

经典数值算法完全解析



C# Scientific Computing in C#

科学计算讲义

宋叶志 徐导 何峰 编著

- 囊括目前常用的科学计算方法
- 所有方法给出算法分析及代码实现
- 采用LaTeX排版，版面清爽
- 作者在线答疑互动

21 0.283648
22 0.283648
23 0.283648
24 0.283648
25 0.283648



附光盘



人民邮电出版社
POSTS & TELECOM PRESS



C#

Scientific Computing in C#

科学计算讲义

宋叶志 徐导 何峰 编著

人民邮电出版社
北京

图书在版编目 (CIP) 数据

C#科学计算讲义 / 宋叶志, 徐导, 何峰编著. -- 北京: 人民邮电出版社, 2012.12
ISBN 978-7-115-29401-2

I. ①C… II. ①宋… ②徐… ③何… III. ①C语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2012)第218258号

内 容 提 要

本书较为详细地介绍了科学计算方法, 并对算法给出了源代码。关于算法部分主要介绍了线性方程组的迭代解法与直接解法、正交变换与最小二乘计算方法、鲁棒估计、随机数的产生、插值法、非线性方程求解、多元非线性最优化算法、微分方程数值方法等内容。

本书还给出了 C#程序设计的基本方法, 并对科学计算中要用到的矩阵向量类的构造做了详细阐述。算法的实现本身不限于具体的语言, 本书对于算法的描述是较为详细的, 所以读者也很容易把算法改用 Fortran、MATLAB、C++、Java 等语言编程实现。

本书适合作为大学理工科本科生或研究生计算方法、数值分析课程的教材或参考书。对于从事相关学科教学的教师, 如果不熟悉现代编程语言, 也可以选择本书作为工具书。本书还可以用作科研人员的工程计算工具书与算法集。另外, 在一些需要进行数据处理与分析的公司, 如数量金融、统计等行业, 也可以选用本书作为培训教材, 或直接应用书上的源代码进行软件开发。

C#科学计算讲义

◆ 编 著 宋叶志 徐 导 何 峰

责任编辑 王峰松

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号

邮编 100061 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京鑫正大印刷有限公司印刷

◆ 开本: 787×1092 1/16

印张: 29.25

字数: 730千字

2012年12月第1版

印数: 1-3500册

2012年12月北京第1次印刷

ISBN 978-7-115-29401-2

定价: 69.00元 (附光盘)

读者服务热线: (010)67132692 印装质量热线: (010)67129223

反盗版热线: (010)67171154

前 言

C# 语言的出现是程序设计语言中一件激动人心的事，它是目前完美编程语言的一个代表，C# 是微软.NET 系列最重要的程序设计语言。

在数值运算领域除了类似于 MATLAB、R 等这一类专门的数学软件之外，最重要的高级语言应该当属 Fortran，其次是 C++ 和 C 语言。以笔者的观点而言，如果直接用这三种语言编写程序，最方便的应该还是现代 Fortran 语言，最不方便的应该是 C 语言。在数值运算中，矩阵运算占有十分重要的地位，而 C 语言在处理这一问题上相当麻烦。然而由于 C 语言在程序发展的历程中非常重要，也有相当多的数值算法库是提供 C 语言版本的。

相比之下，C++ 在矩阵处理上也和 C 类似都较为麻烦，但是 C++ 不同于 C 的一个重要地方是，C++ 用户可以方便地定制自己的对象。所以，虽然它本身并不方便做数值运算，但是用户可以使它变得很方便，也许它本身并不安全，但是熟练的用户也可以使它变得安全。

在 C++ 中，动态矩阵是通过动态数组来实现分配，使用完以后需要逐步地把向量和矩阵分别释放，如果忘记释放将会出现内存泄漏的问题。但是 C++ 可以在自定义的类中采用构造函数来实现动态数组分配，而在类调用结束时释放类中所有需要释放的目标。

C# 吸纳了 Java 与 C++ 的优点，又摒弃了二者中一些容易犯错的地方，所以只要定制合适，C# 就会变得很适合编写数值程序。与 C++ 一样，如果只是简单地编写一个小的数值程序，C# 也许并不如 Fortran 方便，但是如果用户合理地创建自己的类，C# 可以“变得”比 Fortran 更适合做数值运算。

