第六章　金融数量计算技巧与实例

金融数量分析是充满变革与创新的世界，从上个世纪 50 年代的马克维茨模型，到 70 年代的 BS 期权定价公式，到 90 年代 CDOs 的定价模型等等，这些模型无在当时无处是创新的产物。在金融数量分析的学习与研究中，往往遇见没有现成求解工具的模型，需要我们利用基本数学原理或者数值计算软件根据实际的需要进行金融数量模型的建立、模型的求解、模型的验证等。在这个过程中，不仅需要数学原理，可能需要更多的数值处理技巧。或许只有在数学原理与数值技术有效的结合的前提先，才能更有效的求解金融数学模型。本章以 BS 公式的隐含波动率计算、KMV 模型方程组的求解、移动平均 hurst 指数计算与基于优化方法的指追踪技术为例，展示在金融数量分析的步骤与技巧。

6.1 BS 公式隐含波动率计算

6.1.1 隐含波动率概念

如在第二章介绍的 Black-Scholes 期权定价公式，欧式买权解的表达式：

$$c_t = S_t N(d_1) - Xe^{-r(T-t)}N(d_2)$$

$$p_t = Xe^{-r(T-t)} \times [1 - N(d_2)] - S \times [1 - N(d_1)]$$

其中，

$$d_1 = \frac{[\ln(\frac{S_t}{X}) + (r + \frac{\sigma^2}{2})(T-t)]}{[\sigma^2 (T-t)^{1/2}]}$$

$$d_2 = d_1 - \sigma^2 (T-t)^{1/2}$$

Black-Scholes 期权定价模型将股票期权价格的主要因素分为五个：

S_t：标的资产市场价格

X：执行价格

r：无风险利率

σ：标的资产价格波动率

$T-t$：距离到期时间。

一般情况下，已知上述五个参数即可计算出相对应的期权价格。期权可以在交易所进行交易的，但其交易价格不一定为根据历史波动率由 BS 公式计算出的理论价格。主要原因为投资者认为该期权标的证券的波动率与其历史波动率不一致所致。例如，期权标的证券代表的公司可能将发生合并重组、资产注入或者由于投资非理性投资造成。

隐含波动率是将市场上的期权交易价格代入权证理论价格 Black-Scholes 模型，反推出来的波动率数值。由于期权定价 BS 模型给出了期权价格与五个基本参数之间的定量关系，只要将其中前 4 个基本参数及期权的实际市场价格作为已知量代入定价公式，就可以从中解出惟一的未知量，其大小就是隐含波动率。

隐含波动率是一个重要的风险指标。历史波动率反映标的期权标的证券在过去一段时间的波动幅度，期权发行商与投资者在期权发行初期只能利用历史波动率作参考。一般来说，期权的隐含波动率越高，其隐含的风险也就越大。期权投资者除了可以利用期权的正股价格变化方向来买卖权证外，还可以从股价的波动幅度的变化中获利。一般来说，波动率并不是可以无限上涨或下跌，而是在一个区间内来回震荡，投资者可以采取在隐含波动率较低时买入而在较高时卖出期权的方法来获利。

如何判断一个期权的价格是否高估？主要应该看隐含波动率与其标的证券的历史波幅之间的关系。隐含波动率是市场对其标的证券未来一段时间内的波动预期，与期权价格是同方向变化。一般而言，隐含波动率不会与历史波幅相等，但在其标的证券的基本面保持稳健的条件下，应该相差不大。

6.1.2 隐含波动率计算方法

隐含波动率是把权证的价格代入 BS 模型中反算出来的，它反映了投资者对未来标的证券波动率的预期。Black-Scholes 期权定价公式中已知：S_t：标的资产市场价格，X：执行价格，r：无风险利率，$T-t$：距离到期时间，看涨期权 c_t 或者看跌期权 p_t 根据 BS 公式计算出与其相应的隐含波动率 σ_{yin}

