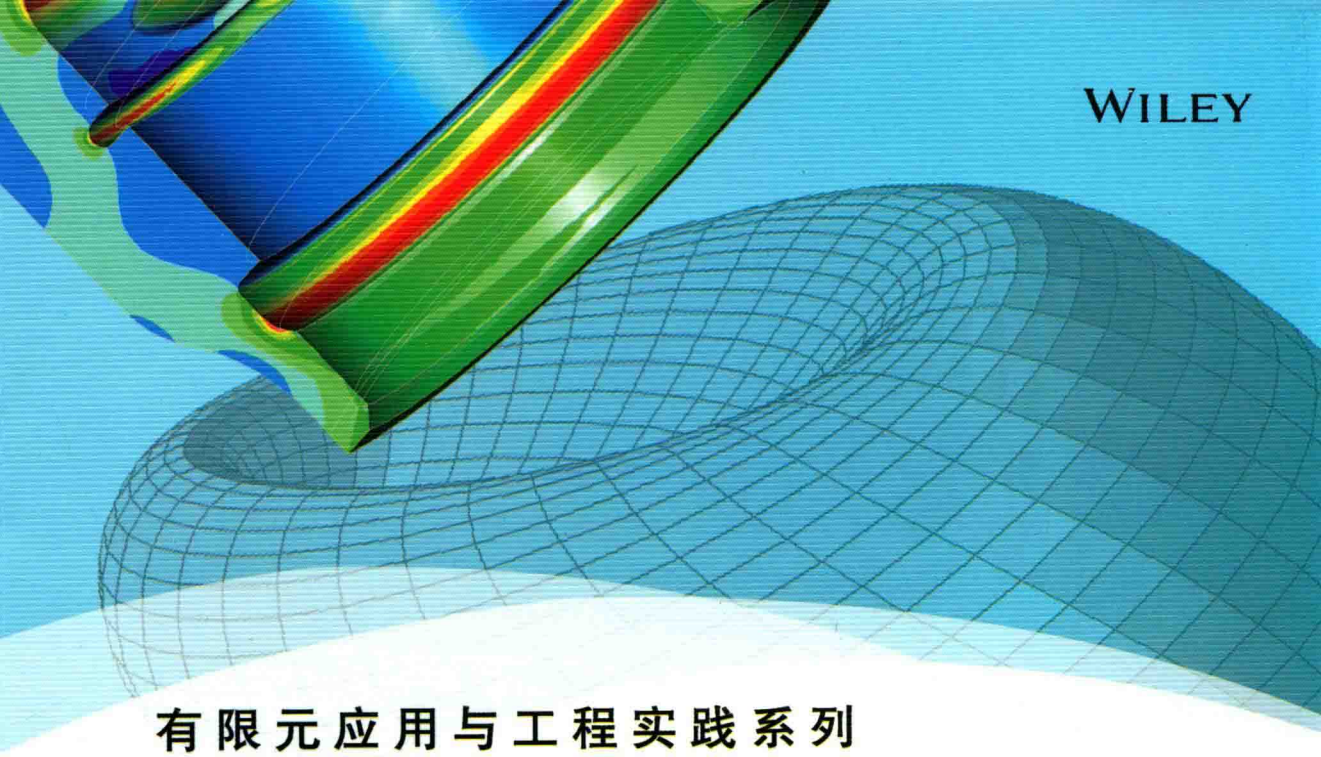


WILEY

有限元应用与工程实践系列

有限元方法编程

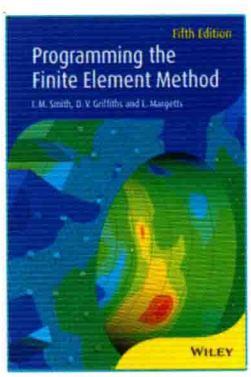
（第五版）

Programming the Finite Element Method

Fifth Edition

[英] I. M. Smith
[美] D. V. Griffiths 著
[英] L. Margetts

张新春 慈铁军 范伟丽 等译



中国工信出版集团

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

有限元应用与工程实践系列

有限元方法编程

（第五版）

Programming the Finite Element Method

Fifth Edition

［英］ I. M. Smith
［美］ D. V. Griffiths 著
［英］ L. Margetts

张新春 慈铁军 范伟丽 等译

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书在前几版的基础上进行了全面的修订，主要围绕三个方面的问题展开有限元程序设计的讨论，即固体力学问题、流体力学（包括热力学）问题及固体与流体的耦合问题（如应力学中的问题），涉及的方程主要有静力平衡方程、传导方程和特征值方程。本书致力于帮助读者通过有限元技术来使用为算法设计的“构建模块”。其重点并不在于程序，而在于过程或子程序的集合，目的在于教会读者编写智能程序并使用它们。