目前，使用 C# 编写数值运算的人还比较少，较多的都是用来编写商业软件。本书详细地给出了一些经典数值算法的 C# 实现方法，并且源代码免费向用户开放。笔者计划编写包含 Fortran、C# 和 C++ 的一系列算法，编写这些算法的目的不在于要出一两本书，而是建立一套较为全面的算法库。预设的算法库主要包括：数值代数、数值逼近与常微分方程、统计与数据处理、神经网络与机器学习、运筹与最优化、最优控制、衍生品计算。当然，以上的想法显然不是短期可以完成的。在编写新的算法系列的同时，也会不断地对旧的算法进行修订。当一个模块相对成熟以后，笔者就会把这些算法发布出来，供任何人免费使用，用于商业目的或者非商业目的都可以不经过我允许直接拿去使用。

需要说明的是，虽然本书的程序是免费向用户开放的，但是与以前一样的观点，本书并非是以提供一套商业程序作为出发点。我们提供算法的实现方法供读者参考，目的还是希望读者了解计算方法。读者可以任意使用或传播书中的程序，但是出版社和笔者不承担由于程序使用造成的各种后果的责任。

原先设想的书的内容要比现在多一些，以下内容起初准备列入本书的：数值逼近及 CAGD 中非常重要的 B 样条方法、神经网络数值逼近与识别、特殊函数、统计学中的部分内容、支持向量机统计学习理论等。这些内容笔者都先后编写了数量不少的算法，可惜这些算法都是用其他语言如 Fortran 等编写的。因时间和精力关系暂时没有把这些算法翻译成 C# 版，这无疑是一个遗憾。另外，笔者用 C# 编写了一些采用差分方法计算偏微分方程的程序，但是觉得这些程序不太成熟，而现在商用的许多有限元软件（如 ANSYS 等）都可以

满足用户对复杂偏微分方程求解的需求，所以在定稿时决定不把这部分内容放在本书中。不过从另一个角度而言，这些内容一般不属于经典数值方法范畴，所以，如果条件允许的话，笔者后续会单独编写书籍分别介绍这些内容。

本书不是一本 C# 语言参考书，而是偏重于算法并且给出了算法实现的书籍。如果需要深入学习 C# 语言，需要通过其他途径。在第 1 章中给出了基本的 C# 语法，至于够不够读者使用，需要读者根据自己情况判断。实际上我认为，对于熟悉 C++ 或 Java 的读者，通过第 1 章结合后面的章节，对于编写数值方法的语法已经足够了。

除本书之外，笔者拟编写以下一些算法书籍：

《Fortran 科学计算讲义》

《Python 科学计算讲义》

《MATLAB 科学计算讲义》

《MATLAB 最优化计算讲义》

《MATLAB 神经网络与支持向量机讲义》

由于笔者精力有限，这些算法不可能一次性编撰完成，所以打算以滚动方式出版，并在条件成熟时，对这些算法进行修订。编写这样一套算法库完全没有商业价值，但笔者认为这是有意义的事。一方面自己的工作中需要用到不同的算法，同时笔者把这些算法系统整理，可满足初学者的学习需求。

感谢我的家人！感谢课题组胡小工研究员、黄勇副研究员、陈国平以及单位的其他同事和部分研究生！全书由笔者用 $\text{\LaTeX}$ 排版，在使用 $\text{\LaTeX}$ 的过程中得到了许多网络上朋友的帮助。本书编辑王峰松老师熟悉 $\text{\LaTeX}$ ，这也使得本书能够较为顺利出版。

对本书而言，从范例选择、算法分析、程序设计到排版编译等过程，工作量甚为庞大，尤其是笔者要面对工作中其他繁重的科研任务，所以虽然笔者已经对本书进行了数次校对，但还会有不少不妥当甚至谬误之处，还请读者指正。读者可以通过 QQ(1142537589) 和笔者联系或交流。

本书的编写得到了上海市空间导航与定位技术重点实验室“月球卫星分析法轨道计算与星载算法优化”项目资助（编号：Y224353002），在此表示感谢。数值算法是数据处理的基本工具，本书的部分算法已经被笔者应用在一些工程项目中，借基金资助的契机，对常用算法进行了系统整理出版。

