计算机数学简介

高小山

中国科学院数学与系统科学研究院

摘要。计算机数学是研究算法的数学，是数学与计算机科学交叉融合产生的新兴学科。计算机数学主要研究内容包括：为算法研究提供数学工具的离散数学，研究算法共性的计算理论，从算法角度研究数学各个分支的机械化数学。本文简要回顾了计算机数学的历史，介绍了其主要内容并展望了未来的主要研究问题。

关键字。计算机数学，算法，离散数学，计算理论，机械化数学

1、什么是计算机数学

计算机数学，顾名思义，是研究应用计算机解决各类问题需要的数学。计算机数学关注“什么是可以计算的”，对于可计算的问题，则关注设计求解该问题的最好算法。所以，我们可以简单地说计算机数学是研究算法的数学。

计算机科学大师 D. Knuth 将计算机科学定义为研究算法的学问。其实，计算机数学是数学与计算机科学的交叉领域：计算机数学是计算机科学的理论基础，也是研究计算与算法的数学分支。

计算机数学大致可以分为以下三部分。

首先，为算法研究提供数学工具的是离散数学。与传统的连续数学或分析数学不同，离散数学研究离散对象的数学结构，主要包括：集合论、图论、组合数学、抽象代数等。需要说明的是，离散数学研究的侧重点与传统数学有所不同。纯粹数学更关心数学对象的结构与分类，而离散数学则侧重研究相关的算法问题。例如，对于数论中的素数，数学家更关心的是素数的分布，而计算机数学则更关心是否存在分解大整数的快速算法。另一方面，两者又密切相关。大整数分解算法的研究需要数论、代数几何等学科的支撑。一个明显的事实是，由于计算机的广泛使用，离散数学在近半个多世纪以来得到了复兴。一些连续数学分支，为了借助计算机求解，也发展了离散化理论。例如，微分方程求解的有限元方法，即通过离散化将微分方程求解变为代数方程求解。又例如，为了处理计算机图形学中出现的离散曲线与曲面，出现了离散微分几何[1]。

其次，关于算法共性的研究已经形成一个专门的学科，即计算理论或理论计

算机科学，其核心内容是判定性问题与计算复杂度理论。从算法角度研究一个问题，首先需要知道是否存在求解给定问题的算法，即判定性问题或可计算问题。许多重大数学问题由于判定性问题的研究得到澄清。例如，一个公理体系内的所有命题是否可以判断？什么是可计算的？特别是，实数是否可以计算？等等。对于一个可判定的问题，我们需要设计求解该问题的“好的算法”。一个算法的好坏，可以从其时间计算复杂度与空间计算复杂度来判断。所谓时间计算复杂度可以简单理解成求解问题所需的步骤数，而空间计算复杂度则是求解问题所需要的存贮空间。计算复杂度理论的主要任务是对各种计算问题根据其计算复杂度进行分类。

最后，数学本身也因为计算机的使用而得到了长足的发展。一些重大的遗留问题，如四色定理与 Kepler 猜想，借助计算机得到了解决。更重要的是，出现了一批借助计算机研究数学自身的分支，如计算数学或数值计算(一般不归在计算机数学，由其他文章介绍)、自动推理、计算机代数、计算数论、计算代数几何、计算拓扑、计算几何、符号分析等。这里，每一个学科的出现都有双重目的。例如，计算数论不仅丰富了数论的内涵，还是密码与编码等重要信息技术的数学基础。如今，算法这一概念，就像方程、公式一样，已经成为日常数学语言的一部分。吴文俊在上世纪 70 年代末就敏锐地指出，计算机的出现使得数学的机械化成为可能，从而会对数学的发展起到重大影响。他将可以借助计算机进行计算与推理的数学称为机械化数学[2]。

