

隐含波动率与波动率微笑数值实验

实验项目开发背景：随着全球经济一体化和金融市场的不断深化，金融学科的实验教学发展面临巨大的挑战，一方面要求真实金融市场数据全面进课堂，另一方面要求在传统专业课基础上紧跟市场发展趋势开设前沿性的实验课，致力于培养能与业界无缝对接的金融投资人才。其中，《期货与期权》实验课成为金融工程专业、投资专业、数理金融等专业的核心课程。在《期货与期权》实验课中，期权定价是重点实验内容，要求学生通过实验全面掌握期权定价离散模型及连续时间模型及其相关数值方法核心内容和算法实现。全部实验均要求学生使用万得数据终端和彭博数据终端获取公司和金融市场的真实数据，同时，考虑到学生的背景和未来的职业选择差异，设计了三种编程的算法实现手段，即轻量级的解决方案——Excel 及其 VBA，中量级的解决方案——Matlab 及其图形用户界面，重量级的解决方案——C++ 及其 Excel 加载宏。既训练了不同知识背景和技能的学生的定价模型编程实现技巧，又考虑到面向用户展示和使用时的友好性和直观性。“隐含波动率与波动率微笑数值实验”是《期货与期权》实验课中一个代表性实验项目。

一、实验目的

要求学生使用 EXCEL 和 MATLAB 计算隐含波动率并利用苹果公司股票期权真实交易数据绘制波动率微笑。通过该实验实现以下目的：

1. 使学生掌握隐含波动率计算方法的原理，并在具有隐含波动率和期权执行价格数据的条件下，通过绘制波动率微笑，使学生更为感性认知波动率微笑现象的真实存在。
2. 促使学生发现 B-S-M 定价模型的历史局限性，从而为定价模型的扩展提供思路和线索，启发学生探索更为先进的定价模型。
3. 由于这两种方法均采用相同的真实数据，学生可以比较数值计算结果，从而验证实验结果的正确性。

二、实验准备（简单列示开展该实验项目需要的知识点以及需掌握的软件或数据终端）

1. 回顾波动率微笑的概念、内涵以及产生的原因。
2. 掌握隐含波动率的理论计算方法。
3. 掌握 EXCEL 的常用绘图功能。
4. 掌握 MATLAB 常用绘图命令。
5. 熟悉 Bloomberg 金融终端有关上市公司股票期权交易数据的查询和下载功能。

三、实验数据或案例

隐含波动率的计算需要基于真实的市场数据。苹果公司的股票期权交易也非常活跃，交易量巨大，而且不同执行价格的股票期权合约数量很多，对于计算隐含波动率和展示波动率微笑现象十分有利。我们可以登录 Bloomberg 金融终端查询下载苹果公司 2013 年 6 月 22 日的股票收盘价，并选择一个最为活跃的短期合约，下载该合约当日所有期权执行价格及对应期权收盘价，以及无风险利率（附录 3 的代码中给出了完整数据，其中 K 为执行价格， P 为相应的执行价格下股票看涨期权的交易价格）。

四、实验过程

1. 基于 EXCEL 的隐含波动率与波动率微笑数值实验过程

1.1 设计并建立数据表格

建立基础数据输入表格，内容要包括股票价格等数据。其中包括一系列的期权执行价格，

以及预留出的隐含波动率数据的填入空间。

1.2 创建期权价格计算函数

在接下来的 1.3 中我们要创建隐含波动率计算函数，其中需要使用一系列的期权交易价格，因此在计算隐含波动率之前我们首先要创建期权价格计算函数，从而有利于 1.3 中隐含波动率函数的创建。参考代码（见附录 1）。

通过附录 1 中的代码我们创建了两个函数 BS_Call 和 BS_Put，每个函数都包括 6 个参数，即股票价格 S、执行价格 X、距离到期日剩余时间 T、无风险利率 r、波动率 σ 和分红率 q 。当用户在 EXCEL 单元格内使用此函数并输入相应的参数后，这些参数将传入这段代码，并计算出 B-S-M 模型下的期权价格，返回到该单元格中。