宋叶志

中国科学院空间飞行器精密定轨创新团组

目 录

引言	1
第 1 章 C# 程序设计基础	9
1.1 计算机、程序设计与算法	9
1.1.1 计算机结构	9
1.1.2 操作系统	10
1.1.3 机器语言与高级语言	10
1.1.4 程序设计与算法	10
1.2 C# 历史与概述	11
1.2.1 C 语言: 结构化编程语言的高峰	11
1.2.2 C++ 语言: 面向对象与大型程序	11
1.2.3 Java 语言: 可移植、安全性与 Internet	11
1.2.4 C#: .NET 主打语言	12
1.3 集成开发环境介绍	12
1.4 面向对象程序设计	16
1.4.1 封装	16
1.4.2 多态	16
1.4.3 继承	17
1.5 数据类型与运算符	17
1.5.1 简单数据类型	17
1.5.2 数组	17
1.5.3 运算符	17
1.5.4 赋值运算符	18
1.6 程序控制结构	18
1.6.1 顺序结构	18
1.6.2 分支结构	18
1.6.3 循环结构	20
1.6.4 控制结构的嵌套	21
1.7 类的设计及对象实现	21
1.7.1 定义类	22
1.7.2 创建对象	22
1.7.3 方法	22
1.7.4 构造函数	23
1.7.5 析构函数与垃圾回收	23
1.8 运算符重载及索引器	24
1.8.1 运算符重载	24
1.8.2 索引器	26

1.8.3 面向对象思想在 C# 程序设计中的重要性	27
1.9 GUI 编程	28
1.10 本章小结	31
第 2 章 线性方程组迭代解法	32
2.1 Jacobi 迭代法	32
2.1.1 基本原理	32
2.1.2 实验内容与数据	33
2.1.3 程序源代码	33
2.1.4 实验结论	37
2.2 Gauss-Seidel 迭代法	38
2.2.1 基本原理	38
2.2.2 实验内容与数据	39
2.2.3 程序源代码	39
2.2.4 实验结论	43
2.3 逐次超松弛迭代法	44
2.3.1 基本原理	44
2.3.2 实验内容与数据	44
2.3.3 程序源代码	45
2.3.4 实验结论	49
2.4 Richardson 迭代法	50
2.4.1 基本原理	50
2.4.2 实验内容与数据	50
2.4.3 程序源代码	50
2.4.4 实验结论	54
2.5 广义 Richardson 迭代法	55
2.5.1 基本原理	55
2.5.2 实验内容与数据	55
2.5.3 程序源代码	55
2.5.4 实验结论	60
2.6 Jacobi 超松弛迭代法	60
2.6.1 基本原理	60
2.6.2 实验内容与数据	61
2.6.3 程序源代码	61
2.6.4 实验结论	65
2.7 最速下降法	66
2.7.1 基本原理	66
2.7.2 实验内容与数据	66
2.7.3 程序源代码	67

2.7.4	实验结论	71
2.8	共轭梯度法	72
2.8.1	基本原理	72
2.8.2	实验内容与数据	72
2.8.3	程序源代码	72
2.8.4	实验结论	77
2.9	本章小结	77
第 3 章	线性方程组的直接解法	78
3.1	三角方程组	78
3.1.1	基本原理	78
3.1.2	实验内容与数据	79
3.1.3	程序代码	79
3.1.4	实验结论	83
3.2	高斯消去法	83
3.2.1	基本原理	83
3.2.2	实验内容与数据	84
3.2.3	程序源代码	84
3.2.4	实验结论	89
3.3	选主元消去法	90
3.3.1	基本原理	90
3.3.2	实验内容与数据	90
3.3.3	程序源代码	90
3.3.4	实验结论	96
3.4	Crout 分解	97
3.4.1	基本原理	97
3.4.2	实验内容与数据	98
3.4.3	程序源代码	98
3.4.4	实验结论	103
3.5	Doolittle 分解	103
3.5.1	基本原理	103
3.5.2	实验内容与数据	104
3.5.3	程序源代码	104
3.5.4	实验结论	108
3.6	追赶法计算三对角方程	109
3.6.1	基本原理	109
3.6.2	实验内容与数据	110
3.6.3	程序源代码	110
3.6.4	实验结论	114

