

## 《系统仿真与 matlab》综合试题

题 目： 战争模型

编 号： 30

姓 名 罗亚文

班 级 人工智能本硕博 2001 班

学 号 U202015237

电话号码 15229571933

电子邮箱 1272773750@qq.com

得分项	创新性	工作量	代码可读性	报告写作	总分
分数					

# 目录

《系统仿真与 matlab》综合试题.....	1
1 引言.....	3
2 模型建立.....	3
2.1 模型假设.....	3
2.2 符号说明.....	3
2.3 模型建立.....	4
2.3.1 正规战.....	4
2.3.2 游击战.....	5
2.3.3 混合战争模型.....	5
3 模型理论求解.....	6
3.1 正规战.....	6
3.2 游击战.....	8
3.3 混合战.....	9
4 系统仿真过程.....	11
4.1 仿真流程.....	11
4.2 仿真关键.....	13
4.2.1 Matlab 仿真界面的设计.....	13
4.2.2 数学模型的仿真.....	13
4.2.3 数学模型的求解.....	13
4.2.4 仿真参数的输入.....	14
4.2.5 仿真模式的选择.....	14
4.2.6 战争场景动态仿真.....	14
4.2.7 仿真结果输出.....	14
4.3 仿真难点.....	15
4.3.1 数值仿真方法选取.....	15
4.3.2 参数传递.....	15
5 程序运行指南.....	15
5.1 程序文件介绍.....	15
5.2 运行方式介绍.....	16
6 仿真模型的亮点.....	16
7 程序运行实例分析.....	16
7.1 正规战实例.....	19
7.2 游击战实例.....	27
7.3 混合战实例.....	30
8 心得体会.....	32
参考文献.....	33
部分代码附录.....	33

# 1 引言

影响一个军队战斗力的因素是多方面的，比如士兵人数、单个士兵的作战素质以及部队的军事装备，而具体到一次战争的胜负，部队采取的作战方式同样至关重要，此时作战空间同样成为讨论一个作战部队整体战斗力的一个不可忽略的因素。

第一次世界大战 Lanchester 提出预测战役结局的模型，战争分为正规战争，游击战争，混合战争三种类型。为了简化分析，本此仿真仅考虑双方兵力多少和战斗力强弱，并假设兵力因战斗及非战斗减员而减少，因增援而增加，战斗力与射击次数以及命中率有关。<sup>[1]</sup>

## 2 模型建立

### 2.1 模型假设

假设 1：假设交战双方每方战斗减员率取决于双方的兵力和战斗力，且每方非战斗减员率与本方兵力成正比。

假设 2：增援率通常取决于一个已经投入战争部队意外的因素，主观性过强，影响因素过多，本次仿真为了简化仿真模型，不考虑增援率的影响，即设置双方增援率均为 0。

假设 3：双方均采用游击战的情况下，甲方战斗减员率还随着甲方兵力的增加而增加。

### 2.2 符号说明

仿真过程中所用符号说明如下：

符号	符号说明
$x_0, y_0$	甲乙交战双方在时刻 0 的初始兵力。
$x(t), y(t)$	甲乙交战双方在时刻 t 的兵力，视为双方的士兵人数。

$f(x, y), g(x, y)$	甲乙双方的战斗减员率，取决于双方的兵力。
$\alpha, \beta > 0$	甲乙双方非战斗减员率，比如由于疾病或逃跑等因素导致一个部队减员。
$u(t), v(t)$	增援率，通常取决于一个已投入战争部队以外的因素。
$r_x, r_y$	甲乙双方单个士兵的射击率，通常主要取决于部队的武器装备。
$p_x, p_y$	甲乙双方士兵一次射击的（平均）命中率。
$s_x, s_y$	甲乙双方的有效活动区域的面积。

假设一场战争中有甲乙俩方部队。则易得：

$$\begin{cases} \frac{dx}{dt} = f(x, y, t) - \alpha x(t) + u(t), \alpha > 0 \\ \frac{dy}{dt} = g(x, y, t) - \beta y(t) + v(t), \beta > 0 \\ x(0) = x_0, y(0) = y_0 \end{cases}$$

