高等院校金融数学丛书

金融资产的定价理论与数值计算

附C++程序

田文昭／著

北京大学出版社
PEKING UNIVERSITY PRESS

金融资产的定价理论与数值计算

——附C++程序

田文昭　著

北京大學出版社
PEKING UNIVERSITY PRESS

内 容 简 介

计算金融学(Computional Finance)是金融学与计算机科学的交叉学科.

本书较为全面地介绍了计算金融学的原理和方法,包括货币的时间价值、简单衍生证券定价(远期、期货和互换)、期权定价理论、基本的数值计算方法(蒙特卡罗法、二叉树法和有限差分法)、利率衍生证券定价、奇异期权定价等,并提供了大量实用定价模型和金融计算的 C++ 源程序(如果需要,请登录博客"http://blog.sina.com.cn/scifinance"留言).本书侧重介绍使用计算金融学的原理和方法求解金融问题,尤其是没有解析解的金融问题.

本书可作为金融研究、金融实务的专业用书,同时也可作为高等院校计算金融学的教学、科研用书,还可作为作者主持开发的"金融衍生证券定价系统"(软著登字第 0170820 号)的指导用书和《期权、期货和衍生证券》(Hull 著)的参考用书.

图书在版编目(CIP)数据

金融资产的定价理论与数值计算:附 C++程序/田文昭著. —北京:北京大学出版社,2010.4

(高等院校金融数学丛书)

ISBN 978-7-301-15990-3

Ⅰ.金… Ⅱ.田… Ⅲ.金融—数值计算—研究 Ⅳ.F830

中国版本图书馆 CIP 数据核字(2010)第 187706 号

书　　名:金融资产的定价理论与数值计算——附 C++程序

著作责任者:田文昭　著

责 任 编 辑:曾琬婷　王　华

标 准 书 号:ISBN 978-7-301-15990-3/F・2319

出 版 发 行:北京大学出版社

地　　址:北京市海淀区成府路 205 号　100871

网　　址:http://www.pup.cn　　电子信箱:zpup@pup.pku.edu.cn

电　　话:邮购部 62752015　发行部 62750672　理科编辑部 62752021　出版部 62754962

印 刷 者:北京宏伟双华印刷有限公司

经 销 者:新华书店

787mm×980mm　16 开本　18 印张　380 千字

2010 年 4 月第 1 版　　2010 年 4 月第 1 次印刷

定　　价:36.00 元

前　言

本书从酝酿到完稿，前后大致经历了四年左右的时间. 在此期间，美国爆发了金融危机，中国股市从 6000 多点一路狂跌至 1600 多点. 对于这场金融危机，目前已经有许多解读. 有相当多的人认为，衍生证券是这场危机的始作俑者. 那么，什么是衍生证券？

衍生证券是"火箭科学家"，运用计算金融学(Computional Finance)原理和方法，通过对简单证券的合成、剥离而开发出来的新型金融工具. 美国康奈尔大学教授黄明认为，衍生证券有简单与复杂之分. 简单的衍生证券可以用诺贝尔经济学奖理论，甚至初中数学就可以解决；复杂的衍生证券则要用比诺贝尔经济学奖理论更加复杂，依靠几百几千行的计算机程序，才能解决.

衍生证券的基本功能是对冲风险，然而滥用衍生证券，将会造成巨额损失，甚至酿成金融危机. 衍生证券的这种双刃剑功能，要求投资者在使用前要具备一定的知识. 本书将向广大读者介绍这方面的知识.

本书以货币的时间价值、资产组合理论、资本资产定价模型和期权定价理论等为主线，向读者介绍如下内容及相应的 C++程序:

(1) 货币的时间价值与股票、债券、远期、期货和互换等基础金融资产的定价.

(2) 投资组合理论、资本资产定价模型和套利定价模型. 在资产组合理论中，仅讨论在等式约束条件下的优化问题，一般性的二次规划问题，因涉及的内容较为复杂，将在作者的博客"http://blog.sina.com.cn/scifinance"中与大家探讨.