3.7	行列式的计算	115
3.7.1	基本原理	115
3.7.2	实验内容与数据	115
3.7.3	程序源代码	115
3.7.4	实验结论	119
3.8	本章小结	120
第 4 章	正交变换与最小二乘计算方法	121
4.1	对称正定阵的 Cholesky 分解	121
4.1.1	基本原理	121
4.1.2	实验内容与数据	122
4.1.3	程序源代码	122
4.1.4	实验结论	126
4.2	不开平方的 Cholesky 分解	127
4.2.1	基本原理	127
4.2.2	实验内容与数据	127
4.2.3	程序源代码	127
4.2.4	实验结论	132
4.3	QR 分解之 Householder 镜像变换方法	133
4.3.1	基本原理	133
4.3.2	实验内容与数据	134
4.3.3	程序源代码	134
4.3.4	实验结论	140
4.4	修正的 Gram-Schmidt 正交化方法	141
4.4.1	基本原理	141
4.4.2	实验内容与数据	142
4.4.3	程序源代码	142
4.4.4	实验结论	147
4.5	求解法方程计算最小二乘问题	147
4.5.1	基本原理	147
4.5.2	实验内容与数据	149
4.5.3	程序源代码	149
4.5.4	实验结论	157
4.6	QR 分解法计算最小二乘问题	158
4.6.1	基本原理	158
4.6.2	实验内容与数据	159
4.6.3	程序源代码	159
4.6.4	实验结论	167
4.7	加权最小二乘与 Gauss-Markov 估计	167

4.7.1	基本原理	167
4.7.2	实验内容与数据	169
4.7.3	程序源代码	169
4.7.4	实验结论	178
4.8	具有先验信息的贝叶斯估计	178
4.8.1	基本原理	178
4.8.2	实验内容与数据	179
4.8.3	程序源代码	179
4.8.4	实验结论	189
4.9	工程应用中最小二乘法的实用方法	191
4.10	本章小结	192
第 5 章	鲁棒估计	193
5.1	M 估计的 IGGI 方案	193
5.1.1	基本原理	193
5.1.2	实验内容与数据	194
5.1.3	程序源代码	196
5.1.4	实验结论	208
5.2	Hampel 函数作标准等价权	210
5.2.1	基本原理	210
5.2.2	实验内容与数据	210
5.2.3	程序源代码	212
5.2.4	实验结论	224
5.3	Huber 估计	227
5.3.1	基本原理	227
5.3.2	实验内容与数据	227
5.3.3	程序源代码	229
5.3.4	实验结论	241
5.4	本章小结	243
第 6 章	随机数	244
6.1	乘同余法均匀分布随机数发生器	244
6.1.1	基本原理	244
6.1.2	实验内容与数据	244
6.1.3	程序源代码	244
6.1.4	实验结论	248
6.2	混合同余法均匀分布随机数发生器	249
6.2.1	基本原理	249
6.2.2	实验内容与数据	249
6.2.3	程序源代码	249

6.2.4 实验结论	253
6.3 正态分布随机数	253
6.3.1 基本原理	253
6.3.2 实验内容与数据	254
6.3.3 程序源代码	254
6.3.4 实验结论	261
6.4 蒙特卡罗方法介绍	261
6.4.1 基本原理	261
6.4.2 实验内容与数据	262
6.4.3 程序源代码	262
6.4.4 实验结论	265
6.5 本章小结	265
第 7 章 插值法	266
7.1 拉格朗日插值	266
7.1.1 基本原理	266
7.1.2 实验内容与数据	266
7.1.3 程序源代码	266
7.1.4 实验结论	270
7.2 牛顿插值法	271
7.2.1 基本原理	271
7.2.2 实验内容与数据	271
7.2.3 程序源代码	271
7.2.4 实验结论	276
7.3 Hermite 插值法	276
7.3.1 基本原理	276
7.3.2 实验内容与数据	277
7.3.3 程序源代码	277
7.3.4 实验结论	281
7.4 本章小结	281
第 8 章 非线性方程数值解法	282
8.1 Picard 迭代法	282
8.1.1 基本原理	282
8.1.2 实验内容与数据	283
8.1.3 程序源代码	283
8.1.4 实验结论	285
8.2 牛顿迭代法	285
8.2.1 基本原理	285
8.2.2 实验内容与数据	286

