交易开拓者公式编写基础（二）

蔡云华

深圳开拓者科技有限公司

内容概要

- 公式编写应注意的问题及解决办法
- 止损止盈、跟踪止盈代码的编写
- TB用户函数的编写
- 常用指标交易系统的实现

信号消失问题及解决办法

- 产生的原因：
  - 使用BUY/Sell指令进行自动交易；
  - 交易（开仓或平仓）判断条件中使用了变化的数据
- 后果：
  - 导致历史回测结果失真；
  - 导致后续交易指令出现问题；
- 解决办法：
  - 用确定不变的数据来做为判断条件；
  - 用能保持结果不变的数据来做为判断条件；
  - 信号消失后，在代码中将最符合实际交易结果的信号补上（解决某些函数在历史测试无效的问题）

用确定不变的数据做判断

比如：用前一根K线做判断举例：

```
Condition = Close > MA;
If (condition[1])
{
        Buy(1, Open);
}
```

或者

```
if (Close[1] > MA[1])
{
        Buy(1, Open);
}
```

用能保持结果不变的数据做判断

比如：用High、Low、Open等做判断

突破代码：

```
If (High>High[1])
{
        buy(1, Max(Open, High[1]));
}
```

止损代码：

```
if (Low < Stopline)
{
        Sell(0, Min(Open, Stopline));
}
```

代码中将消失的信号补上

F1帮助中的收盘平仓的例子：

```
If ((Date[-1]!=InvalidInteger && Date!=Date[-1])||(Date[-1]==InvalidInteger
&& Date < CurrentDate))
 {
            Sell(0,Close);
            BuyToCover(0,Close);
 }
Else If (Date==CurrentDate && Time==0.1455 && CurrentTime>=0.1459)
 {
           Sell(0,Close);
           BuyToCover(0,Close);
}
```

连续建仓的控制

- 全局交易设置 --- 连续建仓的设置
- 通过持仓函数Marketposition在代码中控制

MarketPosition

说明	获得当前持仓状态。
语法	Integer MarketPosition()
参数	无
备注	获得当前持仓状态，返回值为整型。 返回值定义如下： -1 当前位置为持空仓 0 当前位置为持平 1 当前位置为持多仓
示例	if(MarketPosition==1)判断当前是否持多仓 if(MarketPosition!=0)判断当前是否有持仓，无论持空仓或多仓

例7_1（限制连续建仓）

Sample7_1：

```
Params
        Numeric Length1(10);
        Numeric Length2(20);
        Numeric Lots(1);
Vars
        NumericSeries MA1;
        NumericSeries MA2;
Begin
        MA1 = AverageFC(Close,Length1);
        MA2 = AverageFC(Close,Length2);
        PlotNumeric("MA1",MA1);
        PlotNumeric("MA2",MA2);
        If (MarketPosition <> 1 and MA1[1] > MA2[1])
        {
                Buy(Lots,Open);
        }
        If (MarketPosition <> -1 and MA1[1] < MA2[1])
        {
                SellShort(lots,Open);
        }
End
```

如果希望加仓操作

- 加仓规则
  - 均线多头排列，而且已经持有多单，如果价格突破前一根K线的高点，追加多头一手；
  - 均线空头排列，而且已经持有空单，如果价格跌破前一根K线的低点，追加空头一手；
  - 当均线趋势反转时，将原有头寸全部平仓，然后反手开仓

加仓部分代码

```
If (MarketPosition <> 1 and MA1[1] > MA2[1])
{
        Buy(Lots,Open);
} else if (marketposition == 1 and MA1[1] > MA2[1] and high > High[1])
{
        Buy(Lots,Max(Open,High[1]));
}

If (MarketPosition <> -1 and MA1[1] < MA2[1])
{
        SellShort(lots,Open);
} Else if (marketposition == -1 and MA1[1] < MA2[1] and low < Low[1])
{
        SellShort(Lots,Min(Open,Low[1]));
}
```

突破时考虑滑点

- 突破类系统必须考虑两种问题：
  - 跳空高开的情况
  - 突破的价格滑点
- 考虑滑点后的代码（做多部分）

```
Params
        Numeric Offset(2);
……
else if (marketposition == 1 and MA1[1] > MA2[1] and high > High[1])
{
        Buy(Lots,Max(Open,High[1] + Offset * MinMove * PriceScale));
}
```