其中， $f(x, y), g(x, y)$ 主要取决于战争类型。下面根据战争类型对模型进行分类讨论：

## 2.3 模型建立

### 2.3.1 正规战

正规战情形：甲乙双方均以正规部队作战，每一方士兵的活动均公开，处于对方士兵的监视与杀伤范围之内，一旦一方的某个士兵被杀伤，对方的火力立即转移到其他士兵身上。

若双方均以正规部队作战，甲方的战斗减员率只取决于乙方的兵力和战斗力，则 $f(x, y, t) = -r_y * p_y * y(t)$ ，乙方的战斗减员率只取决于甲方的兵力和战斗力，则 $g(x, y, t) = -r_x * p_x * x(t)$ 。建立模型如下：

$$\begin{cases} \frac{dx}{dt} = -r_y * p_y * y(t) - \alpha x(t) + u(t), \alpha > 0 \\ \frac{dy}{dt} = -r_x * p_x * x(t) - \beta y(t) + v(t), \beta > 0 \\ x(0) = x_0, y(0) = y_0 \end{cases}$$

根据假设 2，原模型可以简化如下：

$$\begin{cases} \frac{dx}{dt} = -r_y * p_y * y(t) - \alpha x(t), \alpha > 0 \\ \frac{dy}{dt} = -r_x * p_x * x(t) - \beta y(t), \beta > 0 \\ x(0) = x_0, y(0) = y_0 \end{cases}$$

### 2.3.2 游击战

游击战情形：甲乙双方均以游击作战方式，每一方士兵的活动均具有隐蔽性，对方的射击行为局限在某个范围考虑可以被认为是盲目的。

根据假设 3 可知，乙方射击的准确程度与乙方的有效区域活动面积成正比，与甲方的有效区域活动面积成反比，则有： $f(x, y, t) = -r_y * p_y * y(t) * x(t)$ ，其中，

$$p_y = s_y / s_x。$$

甲方射击的准确程度与甲方的有效区域活动面积成正比，与乙方的有效区域活动面积成反比，则有： $g(x, y, t) = -r_x * p_x * y(t) * x(t)$ ，其中， $p_x = s_x / s_y。$

$$\begin{cases} \frac{dx}{dt} = -r_y * p_y * y(t) * x(t) - \alpha x(t) + u(t), \alpha > 0 \\ \frac{dy}{dt} = -r_x * p_x * y(t) * x(t) - \beta y(t) + v(t), \beta > 0 \\ x(0) = x_0, y(0) = y_0 \end{cases}$$

根据假设 2，游击战的数学模型可以简化如下：

$$\begin{cases} \frac{dx}{dt} = -r_y * (s_y / s_x) * y(t) * x(t) - \alpha x(t), \alpha > 0 \\ \frac{dy}{dt} = -r_x * (s_x / s_y) * y(t) * x(t) - \beta y(t), \beta > 0 \\ x(0) = x_0, y(0) = y_0 \end{cases}$$

### 2.3.3 混合战争模型

混合战情形：正规战和游击战两种情形的混合。

不妨甲方为游击部队，乙方为正规部队，则与上述分析相似，可得，甲方的战斗减员率  $f(x,y,t) = -r_y * p_y * y(t) * x(t)$ ，乙方的战斗减员率只取决于甲方的兵力和战斗力，则  $g(x,y,t) = -r_x * p_x * x(t)$ ，则有：

$$\begin{cases} \frac{dx}{dt} = -r_y * p_y * y(t) * x(t) - \alpha x(t) + u(t), \alpha > 0 \\ \frac{dy}{dt} = -r_x * p_x * x(t) - \beta y(t) + v(t), \beta > 0 \\ x(0) = x_0, y(0) = y_0 \end{cases}$$

根据假设 2，游击战的数学模型可以简化如下：

$$\begin{cases} \frac{dx}{dt} = -r_y * (s_y/s_x) * y(t) * x(t) - \alpha x(t), \alpha > 0 \\ \frac{dy}{dt} = -r_x * p_x * x(t) - \beta y(t), \beta > 0 \\ x(0) = x_0, y(0) = y_0 \end{cases}$$