1.3 创建隐含波动率计算函数

隐含波动率有很多理论解法，为了便于 EXCEL 编程，我们这里采用二分法。二分法的思路是给定某一波动率的区间，取区间的中点计算期权价格，若所得价格与实际价格间差值小于某一给定数值，则取此波动率为该期权的隐含波动率，否则，将所取波动率作为新的区间边界，再次进行计算。参考代码（见附录 2）。

通过附录 2 中的代码我们创建了 IV 函数，该函数有 6 个参数，即股票价格 S、执行价格 X、距离到期日剩余时间 T、无风险利率 r、期权市场价格 Price、分红率 d。函数设定的最初波动率的范围是(0.001,5)，差值给定为 0.0000001。此函数调用了 1.3 中建立的期权价格计算函数，用于计算给定波动率的期权理论价格。

1.4 计算隐含波动率并绘制波动率微笑图形

使用创建的隐含波动率计算函数计算不同执行价格期权的隐含波动率，并将结果输送到 1.1 中建立的表格。利用隐含波动率数据及相应的执行价格数据，绘制二者间的关系图，具体结果见图 1。

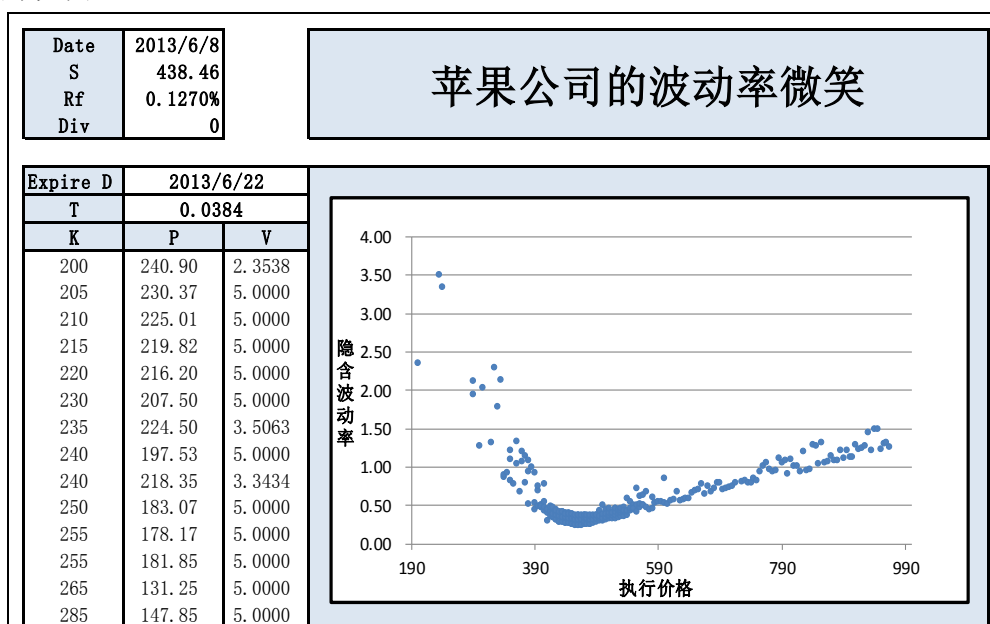


图 1 苹果公司股票期权的波动率微笑——EXCEL

2. 基于 MATLAB 的隐含波动率与波动率微笑数值实验过程

2.1 隐含波动率计算方法

在 B-S-M 公式中，波动率是一个重要参数，在 B-S-M 公式问世不久，就有学者质疑波动率常数这一假设。于是学者们提出，如果 B-S 公式是对的，把市场上观测到的期权价格作为输入量代入 B-S-M 公式，应该可以反推算出波动率这一原本的输入参数。而且如果波

动率常数假设是对的，不同的期权价格输入公式反算出来的应该是相同的结果，即隐含在股票价格中的常数波动率。但事实并非如此，实证研究表明，通过这样计算得到的隐含波动率并非常数，由不同执行价格和不同到期日的期权价格得到的股票的隐含波动率是一个关于执行价格 K 和到期日 T 的函数，即 $\sigma = \sigma(K, T)$ 。