本书结构合理、示例丰富，可作为从事有限元研究与开发的工程技术人员及土木工程和机械工程等相关专业的教师、学生的参考书。

Programming the Finite Element Method, Fifth Edition
ISBN: 978-1-119-97334-8
I. M. Smith　D. V. Griffiths　L. Margetts

图书在版编目(CIP)数据

有限元方法编程：第五版 /(英)I.M.史密斯(I. M. Smith)，(美)D.V.格里菲斯(D. V. Griffiths)，(英)L.玛吉茨(L. Margetts)著；张新春等译. —北京：电子工业出版社，2017.5
(有限元应用与工程实践系列)
书名原文: Programming the Finite Element Method, Fifth Edition
ISBN 978-7-121-31414-8

I. ①有… II. ①I… ②D… ③L… ④张… III. ①有限元法－程序设计－指南 IV. ①O241.82-62

中国版本图书馆 CIP 数据核字(2017)第 085076 号

策划编辑：冯小贝
责任编辑：李秦华
印　　刷：三河市良远印务有限公司
装　　订：三河市良远印务有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编　100036
开　　本：787×1092　1/16　　印张：33　　字数：851.2 千字
版　　次：2003 年 9 月第 1 版（原著第 3 版）
　　2017 年 5 月第 2 版（原著第 5 版）
印　　次：2017 年 5 月第 1 次印刷
定　　价：99.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010)88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：fengxiaobei@phei.com.cn。

译 者 序

随着计算机硬件技术的巨大改进，有限元技术在各个工程领域正日益显示出强大的生命力。有限元技术是一种数值计算技术，它对不能用解析方法求解的问题，有着独特的应用能力。但是，掌握有限元技术又不仅仅是一个纯粹的理论问题，它要求应用人员有一定的程序设计能力，才能将它们应用于工程领域实践，否则只能“望洋兴叹”。本书就是在这样一种背景之下翻译出版的，使读者真正具有编制结构清晰可见、阅读性强的有限元程序。

本书的特点之一是提供了相当多的源程序供读者参考，真正地“站在巨人的肩膀上”。本书提供的各个程序都利用了 FORTRAN 2003 的强大功能，采用模块化编程技术实现，有利于读者阅读。同时，结合具体的工程应用实例，对本书中各个程序的应用原理、变量含义等做了简明扼要的说明。本书的另一特点是，对每一类问题都提供了不同的求解技术，如高斯直接消元法、迭代法、隐式积分法、显式积分法、混合显式/隐式积分法、共轭梯度法等。通常情况下，有限元分析对计算机的存储要求和运行速度要求都是很高的，尤其是对大型问题而言。因此，本书除了提供通常采用的单元组装技术之外，还提供了有关“逐个单元”法或“自由网格”法等求解技术，它们不需要存储大型的总刚度矩阵。

此外，本书的这一版本中，还特别关注与其他开放软件的接口，例如，ParaView 用于计算结果的可视化，ABAQUS 用户子程序用于一些材料本构模型，ARPACK 用于大型特征值分析，METIS 用于网格划分等。本书涉及的知识面相对较广，但主要围绕三个方面的问题展开有限元程序设计的讨论，即固体力学问题、流体力学(包括热力学)问题、固体与流体的耦合问题(如应力学中的问题)，涉及的方程主要有静力平衡方程、传导方程和特征值方程。在本书中，第 2 章和第 3 章是以后各章的基础；第 4 章至第 6 章是针对弹、塑性问题的静力分析，应用的主要是静力平衡方程；第 7 章至第 9 章主要针对流体问题，或者流体与固体的耦合问题，应用的主要是传导方程；第 10 章和第 11 章是弹、塑性实体的动力响应分析，应用的主要是特征值方程；第 12 章是有限元分析的并行处理。当然，其有限元分析过程都是类似的，这有利于读者针对不同问题扩展现有程序。

