河海大学本科数学系列教材之五

计算方法

JISUAN FANGFA

黄健元※主编
徐小明　丁莲珍　姚健康※编

河海大学出版社

河海大学本科数学系列教材之五

计算方法

JISUAN FANGFA

黄健元※主编
徐小明　丁莲珍　姚健康※编者

河海大学出版社

图书在版编目(CIP)数据

计算方法/黄健元主编. —南京：河海大学出版社，2004.8
ISBN 7-5630-2006-3

Ⅰ.计… Ⅱ.黄… Ⅲ.计算方法—高等学校—教材
Ⅳ.0241

中国版本图书馆 CIP 数据核字（2004）第 067976 号

内 容 提 要

本书主要介绍计算方法中基本的算法与理论。内容包括：绪论、插值与拟合、数值积分与数值微分、非线性方程的数值解法、线性代数方程组的数值解法、矩阵特征值问题的计算、常微分方程的数值解法。

本书可作为高等学校工科专业本科生的教材，也可供广大工程技术人员及自学者参考。

书　　名：计算方法
书　　号：ISBN 7-5630-2006-3/O·117
责任编辑：龚　俊
特约编辑：王阶祥
责任校对：梁丽华
封面设计：步江华
出　　版：河海大学出版社出版发行
地　　址：南京市西康路 1 号(邮编：210098)
电　　话：(025)83737852(总编室)　(025)83722833(发行部)
经　　销：江苏省新华书店
印　　刷：常州市武进第三印刷有限公司
开　　本：787×1092mm　1/16　9.5 印张　231 千字
2004 年 8 月第 1 版　2004 年 8 月第 1 次印刷
定　　价：17.00 元

前　言

随着电子计算机的迅速发展与广泛应用，与理论研究、科学实验并列为现代科学三大组成部份之一的科学计算正越来越受到重视。计算方法是科学计算的基础，是研究计算机上求解各种各样数学模型数值解的算法与理论，是从事科学研究的人员必须了解和掌握的知识。

本书是在河海大学原《计算方法》教材(1993 版)的基础上，根据教育部《面向 21 世纪工科数学教学内容和课程体系改革的研究与实践》的要求和《高等工业学校数值计算方法课程教学基本要求》并结合多年该课程教学实践经验后重新编写而成的。全书共分七章：绪论、插值与拟合、数值积分与数值微分、非线性方程的数值解法、线性代数方程组的数值解法、矩阵特征值问题的计算、常微分方程的数值解法。

使用本书约需 48 学时，标 * 号的内容可根据学时多少酌情取舍。学习本教材需具备高等工科院校开设的高等数学与线性代数的知识。本书可作为高等工科院校本科计算方法课程的教材或参考书，也可供工程技术人员及自学者参考。

本书第一章、第二章、第四章由黄健元编写，第三章由丁莲珍编写，第五章由徐小明编写，第六章、第七章由姚健康编写。全书内容由上述四位老师进行了互审，并请中国科学院计算数学与科学工程计算研究所博士生导师陈志明研究员进行了审定。

编者衷心地感谢河海大学理学院数学系姚敬之、王淑云、郁大刚、方保镕等老师的支持。感谢河海大学出版社领导，特别是责任编辑龚俊先生的大力帮助。

由于编者水平有限，书中不当之处在所难免，敬请同行和读者指正。

编　者

2004 年 5 月

目　录

第 1 章

绪　论

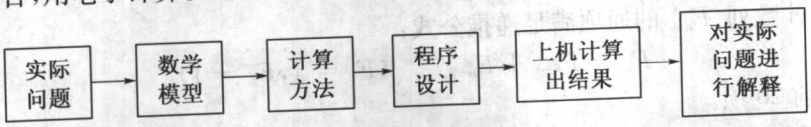
计算方法(或数值计算方法、数值算法)是研究数学问题求数值解的算法和有关理论的一门学科。随着科学技术的不断发展,数值计算越来越显示出其重要的作用。计算机技术的飞速发展极大地促进了数值计算方法的改进,数值计算方法是每一位从事科学研究与应用的人士不可缺少的知识。本章首先介绍计算方法的研究对象、内容、算法的结构、算法的描述方式;然后介绍误差及其相关概念;最后介绍算法设计与选择时应遵循的若干原则。