### 3 模型理论求解

#### 3.1 正规战

数学模型：

$$\begin{cases} \frac{dx}{dt} = -r_y * p_y * y(t) - \alpha x(t), \alpha > 0 \\ \frac{dy}{dt} = -r_x * p_x * x(t) - \beta y(t), \beta > 0 \\ x(0) = x_0, y(0) = y_0 \end{cases}$$

由于该常微分线性方程组求解结果过于复杂，不妨假设  $\alpha = \beta = 0$ ，即每只军队只考虑战争因素导致的死亡，则数学模型可以进一步简化如下：

$$\begin{cases} \frac{dx}{dt} = -r_y * p_y * y(t), \alpha > 0 \\ \frac{dy}{dt} = -r_x * p_x * x(t), \beta > 0 \\ x(0) = x_0, y(0) = y_0 \end{cases}$$

使用第二个式子除以第一个式子，得到可分离变量的微分方程：

$$\frac{dy}{dx} = \frac{r_x * p_x * x(t)}{r_y * p_y * y(t)}$$

即：  $r_y * p_y * y(t) dy = r_x * p_x * x(t) dx$

对俩端进行积分可得： $(r_y * p_y)y(t)^2 - (r_x * p_x)x(t)^2 = k$

将初始值带入可得： $k = (r_y * p_y)y_0^2 - (r_x * p_x)x_0^2$

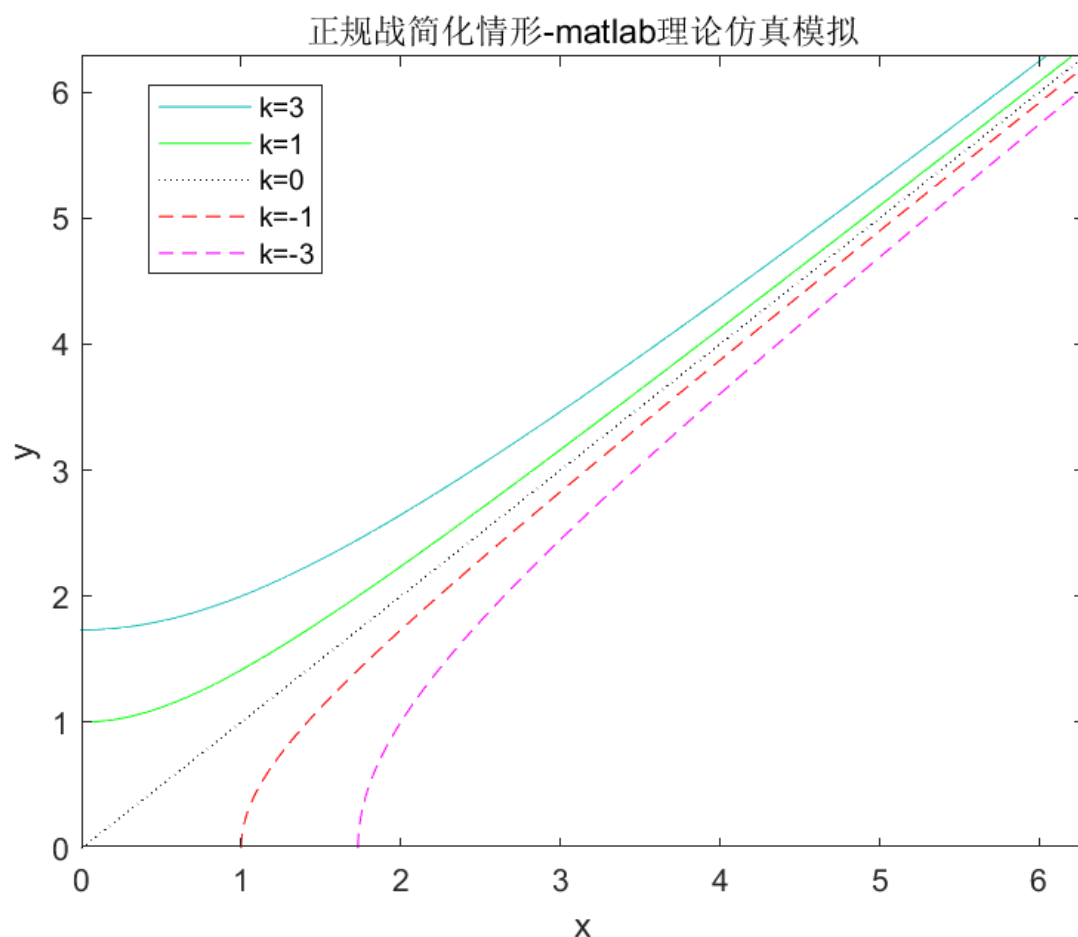
带入 $k$ 可得： $(r_y * p_y)y(t)^2 - (r_x * p_x)x(t)^2 = (r_y * p_y)y_0^2 - (r_x * p_x)x_0^2$

