

飞行器 MDO 中灵敏度计算的自动微分方法*

颜力, 陈小前, 王振国

(国防科技大学 航天与材料工程学院 湖南 长沙 410073)

摘要 :自动微分方法(ADM)是灵敏度计算的一种新方法,是处理飞行器多学科设计优化中灵敏度分析问题的有力工具。将 ADM 与估算灵敏度最常用的有限差分方法(FDM)以及与 ADM 同时期发展起来的另外一种灵敏度计算方法—复变量方法(CVM)从原理上进行了比较,研究了 ADM 前向模式在 Visual C++ 6.0 环境中的实现方法,结合多学科环境中的飞行器设计优化计算实例分析了该模式的优缺点。ADM 在科学计算、工程计算等方面有很大的发展空间。

关键词 :自动微分方法;灵敏度分析;多学科设计优化

中图分类号 :V421.1 **文献标识码** :A

Study of the Automatic Differentiation Method for Calculating Sensitivities in the MDO of Flying Vehicles

YAN Li, CHEN Xiao-qian, WANG Zhen-guo

(College of Aerospace and Material Engineering, National Univ. of Defense Technology, Changsha 410073, China)

Abstract :The automatic differentiation method(ADM) for calculating sensitivities is presented. It is a powerful method for dealing with the system sensitivity analysis of the multidisciplinary design optimization. The theory of this method, a general procedure for the implementation of this method, and the realization in Visual C++ 6.0 of this method were described in detail. Also there is a comparison between the finite differences method(FDM) and complex variable method(CVM) in this paper. The ADM is shown to have some advantages over the others, thus having the potential to be widely used in the future.

Key words :automatic differentiation method; sensitivity analysis; multidisciplinary design optimization

随着飞行器系统复杂性的日益增加,人们对飞行器的多学科设计优化(multidisciplinary design optimization, MDO)给予了更多地关注。1991年,美国航空航天学会(AIAA)成立了专门的 MDO 技术委员会,并发表了“关于 MDO 发展现状的白皮书”,认为灵敏度分析技术在工程设计的优化问题中处于核心地位。1996年,由 Sobieski 和 Haftka 两位 MDO 研究权威撰写的“航空航天领域中的 MDO 研究综述”一文,将灵敏度分析技术作为 MDO 的关键技术之一加以研究,指出灵敏度分析技术与分解技术、近似技术等相结合,将有利于解决复杂系统的 MDO 问题中面临的计算复杂性和组织复杂性问题^[1]。文献[2]对灵敏度分析技术进行了系统研究,对学科灵敏度和系统灵敏度导数计算方法进行了分类研究。

各种学科灵敏度导数计算方法本质上可分为两类^[2]:直接法和伴随法。有限差分方法(FDM)、复变量方法(CVM)和自动微分方法(ADM)等属直接法范畴。ADM 针对程序代码解析地求解函数的导数,其突出的一个优点是无截断误差,具有机器在有效位数字上所能表示的最小精度。众多 ADM 研究小组开发了大量的自动微分工具,并应用这些工具将自动微分技术应用在数值方法、灵敏度分析、设计优化以及数据相关等研究领域。

1 FDM 与 CVM 的基本原理

用于估算学科灵敏度的最常用的近似方法是 FDM,其一阶和二阶计算公式分别为

* 收稿日期 2005 - 12 - 07

基金项目 国家 863 高技术研究发展计划基金资助项目(2004AA721051) 国家自然科学基金资助项目(10302031)

作者简介 颜力(1976—),女,博士生。

$$\frac{dF}{dx} = \frac{F(x+h) - F(x)}{h} + \alpha(h) \quad (1)$$

$$\frac{dF}{dx} = \frac{F(x+h) - F(x-h)}{2h} + \alpha(h^2) \quad (2)$$

其中, F 为所求的光滑实值函数; x 为自变量; h 为实数步长, 是一小扰动参数; $\alpha(h)$ 为式(1)的截断误差, $\alpha(h^2)$ 为式(2)的截断误差。一阶计算公式的精度不如二阶计算公式高, 故在实际应用中使用二阶计算式的情况较多。但是无论应用哪一种计算公式, 为了减小误差, 都希望步长越小越好, 这样就不可避免地会因为两个函数之间的减法运算而产生舍入误差。在具体的工程应用中, 步长究竟取多少还需不停地反复试算, 这一点无论是在时间和精度上都令人无法忍受。