8.2.3	程序源代码	286
8.2.4	实验结论	289
8.3	割线法	289
8.3.1	基本原理	289
8.3.2	实验内容与数据	290
8.3.3	程序源代码	290
8.3.4	实验结论	293
8.4	重根时的迭代改进	293
8.4.1	基本原理	293
8.4.2	实验内容与数据	294
8.4.3	程序源代码	294
8.4.4	实验结论	297
8.5	应用范例：债券到期收益率的计算	297
8.5.1	基本原理	297
8.5.2	实验内容与数据	298
8.5.3	程序源代码	298
8.5.4	实验结论	304
8.6	本章小结	304
第 9 章	非线性最优化	305
9.1	一维搜索之黄金分割法	305
9.1.1	基本原理	305
9.1.2	实验内容与数据	306
9.1.3	程序源代码	306
9.1.4	实验结论	310
9.2	连续抛物线插值法	311
9.2.1	基本原理	311
9.2.2	实验内容与数据	312
9.2.3	程序源代码	312
9.2.4	实验结论	316
9.3	多维非线性最优化牛顿下山法	317
9.3.1	基本原理	317
9.3.2	实验内容与数据	318
9.3.3	程序源代码	318
9.3.4	实验结论	325
9.4	最速下降法	327
9.4.1	基本原理	327
9.4.2	实验内容与数据	327
9.4.3	程序源代码	327

9.4.4 实验结论	333
9.5 变尺度之 DFP 方法	333
9.5.1 基本原理	333
9.5.2 实验内容与数据	335
9.5.3 程序源代码	335
9.5.4 实验结论	341
9.6 拟牛顿之 BFGS 方法	341
9.6.1 基本原理	341
9.6.2 实验内容与数据	342
9.6.3 程序源代码	342
9.6.4 实验结论	349
9.7 本章小结	349
第 10 章 常微分方程(组)的数值方法	350
10.1 经典 Rung-Kutta 方法	350
10.1.1 基本原理	350
10.1.2 实验内容与数据	351
10.1.3 程序源代码	351
10.1.4 实验结论	353
10.2 Gill 方法	354
10.2.1 基本原理	354
10.2.2 实验内容与数据	355
10.2.3 程序源代码	355
10.2.4 实验结论	357
10.3 Rung-Kutta 方法计算微分方程组	358
10.3.1 基本原理	358
10.3.2 实验内容与数据	359
10.3.3 程序源代码	359
10.3.4 实验结论	363
10.4 Adams-Bashforth 三步三阶方法	364
10.4.1 基本原理	364
10.4.2 实验内容与数据	365
10.4.3 程序源代码	365
10.4.4 实验结论	371
10.5 Adams-Bashforth 四步四阶方法	372
10.5.1 基本原理	372
10.5.2 实验内容与数据	372
10.5.3 程序源代码	372
10.5.4 实验结论	379

10.6 三阶 Adams 预测校正方法 (PECE)	380
10.6.1 基本原理	380
10.6.2 实验内容与数据	381
10.6.3 程序源代码	381
10.6.4 实验结论	387
10.7 四阶 Adams 预测校正方法 (PECE)	388
10.7.1 基本原理	388
10.7.2 实验内容与数据	389
10.7.3 程序源代码	389
10.7.4 实验结论	396
10.8 辛结构与哈密顿系统的辛算法介绍	397
10.8.1 基本原理	397
10.8.2 实验内容与数据	400
10.8.3 程序源代码	400
10.8.4 实验结论	405
10.9 本章小结	406
附录 A C# 数值代数类的抽象与设计	408
附录 B 动态链接库与混合编程	428
B.1 静态链接库与动态链接库	428
B.2 C# 调用 Fortran 动态链接库范例	428
B.3 调用可执行函数	433
附录 C Linux 下 C# 开发与跨平台编程介绍	444
C.1 Mono 简介	444
C.2 Linux 下 C#IDE 开发范例	444
参考文献	454