本书作为一本教材，主要针对学习有限元编程技术的本科生和研究生。另外，对相关领域内从事有关有限元程序设计及应用的工程技术人员也有指导作用。

翻译国外的教材是一个再创作过程，本书的翻译过程得到了很多人的帮助和参与，没有他们为此付出的辛勤劳动，就不会在如此短的时间内顺利地翻译完这本书，在此衷心感谢为本书翻译付出努力的每一个人！除了本人之外，慈铁军副教授、范伟丽副教授、叶锋副教授和杨文刚博士等也参与了本书的翻译工作。另外，韩春雨、曹应平、白云灿、郑朝阳、张军磊、李赛赛、李先超和张晨阳等研究生也参与了本书翻译的相关工作，在此对他们的工作表示衷心感谢。全书由张新春副教授统一校核。

由于译者自身的知识局限性和精力有限，译文难免有错漏之处，谨向原书作者和读者表示歉意，并欢迎读者批评指正。

华北电力大学机械工程系

张新春

2016 年 10 月　于保定

第五版前言

本书第五版保持了先前版本成功的主题，即模块化程序设计风格，该风格简洁，易于读取计算机程序，通过求解偏微分方程解决广泛的工程和科学问题。

尽管在计算机硬件上有了巨大的改进，但编程风格本质上是相同的。本书既适合首次接触有限元方法的初学者阅读，也可供使用最新一代并行超级计算机求解大型工程问题的专家参考。

在这一版本中，特别关注的是与其他开放软件的接口，例如，ParaView 用于结果的可视化，ABAQUS 用户子程序用于一些材料本构模型，ARPACK 用于大型特征值分析，METIS 用于网格划分。

考虑到计算机硬件的快速发展，第 1 章已经重新改写，例如，GPU 的可用性和云计算环境。在第 2 章至第 11 章中，增加了大量附件用于提高分析选项。例如，新回归算法用于弹、塑性分析，更多的一般边界条件规范和动态分析的复杂响应选项。

第 12 章已经更新，说明有限元分析在并行计算环境中快速发展的可能性。在第四版中，并行“进程”的最大数目是 64，而在本版本中，这个数目已经增加到 64 000。本书还讲述了加速计算 GPU 的使用。

致　谢

非常感谢许多个人和机构对本书所做的贡献。感谢澳大利亚研究理事会对纽卡斯尔(NSW)大学岩土科学与工程研究中心(CGSE)的大力支持，尤其是黄劲松(Jinsong Huang)对第 6 章、第 8 章和第 9 章中几个新的和改进的程序开发与验证所做的贡献。Louise Lever(曼彻斯特大学)是 ParaFEM 的主要开发者之一，提供了第 1 章、第 5 章、第 6 章和第 12 章中 ParaView 使用的练习题，并建立了社团网站http://parafem.org.uk。

有许多人对第 12 章做出了贡献。Llion Evans，Paul Mummery，Philip Manning，Graham Hall 和 Dimitris Christias(曼彻斯特大学)提供了科学案例研究。Florent Lebeau 和 Francois Bodin(CAPS 公司)评估了 GPU 的使用，Philippe Young (Simpleware 有限公司)提供了基于图像模型的大力支持。

第 12 章中的程序标记利用超级计算机来完成，这些超级计算机属于英国国家高性能计算服务“HECToR”(e107，e254)和英国区域服务“N8 HPC”(EP/K000225/1)。欧盟 FP7 项目“Venus-C”和(西班牙)巴塞罗那超级计算中心提供了使用微软 Azure 的通道、资源和培训。

在本书的准备过程中，我们也要感谢我们的家庭成员对本书的大力支持，包括 Valerie Griffiths，Laura Sanchez 和 Nathan Margetts。

目　录

第1章　预备知识：计算机策略

1.1　引言

现在关于有限元方面的书籍已经有很多，但大多数都是讲述有限元分析方法的原理及其在解决实际工程和科学问题方面的广泛应用。通常，很少有讲授计算机有限元编程方面的书籍。即使有，这些书籍要么假定读者已能使用那些已编制好的程序(也可能是很复杂的“程序包”)，要么假定读者已具备编制计算机程序方面的能力。然而，对于那些在有限元领域缺乏丰富经验的人而言，在有限元分析原理的理解到具体编程的实现，仍有相当大的距离。

本书即为解决上述问题而编写的。它旨在通过一套专门针对有限元计算而设计的“构建模块”策略，帮助读者编制好自己的计算机程序，以解决特定的工程和科学问题。本书随后给出的一些程序实质上并不是某一个程序或者某一组程序，而是一个有关过程或子程序的集合(函数库)，它类似于科学计算机语言中的标准函数(如 SIN，SQRT，ABS 等)。由于有限元方程往往呈现某种矩阵结构形式，所以大多数模块程序都与矩阵运算有关。