俩端同时除以 $r_y * p_y * r_x * p_x$  ,不难得到：

$$\frac{y(t)^2}{r_x * p_x} - \frac{x(t)^2}{r_y * p_y} = \frac{(r_y * p_y)y_0^2 - (r_x * p_x)x_0^2}{r_y * p_y * r_x * p_x}$$

其中， $r_x * p_x > 0$ 且 $r_y * p_y > 0$ 。

下面用 matlab 做出双曲线的图：



根据双曲线的基本性质以及理论仿真结果，不难看出，当 $k = (r_y * p_y)y_0^2 - (r_x * p_x)x_0^2 > 0$ 时， $y > 0, x = 0$ ，则乙方获得最终的胜利，当 $k = (r_y * p_y)y_0^2 - (r_x * p_x)x_0^2 < 0$ 时， $x > 0, y = 0$ ，则甲方获得最终的胜利。当 $k = (r_y * p_y)y_0^2 -$

$(r_x * p_x)x_0^2 = 0$ 时,  $x = 0, y = 0$ , 则双方取得平局。

由此可见, 决定双方谁取胜的因素为  $k$  的正负, 即  $(r_y * p_y)y_0^2$  与  $(r_x * p_x)x_0^2$  的大小, 当  $(r_y * p_y)y_0^2 > (r_x * p_x)x_0^2$  时, 乙方获胜, 当  $(r_y * p_y)y_0^2 < (r_x * p_x)x_0^2$  时, 甲方获胜, 当  $(r_y * p_y)y_0^2 = (r_x * p_x)x_0^2$  时, 双方取得平局。  $r_i * p_i$  中,  $r_i$  代表单个士兵的设计率,  $p_i$  代表一次射击的评论准确率,  $r_i * p_i$  代表单个士兵的有效杀伤率, 因此,  $r_i * p_i$  高的队伍取得最终胜利, 是很合理的理论仿真结果。

### 3.2 游击战

数学模型:

$$\begin{cases} \frac{dx}{dt} = -r_y * (s_y/s_x) * y(t) * x(t) - \alpha x(t), \alpha > 0 \\ \frac{dy}{dt} = -r_x * (s_x/s_y) * y(t) * x(t) - \beta y(t), \beta > 0 \\ x(0) = x_0, y(0) = y_0 \end{cases}$$

由于该常微分线性方程组求解结果过于复杂, 不妨假设  $\alpha = \beta = 0$ , 即每只军队只考虑战争因素导致的死亡, 则数学模型可以进一步简化如下:

$$\begin{cases} \frac{dx}{dt} = -r_y * (s_y/s_x) * y(t) * x(t) \\ \frac{dy}{dt} = -r_x * (s_x/s_y) * y(t) * x(t) \\ x(0) = x_0, y(0) = y_0 \end{cases}$$

使用第二个式子除以第一个式子, 得到可分离变量的微分方程:

$$\frac{dy}{dx} = \frac{r_y * s_y/s_x}{r_x * s_x/s_y}$$

求解微分方程, 可得:

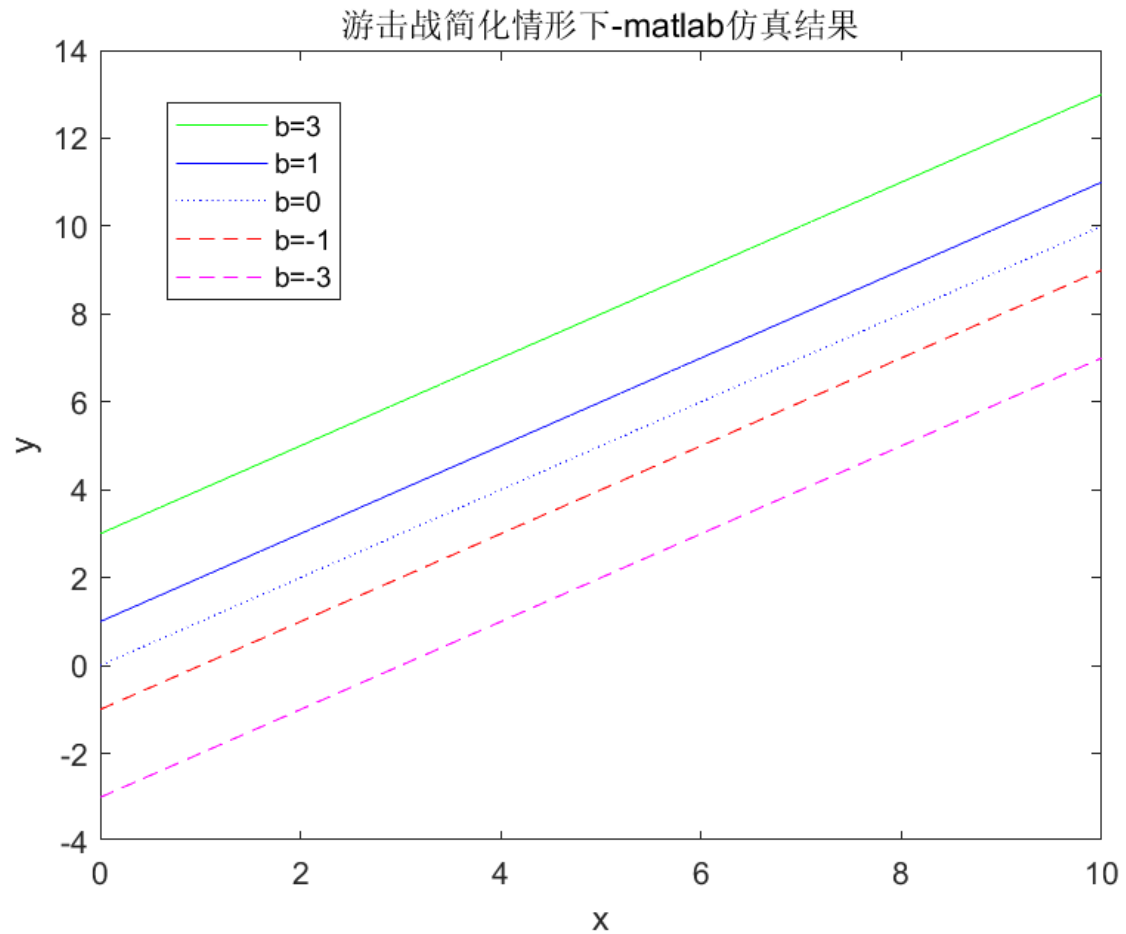
$$y(t) = \frac{r_y * s_y/s_x}{r_x * s_x/s_y} x + b$$

将初值  $x(0) = x_0, y(0) = y_0$  带入即可得到:

$$b = y_0 - \frac{r_y * s_y/s_x}{r_x * s_x/s_y} * x_0$$

不难得到, 该一次型曲线的斜率大于 0, 用 matlab 绘制其曲线可得:





根据一次曲线的基本性质以及理论仿真结果，不难看出，当

$b = y_0 - \frac{r_y * s_y / s_x}{r_x * s_x / s_y} * x_0 > 0$  时， $y > 0, x = 0$ ，则乙方获得最终的胜利，当  $b = y_0 - \frac{r_y * s_y / s_x}{r_x * s_x / s_y} * x_0 < 0$  时， $x > 0, y = 0$ ，则甲方获得最终的胜利。当  $b = y_0 - \frac{r_y * s_y / s_x}{r_x * s_x / s_y} * x_0 = 0$  时， $x = 0, y = 0$ ，则双方取得平局。