本文将从历史回顾、计算理论与机械化数学三个方面简要介绍计算机数学的发展趋势与主要研究内容。

2、历史回顾

电子计算机的出现不过数十年，而算法的概念却源远流长。回顾数学发展史，主要有两种思想：一是公理化思想，另一是算法化或机械化思想[3]。前者源于希腊，后者则贯穿整个中国古代数学。这两种思想对数学发展都曾起过巨大作用。从汉初完成的《九章算术》中对开平方、开立方的机械化过程的描述到宋元时代发展起来的求解高次代数方程组的机械化方法，无一不与数学机械化思想有关，并对数学的发展起了巨大的作用。公理化思想在现代数学，尤其是纯粹数学中占据着统治地位。然而，检查数学史可以发现，数学的多次重大跃进无不与机械化思想有关。数学启蒙中的四则运算由于代数学的出现而实现了机械化。线性方程组求解中的消去法是机械化思想的杰作。对近代数学起着决定作用的微积分也是得益于经阿拉伯传入欧洲的东方数学的机械化思想。在现代纯粹数学研究中，机械化思想也一直发挥着重大作用。Hilbert 倡导的数学判定性问题的研究导致了

数理逻辑的突破性发展并为计算机的设计原理做了准备。E. Cartan 关于微分方程、微分几何及李群的著作中经常显现出机械化特色。H. Cartan 关于代数拓扑学同调群计算的工作可以看作是机械化思想的成功范例。

数学机械化思想的明确提出可以追溯到 17 世纪法国思想家 R. Descartes。Descartes 认为，代数可以将数学机械化，使思维变得简单，不再需要繁复的脑力劳动，数学创造也极可能成为自动。甚至逻辑原理和方法也可以被符号化，进而所有的推理过都实现机械化[4]。Descartes 还将他这一设想具体化，提出一个求解一般问题的具体构想：将任意问题的解答归结为数学问题的解答，将数学问题的解答归结为代数问题的解答，将代数问题的解答归结为方程组求解，最后方程组的求解可以归结为单个方程求解。Polya 评价到：“这一构想虽未成功，但它仍不失为一个伟大的设想。即使失败了，它对于科学发展的影响比起千万个成功的小设想来，仍然要大的多。”[5] 这是因为虽然这一设想不能涵盖所有问题，但却包括了大量有重要意义的问题。

G. Leibniz 发展了 Descartes 的想法，并开始了一个更加雄心勃勃的计划。Leibniz 提出应该发展一种广义计算，这种计算可以使人们在所有的领域都能机械地、不费力地，通过一种像算术与代数那样的演算来达到精确的推理。这种方法将“使真理昭然若揭，颠扑不破，就像是建立在机械化的基础之上。”[4]

Descartes 和 Leibniz 提出的想法是比较笼统的。19 世纪中叶，G. Boole 创立了现在所说的 Boole 代数，把思维在某种程度上形式化，用代数形式加以描述。这一工作比起 Leibniz 和 Descartes 的想法至少有了某种程度的数学化。20 世纪 20 年代，D. Hilbert 正式提出了所谓的“Hilbert 计划”，试图通过公理化建立数学的严格基础。特别是，Hilbert 在其计划中提出了判定性问题，即是否存在一个算法“机械化”地判定每个数学分支中所有命题的正确性[6]。

1931 年，奥地利数理逻辑学家 Goedel 证明，即使是 Peano 算术这样简单的数学系统，也存在定理，尽管我们知道是对的，却不能够证出来 [7]。Hilbert 希望证明数学是圆满无缺的，是相容的，是可以判断的。Goedel 的结论指出，Hilbert 计划太过理想，对于很多数学学科，Hilbert 的数学公理化计划无法实现。Goedel 的结论是革命性的，人们首次严格证明有的知识是不可以推出或计算的。Hilbert 计划虽然不能完整实现，但对数学发展的影响是巨大的。本文将讲到，计算理论与机械化数学都可以说是在 Hilbert 判定性问题的直接影响下产生的。