(3) 期权定价理论与相关内容是本书的核心. 本书将用四章篇幅讨论这类问题，内容包括：Black-Scholes 期权定价理论、Black-Scholes 期权定价理论的拓展模型、蒙特卡罗方法、二叉树方法和有限差分法.

(4) 利率衍生证券是衍生证券家族中的一个重要分支. 本书介绍三类利率衍生证券模型：Black-Scholes 期权定价理论的拓展模型、均衡模型、无套利模型，并且给出了重要模型的 C++程序.

(5) 奇异期权是非常复杂的衍生证券. 奇异期权的种类很多，定价相当复杂，本书仅给出了几种典型的奇异期权定价及相应的 C++程序，以便读者了解复杂衍生证券定价和编程的大致思路和方法.

(6) 在写作本书期间爆发了金融危机，衍生证券受到许多指责，本书专用一章篇幅介绍了一些专家和学者对本次金融危机的解读以及与本次金融危机关系紧密的衍生证券和定价.

本书兼顾理论和实务，即本书在介绍金融理论的同时，还提供约 300 个 C++源程序. 所

有这些C++源程序都在Visual C++6.0版本上经过精心的验证和调试，可以直接应用于金融教学、金融科研和金融实务. 如果需要的话，请登录博客"http://blog. sina. com. cn/scifinance"留言.

作者长期从事计算金融的科研、教学与实务工作，做过证券投资，主持开发过大型金融资产定价和投资决策系统. 作者数年前到高校任教直至现在，在此期间发现，我国高校不仅在金融科研、教学方面与国际上存在着较大差距，而且观念上还远落后于国内金融实务界. 因此，作者萌发了撰写一部反映金融发展趋势的书籍. 在同仁的鼓励下，作者从2004年着手本书的写作和编程，前后花了四年左右的时间. 感谢刘为民同志在本书的编程过程中给予的支持. 另外，还要把诚挚的感谢送给本书的编辑，他们为本书的出版做了大量艰辛的工作. 感谢所有对本书写作和出版给予过支持的同志.

计算金融学是一门交叉学科，在我国涉足这领域的人不多，金融危机或许使人们认识到它的重要性. 本书是国内关于这一方面研究的极少图书之一，希望它的出版可在一定程度上缩短我国在金融科研、金融教学和金融实务等方面与国际上的差距. 由于本人水平有限，本书肯定还存在一些不完善的地方，权且作为抛砖引玉，期待后续类似图书质量越来越高.

本书在使用过程中如果有什么问题，欢迎登录博客"http://blog. sina. com. cn/scifinance"留言，以便进一步提高本书的质量.

田文昭

2009. 5. 20

目　　录

第 1 章　货币的时间价值及应用

货币的时间价值也称资金的时间价值，是指货币经历了一定时间的投资和再投资之后所增加的价值. 货币随着时间的延续而增值，不同的时间货币的价值是不一样的. 所以，不同时间的货币价值需要换算到相同的时间基础上才能进行比较. 货币的时间价值在金融领域有着非常广泛的应用，可以说整个金融学的核心——资产定价，都是以货币的时间价值为基础的. 因此，我们将以货币的时间价值作为本书的开篇.

§1.1　单利计息与复利计息

我们先引进两个最基本的概念：总量函数和利息.

设 $A(t)$ 为本金 $A(0)$ 经过时间 $t(t>0)$ 后的价值，则当 t 变动时，称 $A(t)$ 为**总量函数**. 总量函数 $A(t)$ 在时间 $[t_1, t_2]$ 内的改变量称为本金在时间 $[t_1, t_2]$ 内的**利息**，记为 I_{t_1,t_2}，即

$$I_{t_1,t_2} = A(t_2) - A(t_1). \tag{1.1.1}$$

特别地，当 $t_1 = n-1, t_2 = n(n \in \mathbf{N})$ 时，记

$$I_n = A(n) - A(n-1), \tag{1.1.2}$$

并称 I_n 为第 n 个时间段的利息.

