注意1996-2016 call是0 put是1

2016-2023 反之

## 模型

$$\frac{dY_t}{Y_t} = \mu_Y dt + \sigma_Y d\omega_t$$

$$\frac{dM_t}{M_t} = \theta_t d\omega_t, \quad M_0 = 1$$

$$d\theta_t = -\zeta_\theta(\theta_t - \overline{\theta})dt + \sigma_\theta d\omega_t, \quad \theta_0 = \overline{\theta}$$

$$\xi_j(\Theta) = e^{-r_{f,T_j} \times T_j} \frac{\left(\frac{1}{\tilde{R}_{M,T_j}}\right)\left[(1-\nu_0)\left(\frac{M_{T_j}}{M_0}\right)^p + \nu_0\left(\frac{M_{T_j}}{M_0}\right)^{1-p}\right]}{E\left[\left(\frac{1}{\tilde{R}_{M,T_j}}\right)\left[(1-\nu_0)\left(\frac{M_{T_j}}{M_0}\right)^p + \nu_0\left(\frac{M_{T_j}}{M_0}\right)^{1-p}\right]\right]}$$

待估计参数$\Theta = (\gamma, \nu_0, \overline{\theta}, \zeta_\theta, \sigma_\theta)$

---

## 计算$\mu_Y$，$\sigma_Y$

使用AR(1)  GARCH(1,1)

**均值方程**： 取过去1250天对数收益率 $r_t$ 建立AR(1)模型

$r_t = c + \phi_1 r_{t-1} + \varepsilon_t$

**方差方程** : GARCH(1,1)

当期的波动率取决于常数、昨天的波动率预测误差（ARCH项）和昨天的预测方差（GARCH项）。

$\varepsilon_t = \sigma_t z_t, \quad z_t \sim N(0,1)$

$\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2$

利用上述方程递归地推导未来 $H$ 天（`forecast_horizon`）的期望值。

预测均值 ($\hat{r}_{t+k}$)，预测方差 ($\hat{h}_{t+k}$)

年化波动率 $\sigma_Y$ :

$$\sigma_Y = \sqrt{\underbrace{\left(\frac{1}{H}\sum_{k=1}^{H}\hat{h}_{t+k}\right)}_{\text{平均每日方差}} \times 252}$$

漂移项 $\mu_Y$

$$\mu_Y = \underbrace{\left[\left(\frac{1}{H}\sum_{k=1}^{H}\hat{r}_{t+k}\right) \times 252\right]}_{\text{年化对数收益率}} + \underbrace{\frac{1}{2}\sigma_Y^2}_{\text{Itô 调整}}$$

---

## 向量化蒙特卡洛模拟

$$Y_{i+1} = Y_i \exp\left(\left(\mu_Y - \frac{1}{2}\sigma_Y^2\right)\Delta t + \sigma_Y \Delta\omega_i\right)$$

$$\theta_{i+1} = \overline{\theta} + (\theta_i - \overline{\theta})e^{-\zeta_\theta \Delta t} + \sigma_\theta \sqrt{\frac{1 - e^{-2\zeta_\theta \Delta t}}{2\zeta_\theta}} Z_{i+1}$$

$$M_{i+1} = M_i \exp\left(\theta_i \Delta\omega_i - \frac{1}{2}\theta_i^2 \Delta t\right)$$

**(0)参数设定**

1. 时间步长$\Delta t = 1/252$;
2. 第一个期权到期时间$T_1$= floor((options(t_1(d), 2) - options(t_1(d), 1)) / 7 * 5);

   第一个期权到期时间$T_2$= floor((options(t_2(d), 2) - options(t_2(d), 1)) / 7 * 5);
3. 步数$numsteps_1$=$T_1$;
4. 步数$numsteps_2$=$T_2$;
5. 蒙特卡洛模拟路径数$numPaths = 200000$;

**(1)随机冲击项**

为了保证路径一致

Z = randn(max(num_steps2, num_steps1), numPaths);

Z1 = Z(1:num_steps1, :);

Z2 = Z(1:num_steps2, :);

$\Delta\omega_1 = \sqrt{\Delta t} * Z1$=;