采用与 FDM 相同的思路推导出的另外一种学科灵敏度计算方法 CVM, 其灵敏度计算公式

$$\frac{dF}{dx} = \frac{\text{Im}[F(x+ih)]}{h} + \alpha(h^2) \quad (3)$$

$$\frac{d^2F}{dx^2} = \frac{2\{F(x) - \text{Re}[F(x+ih)]\}}{h^2} + \alpha(h^4) \quad (4)$$

其中, i 为虚数单位。由计算公式可知, 与 FDM 不同, 用 CVM 求函数的一阶导数时无需进行函数的相减操作, 避免了 FDM 中由于步长选择过小而带来的舍入误差, h 的取值不受限制, 可选择尽可能小的值。但用 CVM 计算函数的二阶导数(如实际应用中的海赛矩阵的计算)时会遇到步长选择的问题, 且 CVM 由于需处理复数, 占用较多的计算机内存, 耗时, 计算量大^[2]。

2 ADM 的基本原理

自动微分技术的基本思想是^[3], 对于任一光滑函数 F , 在计算机程序运行过程中, 无论函数的计算有多复杂, 都可分解为一系列的初等计算(如加、减、乘、除)和初等函数(如三角函数、对数函数、指数函数等)运算的有序复合, 这种有序复合可由一组代码表(code list, 类似于计算机代码段, 用于估计一个表达式)表示。通过对这些初等函数迭代地运用链式规则, 计算机就可以应用 ADM 自动精确地得到函数 F 的任意阶导数。将如式(5)所示的链式规则反复应用于已分解的初等运算上,

$$\frac{dF(g(x), h(x))}{dx} = \frac{\partial F(s, r)}{\partial s} \times \frac{dg(x)}{dx} + \frac{\partial F(s, r)}{\partial r} \times \frac{dh(x)}{dx} \quad (5)$$

就可以得到函数 F 的导数信息。有学者已经证明^[4], 对于标量函数一阶导数的计算复杂性, ADM 不超过所求函数计算量的 5 倍, 与函数的变量个数 n 无关, 而 FDM 则是 $(n+1)$ 倍。

自动微分技术可以直接用于任意长度的计算机程序, 可以包含分支、循环和子程序等程序结构, 针对程序模块求取解析导数。自动微分技术对任意子程序的求导算法为^[2](1)将该子程序分解为一系列的初等函数(2)对初等函数求导(3)将前一步中所求的初等偏导数累加起来。

这三步可以同时进行。在程序实现过程中, 对函数进行算法定义的最简单的形式是代码表。由于分解出的初等函数的种类是有限的, 所以第(2)步的实现代码是固定的。第(1)步可以有多种实现方法, 目前主要有两种实现方法: 源代码转换方法(STM)和操作符重载方法(OOM)。第(3)步的累加方法也有两种基本模式: 前向模式和反向模式。这两种模式的区别在于怎样运用链式规则通过计算传递导数: 前向模式传递中间变量关于独立变量的导数; 反向模式则传递最终结果关于中间变量的导数。使用反向模式时, 为了传递反向, 需引入中间变量, 逆转程序流。

ADM 的一些基本概念可通过一个计算图来阐述。每个函数的计算图代表了一组函数的计算, 对应于可编程的不同的代码表。下面结合一个实例对函数计算图和 ADM 的两种模式进行概念性介绍。待计算的函数具体表达式如下:

$$F(x_1, x_2) = \sin x_1 + \ln(x_1, x_2) \quad (6)$$

式(6)的一种代码表定义为^[5]

$$\begin{cases} t_1 = x_1 & t_4 = t_1 t_2 \\ t_2 = x_2 & t_5 = \ln t_4 \\ t_3 = \sin t_1 & t_6 = t_3 + t_5 = F \end{cases} \quad (7)$$

其计算图如图 1 所示。将自动微分规则应用于式(7)所示的代码表,对函数(6)求导,可分别得到其前向模式和反向模式的计算式,如表 1 所示,表中箭头表示两种模式的计算方向。

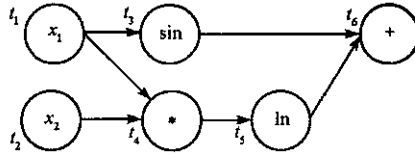


图 1 式(6)的计算图

Fig.1 Computational graph of expression (6)

表 1 函数(6)的前向模式与反向模式导数计算

Tab.1 Differentiation calculation of forward and reverse mode of expression (6)