注释语句-- Commentary

- TB的信息输出，除了可以通过FileAppend输出到文件外，也可以将信息输出显示到图表上；
- Commentary的用法：
  - 在超级图表的当前BAR添加一行注释信息；
  - 参数：String strTip; // 提示的信息

CurrentContract函数

CurrentContracts

说明	获得当前持仓的持仓合约数。
语法	Numeric CurrentContracts()
参数	无
备注	获得当前持仓的持仓合约数，返回值为整型。 只有当MarketPosition != 0时，即有持仓的状况下，该函数才有意义，否则返回0。 多头持仓返回正数，空头持仓返回负数。
示例	当前空头持仓2手，CurrentContracts则返回-2。

止损止盈的编写

以多单的止盈和止损为例

- 固定跳数的止盈或止损的写法：

```
if (MarketPosition==1)
{
        TargetPrice = EntryPrice + TakeProfit * MinMove * PriceScale;
        StopPrice = EntryPrice – Stoploss * MinMove * PriceScale;
        if (High >= TargetPrice) Sell(0, Max(Open, TargetPrice));
        Else if ( Low <= StopPrice) Sell(0, Min(Open, StopPrice));
}
```

- 价格百分比的止盈或止损的写法：

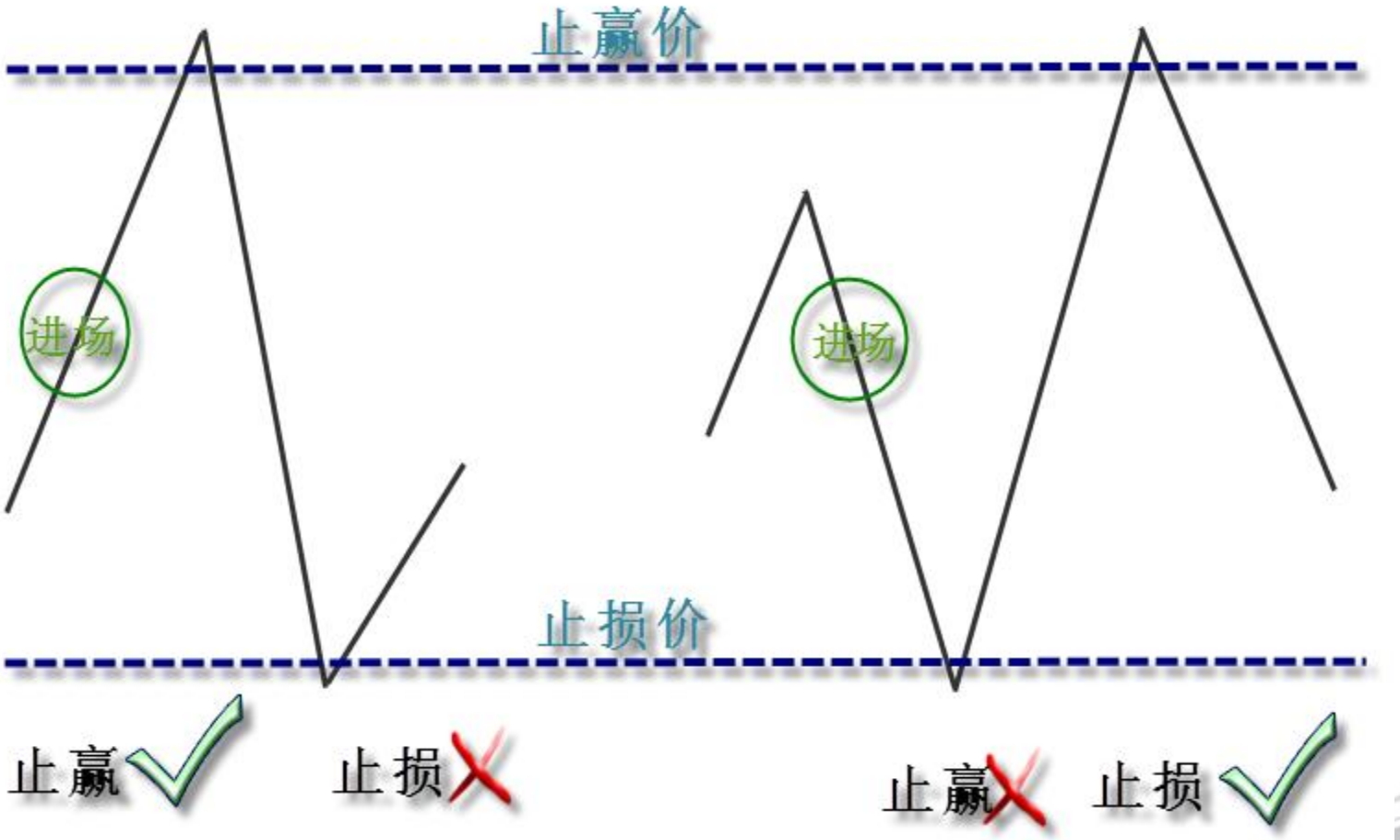
```
        TargetPrice = EntryPrice * (1+ TakeProfit * 0.01);
        StopPrice = EntryPrice * (1 – Stoploss * 0.01);
```