数学模型为：

$$f_c(\sigma_{yin}) = S_t N(d_1) - Xe^{-r(T-t)} N(d_2) - c_t = 0$$

$$f_p(\sigma_{yin}) = Xe^{-r(T-t)} \times [1-N(d_2)] - S \times [1-N(d_1)] - p_t = 0$$

其中

$$d_1 = \frac{[\ln(\frac{S_t}{X}) + (r + \frac{\sigma_{yin}^2}{2})(T-t)]}{[\sigma_{yin}^2 (T-t)^{1/2}]}$$

$$d_2 = d_1 - \sigma_{yin}^2 (T-t)^{1/2}$$

求解方程 $f_c(\sigma_{yin}) = 0,\ f_p(\sigma_{yin}) = 0$ 的根。

6.1.3 隐含波动率计算程序

利用 fsolve 函数计算隐含波动率，fsolve 是 matlab 最主要内置的求解方程组的函数，具体 fsolve 的使用方法可以参看 Appendix B：相关函数说明。

例 6.1 假设欧式股票期权，一年后，执行价格 95 元，现价为 100 元，无股利支付，股价年化波动率为 50%，无风险利率为 10%，则期权价格为：

```
>>[Call, Put] = blsprice(100, 95, 0.1, 0.25, 0.5)
>> Call = 13.6953     Put =    6.3497
```

假设目前其期权交易价格为 Call = 15.00，Put = 7.00 分别计算其相对应的隐含波动率。

步骤 1：建立方程函数

看涨期权隐含波动率方程：M 文件 ImpliedVolatitityCallObj.M

f=ImpliedVolatitityCallObj(Volatility, Price, Strike, Rate, Time, Callprice)
输入参数：
Volatility：标的资产价格波动率
Price：标的资产市场价格
Strike：执行价格
Rate：无风险利率
Time：距离到期时间
Callprice：看涨期权价格
输出函数：

f: $f_c(\sigma_{yin})$ 的函数值

程序源码：

```
function f=ImpliedVolatitityCallObj(Volatility, Price, Strike, Rate, Time, Callprice)
%ImpliedVolatitityCallObj
%code by ariszheng@gmail.com 2009-8-3
[Call,Put] = blsprice(Price, Strike, Rate, Time, Volatility);
%fc(ImpliedVolatitity)=Call-Callprice=0
f=Call-Callprice;
```

看跌期权隐含波动率方程：M 文件 ImpliedVolatitityPutObj.M
f=ImpliedVolatitityPutObj(Volatility, Price, Strike, Rate, Time, Putprice)
输入参数：
Volatility：标的资产价格波动率
Price：标的资产市场价格
Strike：执行价格
Rate：无风险利率
Time：距离到期时间
Putprice：看跌期权价格
输出函数：

f: $f_p(\sigma_{yin})$ 的函数值

程序源码：

```
function f=ImpliedVolatitityPutObj(Volatility, Price, Strike, Rate, Time, Putprice)
%ImpliedVolatitityCallObj
%code by ariszheng@gmail.com 2009-8-3
[Call,Put] = blsprice(Price, Strike, Rate, Time, Volatility);
%fp(ImpliedVolatitity)=Put-Putprice=0
f=Put-Putprice;
```

步骤 2：求解方程函数：M 文件 ImpliedVolatility.m
[Vc,Vp,Cfval,Pfval]=ImpliedVolatility(Price,Strike,Rate,Time,CallPrice,PutPrice)
输入参数：
Price：标的资产市场价格
Strike：执行价格

Rate：无风险利率
Time：距离到期时间
Callprice：看涨期权价格
Putprice：看跌期权价格
输出函数：
Vc：看涨期权的隐含波动率
Vp：看跌期权的隐含波动率