3、计算理论

A. Turing 因为提出计算理论的基本概念 Turing 机，被誉为现代计算机奠

基人之一。Turing 这一研究的起因是希望回答 Hilbert 计划中的可判定性问题[8]。为了回答可判定性问题，首先需要明确可以用于判断的计算手段。为此，Turing 改进了 Goedel 的想法，提出著名的 Turing 机。Turing 机不是一种具体的机器，而是一种计算模型。根据 Church-Turing 假设，用任意算法可以计算的问题，都可以用 Turing 机计算。换句话说，Turing 机为计算这个抽象术语提供了一个严格的数学模型，从而为算法的严格研究奠定了基础。在其论文中，Turing 证明 Turing 机的停机问题是不可判定的，即不存在一个算法，判定 Turing 机对于所有可能的输入是否停机，从而给出了 Hilbert 可判定性问题的又一个反例。

著名的不可判定的数学问题，除了 Goedel 与 Turing 所给的例子之外，还有 Hilbert 第十问题：任意整数系数多项式方程是否存在整数解是不可判定的。

在计算理论中，关于不可判定性的研究属于基础理论范畴。如前所述，计算机数学更关心的是如何实现高效计算。所以，计算复杂性理论成为计算理论的主流。

计算复杂性主要研究算法的复杂度，包括时间与空间复杂度，即对于给定的输入，在 Turing 机上用多少步骤与多少存储空间可以完成所给的计算。其中又以时间复杂度最为重要。设输入的大小是 n，则在 n 的多项式时间内可以用 Turing 机求解的问题的集合记为 P，在 n 的指数时间内可以用 Turing 机求解的问题的集合记为 EXPTIME。显然，P 是 EXPTIME 的真子集。一般认为，P 中的问题是可以在计算机上大规模高效求解的。所以对于 P 中问题的界定变为计算复杂性的核心问题。

一类介于 P 与 EXPTIME 之间的计算问题，NP 类，在计算复杂性中扮演关键角色。NP 类是指由非确定型 Turing 机在多项式时间内可以解决的判定问题。所谓非确定型 Turing 机，是指在一步计算中可以根据不同条件选择多种执行步骤的 Turing 机。通俗地讲，P 问题是指能够在普通计算机上多项式时间内求解的判定问题，而 NP 问题则是指那些在普通计算机上能够在给定正确信息下，可以在多项式时间内验证的判定问题。很显然，P 属于 NP，那么 NP 是否属于 P？这就是著名的“P=NP？”问题，是计算复杂性的核心研究问题。

20 世纪 70 年代，S.A. Cook 发现 NP 问题中最困难的问题具有特殊性质。Cook 证明，如果这类问题中的一个属于 P，则所有其他 NP 问题都属于 P [9]。由此，Cook 引入了 NP 完全的概念，第一次明确提出了“P=NP？”问题。

人们之所以关注 NP 完全问题，是因为在各个领域遇到的大多数自然的难解问题，最终都发现是 NP 完全问题。NP 完全问题类非常丰富，存在于数学、优化、人工智能、生物、物理、经济、工业等各个领域。如果能够解决 P 与 NP 两个问

题类之间的关系，则解决了这些问题的算法复杂度问题，具有非常重大的实际意义。

70 年代人们对 NP 完全问题的研究主要是横向发展，也就是以许多不同的计算模型来分析难解问题的本质。已经发现的 NP 完全问题超过千个，涉及几乎所有数学学科的计算问题，如数理逻辑、图论、数论、拓扑、代数、组合优化、几何、对策论等。因此，我们可以应用多个数学学科的知识以多种不同的手段研究这一问题[10，11]。例如，在 Smale 提出的本世纪 18 个重大数学未决问题中，他选择了下列源自传统数学的 NP 完全问题作为“P=NP？”问题的代表：给定 Z_2 上关于 n 个变量的 k 个多项式，问是否存在多项式时间的算法判定它们在 $(Z_2)^n$ 上有公共零点[12]。对多种 NP 完全问题的研究使我们对难解问题有了更深的认识，另一方面也产生了一些预想不到的应用。例如，基于 NP 完全理论，密码学取得了革命性突破，建立了公钥密码体系[13]。