应注意的问题

- 如果单根K线的最高价和最低价相差很大，有可能出现止盈和止损同时满足的情况，解决办法：
  - 切换到更小的时间周期上进行交易；
  - 扩大止盈和止损的幅度

- 在开仓BAR，因无法判断开仓价和最高价最低价的先后顺序，因此一般是在开仓BAR的后一根BAR才开始判断是否满足止盈止损或跟踪止盈的的条件。如交易策略需要及时的止损，同样需要切换到更小的时间周期上进行交易。

K线波动太大的问题



跟踪止盈的编写

- 跟踪止盈策略有很多变种，常见的有：
  - 峰值价回落固定点数
  - 峰值价回落一定的百分比
    - 峰值价的一定比例；
    - ATR的一定比例；
    - 进场价（开盘价）的一定比例。
  - 是否盈利达到一定幅度才启用追踪止盈；
  - 动态的回落点数或比例。
- 下面以峰值价回落ATR一定比例止盈来举例

使用适当的变量类型

- 记录峰值价 – 序列变量；
- 变量的生存期问题
  - 简单变量和序列变量变化的不同
  - 函数返回值的变化规律
- 全局变量和读写数据库文件

全局变量

- 序列变量的缺陷
  - 序列变量在每个BAR只能有一个值，这个值在行情更新时，会不断刷新，直到最后一个Tick才能将值保存下来；
  - 因此，序列变量无法记录盘中每个Tick运行公式产生的数据；
    比如：我们要对每个Tick计数，用序列变量就做不到。
- 全局变量
  - 全局变量通过SetGlobalVar和GetGlobalVar函数来设置和读取，TB V4中单个 公式应用可以支持500个全局变量；
  - Bool SetGlobalVar(Integer nIndex,Numeric fVal)

    参数：nIndex --- 全局变量的索引值

    fVal --- 要设置的变量的值

    如： SetGlobalVar(0,1) 将0号全局变量设置为1；
  - Numeric GetGlobalVar(Integer nIndex)获取某个索引的全局变量值

- 全局变量的初始值为无效值，它的值不会因为当前BAR的变化而变化，而只能由SetGlobalVar函数来设置；
- 全局变量依附在超级图表上，一旦关掉超级图表后，所有与该图表有关的全局变量将不复存在；
- 全局变量值的变化只跟SetGlobalVar的执行顺序有关，因此在图表上进行刷新时，必须考虑因公式重新运行导致的全局变量值的变化。

写数据库文件

- SetTBProfileString 写信息文件

  参数1：String strSection --- 指定的信息块的块名

  参数2：String strKey --- 指定的信息的键名

  参数3：String strValue --- 写入的字符串信息

- GetTBProfileString 读信息文件

  参数1：String strSection

  参数2：String strKey

例7_2(加入跟踪止盈）

```
Params
        Numeric Length1(10);
        Numeric Length2(20);
        Numeric Lots(1);
        Numeric ATRLength(20);          // ATR周期
        Numeric TrailStop(2);           // 追踪止损,回撤ATR的倍数
Vars
        NumericSeries MA1;
        NumericSeries MA2;
        NumericSeries ATRValue;
        NumericSeries HiAfterEntry;
        NumericSeries LoAfterEntry;
        Numeric Stopline;
```

```
Begin
        ATRValue = AvgTrueRange(ATRLength);
        MA1 = AverageFC(Close,Length1);
        MA2 = AverageFC(Close,Length2);
        PlotNumeric("MA1",MA1);
        PlotNumeric("MA2",MA2);
        If (MarketPosition <> 1 and MA1[1] > MA2[1])
        {
                Buy(Lots,Open);
        }
        If (MarketPosition <> -1 and MA1[1] < MA2[1])
        {
                SellShort(lots,Open);
        }
```

```
If (MarketPosition == 1 and BarsSinceEntry == 0)
        HiAfterEntry = High;
If (MarketPosition == 1 and BarsSinceEntry >= 1)
        HiAfterEntry = Max(HiAfterEntry,High);
If (MarketPosition == -1 and BarsSinceEntry == 0)
        LoAfterEntry = Low;
If (MarketPosition == -1 and BarsSinceEntry >= 1)
        LoAfterEntry = Min(LoAfterEntry,Low);
if (BarssinceEntry > 0 and MarketPosition == 1)
{
        StopLine = HiAfterEntry[1] - TrailStop * ATRValue[1];
        If (Low <= StopLine)
        {
                Sell(0, Min(Open, Stopline));
        }
}
```

```
        Else If(BarsSinceEntry > 0 and MarketPosition == -1)
    {
            StopLine = LoAfterEntry[1] + TrailStop * ATRValue[1];
            If(High >= StopLine)
            {
                    BuyToCover(0, Max(Open, Stopline));
            }
    }
End
```