Cfval： $f_c(\sigma_{yin})$ 的函数值，若为 0，则隐含波动率计算正确；

Pfval： $f_p(\sigma_{yin})$ 的函数值，若为 0，则隐含波动率计算正确；

程序源码：

```
function [Vc,Vp,Cfval,Pfval]=ImpliedVolatility(Price,Strike,Rate,Time,CallPrice,PutPrice)
%ImpliedVolatility
%code by ariszheng@gmail.com 2009-8-3
%优化算法初始迭代点;
Volatility0=1.0;
%CallPrice 对应的隐含波动率
[Vc,Cfval] =fsolve(@(Volatility) ImpliedVolatitityCallObj(Volatility, Price, Strike,...
    Rate, Time, CallPrice),Volatility0);
%CallPrice 对应的隐含波动率
[Vp,Pfval] =fsolve(@(Volatility) ImpliedVolatitityPutObj(Volatility, Price, Strike, ...
    Rate, Time, PutPrice),Volatility0);
```

步骤 3：函数求解：M 文件 TestImpliedVolatility.M

```
%TestImpliedVolatility
Price=100;
Strike=95;
Rate=0.10;
Time=1.0;
CallPrice=15.0;%看涨期权交易价格
PutPrice=7.0; %看跌期权交易价格
[Vc,Vp,Cfval,Pfval]=ImpliedVolatility(Price,Strike,Rate,Time,CallPrice,PutPrice)
```

计算结果

```
>> Optimization terminated: first-order optimality is less than options.TolFun.
Optimization terminated: first-order optimality is less than options.TolFun.
Vc =
    0.1417
Vp =
    0.3479
Cfval =
  3.7957e-011
Pfval =
  7.1054e-015
```

结果说明 Cfval 与 Pfval 函数值为 0 说明计算出 Vc 与 Vp 为方程的解，即期权交易价格

为 Call = 15.00，Put =　7.00 分别计算其相对应的隐含波动率为 14.17%与 34.79%。

波动率与价格关系图像：

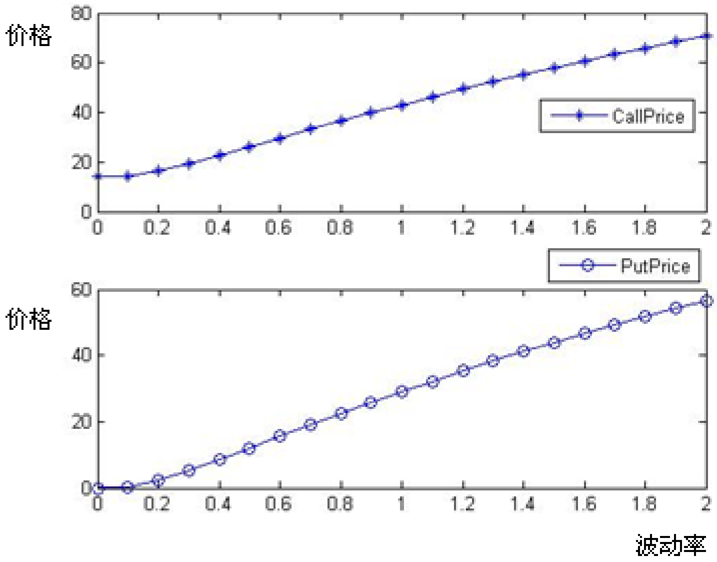

图 6.1 波动率与价格关系图

在其他条件不变的情况下，图 6.2 为期权价格与波动率关系图，横轴为波动率，纵轴为交易价格。

画图程序 M 文件 VolatilityPrice.m

```
Price=100;
Strike=95;
Rate=0.10;
Time=1.0;
Volatility=0:0.1:2.0;
n=length(Volatility);
Call=zeros(n,1);
Put=zeros(n,1);
for i=1:n
    [Call(i),Put(i)] = blsprice(Price, Strike, Rate, Time, Volatility(i));
end
subplot(2,1,1)
plot(Volatility,Call,'-*');
legend('CallPrice')
subplot(2,1,2)
plot(Volatility,Put,'-o');
legend('PutPrice')
```

程序中引用[Call,Put] = blsprice(Price, Strike, Rate, Time, Volatility)函数中 Volatility 参数不能为负数，在特殊情况下，某些期权价格不符合 BS 公式，即某些期权价格不能使用上述公式计算出隐含波动率。