这些构建模块可以按不同的方式进行组合，生成解决各种工程、科学问题的测试程序。这种编程方式的意义在于：一个这样的测试程序就可以作为一个开发平台，有兴趣的用户可以在该平台上开发新的应用程序。

本书的目的是帮助读者编写巧妙的程序并会使用它们。作者处理了串行和并行计算环境的问题，本书中所有的模块程序(超过 100 个)和测试程序(超过 70 个)已经在多种计算机上测试通过。另外，也考虑了计算效率。

本书所选用的编程语言是 FORTRAN 语言。毫无疑问，FORTRAN 语言仍然是大型工程和科学计算中最常用的程序语言。在本章后面，将对 FORTRAN 语言与有限元法编程密切相关的一些特点进行全面阐述。FORTRAN 语言最新的版本升级是 2008 年(ISO/IEC 1539-1:2010)。在并行环境中，尽管编程方式已经被 OpenMP 测试过，但依然会用到 MPI，或两者都会用到。

1.2　计算机硬件

原则上，任何能够编译和运行 FORTRAN 程序的计算机均可以执行本书中所给出的有限元程序。但实际上，从用于普通计算和教学目的的个人计算机到用于大型计算(尤其是非线性的三维分析)的超级计算机(通常具有并行处理能力的超级计算机)，计算机硬件的差别非常大。对于那些没有条件使用超级计算机但希望进行大型计算机的用户来说，通过云计算在基本计算机硬件的基础上能够获得超级计算支持的能力非常重要(详见第 12 章)。在此向读者推荐一个重要的编程策略，即同一个软件能够运行于所有类型的机器之上。至于向量、多核、图形、并行处理器的特性将在后面讲述(详见 1.4 节至 1.7 节)。

1.3 存储管理

在本书所有的程序中，都假定有足够的内存用以存储数据和执行程序。但是，在有限元计算中需要处理的数组可能非常大，比如，1 000 000×10 000，这样计算机就需要 10^{10} 个字(word)(数十 GB)的内存来存储这些信息。尽管有这样的计算机存在，但也相当少，目前典型的计算机内存容量仍然是在 10^9 个字(即 1 GB)这样的数量级上。

避免这个问题的一种办法是让程序员编写一个能“溢出内存”或“核外计算”的子程序，以便能处理内存中大量的数组，以及在外存和内存之间传输适当的数据。

另外一种办法是将存储管理的控制权从程序员手中交给系统硬件和软件。在这种情况下，程序员看到的仅仅是一个容量很大的单一虚拟内存，数据信息在控制程序或执行程序的控制下，在二级存储器和主存之间来回传输。这种计算机系统必须能够将变量的虚拟地址转换成在内存中的真实地址。这一转换通常包含了一个复杂的位模式匹配过程，称为“分页”。将虚拟存储器划分成大小固定或可变的段或页，通过页表索引，监控程序就能够根据用户存取数据的方式“学习”，以便能按照某种预测的方式管理存储空间。但是，存储管理是不可能完全脱离程序员控制的，而且总是假定程序员能按照某种合理的逻辑来进行操作，譬如按顺序访问数组单元(就像编译器和编程语言按行或列那样组织)。如果程序员以随机方式访问一个容量为 10^{10} 个字的虚拟内存，则换页请求会使程序的执行效率非常之低(参见 Willé，1995)。

在不久的将来，“大型”的有限元分析——比如含有 1000 万个以上的未知量——可能要用下一节即将介绍的向量计算机或并行计算机进行处理。当采用这种计算机时，如果程序员在程序中中断计算过程去执行内存与外存之间的数据传输，或者处理过程中自动出现换页的情况，则计算过程仍会耗费相当多的时间。因此，在本书的第 3 章将介绍一些特殊的策略，使得大型的有限元分析无须内外存数据交换就可以处理。但是，随着问题规模的增大，在大多数计算机上会存在主存和闪存被过度使用而性能退化的风险。

1.4 向量处理器

早期的数字计算机是以“串行”方式运行的，通俗地讲，如果有 1000 个操作要执行，仅当第一个操作完成第二个操作才会开始。不过，当操作是在数组上进行时，完全可以想象，如果对两个数组单元进行操作的结果不会影响对另外两个数组单元的操作，那么这样的计算完全可以同时执行。计算机中实现这种功能的硬件称为“管道”，一般而言，所有的现代计算机都或多或少地使用了这种硬件技术。由可实现管道化操作的专用硬件组成的计算机称为向量计算机。由于“管道”的长度毕竟有限，所以，为了使操作能同时进行，就必须使相关的操作数在正确的时间内放到管道中。此外，还必须遵守一个原则，即同时进行的两个操作不能互相依赖。这两点要求(众多要求的一部分)意味着：为使计算机的向量处理能力得到最大限度的发挥，在编写程序时必须采取一些必要的措施。另外，还有一个很有趣的“副作用”，即为向量计算机编制好的程序在其他任何计算机上都会运行得更好，这是因为计算所需的数据信息能在恰当的时间出现在恰当的位置上(如在一个特殊的缓存中)。