例7_2中存在的问题

- 跟踪止盈出场后，有可能均线的排列趋势并没有改变，会导致出场后又立即进场，破坏了规则的完整性。

完善交易规则

- 增加跟踪止盈出场后的状态记录，以避免出场后，立即再进场；
- 增加再进场的条件，避免大趋势的踏空；
- 我们以价格突破出场前高点或低点，作为再进场的条件，来完善我们的代码。

例7_3(完整的代码）

```
Params
        Numeric Length1(10);
        Numeric Length2(20);
        Numeric Lots(1);
        Numeric ATRLength(20);          // ATR周期
        Numeric TrailStop(2);           // 追踪止损,回撤ATR的倍数
Vars
        NumericSeries MA1;
        NumericSeries MA2;
        NumericSeries ATRValue;
        NumericSeries HiAfterEntry;
        NumericSeries LoAfterEntry;
        Numeric Stopline;
        BoolSeries bLongStoped(false);
        BoolSeries bShortStoped(false);
```

Begin

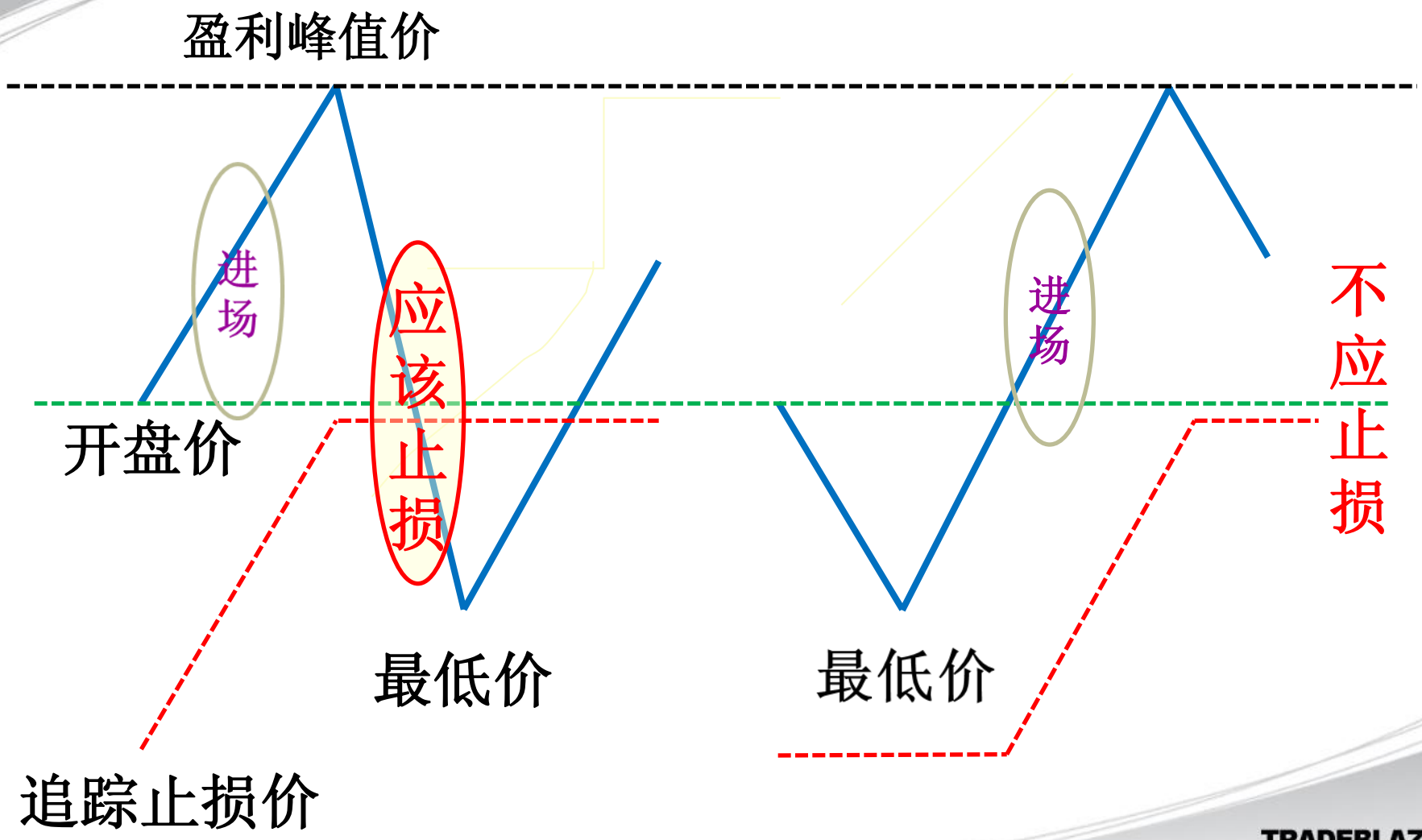
```
ATRValue = AvgTrueRange(ATRLength);
MA1 = AverageFC(Close,Length1);
MA2 = AverageFC(Close,Length2);
PlotNumeric("MA1",MA1);
PlotNumeric("MA2",MA2);
If (!bLongStoped and MarketPosition <> 1 and MA1[1] > MA2[1])
{
        Buy(Lots,Open);
        bShortStoped = false;
}
If (!bShortStoped and MarketPosition <> -1 and MA1[1] < MA2[1])
{
        SellShort(lots,Open);
        bLongStoped = false;
}
```

```
If (bLongStoped and MarketPosition <> 1 and High >= HiAfterEntry)
{
        Buy(Lots, Max(Open, HiAfterEntry));
        bLongStoped = false;
}
If (bShortStoped and MarketPosition <> -1 and Low <= LoAfterEntry)
{
        SellShort(Lots, Min(Open, LoAfterEntry));