第一节　计算方法简介

一、计算方法的研究对象、内容

当今科技领域中所提出的三大环节是:科学理论、科学试验以及科学计算方法。计算机的出现与发展,使科学计算在科研与工程设计中的作用越发显得重要。由试验向计算的重大转变,也促使一些边缘学科相继产生并获得蓬勃发展,例如计算物理、计算力学、计算化学、计算生物学、计算经济学等各个领域,计算机上使用的数值计算方法已浩如烟海。

一般而言,用电子计算机进行科学计算,解决实际问题,其基本的过程如下:



由此可见,计算方法是处于一种承上启下的地位,是科学计算中不可缺少的重要环节。

计算方法研究的对象就是在计算机上求解各种数学模型实用、有效的方法并进行方法的有关理论分析。具体来说,计算方法内容中最基本的有插值法、数值积分与数值微分、数值求解线性代数方程组、数值求解非线性方程、矩阵特征值问题的计算、常微分方程数值解法以及偏微分方程数值解法等等。对于每一种数学问题的数值解法,计算方法除了要讲明它的基本原理、基本公式外,还必须研究它的具体实施方法(算法)以及近似解与准确解之间的近似程度(误差分析)。

二、算法的结构

为了能设计出优良的算法,方便阅读和交流,计算机专家们提出了算法结构的思想。算法的基本结构可分成三种,即顺序结构、分支结构和循环结构。三种基本结构可以组合成其

它各种各样结构的算法。下面分别举例说明。

例 1.1 顺序结构(或称简单结构)算法

已知 $\triangle ABC$ 三条边的长度 a、b、c,用海伦公式计算三角形面积 S。

算法 1.1

第一步:输入三角形三条边的长度:a、b、c

第二步:计算三角形的半周长:

$$p=\frac{a+b+c}{2}$$

第三步:计算三角形面积:

$$S=\sqrt{p(p-a)(p-b)(p-c)}$$

第四步:输出计算结果 S,结束。

例 1.2 分支结构算法

用求根公式求一元二次方程 $ax^2+bx+c=0$ 的根。

算法 1.2

第一步:输入二次项的系数,一次项的系数和常数项:a、b、c

第二步:计算判别式的值:$p=b^2-4ac$

第三步:判断:如果 $p\geqslant 0$,则

$$x_1=\frac{-b-\sqrt{p}}{2a},x_2=\frac{-b+\sqrt{p}}{2a}$$

否则,

$$x_1=\frac{-b-i\sqrt{-p}}{2a},x_2=\frac{-b+i\sqrt{-p}}{2a}$$

第四步:输出方程的两个根 x_1、x_2,结束。

例 1.3 循环结构算法

Fibonacci 数列$\{F_n\}$的通项满足递推公式:

$$F_n=F_{n-1}+F_{n-2}\qquad (F_1=1,F_2=1)$$

求该数列的前 20 项。

算法 1.3

第一步:置初值 $k=2,F_1=1,F_2=1$

第二步:$k\leftarrow k+1,F_k=F_{k-1}+F_{k-2}$

第三步:判断,若 $k<20$ 则转第二步,否则输出 $F_k(k=1,2,\cdots,20)$,结束。

由上面的例子可知,顺序结构算法按顺序执行每一步操作;分支结构算法带有逻辑判断条件,当条件成立时执行某些操作,条件不成立时则执行另一些操作;而循环结构算法带控制条件,当条件成立时重复执行循环体内的操作,条件不成立时退出循环体。一个解决实际问题的算法往往是由计算机可执行的基本操作(+、−、×、÷、置数、逻辑判断以及调用内部函数等)按照顺序、分支、循环这三种基本结构组成。

三、算法的描述方式

算法可以用解题操作步骤来表达，也可以用计算机程序来表示，还可以用算法流程图或框图来描述。由于算法流程图或框图直观、方便、便于交流，特别是在设计算法时能较好地考虑算法执行时的动态性，因而为大多数人所使用。例如：

