要用到正定二次型理论等。难怪当时在学习线性代数的时候那么吃力, 对这些抽象概念不理解, 学习的时候就只能是机械的记忆, 如果以后用不到, 肯定都还给老师了。OPEN CASCADE 中与 Hessian Matrix 对应的类是 `math_MultipleVarFunctionWithHessian`, 其类图如下所示:

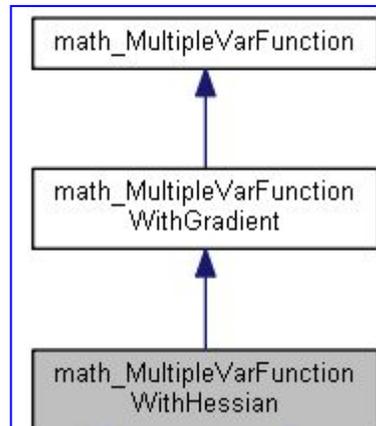


Figure 4.1 `math_MultipleVarFunctionWithHessian` class diagram

由上面类图可知，`math_MultipleVarFunctionWithHessian` 派生自带梯度的多元函数类 `math_MultipleVarFunctionWithGradient`。因此，比带梯度的多元函数类多个求值的虚线函数：

`Values(const math_Vector&X, Standard_Real&F, math_Vector&G, math_Matrix&H) = 0`

通过这个虚函数计算出多元函数在指定变量 X 处的函数 F，梯度值 G 和 Hessian Matrix H。

因为 `math_MultipleVarFunctionWithHessian` 有纯虚函数，所以也不能直接实例化，需要根据实际情况从其派生类来使用。正如本文开头所示的类图中几个类一样，从带有 Hessian Matrix 的多元函数派生出新类来计算全局极值和曲线光滑，如下类图所示：

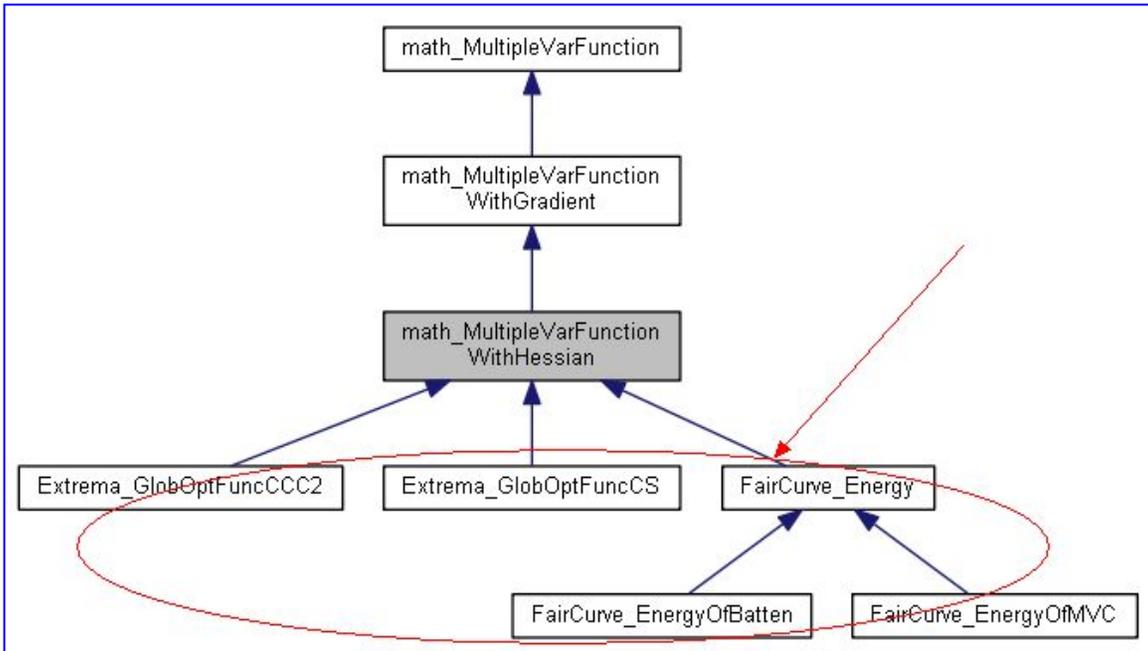


Figure 4.2 `math_MultipleVarFunctionWithHessian` class diagram

5. Conclusion

在 OPEN CASCADE 的数学包 Package math 中不仅描述了只有一个自变量的一元函数 math_Function, 也描述了具有多个自变量的多元函数, 及其一阶导数 (梯度) 和二阶导数 (Hessian Matrix)。通过应用多元函数的积分计算, 可以看出 OCC 对这些概念进行的类封装还是很清晰, 便于理解和使用。对于这些基础概念的介绍, 便于理解后面具体的优化算法的使用。

从 OPEN CASCADE 中的优化算法可以看出, 核心是数学这个精巧工具的熟练应用。可以找出《高等数学》、《线性代数》、《最优化方法》、《非线性最优化理论与方法》、《数值分析》等相关书籍, 通过对比 OPEN CASCADE 中的程序实现有效的理论联系实践, 提高学习效率。而对高等数学和线性代数可以做到温故知新, 对原来百思不解的问题可以有个 Aha! 的恍然大悟, 找到学习的乐趣。

6. References

1. 同济大学数学教研室. 高等数学. 高等教育出版社. 1996
2. 同济大学应用数学系. 线性代数. 高等教育出版社. 2003
3. 易大义, 陈道琦. 数值分析引论. 浙江大学出版社. 1998
4. 《运筹学》教材编写组. 运筹学. 清华大学出版社. 2012
5. 何坚勇. 最优化方法. 清华大学出版社. 2007
6. 杨庆之. 最优化方法. 科学出版社. 2015
7. 王宜举, 修乃华. 非线性最优化理论与方法. 科学出版社. 2012