        bShortStoped = false;
}
If (MarketPosition == 1 and BarsSinceEntry == 0) HiAfterEntry = High;
If (MarketPosition == 1 and BarsSinceEntry >= 1)
        HiAfterEntry = Max(HiAfterEntry,High);
If (MarketPosition == -1 and BarsSinceEntry == 0) LoAfterEntry = Low;
If (MarketPosition == -1 and BarsSinceEntry >= 1)
        LoAfterEntry = Min(LoAfterEntry,Low);
```

```
if (BarssinceEntry > 0 and MarketPosition == 1)
{
        StopLine = HiAfterEntry[1] - TrailStop * ATRValue[1];
        If (Low <= StopLine)
        {
                Sell(0, Min(Open, Stopline));
                bLongStoped = true;
        }
} Else If(BarsSinceEntry > 0 and MarketPosition == -1)
{
        StopLine = LoAfterEntry[1] + TrailStop * ATRValue[1];
        If(High >= StopLine)
        {
                BuyToCover(0, Max(Open, Stopline));
                bShortStoped = true;
        }
}
```

End

进场位置决定是否应该止损



TB用户函数

- 用户函数是可以通过名称进行调用的一组语句的集合，实际应用中一般将某些经常需要用到的功能做成用户函数以方便以后编程时调用；
- 用户函数一般有一个返回值，类型可以是三种基本类型之一；
- 用户函数通过参数传入数据，通过返回值或引用型变量返回值；
- 用户函数间可以相互调用，也可以递归调用；
- 用户函数分为内建用户函数和其他用户函数，内建用户函数可以查看和调用，不能修改；
- 用户函数实例：summation,Extremes

序列函数

- 序列函数是一种特殊的用户函数，当它的参数或变量中使用了序列变量时，我们就称之为序列函数；
- 序列数据作为普通计算机语言和TB语言的重要区别，是进行金融序列数据计算的的核心；
- 为保证序列数据的正确计算，序列函数需要每个BAR都调用，否则序列函数中的序列数据会不正确；
- 除非算法需要，否则建议不要在条件语句内、循环语句内以及包含逻辑运算符的条件表达式中使用序列函数。