根据 S, K, r, T 和期权价格 C 并不能将波动率 σ 明显地求出，而需要像对待非线性方程的通常求根步骤那样，通过迭代算法才能求出隐含波动率。计算隐含波动率常用的迭代算法是牛顿——拉夫森方法。具体计算公式如下：

$$\sigma_{n+1} = \sigma_n - \frac{C(\sigma_n) - C_m}{C'(\sigma_n)}$$

其中 σ_n 表示隐含波动率 σ_{imp} 的第 n 次迭代。适当的选择初始值 σ_1 ，那么 $\{\sigma_n\}$ 的极限将收敛到隐含波动率的唯一解。MATLAB 金融函数工具箱中的隐含波动率的计算正是采用该方法。计算隐含波动率只需要直接调用隐含波动率函数即可，非常便利。

2.2 绘制波动率微笑

有了隐含波动率计算结果和其他相应数据，就可以运用 MATLAB 绘制苹果公司股票期权的波动率微笑现象。具体代码见附录 3。建议使用 importdata 函数以实现数据的调用，这样便可将股票等数据保存在其他 M 文件或者直接引用 MATLAB 支持的其他文件中的数据，从而可以很好的保全数据的完整性、可靠性，也增加了程序的灵活性。应当注意，在使用苹果公司全部数据的时候，程序运行将很慢，不方便将计算过程与画图过程二者放在同一个程序中进行。这时我们可以选择通过双击 work space 窗口下的 `ImpvVol` `<1x350 dout`，获得期权全部的隐含波动率，另外储存在一个矩阵中，单独画图，以实现计算与画图二者功能的分离，提高运行效率。为了更好地展现经典的实验结果，剔除个别不能计算期权隐含波动率的点（在理论研究中，我们不能随意剔除数据，这里只做展示用，故剔除）之后，绘制的波动率微笑见图 2。