设计算法用以计算 n 次多项式

$$p_n(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$$

在自变量 x 处的值。

在计算机里，由于加减操作的机时比乘除操作的机时要少得多，因而机时主要是进行乘除法的运算。如果直接逐项求和，则需要 $\frac{n(n+1)}{2}$ 次乘法。为节省计算量，现介绍著名的秦九韶算法，以 4 次多项式为例说明算法思想。

$$\begin{aligned} P_4(x) &= a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 \\ &= a_0 + x(a_1 + x(a_2 + x(a_3 + xa_4))) \end{aligned}$$

上式右端有三层括号，规定内层括号的计算优先，即最先计算 $a_3 + xa_4$ 的值，求一个四次多项式只需四次乘法，同理，求 n 次多项式只需 n 次乘法，大大节省了运算量。

$$\begin{cases} S_n = a_n \\ S_{k-1} = a_{k-1} + xS_k, \quad (k = n, n-1, \cdots, 1) \\ P_n(x) = S_0 \end{cases}$$

S_0 即为所求 n 次多次式 $P_n(x)$ 的值，算法框图见图 1.1。

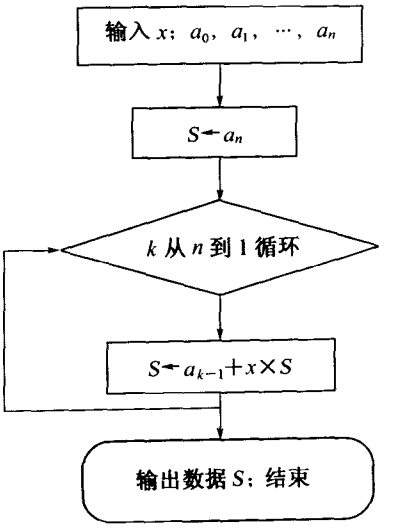


图 1.1　秦九韶算法

也可以按解题步骤方法，描述如下：

算法 1.4　秦九韶算法

第一步：输入多项式的系数 $a_0, a_1, \cdots, a_n$ 及自变量的值 x

第二步：置 $k \leftarrow n; S \leftarrow a_n$

第三步：计算 $S \leftarrow a_{k-1} + xS$

第四步：判断，若 $k > 1$，则置 $k \leftarrow k-1$，转第三步；否则，输出 S，结束。

第二节　误差及其相关概念

一、误差来源和分类

在科学与工程计算中，估计计算结果的精确度是十分重要的，而影响精确度的是各种各样的误差。误差按照它们的来源可分为模型误差、观测误差、截断误差和舍入误差四种。

(1) 模型误差：由实际问题抽象、简化为数学问题(建立数学模型) 时所引起的误差；

(2) 观测误差：测量工具的限制或在数据的获取时随机因素所引起的物理量的误差；

(3) 截断误差：又称为方法误差，用数值方法求解数学模型时，得到的正确解与模型准确解之间的误差，通常是用有限过程替代无限过程所引起，例如，利用 Taylor 公式，函数 e^x 可表示为：

$$e^x = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} + \frac{x^{n+1}}{(n+1)!}e^{\theta x}, \quad 0 < \theta < 1$$

对给定的 x，计算相应的函数值 e^x 时，可采用近似公式：

$$e^x \approx I = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!}$$

此近似公式的截断误差为

$$R = e^x - I = \frac{x^{n+1}}{(n+1)!}e^{\theta x}, \quad 0 < \theta < 1$$

由于科学计算经常要把一些数学函数变成计算机易于处理的形式，这样总会产生截断误差，因此，截断误差是计算方法主要研究的误差，很多好的数值计算方法都是巧妙地处理截断误差得出的，截断误差主要涉及到计算方法的收敛性。

(4) 舍入误差：由于计算机所表示的数的位数有限，通常会按舍入原则取近似值，由此引起的误差。如用 3.14159 作为 π 的近似值产生的误差就是舍入误差。值得注意的是少量运算的舍入误差一般是微不足道的，但在计算机上要完成千百万次运算后舍入误差的积累就可能是惊人的，不可忽视的。

根据误差来源，可以将误差分为两大类：一类是固有误差，包括模型误差和观测误差；另一类是在计算过程中引起的计算误差，包括截断误差和舍入误差，计算方法所考虑的主要是计算误差。