由此可见，决定双方谁取胜的因素为  $b$  的正负，即  $y_0$  与  $\frac{r_y * s_y / s_x}{r_x * s_x / s_y} * x_0$  的大小，

当  $(r_x * s_x^2) * y_0 > (r_y * s_y^2) * x_0$  时，乙方获胜，当  $(r_x * s_x^2) * y_0 < (r_y * s_y^2) * x_0$  时，甲方获胜，当  $(r_x * s_x^2) * y_0 = (r_y * s_y^2) * x_0$  时，双方取得平局。

### 3.3 混合战

数学模型：

$$\begin{cases} \frac{dx}{dt} = -r_y * (s_y/s_x) * y(t) * x(t) - \alpha x(t), \alpha > 0 \\ \frac{dy}{dt} = -r_x * p_x * x(t) - \beta y(t), \beta > 0 \\ x(0) = x_0, y(0) = y_0 \end{cases}$$

由于该常微分线性方程组求解结果过于复杂，不妨假设  $\alpha = \beta = 0$ ，即每只军队只考虑战争因素导致的死亡，则数学模型可以进一步简化如下：

$$\begin{cases} \frac{dx}{dt} = -r_y * (s_y/s_x) * y(t) * x(t) \\ \frac{dy}{dt} = -r_x * p_x * x(t) \\ x(0) = x_0, y(0) = y_0 \end{cases}$$

使用第二个式子除以第一个式子，得到可分离变量的微分方程：

$$\frac{dy}{dx} = \frac{r_x * p_x * s_x}{r_y * s_y * y(t)}$$

即：  $r_y * s_y * y(t) dy = r_x * p_x * s_x dx$

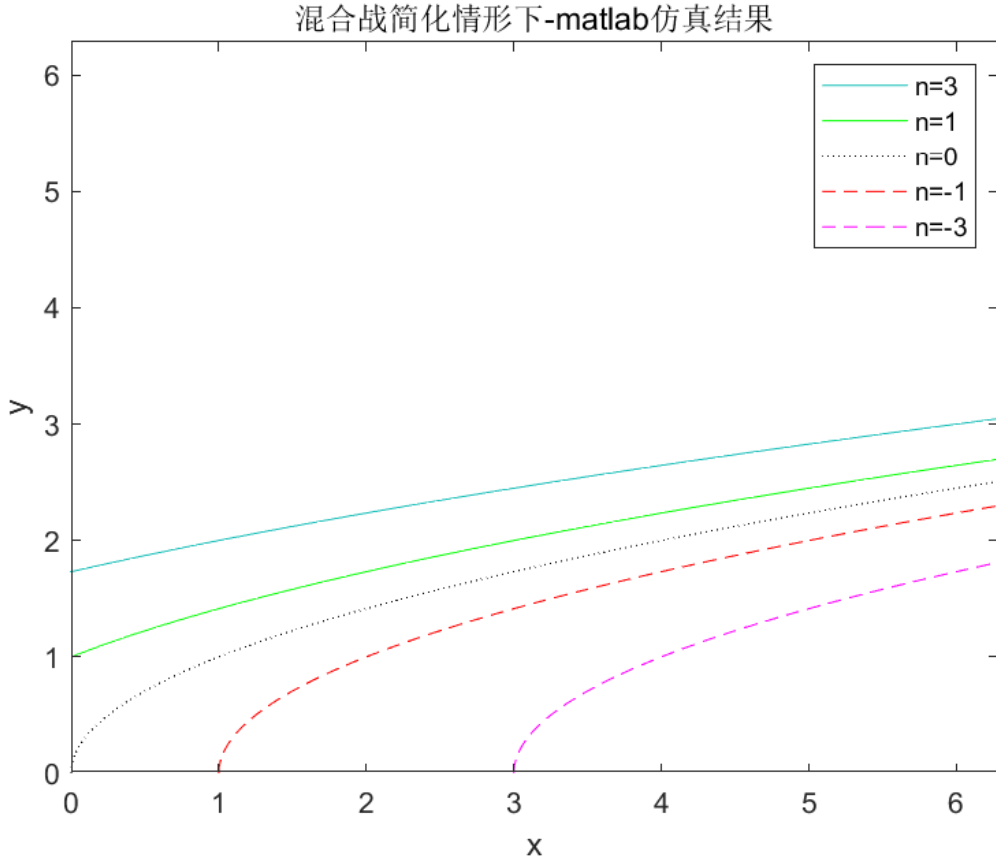
对两端进行积分可得：  $(r_y * s_y)y(t)^2 - (r_x * p_x * s_x)x(t) = n$