代码表	前向模式	反向模式
$t_1 = x_1$	$\dot{t}_1 = \dot{x}_1$	$\bar{t}_1 = \cos t_1 \times \bar{t}_3 + t_2 \times \bar{t}_4$
$t_2 = x_2$	$\dot{t}_2 = \dot{x}_2$	$\bar{t}_2 = t_1 \times \bar{t}_4$
$t_3 = \sin t_1$	$\dot{t}_3 = \cos t_1 \times \dot{t}_1$	$\bar{t}_3 = \bar{t}_6$
$t_4 = t_1 \times t_2$	$\dot{t}_4 = \dot{t}_1 \times t_2 + t_1 \times \dot{t}_2$	$\bar{t}_4 = \bar{t}_5 / t_4$
$t_5 = \ln t_4$	$\dot{t}_5 = \dot{t}_4 / t_4$	$\bar{t}_5 = \bar{t}_6$
$t_6 = t_3 + t_5$	$\dot{t}_6 = \dot{t}_3 + \dot{t}_5$	$\bar{t}_6 = 1$

3 ADM 的实现

采用 OOM 来实现 ADM 的前向模式,为了将 ADM 前向模式与 FDM 和 CVM 在计算导数方面进行比较,在现有的 FDM 代码基础上对程序进行改造,定义专门用于 ADM 前向模式计算的数据类型 fadm(包括数值、一阶灵敏度导数和二阶灵敏度导数三部分),按如下三步在计算机程序中实现 ADM 前向模式:(1)将所有 double 类型的变量声明替换为 fadm 类型的变量声明(2)为 fadm 变量定义所没有的关系运算符、函数和算术运算符(3)分别按前向模式和反向模式的计算式来编写计算导数的程序。

面向对象编程语言 C++ 具有封装性、继承性和多态性等特性,为自动微分的软件实现提供了较 Fortran 语言更为方便、快捷的途径。利用高级编程语言 C++ 的面向对象特性,借鉴文献[5]的思路,在 Visual C++ 环境中可编写出 ADM 前向模式的实现程序。表 2 给出了 C++ 中用于自动微分数据类型 +、*、sin 计算的函数重载和运算符重载的定义,表中 z、x、y、t 是 fadm 类型的变量,z.v 是 z 变量的数值部分,z.d 是 z 变量的一阶灵敏度导数部分,z.dd 是 z 变量的二阶灵敏度导数部分,其它类似。

表 2 Visual C++ 中用于 ADM 前向模式的函数与运算符重载(+、*、sin)

Tab.2 The OOM(+、*、sin) used in forward mode in Visual C++

重载的函数	一阶灵敏度计算定义	二阶灵敏度计算定义
$t = x + y$	$t.d = x.d + y.d$	$t.dd = x.dd + y.dd$
$t = x * y$	$t.d = x.d * y.v + x.v * y.d$	$t.dd = x.dd * y.v + x.d * y.d + x.d * y.d + x.v * y.dd$
$t = \sin z$	$t.d = z.d * \cos z$	$t.dd = -z.d * t.v * z.d + z.dd * \cos z$

4 实例计算

在 Visual C++ 环境下定义了用 ADM 前向模式计算灵敏度导数的数据类型“fadm”,并将其放在“FADM.h”中。fadm 类的定义包括了数据的赋值、读取、不同数据类型之间的转换及计算以及表 2 所示的函数重载等内容。

以某无控飞行器多学科设计优化为例来介绍 ADM 在学科灵敏度计算中的实际应用。这是一个包括了质量、气动、结构与弹道等多个学科的复杂问题,其多学科设计的模块包括质量计算、气动力计算、

结构分析和弹道计算等模块。在此基础上,可进行飞行器的总体优化。其中,在气动学科的计算中采用 FDM 来计算其升力系数、阻力系数等参数相对于马赫数、攻角等参数的一阶灵敏度导数。在 Visual C++ 环境下仅需对源程序所有 double 类型的变量声明替换为 fadm 类型的变量声明和在主程序中包含“FADM.h”文件,就可应用 ADM 前向模式来计算优化过程所需的灵敏度导数。图 2 是在马赫数 $Ma = 0.7$ 、攻角 $\alpha = 4.0^\circ$ 时,分别用 ADM 前向模式、FDM 和 CVM 三种方法计算得出无控飞行器的升力系数相对于马赫数的灵敏度导数($\partial C_L / \partial Ma$)的相对误差与计算步长的关系图。