二、绝对误差与相对误差

定义 1.1　设 x 是精确值 x^* 的一个近似值，记

$$e = x^* - x$$

称 e 为近似值 x 的绝对误差，简称误差，绝对误差 e 可正可负。

如果 ε 为 $|e|$ 的一个上界，即

$$|e| \leqslant \varepsilon$$

则称 ε 为近似值 x 的绝对误差限，简称误差限。

例如，$x = 1.414$ 作为无理数 $\sqrt{2}$ 的一个近似值，它的绝对误差为

$$e = \sqrt{2} - 1.414$$

易知 $|e| \leqslant 0.00022$

故 $x = 1.414$ 作为 $x^* = \sqrt{2}$ 的近似值，它的一个绝对误差限为 $\varepsilon = 0.00022$。

用绝对误差来刻划一个近似值的准确程度是有局限性的，如测量 100m 和 1m 两个长度，若它们的绝对误差都是 1cm，显然前者的测量结果比后者的准确。因而除了要考虑绝对误差的大小外，还需要考虑该量本身的大小，为此引入相对误差的概念。

定义 1.2 设 x 是精确值 x^* 的一个近似值，记

$$e_r = \frac{e}{x^*} = \frac{x^* - x}{x^*}$$

称 e_r 为近似值 x 的相对误差，由于 x^* 未知，实际使用时总是将 x 的相对误差取为

$$e_r = \frac{e}{x} = \frac{x^* - x}{x}$$

$|e_r|$ 的上界，即：

$$\varepsilon_r = \frac{\varepsilon}{|x|}$$

称为近似值 x 的相对误差限，显然有 $|e_r| \leqslant \varepsilon_r$。

例 1.4 设 $x = 2.18$ 是由精确值 x^* 经过四舍五入得到的近似值，问 x 的绝对误差限 ε 和相对误差限 ε_r 分别是多少？

解 根据四舍五入原则，应有

$$x - 0.005 \leqslant x^* \leqslant x + 0.005$$

即 $|x^* - x| \leqslant 0.005$

所以 $\varepsilon = 0.005$

$$\varepsilon_r = \frac{\varepsilon}{|x|} = \frac{0.005}{2.18} \approx 0.23\%$$

凡是由精确值经过四舍五入得到的近似值，其绝对误差限等于该近似值末位的半个单位。

三、有效数字

设数 x 是数 x^* 的近似值，如果 x 的绝对误差限是它的某一数位的半个单位，并且从 x 左起第一个非零数字到该数位共有 n 位，则称这 n 个数字为 x 的有效数字，也称用 x 近似 x^* 时具有 n 位有效数字。

例如，$\pi = 3.14159265\cdots$，可用 3.14 或 3.1416 等作为它的近似值，而

$$|\pi - 3.14| = 0.00159265\cdots \leqslant \frac{1}{2} \times 10^{-2}$$

$$|\pi - 3.1416| = 0.00000734\cdots \leqslant \frac{1}{2} \times 10^{-4}$$

故 π 的近似值 3.14 及 3.1416 分别有三位及五位有效数字。而 π 的近似值 3.1415 只有四位有效数字(请读者思考,这是为什么)。

为了更具体地描述有效数字的概念,我们将近似值 x 写成规格化形式

$$x = \pm 0.a_1 a_2 \cdots a_n \cdots a_l \times 10^m \tag{1.1}$$

式中 a_1 是 1 到 9 之间的一个正整数;$a_2, \cdots, a_l$ 都是 0 到 9 之间的自然数;m 为整数。

定义 1.3 若准确值 x^* 的近似值 x 的绝对误差满足不等式

$$|x^* - x| \leqslant \frac{1}{2} \times 10^{m-n} \qquad (1 \leqslant n \leqslant l) \tag{1.2}$$