**(2)求$Y_{T_j}$**

对数后累加与计算M的思路相同

**(3)求$\theta_i$**

$\theta_{i+1} = \overline{\theta} + (\theta_i - \overline{\theta})e^{-\zeta_\theta \Delta t} + \sigma_\theta \sqrt{\frac{1-e^{-2\zeta_\theta \Delta t}}{2\zeta_\theta}} Z_{i+1}$

这里将$y_i = \theta_i - \overline{\theta}$，使用信号处理中的数字滤波器 ( `filter` 函数)来高效地进行向量化递归计算，MATLAB 的 `filter` 是内置函数，底层由编译好的 C 或 C++ 代码执行。每一次输入行向量进行并行计算,替代了循环结构。

```
a1 = exp(-zeta_theta * dt1);
coef1 = sigma_theta * sqrt((1 - exp(-2 * zeta_theta * dt1)) / (2 * zeta_theta));
inc1 = coef1 * Z1;
theta_dev1 = filter(1, [1, -a1], inc1);
theta_t_1 = theta_bar + theta_dev1;
```

矩阵 `inc1` (输入项) 的长相如下，大小Steps x Paths：

$$\text{inc1} = \begin{bmatrix} \delta_1^{(1)} & \delta_1^{(2)} & \delta_1^{(3)} & \delta_1^{(4)} \\ \delta_2^{(1)} & \delta_2^{(2)} & \delta_2^{(3)} & \delta_2^{(4)} \\ \cdots \end{bmatrix}$$

filter函数是对每一行进行并行计算，计算公式如下,n=1,...,Steps,y_0=0

$1 \cdot y_n + (-a1) \cdot y_{n-1} = 1 \cdot x_n$

$y_n = a1 \cdot y_{n-1} + x_n$

这里$y_n$，$x_n$是1×200000行向量

输出 `theta_dev1` 依然是一个 `[Steps x Paths]` 的 GPU 矩阵。

**(4)求$M_i$**

$\ln(M_{i+1}) = \ln(M_i) + \left(\theta_i \Delta\omega_i - \frac{1}{2}\theta_i^2 \Delta t\right)$ 令 $L_i = \ln(M_i)$，则 $L_{i+1} = L_i + \text{增量}_i$，累加即可

```
theta_prev1 = [theta_0_vec_1; theta_t_1(1:end-1, :)]; %数据对齐%
domega1 = sqrt(dt1) * Z1;
incM1 = theta_prev1.* domega1 - 0.5 * (theta_prev1.^ 2) * dt1;
logM1 = cumsum(incM1, 1);
M_t_1 = exp(logM1);
```

incM1是 `[Steps x Paths]` 的 GPU 矩阵。

cumsum(A, 1)的输出依然是 `[Steps x Paths]` 的 GPU 矩阵。逻辑是：

第1行 =A_1

第2行 =A_1+A_2

第3行 =A_1+A_2+A_3

......

这里A_i表示矩阵A第i行行向量

---

## 计算定价核

$$m_j(\Theta)^{(k)} = \left(\frac{1}{\tilde{R}_{M,T_j}^{(k)}}\right)\left[(1-\nu_0)\left({M_{T_j}}^{(k)}\right)^p + \nu_0\left({M_{T_j}}^{(k)}\right)^{1-p}\right]$$

$$E[m_j(\Theta)] = \sum_{k=1}^{2000000} m_j(\Theta)^{(k)} \times \frac{1}{2000000}$$

$$\xi_j(\Theta)^{(k)} = \frac{e^{-r_{f,T_j} \times T_j} \times m_j(\Theta)^{(k)}}{E[m_j(\Theta)]}$$

---

### 估计理论价格

### 反推隐含波动率

### 计算均方差

---

### GMM估计参数

基于中心差分梯度估计的内点法，

% 优化参数范围设置

gamma0= 1.2;nu0_range = 0.5;theta_bar= 0; zeta_theta= 1.2;  sigma_theta = 1.2;

lb = [1.001,  0.0, -20,  0.001, 0.001];

ub = [20.0, 1.0,  20,  10, 10];