到了 80 年代中，NP 完全问题的研究有了纵向的突破，在许多表面看来并不相关的计算问题之间发现了深刻的刻画关系。这些刻画关系不但解决了几个令人困扰多年的问题，同时也刺激了其它相关领域的发展。其中一个重大的结果是概率可验证明(PCP)对 NP 类的刻画，由此得出了许多组合优化问题近似解的 NP 完全难近似性，从而刺激了近似算法的研究[14]。90 年代，人们又研究了新的计算模型比如量子计算机和命题证明系统。新工具的引入无疑增加了这一问题的内涵。例如，可以问：在量子计算机模型下是否存在类似“P=NP？”的问题；是否具有量子多项式时间算法求解某类 NP 完全问题？另一个研究热点是参数复杂性理论[15]，将 NP 完全问题的参数区别对待。例如图的顶点覆盖问题，它的参数有图的顶点个数 n 和覆盖子集尺寸 k，而如果固定参数 k，则顶点覆盖问题是多项式时间可解的，但是其运行时间是 k 的指数函数。于是研究 NP=P 问题，可以分解为各个 NP 完全问题的固定参数如何影响其 NP 完全性的问题。

一般认为，“P=NP”是不成立的。如果这一结论正确，那么希望通过研究某个具体的 NP 完全问题解决 P=NP？问题的途径是不可行的。P=NP？问题的解决可能需要全新的数学理论。因此，Smale 将 P=NP？问题称为“计算机科学为数学带来的一个礼物。”[12] 这一问题被 Clay 研究所列为七个千僖问题之一。

既然“P=NP”被认为是不成立的，而现实应用中又存在如此之多的 NP 完全问题，怎样有效求解这些 NP 完全问题变为计算复杂性的又一核心问题。NP 完全问题的计算困难性是对于 Turing 机的确定型算法而言，因而寻找高效算法必须使用新的计算模型。下面介绍几种常用的计算模型,主要是并行算法、随机算法、近似算法与量子算法。

NP 完全问题应用最为广泛的求解算法是各种随机搜索算法。例如，由 J. Holland 发明的遗传算法，是模仿动物基因进化过程设计的优化问题的求解算法[16]。遗传算法是一种基于随机选择与并行计算的贪心算法。这类算法已被广泛应用于各种问题的求解，但是由于算法的复杂性，其理论分析非常困难，还远未取得实质性进展。随机算法虽然在 NP 完全问题求解方面未取得实质性理论进展，却为很多其他计算问题提供了有效的途径，已经成为一种非常有效的计算手段[17]。

近似算法用于求解优化问题，希望用多项式时间算法给出某个 NP 完全优化问题真正最优值的某个百分比[18]。对于近似算法，NP 完全问题可以分为三类。最容易的一类，例如背包问题，我们可以设计一系列算法在多项式时间内求的任意接近其最优解的近似解。中间的一类，我们可以设计近似算法求的其最优值的某个固定比例 r (0<r<1)。最困难的一类，对于任意的 r (0<r<1)，近似求解其 r 最优值也是 NP 困难的。近似算法的主要目标是对计算问题根据其近似计算的复杂度进行分类。

另一种思路是采取完全不同于 Turing 机的计算模型，例如量子计算。量子计算是借助于量子力学原理设计的一种计算模型，其主要特点是高度并行与随机性。1994 年 P. Shor 证明可以用量子算法在多项式时间内分解大整数[19]。由于大整数分解的困难性是 RSA 密码体制安全性的基础，这一工作引发了人们关于量子算法的关注。有趣的是，到目前为止，利用量子算法可以实质性提高速度的问题并不多。特别地，量子算法还不能实质性降低求解 NP 完全问题计算复杂度。