1.1.1　累积函数

在货币的价值增值过程中，本金只是一种名义值，而真正起作用的是单位本金在整个过程中价值的增值. 为了揭示这个规律，我们引入累积函数的概念.

定义 1.1.1　设单位本金在 $t(t>0)$ 时刻的价值是 $a(t)$，则当 t 变动时，称 $a(t)$ 为**累积函数**.

累积函数 $a(t)$ 有如下性质：

(1) $a(0)=1$；

(2) $a(t)$ 为递增函数.

注　若 $a(t)$ 出现下降的趋势，将产生负的利息，这在实际上是没有意义的. 另外，累积函数为常数表示无利息.

1.1.2　利率

为了反映货币价值的相对变化，引入利率的概念.

定义 1.1.2　总量函数 $A(t)$ 的增量与本金的比值称为在计息期 $[t_1, t_2]$ 内的**利率**，记为

r_{t_1,t_2}，即

$$r_{t_1,t_2}=\frac{A(t_2)-A(t_1)}{A(t_1)}. \tag{1.1.3}$$

特别地，当 $t_1=n-1, t_2=n(n\in\mathbf{N})$时，记

$$r_n=\frac{A(n)-A(n-1)}{A(n-1)}=\frac{I_n}{A(n-1)}. \tag{1.1.4}$$

结论 1.1.1 某个计息期$[t_1,t_2]$内的利率为单位本金在该计息期内利息与本金的比值，即

$$r_{t_1,t_2}=\frac{a(t_2)-a(t_1)}{a(t_1)}. \tag{1.1.5}$$

证明 假设本金为 $A(0)$，则

$$A(t_1)=A(0)a(t_1),\quad A(t_2)=A(0)a(t_2).$$

所以，由式(1.1.3)有

$$r_{t_1,t_2}=\frac{A(t_2)-A(t_1)}{A(t_1)}=\frac{a(t_2)-a(t_1)}{a(t_1)}.$$

1.1.3 单利计息与复利计息

由结论 1.1.1，利息和利率的计算实质上是对累积函数进行的计算，按照累积函数的不同形式，有不同的计算方法. 下面介绍两种常见的利息计算方法.

1. 单利计息

单利计息的基本思想是：只要本金在一定期限内有利息，不管时间多长，所产生的利息均不加入本金重新计息.

定义 1.1.3 如果单位本金经历了任意一个单位计息期的投资所产生的利息为常数，则称对应的计息方式为**单利计息**，而对应的利息和利率分别称为**单利**和**单利率**.

结论 1.1.2 在单利计息下，有

$$a(t)=1+rt, \tag{1.1.6}$$

式中，r 是单位本金在经过了一个单位计息期后产生的利息，通常称之为**单利率**.

证明 在单利计息下，单位本金在第一个计息期末价值为 $1+r$，在第二计息期末价值为 $1+2r$，依此类推，累积函数为

$$a(t)=1+rt.$$

分析：式(1.1.6)包含最简单的“+(加)”、“-(减)”、“*(乘)”运算，这些运算都是C++的最基本运算，使用 C++中的相关运算符很容易实现.

程序 1.1.1 单利计息下的累积函数值.

```
#include<iostream.h>

double simple_interest_discrete (const double &times,    // 计息期数;
```

```
                                    const double &r)        // 利率;
{
    double a_t = 1.0;
    a_t = (1.0 + r* times);
    return a_t;
}
```

2. 复利计息

复利计息的基本思想是：在投资期间的每个时期，过去的本金和利息之和都将用于下一时期的再投资. 这就是"利滚利"的含义. 例如，面值为 1000 元，年利率是 10%，期限是三年的债券，第一年年底的价值是 1000×1.10=(1100)元，第二年年底的价值是 1000×1.10×1.10=(1210)元，第三年年底的价值是 1000×1.10×1.10×1.10=(1331)元.

