

*论文题目: Software Numerical Instability Detection and Diagnosis by Combining Stochastic and Infinite-precision Testing (随机方法与高精度计算相结合的软件数值错误自动检测与自动诊断)

*作者: 汤恩义, 张翔宇, Norbert Th. Müller, 陈振宇, 李宣东

*单位: 南京大学软件新技术国家重点实验室 汤恩义助理研究员、陈振宇教授、李宣东教授, 美国普渡大学计算机科学系张翔宇教授, 德国特里尔大学计算机科学系 Norbert Th. Müller 教授

联系方式: eytang@foxmail.com

*文章发表信息: (论文集, 请提供论文集名称, 出版社, 出版时间; 杂志, 请提供杂志名称, 年, 卷, 期, 页码) IEEE Transactions on Software Engineering 2017 已录用待发表

*原文链接地址: <http://ieeexplore.ieee.org/document/7792694>

*正文:

由数值稳定性问题而引发的软件数值错误是高可信软件关注的重要缺陷之一。随着软件规模的增加, 依靠人工来排查与修复软件的数值错误已经难以满足实际需求。基于这一现状, 我们着重研究了软件数值错误的自动检测与自动诊断方法。由于该问题近年来引起了软件工程领域的重视, 因此, 相关工作中已经有一些数值稳定性错误的自动检测方法被陆续提出, 然而, 业界对软件中数值稳定性错误的来源及自动诊断方法的认识在本文发表前仍然较为模糊。

我们按照软件开发过程, 将软件中数值稳定性错误的来源归为两类: 1) 来源于软件需求, 由问题背景引起的数值稳定性错误。该类错误在软件需求不变, 数值类型字长不变的情况下不可修复。2) 来源于软件的设计与实现。该类错误由软件本身的实践而引发, 由于理论上会存在更稳定的数值算法, 该类错误可以通过改进软件的实现来直接修复。图 1 显示了这两种不同类型的数值稳定性错误, 由于修复这两类数值稳定性错误的方式截然不同, 因此我们不仅需要检测数值软件中的数值稳定性错误, 且需要对不同类型的稳定性错误进行自动诊断, 区分不同的错误来源, 以便对数值稳定性错误进行进一步修复。

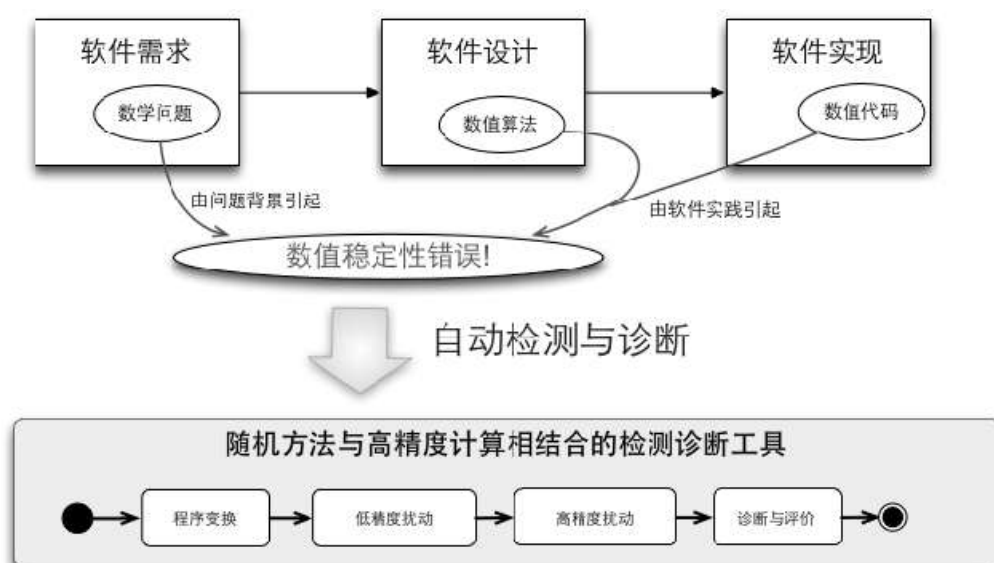


图 1 数值稳定性错误的来源、检测与诊断

我们设计与开发了一组随机方法与高精度计算相结合的检测诊断工具来自动分析与诊断软件中的数值稳定性错误。它首先通过程序变换技术将待测软件中的数值模块自动转为高精度数值计算。原始待测软件的数值模块一般是按照 IEEE-754 标准来构建的, 由于精度限制, 其数值表示与计算一般会引入误差, 在我们的检测诊断框架中我们称之为低精度(或者普通精度)计算。通过计算的代换, 我们将原始待测软件转换成为能获得精确结果的高精度数值计算。它的基本思想是将软件中每一个浮点数值在运行时动态改变占用的内存大小, 每当其精度不足时, 高精度计算会自动增加对应数值的内存占用, 从而保证软件输出精确结果。即当用户需要有限精度的软件输出时, 高精度计算会保证输出中的每一位有效数字都是准确的, 但这是以消耗数百倍, 甚至数千倍系统计算资源为代价的。

经过第一步的程序变换, 我们获得了和原始普通精度软件相对应的高精度程序。对于原始软件, 我们引入了低精度扰动技术来模拟程序的计算误差, 并分析误差对软件输出结果的影响。该扰动技术在测试过程中随机修改浮点数值有效数字末尾的比特串, 并观察软件输出的波动状况。如果软件输出结果的波动很大, 则意味着软件中各变量的数值误差对其输出结果的影响很大, 即数值软件在稳定性上存在不足。

我们进一步对高精度程序进行扰动。由于高精度程序会输出准确结果, 如果高精度扰动的输出波动仍然很大, 则意味着系统需求所隐含的数学性质导致了软件的不稳定性, 即软件中的数值稳定性错误来源于软件需求。在这种情况下, 软件设计开发人员应当积极同软件客户讨论软件需求, 以避免特殊性质的数学问题。当软件需求不可改变时, 软件的设计与开发人员应当对该软件采用高精度计算等措施, 通过增加计算资源来保障软件的正确性。如果高精度扰动的输出结果波动较小, 但低精度扰动的输出结果的波动很大, 则意味着原始待测软件存在数值稳定性问题, 且该问题是由软件的设计开发实践引入的。在这种情况下, 软件设计开发人员应当积极寻求完成该软件的其它稳定算法, 即寻找数学上等价但浮点数计算上不完全相等的算法, 来改进软件的数值稳定性。

在我们工具的最后一步, 我们通过自动获得数值计算的条件数来度量低精度扰动与高精度扰动的波动状况, 并进一步由比对普通精度程序与高精度程序的执行路径来定位发生稳定性问题的实现模块, 从而为软件的设计开发人员提供准确的检测诊断信息与有效的修复提示。

作者简介: 汤恩义, 1982 年生, 于 2013 年起在南京大学软件学院任助理研究员, 主要研究方向为数值分析与保障技术、实时软件时延分析、新型软件测试与程序分析技术、软件工程。

说明: *为必填项。