由图 2 可知,三种方法初始时均收敛。当步长小于 10^{-7} 时,FDM(点划线)会产生严重的舍入误差而导致所求的一阶灵敏度导数近似值不可靠。而当步长小到式(2)分母的减数与被减数趋于相同时(图中小于 10^{-15}),用 FDM 计算的一阶灵敏度导数近似值将趋近于 0 相对误差趋近于 1。与之相反,用 CVM(虚线)计算的一阶灵敏度导数值,由于无需用到差分运算,不存在舍入误差问题且计算结果不受步长过小的影响,当步长小于 10^{-7} 时,所求得的一阶灵敏度导数值能维持相当高的精度。而用 ADM 前向模式(实线)计算的一阶灵敏度导数既没有用到差分运算,又没有用到步长,计算精度仅受限于机器精度,故在任意时刻,所求得的一阶灵敏度导数值都能有相当高的精度。另一方面,在时间消耗上,ADM 前向模式介于 FDM 和 CVM 之间。

5 结束语

对于任一光滑函数,在选择其灵敏度导数计算方法时,最重要的几个指标通常是计算时间、计算结果的正确性、数值精度以及开发成本。FDM 是最常用的灵敏度计算方法,最突出的优点是易于计算机实现,开发成本较低,但求解函数梯度时相对于计算函数本身具有较高的计算复杂性,数值精度较低,灵敏度导数的计算结果与步长选取相关,而步长的确定往往是实际应用中的一个难题。CVM 是较新的一种灵敏度分析方法,在计算一阶灵敏度导数时可选用尽可能小的步长,在大多数实际应用中较 FDM 方便,数值精度较高,但 CVM 存在占用较多的计算机内存、计算量大、耗时等不足,且计算二阶灵敏度导数时也会遇到步长选择的问题。ADM 也是计算机数值计算与分析领域内的一种新方法。与 FDM 和 CVM 相比,ADM 由于对任何光滑函数,在数值求解的同时可将其任意阶导数“自动”求出,其计算精度仅受限于机器精度^[7],为线性、非线性规划中依赖导数的优化方法提供了有效的求导工具。对中大规模的优化问题来说,目标函数越复杂,越能体现其优越性。目前,ADM 已在并行计算、系统参数辨识、多学科设计优化、电子科学、化工生产、天气预报、海洋环流等实际工程中得到了成功应用^[8]。

参考文献:

- [1] Sobieszczanski-Sobieski J, Haftka R T. Multidisciplinary Aerospace Design Optimization, Survey of Recent Developments[R]. AIAA Paper 96-0711, 1996.
- [2] 颜力,陈小前,王振国. 飞行器多学科设计优化中的灵敏度分析方法研究[J]. 航空计算技术, 2005(1): 1-6.
- [3] Barthelemy J F, Hall L E. Automatic Differentiation as a Tool in Engineering Design[J]. Structural Optimization, 1995, 9: 76-82.
- [4] Griewank A. On Automatic Differentiation[A]. Mathematical Programming: Recent Developments and Application. Kluwer Academic Publishers Dordrecht, 1989: 83-108.
- [5] Rall L B. Automatic Differentiation: Techniques and Applications[M]. Springer-Verlag, Berlin Heidelberg New York, 1981.
- [6] Walther A. Basics of Automatic Differentiation[R]. AD Short Course, SIAM Optimization Conference 19, May, 2002.
- [7] 程彬杰,王莉萍,邵志标,等. 数据拟合中自动微分技术的应用[J]. 西安交通大学学报, 1999, 33(7): 19-33.
- [8] Griewank A, Corliss G F. Automatic Differentiation of Algorithms: Theory, Implementation, and Application[M]. SIAM, Philadelphia, 1991.

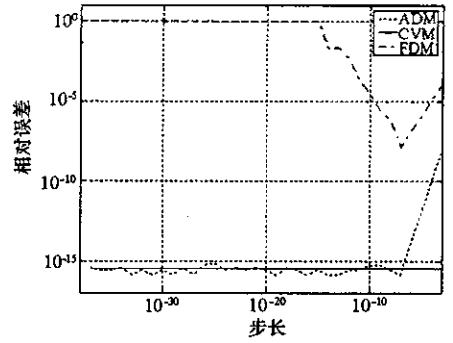


图 2 三种灵敏度计算方法相对误差比较

Fig.2 Contrast of relative error of three sensitivity calculation methods