引 言

★科学计算方法有何应用？

科学计算方法在自然科学研究中起基础性作用，诚如许多学者指出，科学计算和理论研究及实验共同构成当代科学研究的三大基本支柱。

图 1~ 图 6 列出了几个常见的应用领域。实际上，科学计算对现代生活的影响远远不止图示中的这些领域。

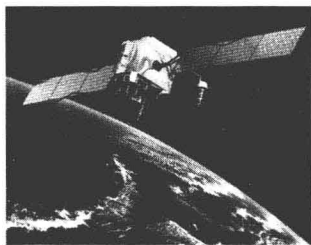


图 1 航天科技

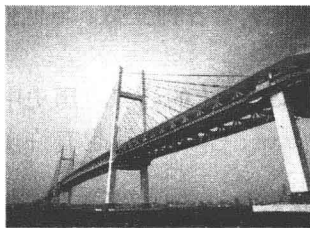


图 2 大型建筑结构力学

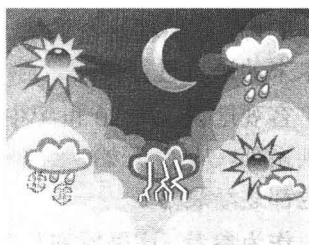


图 3 数值气象预报

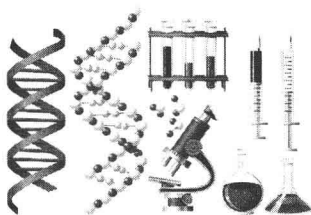


图 4 生物、医学统计



图 5 数量金融工程

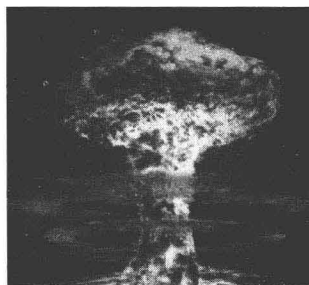


图 6 高能试验

当今社会科技日益发展，而许多高新技术都离不开科学计算的支持。20 世纪中叶以来，计算机软硬件的发展为科学计算提供了有利条件。目前，各国科技领域都非常重视科学计算理论与技术的发展。

历史上，数学发达中心的迁移，同社会政治、经济重心的迁移基本上是吻合的。纵观文艺复兴时期，数学中心从意大利到英、法、德的转移，基本同时伴随着这些帝国的盛衰。二战以后，德国战败，美、前苏联取代德国的哥廷根成为世界数学重地。现今，美国等西方国家在科学、经济领域相对于发展中国家长期处于领先地位，其科学计算水平基本也与之相符。

在不严格的情况下，有时我们不去区分计算方法、科学计算、数值分析几个名词的差别。目前，我国理工科院校的计算数学、计算机等专业在本科生就开设了数值分析课程。而绝大部分院校理工科研究生和博士生都开设了计算方法的课程。科学计算对于个人而言，也是从事科研工作或工程技术的一项基本技能。

科学计算具有一般性的理论意义，并不局限于某具体工程技术，而许多具体的科学实践反过来对促进科学计算的发展又提出了新的课题和需求，从而两者互相促进。

★该选择何种编程语言？

长久以来，人们总是热衷于各种排名及其对未来的预测。许多预测都曾出现过传奇般的精确断言，然而我们往往搞不清是他们使用的方法有效，还是他们的断言仅仅是偶然。英国金融学家 Wilmott 认为：“惨败者倾向于保持沉默。”

关于数值计算领域的语言问题，我们曾先后出版了 MATLAB、Fortran 等版本的书籍，似乎使用每种语言都有自己的原因。在 *Numerical Recipes* 中，笔者用诙谐的语气提到在过去的 20 年中，他们每次对未来编程语言的预测都是错误的，并且没有一次是正确的。该系列以前有过 Fortran、Pascal 等版本，现在有些版本笔者已经停止修订。

的确，在编程这个领域一直都充满着各种变数。与他们的观点一致，我们这里也不妄自对未来的语言进行预测。

客观地说，从 20 世纪中叶到目前为止的计算机历史中，编程语言影响力最大的还是 C 家族系列，从 C 开始，到 C++、Java、C#，中途也有许多曾经被认为很有前途的语言，后来也都逐渐衰落。

作为参考，这里罗列几个主流语言的 TIOBE 指数从 2001—2012 年间的变化情况。图 7 为 C# 语言 TIOBE 指数近年变化情况，显然呈稳步上升的趋势。