当然，计算复杂性学科也还有很多其他的重要问题。例如，我们还不知道若干具有重要应用背景的计算问题的计算复杂性。下面举例说明。

目前，我们还不知道大整数因子分解问题是否属于 P。由于大整数因子分解计算困难性是现在广泛使用的密码 RSA 的安全性基础，所以研究大整数因子分解的快速算法及其计算复杂度具有重要意义。

两个n×n矩阵如果按照通常的方法做乘法，其计算复杂度大致是n^3。1969年Strassen提出“快速乘法”算法，其计算复杂度大致为$n^{2.807}$。现在已知的计算复杂度最好的算法的复杂度大致是$n^{2.37}$。人们猜想，矩阵乘法的计算复杂度可以接近n^2。由于矩阵乘法在各种计算中大量使用，寻找快速有效的矩阵乘法算法有重要的意义。

4、机械化数学

计算机最初(现在也仍然是)主要应用于工程计算，其中主要用到的是近似计算。一个自然的问题是：计算机是否可以通过进行精确的计算与推理用于数学研

究？我们是否可以利用计算机的强大计算能力自动或半自动地解决数学问题？由于定理证明是数学最核心的内容，我们是否可以用计算机证明定理？

前面提到，从 Descartes 到 Hilbert，都是机械化数学的支持者与倡导者。机械化数学发展的相对滞后与相关问题的计算复杂性密切相关。首先，Goedel、Turing 的结果否定了整个数学学科机械化的可能性。这些反面结果影响巨大，以至于形成了数学不可以机械化的固定思维。实际上恰恰相反，与 Goedel 的著名结果几乎同时，法国数学家 J. Herbrand 在 1931 年写出了题为“论算术的相容性”的论文[20]。Herbrand 创立了一种证明定理的算法。这种算法提供了一种进行推理的途径，如果一个命题存在一个证明，则算法在有限的步骤之内结束并给出命题的证明。这一算法是半判定性的，即算法对于某些输入可能不中止，从而不能得出结论。结合 Goedel 的结果，我们可以看到，Herbrand 实际上已经给出了 Hilbert 判定问题理论上的完整解答。由 Goedel 的结果，有些定理是不能够由公理推出的。此时，Herbrand 的算法将不中止。其余的定理都可以由公理推出，而对于这些定理，Herbrand 的算法将给出证明。那么，数学定理的机器证明问题是否解决了？答案当然是否定的。Herbrand 算法的主要问题在于，其计算复杂度是指数的。虽然理论上可行，但实际上不能用于在计算机上证明非平凡的数学定理。

真正在计算机上自动证明定理始于上世纪 50 年代中期。一些计算机科学家，包括 Newell、Simon、Shaw 等人，创立了人工智能学科，尝试利用计算机进行某种脑力劳动，特别地证明数学定理。由此成长起来一门新的学问—自动推理或机器证明。自动推理前期的主流工作是对 Herbrand 算法的改进，希望通过发展各种技巧简化 Herbrand 算法的计算复杂度。但是，一般机器证明算法的发展并不理想，因为定理证明是一个计算复杂度非常高的问题。机器证明的主流逐渐演变为机器验证。

机器验证的主要思路是使用一些高效但不完全的自动推理工具进行自动推理。在自动推理不能进行下去的时候，允许用户通过增加引理等手段提供证明思路。如此多次反复，最后由计算机将证明自动生成。由此生成的证明，虽然不是完全自动的，却是严格验证的。机器验证的思路是成功的。一些重要的数学猜想，借助于计算机验证得到解决。基于这一思路开发的软件已经是计算机芯片正确性验证软件的核心技术。

1976 年 K. Appel 与 W. Haken 宣布借助计算机证明了图论中的四色定理。这一证明由于“不可读”，未能被广泛接收。1997 年，Robertson 等人基于 Appel 与 Haken 的思路，给出了四色定理一个更简单的证明，使得四色定理的证明得到

了初步承认。2005 年，G. Gonthier 借助通用机器验证软件平台 Coq 给出了四色定理的第一个真正的“机器证明，”即这一证明是经过计算机自动检验的，因此可信度非常高[21]。