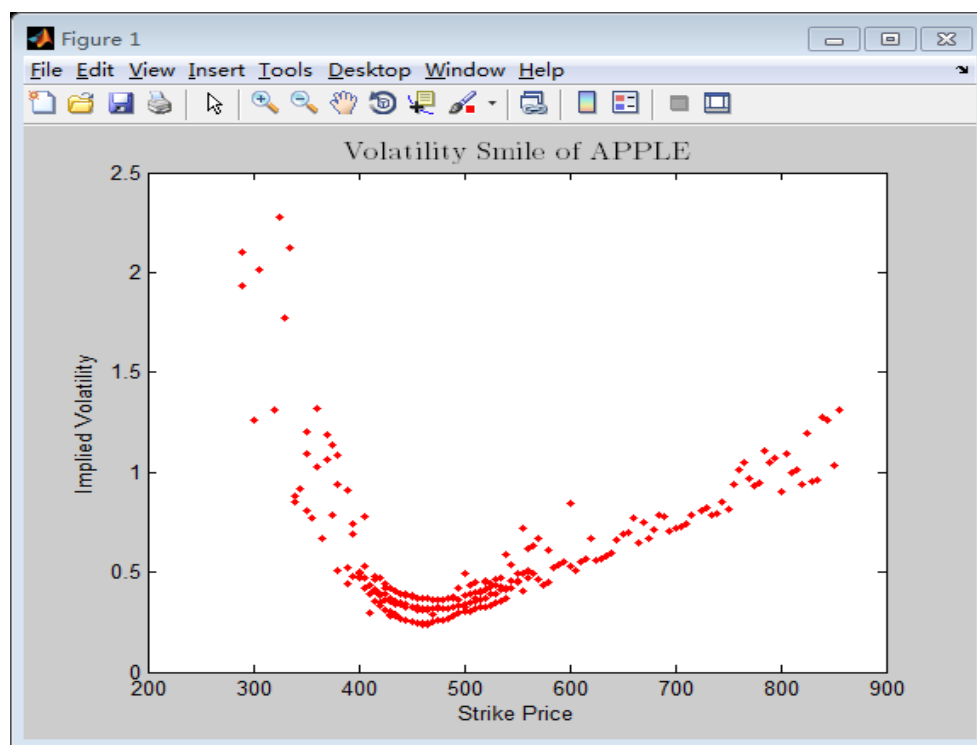


图 2 苹果公司股票期权的波动率微笑

五、实验结果分析

从图 1 和图 2 可知,无论是通过 EXCEL 绘制的苹果公司 2013 年 6 月 22 日到期的股票期权执行价格与隐含波动率关系的图形,还是通过 MATLAB 绘制的苹果公司同一时期的股票期权执行价格与隐含波动率关系的图形,随着执行价格的增加,隐含波动率呈现先下降后上升的情况,呈现出著名的“波动率微笑”现象。而且由于原始数据相同,两者的图形高度一致。当然由于真实的股票期权的执行价格是有一定报价间隔的,因此绘制的图形并不太光滑,将 MATLAB 绘制的波动率微笑与在通过 EXCEL 绘制的波动率微笑相比, MATLAB 绘制的图形的质量更好一些。因此,在对图形质量有较高要求的情况下,我们通常多采用 MATLAB 绘制图形。当然在计算量不大,图形要求质量不高的情况下,与 MATLAB 相比, EXCEL 的使用范围更为广泛。

附录 1:

```
Function BS_Call( _
    ByVal S As Double, _
    ByVal X As Double, _
    ByVal T As Double, _
    ByVal r As Double, _
    ByVal v As Double, _
    ByVal d As Double) As Double
    Dim d1 As Double
    Dim d2 As Double
    d1 = (Log(S / X) + (r - d + v ^ 2 / 2) * T) / v / Sqr(T)
    d2 = d1 - v * Sqr(T)
    BS_Call = Exp(-d * T) * S * Application.NormSDist(d1) - X * Exp(-r * T) *
Application.NormSDist(d2)
End Function
Function BS_Put( _
    ByVal S As Double, _
    ByVal X As Double, _
    ByVal T As Double, _
    ByVal r As Double, _
    ByVal v As Double, _
    ByVal d As Double) As Double
    Dim d1 As Double
    Dim d2 As Double
    d1 = (Log(S / X) + (r - d + v ^ 2 / 2) * T) / v / Sqr(T)
    d2 = d1 - v * Sqr(T)
    BS_Put = -Exp(-d * T) * S * Application.NormSDist(-d1) + X * Exp(-r * T) *
Application.NormSDist(-d2)
End Function
```

附录 2

```
Function IV( _
```

```

ByVal S As Double, _
ByVal X As Double, _
ByVal T As Double, _
ByVal r As Double, _
ByVal Price As Double, _
ByVal d As Double) As Double
Dim epsilonABS As Double
Dim epsilonSTEP As Double
Dim volMid As Double
Dim volLower As Double
Dim volUpper As Double
minimum = 0.0000001
volLower = 0.001
volUpper = 5
Do While volUpper - volLower >= minimum
    volMid = (volLower + volUpper) / 2
    If Abs(BS_Call(S, X, T, r, volMid, d) - Price) <= minimum Then
        Exit Do
    ElseIf ((BS_Call(S, X, T, r, volLower, d) - Price) * (BS_Call(S, X, T, r, volMid, d) - Price) < 0) Then
        volUpper = volMid
    Else
        volLower = volMid
    End If
Loop
IV = volMid
End Function

```

附录 3:

K=[290 290 300 305 320 325 330 335 340 340 345 350 350 350 355 360 360...
365 370 370 375 375 380 380 380 390 390 390 395 395 395 400 400 400 ...
400 405 405 405 405 410 410 410 410 415 415 415 415 415 420 420 420 ...
420 420 425 425 425 425 425 425 430 430 430 430 430 430 435 435 435 ...
435 435 435 440 440 440 440 440 440 445 445 445 445 445 445 450 450 ...
450 450 450 450 455 455 455 455 455 460 460 460 460 460 460 465 465 ...
465 465 465 470 470 470 470 470 470 475 475 475 475 480 480 480 480 ...
480 480 485 485 485 485 490 490 490 490 490 490 495 495 495 495 500 ...
500 500 500 500 500 505 505 505 505 510 510 510 510 510 515 515 515 ...
515 520 520 520 520 520 525 525 525 525 530 530 530 530 535 535 535 ...
535 540 540 540 540 545 545 545 550 550 550 555 555 555 560 560 560 ...
565 565 565 570 570 575 580 580 585 590 595 600 600 605 610 615 620 ...
625 630 635 640 645 650 655 660 665 670 675 680 685 690 695 700 705 ...
710 715 725 730 735 740 745 750 755 760 765 770 775 780 785 790 795 ...
800 805 810 815 820 825 830 835 840 845 850 855 860 865 870 875 880 ...
885 890 895 900 905 910 915 920 925 930 935 940 945 950 955 960 965];

```

P=[160.7 158 140.99 147.5 123.7 138.72 124 128.5 100.35 100.65 96.55...
90.75 97 95 85.8 91.78 85.61 75.5 81.45 78.5 ...
68.72 76.5 71.55 68 59.98 51.3 50 60 46.15 50.75 ...
51.95 41.9 42.3 42.5 42.05 36.75 37.85 46 39.3 ...
32 33.05 33.05 30.05 27.2 28.7 28.5 30.5 30.1 ...
22.85 23.25 24.5 24.25 26.85 18.75 18.65 21.1 20.3...
22.65 22.1 15.2 14.7 17.35 16.8 18.95 14.5 11.7...
11.5 14.15 13.6 15.75 15.9 8.6 8.42 11.05 11.2 ...
12.9 12.7 6.19 6.25 8.8 8.31 10.6 10.45 4.3 ...
4.2 6.65 6.5 8.5 8.2 2.85 2.76 5 4.6 6.4 1.86 1.65 ...
3.58 3.46 4.98 5 1.22 1.13 2.64 2.49 3.95 ...
0.82 0.84 1.88 1.4 2.91 2.85 0.56 1.35 1.44 ...
2.17 0.37 0.35 0.93 0.92 1.56 1.65 0.28 0.69...
0.63 1.25 0.22 0.51 0.52 0.53 0.9 1 0.18 0.37...
1.25 0.6 0.14 0.24 0.3 0.24 0.59 1.88 0.09 0.23...
0.85 0.46 0.08 0.18 0.24 0.37 0.79 0.06 0.15 ...
0.3 0.28 0.04 0.12 0.44 0.26 0.48 0.03 0.12 ...
0.23 0.29 0.03 0.09 0.19 0.3 0.02 0.09 0.27 ...
0.13 0.02 0.07 0.8 0.08 0.06 0.39 0.11 0.07 ...
0.17 0.09 0.02 1.41 0.13 0.06 0.51 0.12 0.07 ...
0.5 0.07 0.03 0.59 0.01 0.01 0.2 0.04 0.04 0.04...
0.02 1 0.01 0.02 0.02 0.09 0.01 0.01 0.01 ...
0.01 0.03 0.04 0.04 0.09 0.01 0.05 0.01 0.02 ...
0.05 0.04 0.01 0.01 0.01 0.01 0.02 0.02 0.02...
0.01 0.01 0.02 0.01 0.05 0.1 0.13 0.05 0.03 ...
0.03 0.15 0.08 0.09 0.01 0.09 0.03 0.03 0.01...
0.16 0.01 0.01 0.24 0.2 0.02 0.25 0.02 0.02 ...
0.04 0.02 0.02 0.07 0.02 0.06 0.02 0.02 0.09 ...
0.05 0.05 0.06 0.23 0.03 0.26 0.25 0.03 ...
0.05 0.06 0.03];

```

N=250; % less than 260 please

```
ImpvVol=zeros(1,N+100);
```

```
for i=1:1:N; %you may change this N to a little number and delete the last '%' to speed up the programme
```

```
AssetPrice =438.36;
```

```
Settlement = 'June-08-2008';
```

```
Maturity = 'June-022-2008';
```

```
Strike =K(i);
```

```
Rates = 0.0127;
```

```
OptionPrice = P(i);
```

```
OptSpec = {'call'};
```

```
%Define RateSpec and StockSpec :
```

```
RateSpec = intenvset('ValuationDate', Settlement, 'StartDates', Settlement,...
```

```
'EndDates', Maturity, 'Rates', Rates, 'Compounding', -1, 'Basis', 1);
```

```
StockSpec = stockspeg(NaN, AssetPrice);
```

```
%Calculate the implied volatility of the options:  
ImpvVol(i) = impvbybls(RateSpec, StockSpec, Settlement, Maturity, OptSpec,...  
Strike, OptionPrice);  
end  
plot(K(1:N),ImpvVol(1:N),'.r');  
xlabel('Strike Price');  
ylabel('Implied Volatility');  
title('Volatility Smile of APPLE','interpreter','latex','FontSize',13);
```