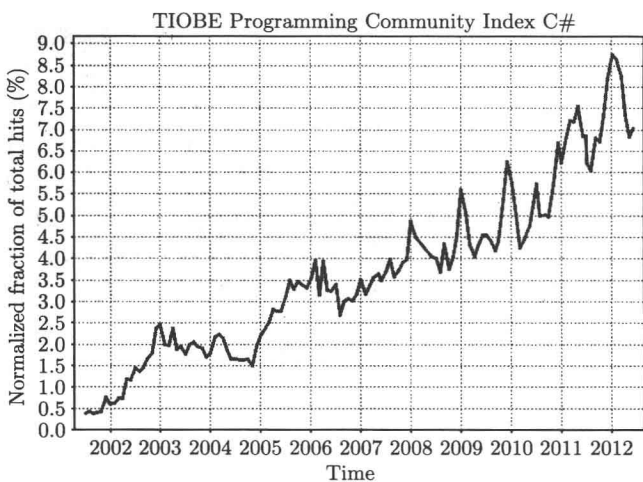


图 7 C# 语言 2001—2012 年间 TIOBE 编程语言指数

图 8~图 11 则分别为 C、C++、Java、Python 几门语言的变化趋势。这几门语言都有一定的起伏。当然，这也不能说明 C# 相对于以上的其他几门语言有绝对的优势。

应当多角度地看待数据变化，以 C++ 语言为例，在面向对象的早期语言中，C++ 几乎处于绝对的优势地位，其使用基础人群极为庞大。而今天，编程者可选择的语言呈现明显的多样性，这种绝对优势也就不再明显，但 C++ 依然基本保持在最热门的语言前几名。

C# 的稳步上升趋势也容易理解，因 C# 是近 10 多年来微软推出的新语言，使用群体也是逐渐扩大。不过就在几年间，C# 已上升为最主流的编程语言之一，其速度也的确是非常之快。

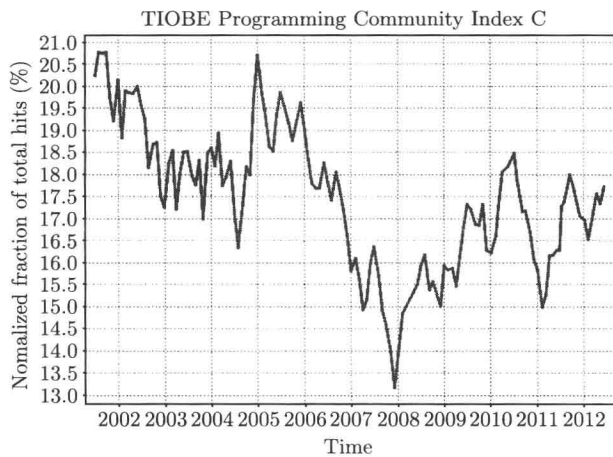


图 8 C 语言 TIOBE 指数

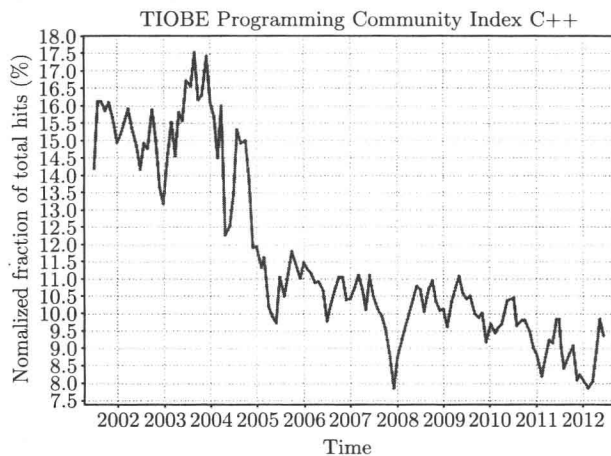


图 9 C++ 语言 TIOBE 指数

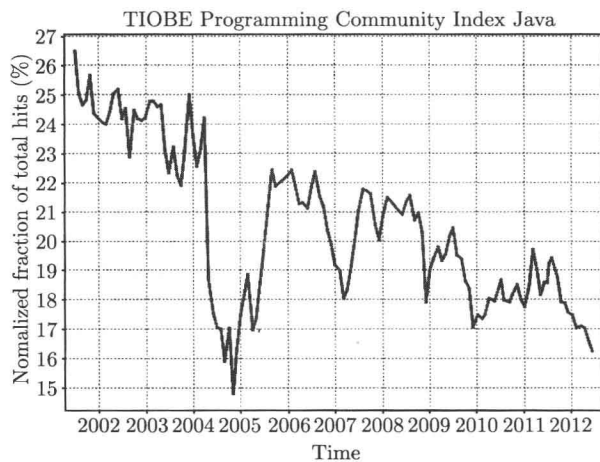


图 10 Java 语言 TIOBE 指数