把技术指标改写为用户函数

- TB的技术指标源代码是公开的；
- 所以编写一个基于技术指标的交易系统在TB中是非常简单的
  - 第一步，复制技术指标的代码，粘贴到新建的公式应用中；
  - 第二步，增加交易部分的代码
- 为了便于以后的调用，我们也可以把技术指标改写成用户函数的形式，以方便写公式时调用；
- 我们下面以RSI指标来举例;

myRSI用户函数

```
Params
        Numeric Length(14) ;
Vars
        NumericSeries NetChgAvg( 0 );
        NumericSeries TotChgAvg( 0 );
        Numeric SF( 0 );
        Numeric Change( 0 );
        Numeric ChgRatio( 0 ) ;
        Numeric RSIValue;
Begin
        If(CurrentBar <= Length - 1)
        {
                NetChgAvg = ( Close - Close[Length] ) / Length ;
                TotChgAvg = Average( Abs( Close - Close[1] ), Length ) ;
        }Else
```

```
        {
                SF = 1/Length;
                Change = Close - Close[1] ;
                NetChgAvg = NetChgAvg[1] + SF * ( Change - NetChgAvg[1] ) ;
                TotChgAvg = TotChgAvg[1] + SF * ( Abs( Change ) - TotChgAvg[1] ) ;
        }
        If( TotChgAvg <> 0 )
        {
                ChgRatio = NetChgAvg / TotChgAvg;
        }else
        {
                ChgRatio = 0 ;
        }
        RSIValue = 50 * ( ChgRatio + 1 );
        Return RSIValue;
End
```

RSI指标交易系统

```
Params
        Numeric Length(14) ;
        Numeric Lots(1);
Vars
        NumericSeries RSIValue;
Begin
        RSIValue = myRSI(Length);
        PlotNumeric("RSI",RSIValue);
        if (Marketposition <> 1 and RSIValue[1] > 50)
        {
                Buy(lots,open);
        }
        if (Marketposition <> -1 and RSIValue[1] < 50)
        {
                SellShort(Lots,open);
        }
End
```

SAR 指标交易系统

```
Params
        Numeric AfStep( 0.02);
        Numeric AfLimit( 0.2 ) ;
        Numeric Lots(1);
Vars
        Numeric oParCl( 0 );
        Numeric oParOp( 0 );
        Numeric oPosition( 0 );
        Numeric oTransition( 0 );
        NumericSeries ParOp;
        NumericSeries ParCl;
Begin
        ParabolicSAR( AfStep, AfLimit, oParCl, oParOp, oPosition, oTransition ) ;
        PlotNumeric( "ParCl" , oParCl) ;
        ParOp = oParOp;
        ParCl = oParCl;
```

```
        if (MarketPosition <> 1 and High[1] < ParCl[1] and High > ParOp[1])
        {
                Buy(Lots, Max(Open, ParOp[1] + MinMove * PriceScale));
        }
        if (MarketPosition <> -1 and Low[1] > ParCl[1] and Low < ParOp[1])
        {
                SellShort(Lots, Min(Open, ParOp[1] - MinMove * PriceScale));
        }
End
```

MACD指标交易系统

```
Params
        Numeric FastLength(12);
        Numeric SlowLength(26);
        Numeric MACDLength(9);
        Numeric Lots(1);
Vars
        NumericSeries MACDValue;
        NumericSeries AvgMACD;
        Numeric MACDDiff;
Begin
        MACDValue = XAverage( Close, FastLength ) - XAverage( Close,
SlowLength ) ;
        AvgMACD = XAverage(MACDValue,MACDLength);
        MACDDiff = MACDValue - AvgMACD;
        PlotNumeric("MACD",MACDValue);
        PlotNumeric("MACDAvg",AvgMACD);
```

```
If (MACDDiff >= 0)
        PlotNumeric("MACDDiff",MACDDiff,0,Red);
Else
        PlotNumeric("MACDDiff",MACDDiff,0,Green);
PlotNumeric("零线",0);

If(MarketPosition != 1 And MACDValue[1] > AvgMACD[1] And
MACDValue[1] > 0)
{
        Buy(Lots,Open);
}
If(MarketPosition == 1 And MACDValue[1] < AvgMACD[1])
{
        Sell(Lots,Open);
}
```

```
        If(MarketPosition != -1 And MACDValue[1] < AvgMACD[1] And
MACDValue[1] <0)
        {
                SellShort(Lots,Open);
        }
        If(MarketPosition == -1 And MACDValue[1] > AvgMACD[1])
        {
                BuyToCover(Lots,Open);
        }

End
```

谢谢大家！