另外一个著名的例子是Kepler猜想的解决[22]。Kepler猜想是关于球在空间中最佳堆积的猜想，已经有四百多年的历史。H. Thomas使用计算机验证了大量的情形，并最终宣称证明了这一猜想。与Appel与Haken的遭遇不同，Thomas的结果基本得到数学界的承认，并发表在数学顶级杂志数学年刊上。

在以上两个例子中，虽然著名的猜想被证明,但是用于证明的方法仅仅是针对这两个问题，似乎并未产生广泛的应用。现在，我们介绍了两种极端情形。Herbrand 算法非常一般，但是不能解决具体问题。四色定理与 Kepler 猜想的证明方法又非常特殊，不能用于其他问题。那么，有没有一条可行的中间之路呢？回答是肯定的。我们用吴文俊关于几何定理机器证明的工作给予说明。

几何定理机器证明是人工智能创始时即最早尝试的数学问题，主要原因是几何推理自古被认为是严格推理的典范，而且一般认为几何定理的证明技巧性很强。但是，基于人工智能方法所开发的软件效率不高，只能证明非常简单的几何定理。1950 年，波兰数学家 A. Tarski 证明初等代数和初等几何定理可以用一种代数算法来证明或否定，即初等几何是可以判定的[23]。但是 Tarski 算法的复杂度太高，以至于不能用来证明有意义的定理。吴文俊于 1978 年发表了几何定理机器证明的代数方法，在几何定理机器证明方面取得突破[2]。“吴继续深化、推广他的方法，并将这一方法用于一系列几何。包括平面几何，代数微分几何，非欧几何，仿射几何，与非线性几何。不仅限于几何，吴还将他的方法用于由 Kepler 定律推出 Newton 定律；用于解决化学平衡问题；与求解机器人方面的问题。吴的工作将几何定理证明从自动推理的一个不太成功的领域变为最成功的领域之一。在很少的领域中，我们可以讲机器证明优于人的证明。几何定理证明就是这样的一个领域。”[24]

受到自己工作的启发，吴文俊在写于 1979-1981 年期间的几篇文章中明确指出数学机械化的重要性，并给出了后来称之为“数学机械化纲领”的研究思路[2]：“在数学的各个学科选择适当的范围，即不至于太小以致失去意义，又不至太大以至于不可机械化，提出切实可行的方法，实现机械化，推动数学发展，并以此为基础解决高科技问题。” 吴文俊的基本想法是 Herbrand 的方法太广，以至于不够有效，而 Appel 与 Haken 类型的方法又应用范围太窄，不能为他人所用。数学机械化正确之路应该是选择有意义的一类问题，发展统一求解的高效算法，逐步实现数学的机械化。近年来蓬勃发展的符号计算[25]、计算代数几何

[26]、计算数论[27]、计算群论[28]、计算拓扑[29]、符号分析等新兴学科无疑说明了吴文俊以上观点的正确性。

符号计算是计算机数学的基础，它对于计算机数学的作用正如数值计算对于计算数学。符号计算形成于 20 世纪 60 年代，当时的标志性成果是多项式 GCD 与因式分解的快速算法。符号计算主要研究内容包括：基本代数运算的符号算法、矩阵的符号算法、多项式系统的符号算法、微分与差分方程的符号算法、符号分析等。以符号计算为基础的数学软件 Mathematica 与 Maple 已经被广泛使用。代数与微分非线性方程组的求解算法一般是指数的。为了提高符号算法解决实际问题的能力，人们提出混合计算方法，通过将符号计算、数值计算、优化算法等结合，得到速度快又能保证计算结果正确的可信算法。