则称近似值 x 有 n 位有效数字。

例如,上面讲到的 π 的近似值 3.14、3.1416 及 3.1415 的规格化形式分别为:0.314×10^1、0.31416×10^1 及 0.31415×10^1,它的误差的绝对值分别满足不等式

$$|\pi - 3.14| \leqslant \frac{1}{2} \times 10^{1-3}$$

$$|\pi - 3.1416| \leqslant \frac{1}{2} \times 10^{1-5}$$

$$|\pi - 3.1415| \leqslant \frac{1}{2} \times 10^{1-4}$$

故它们分别有三位、五位及四位有效数字。

四、有效数字与误差限的关系

设近似值 x 有 n 位有效数字,由式(1.2)可见,在 m 相同的情况下,n 越大则 10^{m-n} 越小,即有效数字的位数越多,绝对误差限则越小。对于有效数字与相对误差限之间的关系,有下面的定理。

定理 1.1 若用规格化形式(1.1)表示的近似值 x 具有 n 位有效数字,则其相对误差限为 $\frac{1}{2a_1} \times 10^{-n+1}$,即

$$|e_r| \leqslant \frac{1}{2a_1} \times 10^{-n+1} \tag{1.3}$$

证明 由式(1.1)可得

$$a_1 \times 10^{m-1} \leqslant |x| \leqslant (a_1 + 1) \times 10^{m-1} \tag{1.4}$$

联系式(1.2)可得

$$|e_r| = \frac{|x^* - x|}{|x|} \leqslant \frac{\frac{1}{2} \times 10^{m-n}}{a_1 \times 10^{m-1}} = \frac{1}{2a_1} \times 10^{-n+1}$$

由此可见,只要知道了近似值 x 的有效数字的位数 n 和第一个非零数字 a_1,就能估计出它的相对误差限。反之,还可以从近似值的相对误差限来估计有效数字的位数。

定理 1.2 若近似值 x 的相对误差限为

$$|e_r| \leqslant \frac{1}{2(a_1 + 1)} \times 10^{-n+1} \tag{1.5}$$

则它至少具有 n 位有效数字。

证明 由于

$$|e| = |x^* - x| = |x| \cdot |e_r|$$

根据式(1.4)及式(1.5),可得

$$|e| = |x| \cdot |e_r| \leqslant \frac{1}{2} \times 10^{m-n}$$

故 x 有 n 位有效数字。

例 1.5 为使 $\sqrt{20}$ 的近似数的相对误差限不超过 0.1%,问至少要取多少位有效数字。

解 由定理 1.1

$$|e_r| \leqslant \frac{1}{2a_1} \times 10^{-n+1}$$

因为 $\sqrt{20}$ 的第一个非零数字 $a_1 = 4$ 即

$$|e_r| \leqslant \frac{1}{2 \times 4} \times 10^{-n+1} \leqslant 0.1\%$$

故取 $n = 4$ 即可满足,也就是说只需 $\sqrt{20}$ 的近似值具有 4 位有效数字,其相对误差限就不超过 0.1%。

第三节 算法设计与选择时应遵循的若干原则

在计算机上进行数值运算时,由于计算机的字长有限,只能保留有限位有效数字,因而每一步计算都可能产生误差。比如计算过程中的舍入误差,在反复多次的计算过程中,将产生误差的传播和积累,当误差积累过大时,会导致计算结果失真。因而,基于减少舍入误差的影响等原因,算法设计与选择时应遵循如下一些原则。

一、避免两个相近的数相减

在数值计算中,两个相近的数相减会使有效数字严重损失,例如,$\sqrt{1001} \approx 31.64$ 与 $\sqrt{1000} \approx 31.62$ 都具有四位有效数字,但它们的差 $\sqrt{1001} - \sqrt{1000} \approx 31.64 - 31.62 = 0.02$ 至多具有一位有效数字,事实上 $\sqrt{1001} - \sqrt{1000}$ 的准确值是 0.015807437…,可见相减的结果有一位有效数字,从而误差很大,严重影响了计算结果的精度。这说明必须尽量避免这种运算,改变计算公式可以防止这种情形的出现。可把算式变形为

$$\sqrt{1001} - \sqrt{1000} = \frac{1}{\sqrt{1001} + \sqrt{1000}} \approx 0.01581$$