将初始值带入可得：  $n = (r_y * s_y)y_0 - (r_x * p_x * s_x)x_0$

带入  $k$  可得：

$$(r_y * s_y)y(t)^2 - (r_x * p_x * s_x)x(t) = (r_y * s_y)y_0 - (r_x * p_x * s_x)x_0$$

用 matlab 绘制其曲线可得：



根据双曲线的基本性质以及理论仿真结果，不难看出，当  $n = (r_y * s_y)y_0 - (r_x * p_x * s_x)x_0 > 0$  时， $y > 0, x = 0$ ，则乙方获得最终的胜利，当  $n = (r_y * s_y)y_0 - (r_x * p_x * s_x)x_0 < 0$  时， $x > 0, y = 0$ ，则甲方获得最终的胜利。当  $n = (r_y * s_y)y_0 - (r_x * p_x * s_x)x_0 = 0$  时， $x = 0, y = 0$ ，则双方取得平局。

由此可见，决定双方谁取胜的因素为  $k$  的正负，即  $(r_y * s_y)y_0$  与  $(r_x * p_x * s_x)x_0$  的大小，当  $(r_y * s_y)y_0 > (r_x * p_x * s_x)x_0$  时，乙方获胜，当  $(r_y * s_y)y_0 < (r_x * p_x * s_x)x_0$  时，甲方获胜，当  $(r_y * s_y)y_0 = (r_x * p_x * s_x)x_0$  时，双方达到了平局。