定义 1.1.4　如果单位本金经过任何一个单位计息期所产生的利率为常数，则称对应的计息方式为**复利计息**，而对应的利息和利率分别称为**复利**和**复利率**.

结论 1.1.3　在复利计息下，有

$$a(t) = (1+r)^t, \quad t \in \mathbf{N}, \tag{1.1.7}$$

其中，r 是一个单位计息期内的利率，即复利率.

证明　由累积函数的定义有

$$a(t) = \prod_{n=1}^{t}(1+r_n), \quad t \in \mathbf{N}.$$

再由定义 1.1.4 知，在复利计息下，各个计息期间内的利率相同，即

$$r_n = r, \quad n = 1,2,\cdots,t.$$

所以，累积函数为

$$a(t) = (1+r)^t, \quad t \in \mathbf{N}.$$

显然，上式对 $t=0$ 同样成立，故结论 1.1.3 成立.

分析：式(1.1.7)中包含了"+(加)"和指数运算，指数运算不是 C++ 的基本运算，需另外编写函数或者调用 C++ 系统函数来实现. 我们采用调用 C++ 系统函数的方法，即：(1) 在程序的开头部分用指令"include"嵌入头文件"math.h"；(2) 在函数体中直接书写函数"pow(a,b)"完成相应的运算.

程序 1.1.2　复利计息下的累积函数值.

```
#include <math.h>
#include<iostream.h>

double compound_interest_discrete (const double &times, // 计息期数;
                                   const double &r) // 利率;
{
```

```
    double a_t = 1.0;
    a_t = pow(1.0 + r, times);
    return a_t;
}
```

例 1.1.1 假设有单位本金，计息期是 5 年，年利率是 5%，每年计息两次，试比较单利计息和复利计息的实际收益.

解 在本例中，$r=0.05$，$t=10$. 将它们代入式(1.1.6)和式(1.1.7)，有结果：

在单利计息下，$a(t)=1+rt=1+0.05\times 10$；

在复利计息下，$a(t)=(1+r)^t=(1+0.05)^{10}$.

```
// 程序调用;
void main()
{
    double r = 0.05;
    double times = 10;
    cout<<"单利计息：";
    cout<<simple_interest_discrete(times,r)<<endl;
    cout<< "复利计息：";
    cout<< compound_interest_discrete(times,r)<<endl;
}
```

输出结果：

```
单利计息：1.5
复利计息：1.62889
```

所以在单利计息下的实际收益为 1.5－1＝0.5；在复利计息下的实际收益为 1.62889－1＝0.62889. 显然，较之单利计息，复利计息更合算. 复利计息在金融资产定价中有着广泛的应用，以后如无特殊说明，本书所指的均为复利计息.

1.1.4 贴现函数

前面介绍的累积函数是用来计算单位本金在一段时期结束时刻的价值的，下面我们将讨论这个过程的反过程.

定义 1.1.5 若 t 时刻的单位资金在 0 时刻的价值记为 $a^{-1}(t)$，则当 t 变动时，称$a^{-1}(t)$为**贴现函数**.

由该定义 1.5 可知，在单利计息下，有

$$a^{-1}(t) = (1+rt)^{-1} \quad (t \geqslant 0), \tag{1.1.8}$$

其中 r 为单利率；在复利计息下，有

$$a^{-1}(t) = (1+r)^{-t} \quad (t \geqslant 0), \tag{1.1.9}$$

其中 r 为复利率. 这说明,贴现与累积是相互对称的计算货币时间价值的方法. 在贴现时使用的利率通常称为**贴现率**.

分析:式(1.1.8)和式(1.1.9)中均包含“+(加)”、“*(乘)”和指数运算,处理方法与程序 1.1.2 相同.

程序 1.1.3　贴现函数.

```
#include <math.h>
#include<iostream.h>
void discount_function_discrete (double &times,     // 计息期数;
                                 double &r,         // 贴现率;
                                 double &at_1,      // 单利贴现函数;
                                 double &at_2)      // 复利贴现函数;
{
   at_1 = pow(1.0 + r* times, - 1);
   at_2 = 1/pow(1.0 + r,times);
}
```

例 1.1.2　假设贴现率为 5%,试求未来 10 年末单位资金的贴现函数值.