此例说明,在相近两数相减时,有时可通过改变计算公式以避免或减少有效数字的损失,例如,

(1)x_1 与 x_2 接近时,$\ln x_1 - \ln x_2 = \ln \frac{x_1}{x_2}$

(2)$x > 0$ 很大时,$\sqrt{x+1} - \sqrt{x} = \frac{1}{\sqrt{x+1} + \sqrt{x}}$

(3)x 的绝对值很小时，$1-\cos x=2\sin^2\frac{x}{2}$

$$e^x-1\approx x+\frac{1}{2!}x^2+\frac{1}{3!}x^3+\cdots+\frac{1}{n!}x^n$$

有些计算如无法改变算式，可采用增加有效数字位数再进行运算；或在计算机上采用双倍字长运算。

二、绝对值太小的数不宜作除数

一方面，在计算机上若用绝对值很小的数作除数可能会造成溢出；另一方面，绝对值很小的除数稍有一点误差时，对计算结果的影响很大，例如：

$$\frac{2.7182}{0.001}=2718.2$$

如分母变为 0.0011，即分母只有 0.0001 的变化时，

$$\frac{2.7182}{0.0011}\approx 2471.1$$

商却引起了很大的变化。因此，在计算过程中不仅要避免两个相近的数相减，更要避免再用这种差作为除数。

三、防止“大数”吃掉“小数”

参加计算的数，有时数量级相差很大，如果不注意采取相应措施，在它们的加、减运算中，绝对值很小的数往往会被绝对值较大的数“吃掉”，造成计算结果失真，这主要是由计算机表示的数位数有限这一客观现实引起的。

例如：$a=10^{13}$，$b=5$，设想这两个数在具有 12 位浮点数计算机系统中相加，在机器数系统中相加的原则是先对阶，后相加，对阶时，

$a+b=10^{13}+5=1.00000000000\times10^{13}+0.00000000000\boxed{05}\times10^{13}$，由于系统只保留前 12 位作为有效数，方框中数据被舍去，实际加法操作如下：

$$1.00000000000\times10^{13}+0.00000000000\times10^{13}$$

最后 $a+b$ 的计算结果是 $1.00000000000\times10^{13}$，即 a 的值作为计算结果赋给了 $a+b$，这显然是不合理的。

由于“大数吃小数”现象，考虑当绝对值悬殊的一系列数相加时，如果有

$$|x_1|>|x_2|>|x_3|>\cdots>|x_n|$$

成立，则应该按绝对值由小到大的顺序确定累加的先后次序。

四、注意简化计算过程，减少运算次数

同样一个计算问题，若选用的计算公式简单，运算次数少，可以减少舍入误差的传播影响，还可节省大量的计算机时，提高计算速度。例如，前面例 1.4 中介绍的秦九韶算法就是一个典型的例子。再如，设 A、B、C 分别为 10×20、20×50、50×1 的矩阵，计算 $D=ABC$ 可有如下不同的算法：

算法 1 $D=(AB)C$，需要做 10500 次乘法；

算法 2 $D=A(BC)$，需要做 1200 次乘法。

显然算法 2 的计算量比算法 1 小，因而算法 2 比算法 1 要来得好。

五、选用数值稳定性好的算法

一种数值算法，如果其计算舍入误差积累是可控制的，则称其为数值稳定的，反之称为数值不稳定的，数值不稳定的算法没有实用价值。

例 1.6 利用递推式计算定积分 $I_n = e^{-1}\int_0^1 x^n e^x dx (n = 0,1,2,\cdots,20)$ 的值。

解

$$I_0 = e^{-1}\int_0^1 e^x dx = e^{-1}(e-1) = 1 - e^{-1}$$

$$I_n = e^{-1}\int_0^1 x^n e^x dx = e^{-1}\left(x^n e^x \Big|_0^1 - n\int_0^1 x^{n-1} e^x dx\right) = 1 - nI_{n-1}$$

由此可得带初值的递推关系式

$$\begin{cases} I_0 = 1 - e^{-1} \\ I_n = 1 - nI_{n-1} \qquad (n = 1,2,\cdots) \end{cases}$$