参考文献

1. Bobenko，A I，Schröder，P，Sullivan，J M, Ziegler，G M. Discrete differential geometry，Birkhäuser Basel，2008.
2. 吴文俊. 吴文俊论数学机械化. 山东教育出版社，1995.
3. 吴文俊. 数学机械化研究回顾与展望，吴文俊荣获邵逸夫数学奖庆祝会，2006. 本文第 2 节摘自此报告。
4. 莫里斯.克莱因. 古今数学思想. 上海科学技术出版社，2002.
5. 乔治.波利亚. 数学的发现. 内蒙古人民出版社，1980.
6. Hilbert D. Die grundlagen der elementaren zahlentheorie. Mathematische Annalen. 1928, 104，485–94. Translated by W. Ewald as The Grounding of Elementary Number Theory.
7. Gödel K. On formally undecidable sentences of Principia Mathematica and related systems，Imontshefte fur Mathematik，1930, 38，173-198.
8. Turing，A M. On computable numbers，with an application to the Entscheidungs Problem，Proc. of the London Mathematical Society，1936, 42(2): 230–65.
9. Cook S A. The complexity of theorem-proving procedures. Proceedings of the 3rd Annual ACM Symposium on Theory of Computing，1971, 151-158.
10. Garey M R and Johnson D S. Computers and intractibility，a guide to the Theory of NP-completeness，W.H. Freeman and Co.，1979.
11. Wigderson A. P，NP and mathematics: a computational complexity perspective，Proc. of the 2006 International Congress of Mathematicians，vol. 1, EMS Publishing House, Zurich, 665-712, 2007.
12. Smale S. Mathematical problems for the next century，Mathematical Intelligencer，1998, 20，7-15.
13. Diffie W. and Hellman M. New directions in cryptography，IEEE Trans. Inf. Theory，1976, 22，644.

14. Arora S，Lund C，Motawani R，Sudan M，Szegedy M. Proof verification and the hardness of approximation problems. Journal of ACM，1998，45(3): 501-555.
15. Flum J and Grohe M. Parameterized complexity theory，Springer，2006.
16. Holland J. Genetic algorithms，Scientific American, 1992, 267(1), 66-72.
17. Mitzenmacher M and Upfal E. Probability and computing: randomized algorithms and probabilistic analysis，Cambridge University Press, 2005.
18. Hochbaum D S. Approximation algorithms for NP hard problems. PWS Pub.，1997.
19. Shor P W.，Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer，1997, SIAM J. Comput. 26，1484.
20. Herbrand J. Logical writings. D. Kreidel，1971.
21. Gonthier G. A Computer-checked proof of the four color theorem，Microsoft Research Cambridge，2005.
22. Thomas H.C. A proof of the Kepler conjecture，Annals of Mathematics，Second Series, 2005, 162(3):1065–1185.
23. Tarski A. A decision method by which the truth of sentences of the elementary algebra and geometry，Univ. of California Press，Berkeley，1951.
24. McCune W. Automated deduction，LNAI 1246, Springer, 1997.
25. Grabmeier J，Kaltofen E，and Weispfennig V. Computer algebra handbook, 2003, Springer.
26. Vasconcelos W V. Computational methods in commutative algebra and algebraic geometry，Springer–Verlag，1998.
27. Bach E and Shallit J. Algorithmic number theory, MIT Press, 1996.
28. Golumbic M C. Algorithmic graph theory and perfect graphs，North-Holland, 2004.
29. Matveev, S V. Algorithmic topology and classification of 3-manifolds, Springer, 2003.

A Brief Introduction to Computer Mathematics

Xiao-Shan Gao

Academy of Mathematics and Systems Science, Chinese Academy of Sciences

Abstract. Computer mathematics is the mathematics discipline to study algorithms, which is an interdisciplinary research field originated from the interaction between mathematics and computer science. The main research domains of computer mathematics include discrete mathematics which provides basic tools for studying

algorithms, theory of computation which studies the common properties of algorithms, and mechanized mathematics which focuses on the algorithmic study of various disciplines of mathematics. In this paper, the history, main research achievements, and future directions of computer mathematics are briefly introduced.

Keywords. Computer mathematics, algorithm, discrete mathematics, theory of computation, mechanized mathematics