但是，真正的向量计算机是非常昂贵的。在编写程序时，一种更加普遍提高处理速度的

方法是在多台计算机上并行处理计算。这样做的原因是使个人计算机也能用于计算，从而降低了价格。对于大规模计算，向量计算机和并行计算机混合使用是最理想的。

1.5　多核处理器

在 20 世纪 80 年代以前，个人计算机只有一个独立的中央处理器。大约每 18 个月，制造商就能将处理器上的晶体管数量加倍，并提高计算机每秒计算的次数(即时钟频率)。到了 21 世纪，电路的微型化达到了可以可靠生产的物理极限。另外一个问题是，保持处理器冷却和节能变得越来越困难。随着多核处理器的发展，这些问题已逐一被解决。制造商不再增加晶体管的数量和提高时钟频率，而是开始将两个或多个中央处理器(核心)组装到单个硅片或封装芯片的多个硅片上。在过去 10 年内，多核处理器很快取代了所有计算机上的单核处理器。

多核处理器的性能取决于同时应用多个核心的能力。程序员需要将软件编写成并行处理的形式，这将在后面进行介绍。这些现在所谓的“标量”计算机也包含一些向量型硬件。最新的 Intel 处理器每个内核有 256 位向量单元，足够同时计算 4 个 64 位浮点操作(与真正的向量处理器相比)。从第 5 章开始，本书将阐述“向量化”程序。

1.6　协处理器

协处理器是第二处理器、辅助主处理器，用于完成特殊的任务，如图像操作，比主机“一般用途”处理器要快得多。它的原理和向量处理器类似。历史上，计算机曾采用过协处理器，也将协处理器去除过。

在写入时，图形处理单元(GPU)是加快数值计算速度的常用方法。GPU 是有上百个内核的专用高效处理器。它们以插板的形式安装在标准计算机上。这种协处理器的主要任务是将数据从计算机内存到 GPU 主板之间来回传送。如果软件安装启动不能将内存转移次数降到最低，运行速度将会大大降低。为了克服这一缺点，处理器开始出现，把图形处理器安装到相同的硅模具上。由多核、分层内存和专门的 GPU 单元组成的处理器被称为“芯片上的系统，即片上系统”，这也是现代计算机下一步发展的方向。

主要有两种编写图形处理单元软件的方法：(1)开放式计算语言(OpenCL)；(2)统一计算设备架构(CUDA)。OpenCL(http://www.khronos.org/opencl)是一种编写软件的开放式结构，它可以应用到任何用户的图形处理单元和其他类型的处理器上。CUDA(http://developer.nvidia.com/category/zone/cuda-zone)是一种专用于 NVIDIA 硬件上的结构。图形处理单元的应用将在第 12 章中进一步讲解。

1.7　并行处理器

在这个概念中(其中有很多变体)，有几个物理意义完全不同的处理单元(一个处理器有几个核或一台计算机的多个多核处理器)。这些处理器也可以使用协处理器。程序和/或数据可以驻留在需要相互联系的不同的处理单元上。

目前，有两种可预知的通信组织方式(就像前面讲述的内存管理那样)。一种是由程序员控制通信进程——使用一个称之为“信息传送”的程序设置机制，另一种是由程序自动执行

而不需要程序员的控制。第二种策略毫无疑问是很吸引人，但到目前为止还未能成功实施。

对于一些特殊的硬件，制造商给用户提供一些可嵌入到程序中的“指令”，编译者也可以将这些指令应用到编码的并行化部分(通常和 DO 循环有关)。Smith(2000)表明这种方法对于适当数量的并行处理器的效果非常显著。然而，这些程序不方便应用于其他计算机上。