直接用 $I_0 = 1 - e^{-1}$ 在 15 位有效数的计算机上计算得

$$I_0 \approx 0.63212055882856$$

利用递推式逐次计算可得 20 个数据如下(表 1.1)：

表 1.1

S_1	0.36787944117144	S_{11}	0.07735222935878
S_2	0.26424111765712	S_{12}	0.07177324769464
S_3	0.20727664702865	S_{13}	0.06694777996972
S_4	0.17089341188538	S_{14}	0.06273108042387
S_5	0.14553294057308	S_{15}	0.05903379364190
S_6	0.12680235656152	S_{16}	0.05545930172957
S_7	0.11238350406936	S_{17}	0.05719187059731
S_8	0.10093196744509	S_{18}	−0.02945367075154
S_9	0.09161229299417	S_{19}	1.55961974427919
S_{10}	0.08387707005829	S_{20}	−30.19239488558378

表中 $S_1,\cdots,S_{20}$ 是积分值 $I_1,\cdots,I_{20}$ 的计算近似值。由于被积函数非负，对正整数 n，有 $I_n \geqslant 0$ 成立。但是表 1.1 中有两个负数 S_{18} 和 S_{20}，这显然是错误的，导致这一错误的直接原因是初始数据的误差，正向递推公式使得误差在计算过程中增大。

考虑另一种算法，由递推公式 $I_n = 1 - nI_{n-1}$ 解得 $I_{n-1} = \frac{1}{n}(1 - I_n)$，这是逆向递推公式，对 I_n 的值作估计，有

$$I_n = e^{-1}\int_0^1 x^n e^x dx \leqslant e^{-1}e^1\int_0^1 x^n dx = \frac{1}{n+1}$$

取 $I_{30} \approx S_{30} = \frac{1}{31}$，利用递推公式

$$S_{n-1}=\frac{1}{n}(1-S_n),\quad n=30,29,28\cdots,2$$

计算出 $S_{29},S_{28},\cdots,S_1$，并选取前 20 个数据，列表如下(表 1.2)：

表 1.2

S_1	0.36787944117144	S_{11}	0.07735222886266
S_2	0.26424111765712	S_{12}	0.07177325364803
S_3	0.20727664702865	S_{13}	0.06694770257562
S_4	0.17089341188538	S_{14}	0.06273216394138
S_5	0.14553294057308	S_{15}	0.05901754087930
S_6	0.12680235656153	S_{16}	0.05571934593124
S_7	0.11238350406930	S_{17}	0.05277111916899
S_8	0.10093196744559	S_{18}	0.05011985495809
S_9	0.09161229298966	S_{19}	0.04772275579621
S_{10}	0.08387707010339	S_{20}	0.04554488407582

表 1.2 中没有出现负数，数据变化表明实数序列$\{S_n\}$随n的增加单调下降趋于零，这与积分值数列$\{I_n\}$的变化规律相吻合。

前一种算法用正向递推，尽管初值的近似值精确度很高，但是在计算过程中造成了误差扩散，这是数值不稳定的算法。

后一种算法用逆向递推，虽然初始数据 S_{30} 带有明显误差，但计算结果却比较可靠，由误差传播规律

$$E_{n-1}=I_{n-1}-S_{n-1}=\frac{1}{n}(1-I_n)-\frac{1}{n}(1-S_n)=-\frac{1}{n}E_n\qquad(n=30,29,\cdots,2)$$

可知，逆向递推计算过程中初始数据误差 E_{30} 的传播规律为

$$|E_{n-1}|=\frac{1}{n(n+1)\cdots29\cdot30}|E_{30}|\qquad(n=29,28,\cdots,2)$$

所以，逆向递推使误差绝对值逐次减小，故这是一种数值稳定的算法。

在实际应用中应选用数值稳定的公式，避免使用数值不稳定的公式。

习题 1

1. 若 $x=3587.64$ 是 x^* 的具有六位有效数字的近似值，求 x 的绝对误差限。

2. 若用 $x=2.72$ 来表示 e 具有三位有效数字的近似值，求 e 的相对误差限。

3. 为使积分 $I=\int_0^1 e^{-x^2}dx$ 的近似值的相对误差不超过 1%，问至少要取几位有效数字？

4. 在下面计算 y 的公式中哪一个算得准？为什么？

(1) 已知$|x|\ll1$

$(a)y=\frac{1}{1+2x}-\frac{1-x}{1+x}$　　$(b)y=\frac{2x^2}{(1+2x)(1+x)}$