解　在本例中,贴现率 $r=0.05$,时间 $t=10$. 将它们代入式(1.1.8)和(1.1.9),有

在单利计息下,$a^{-1}(t)=(1+rt)^{-1}=(1+0.05\times10)^{-1}$;

在复利计息下,$a^{-1}(t)=(1+r)^{-t}=(1+0.05)^{-10}$.

```
// 程序调用;
void main()
{
   double r = 0.05;
   double times = 10;
   double at_1,at_2;
   discount_function_discrete ( times,r,at_1,at_2);
   cout<<"单利贴现函数值: "<<at_1<<endl;
   cout<<"复利贴现函数值: "<<at_2<<endl;
}
```

输出结果:

　　单利贴现函数值: 0.666667

　　复利贴现函数值: 0.613913

1.1.5　复利的终值和现值

在金融领域,人们最为关心的是在复利计息下一定数量的资金在投资开始和结束时的

价值.这两个值分别是复利的终值和现值.它们的具体定义如下:

定义 1.1.6 称本金 A 与复利累积函数的乘积 $A(1+r)^t$ 为在第 t 个计息期末的**复利终值**,简称**终值**(FV),其中 r 为利率.

分析:根据定义 1.1.6,复利终值包含"+(加)"、"*(乘)"和指数运算,处理方法与程序 1.1.3 相同.

程序 1.1.4 复利终值.

```
#include <math.h>
#include<iostream.h>
double compound_interest_fv_discrete (const double &times,     // 计息期数;
                                      const double &amounts,   // 本金;
                                      const double &r)         // 利率;
{
   double FV = 1.0;
   FV =  amounts * pow(1.0 + r, times);
   return FV;
}
```

例 1.1.3 某人购入面值 100 元的复利债券一张,年利率是 8%,期限是 10 年,试计算 10 年末的终值.

解 在本例中,$r=0.08$,$t=10$,$A=100$.根据复利终值定义,有

$$\mathrm{FV} = A(1+r)^t = 100 \times (1+0.08)^{10}.$$

```
// 程序调用;
void main()
{
   double r = 0.08;
   double  times = 10;
   double  amounts = 100;
   cout<< "复利终值:";
   cout<< compound_interest_fv_discrete(times, amounts,r)<<endl;
}
```

输出结果:

复利终值:215.89

定义 1.1.7 称第 t 期资金量 A 与复利贴现函数的乘积 $A(1+r)^{-t}$ 为**复利现值**,简称**现值**(PV),其中 r 为贴现率.

分析：根据定义 1.1.7，复利现值包含"＋(加)"、"＊(乘)"和指数运算，处理方法与程序 1.1.3 相同.

程序 1.1.5　复利现值.

```
#include <math.h>
#include<iostream.h>
double compound_interest_pv_discrete (const double &times,   // 计息期数；
                                      const double &amounts, // 资金量；
                                      const double &r)       // 贴现率；
{
    double PV = 0.0;
    PV =  amounts * pow(1/(1.0 + r), times);
    return PV;
}
```

例 1.1.4　某人计划 5 年后得到 3000 元钱，已知年利率为 8%，按复利计息，问：该人现在应该存入多少钱?

解　由复利现值的定义有

$$PV = A(1+r)^{-t} = \frac{3000}{(1+0.08)^5}.$$

```
// 程序调用；
void main()
{
    double r = 0.08;
    double times = 5;
    double amounts = 3000;
    cout<< "复利现值：";
    cout<< compound_interest_pv_discrete(times,amounts,r)<<endl;
}
```

输出结果：

复利现值：2041.75

1.1.6　计息次数

复利计息不一定总是一年一次，有可能是季度、月或日一次. 当利息在一年内要复利计息几次时，相应的年利率又叫做名义利率.