另外一种方法是使用 OpenMP，即一系列简便的指令，这些指令只能用于一类所谓“共享内存”的并行计算机上。尽管本书中的编码非常适用于使用 OpenMP 的并行处理器(Pettipher 和 Smith，1997)，但这种方法对“共享内存”和“分散内存”系统同样适用，具体将在第 12 章中详细介绍。程序在多核处理器上成功运行，多台个人计算机通过以太网进行通信，程序在共享和分散内存的超级计算机上运行时通过更加昂贵的通信系统进行通信。这种由程序员控制下的消息传递策略可以在诸如 MPI(消息传递接口)这种环境中实现，MPI 是一种 de facto 标准，确保了策略在各种计算机之间的可移植性(MPI Web 协议，2003)。

一个共享内存机器最简单的形式是一个可接入主内存单存储库的多核处理器。并行计算机中包含多个多核处理器，使用混合编程策略有时是一个优势，其中 OpenMP 用来促进本地内核(一个处理器内)之间的通信，MPI 用来和远程内核(在其他处理器上)通信，这种策略往往能达到很好的效果。

1.8 应用软件

因为计算机硬件(指令格式，向量计算能力等)不同，存储管理方式也有所不同，所以，为了最有效地利用这些不同的设备，运行于其上的程序结构当然应该因机器的不同而不同。然而，为了便于程序的移植和程序员的培训，所有计算机上的工程与科学计算程序通常都是用“高级”语言编写的，这种高级语言独立于计算机的硬件。FORTRAN 是目前为止用于工程与科学计算中最为普遍的语言。本节将重点介绍 FORTRAN 语言的最新标准中用于有限元计算的一些主要特点。

图 1.1 是一个典型的 FORTRAN 程序(Smith，1995)，用于处理有关的民意测验，在这里主要向习惯于 FORTRAN 或其他类似语言的读者来说明 FORTRAN 的基本结构。

图 1.1 一个典型的 FORTRAN 程序

从这个例子可以看到，程序可以用自由格式书写，也就是说，程序员可以在页面或屏幕上随意安排语句的位置。其他需要注意的特点还包括：

- 大小写字符可以任意组合。在本书中，大写字符用来标识 FORTRAN 的本征库函数和“关键词”。
- 多条语句可以置于一行，仅需以“;”相隔。
- 长语句行可以在行尾使用“&”进行续行，并同时在续行的开头加上“&”。
- 注释以“!”开头，在编译时“!”后的语句被编译器忽略。
- 长名字(包括下画线在内最多 31 个字符)允许使用有表征意义的标识符。
- “IMPLICIT NONE”语句强制要求所有的变量名和常量名都必须显式声明。这在调试程序时有极大的帮助。
- 变量声明使用双冒号“::”。
- 程序中没有带标号的语句。

1.8.1　编译器

图 1.1 所示的可读文本通过称为“编译器”的程序转换为计算机指令。目前，有许多适合于学生用的免费编译器，如 G95(www.g95.org)和 GFORTRAN(http://gcc.gnu.org /fortran/)。本书中使用的是商业 FORTRAN 编译器，这些编译器具体由 Intel，Cray，NAG 和 Portland 等组织所提供。当在一个超级计算机上建立应用程序时，强烈建议采用商家提供的编译器。这些编译器生成的程序很有特色，比免费的编译器版本能更好地利用计算机硬件。

图 1.1 展示了一个基于 Windows 的编程环境。在这个直观的图形用户界面支持下，可以编写、编译并执行 FORTRAN 程序。FORTRAN 程序也可以使用一个文本编辑器进行编写，在 Windows 或 Linux 终端上用简单命令编译。在“命令行”如何进行编译的例子如下所示。这个编译器采用的是 G95。

```
g95 -c hello.f90          创建一个名为 hello.o 的目标文件
g95 -o hello hello.f90    编译并链接以生成一个可执行文件 hello
```

1.8.2　算术精度

有限元分析是一门计算性很强的技术(参见第 6 章和第 10 章的例子)，它能达到的最大数值精度由 64 位的机器字长所决定。FORTRAN 中有一些有用的内部函数可用于确定和改变处理器的精度。例如，下面的语句：

```
iwp = SELECTED_REAL_KIND(15)
```

将返回一个整型变量 iwp，使处理器上的 KIND 变量取得 15 位十进制精度。如果处理器达不到这个精度，iwp 将返回一个负值。

在为 iwp 赋予必要的值之后，FORTRAN 将采用如下形式声明 REAL 型变量：

```
REAL(iwp)::a,b,c
```

并使它们获得如下的赋值形式：

```
a=1.0_iwp; b=2.0_iwp; c=3.0_iwp
```

为了简明起见，本书给出的大多数程序，在声明变量时指定了常数，例如