## 4 系统仿真过程

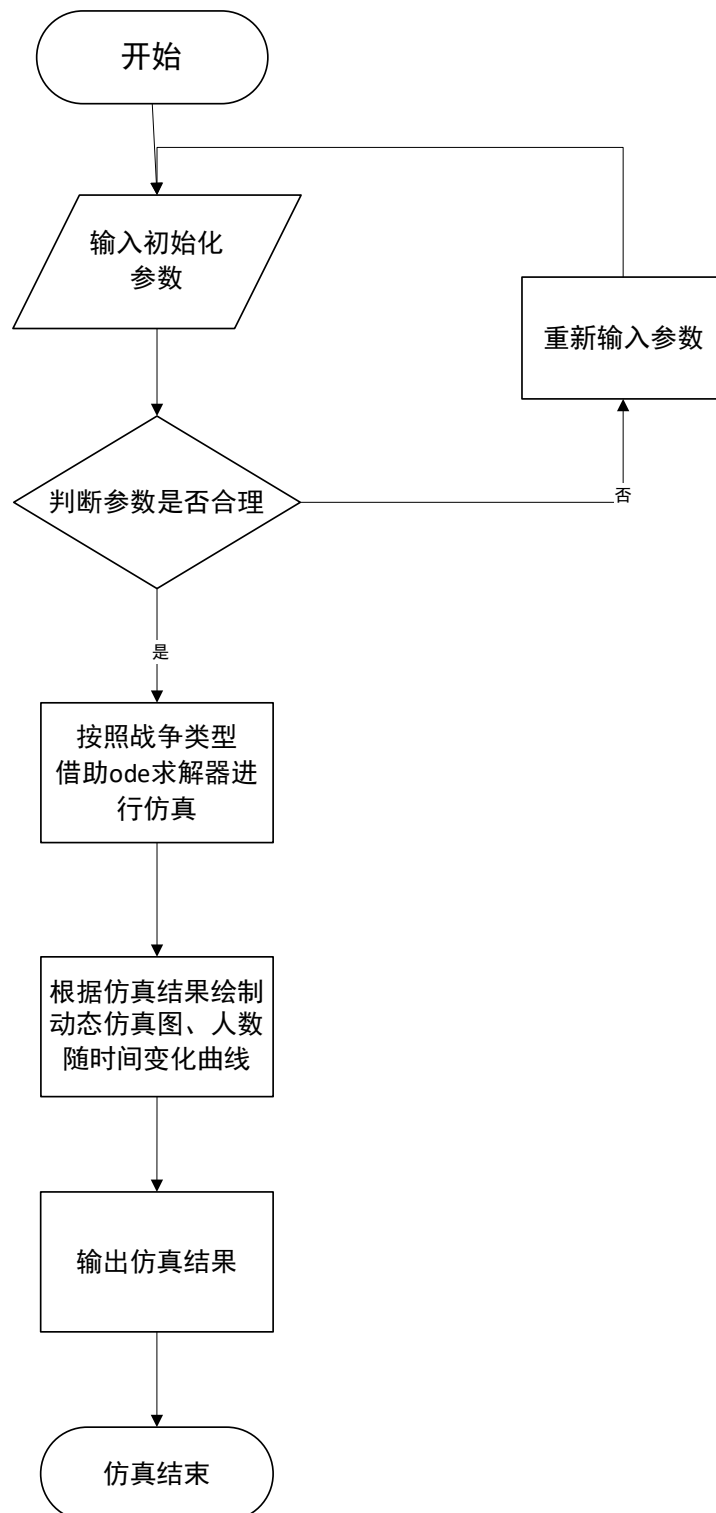
### 4.1 仿真流程

仿真过程的流程主要有：

- (1) 参数初始化步骤：用户输入仿真所需要的参数，对参数的合理性进行检验

后，初始化对应的参数。

- (2) 仿真求解步骤：根据所选择的战争类型对应的数学模型，使用 ode45 变步长仿真求解器来求解所建立的数学模型（带有初始化值的常微分方程组）。
- (3) 结果输出步骤：根据求解器求解出的结果，绘制战争动态仿真图以及人数-时间变化曲线图。



## 4.2 仿真关键

### 4.2.1 Matlab 仿真界面的设计

通过 guide 工具进行界面的设计，在设计完仿真之后，matlab 会自行产生对应的 m 文件，同时生成回调函数。设计的过程中，使用了 uicontrol 辅助添加了各种界面操作部件，实现仿真参数的输入、仿真结果的可视化输出。核心是编写回调函数以及仿真过程的编写。

### 4.2.2 数学模型的仿真

根据前面所建立的数学模型，在模拟的过程中，需要针对建立的数学模型利用求解器 ode45 进行求解，在实现数学模型的过程中使用三个函数 odefun1、odefun2、odefun3 来对数学模型进行仿真。具体的代码如下：

```
function dydt = odefun1(t,y,rx,px,ry,py,a,b)%正规战的仿真模型
dydt = zeros(2,1);
dydt(1) = -ry*py*y(2)-a*y(1);
dydt(2) = -rx*px*y(1)-b*y(2);
end
```

图 1 正规战数学模型实现

```
function dydt = odefun2(t,y,rx,sx,ry,sy,a,b)%游击战的仿真模型
dydt = zeros(2,1);
dydt(1) = -ry*(sy/sx)*y(2)*y(1)-a*y(1);
dydt(2) = -rx*(sx/sy)*y(1)*y(2)-b*y(2);
end
```

图 2 游击战数学模型实现

```
function dydt = odefun3(t,y,rx,sx,ry,sy,px,a,b)%混合战的仿真模型
dydt = zeros(2,1);
dydt(1) = -ry*(sy/sx)*y(2)*y(1)-a*y(1);
dydt(2) = -rx*(px)*y(1)-b*y(2);
end
```

图 3 混合战数学模型实现

### 4.2.3 数学模型的求解

由于该问题为连续问题，对建立的数学模型的求解采用了 ode45 求解器。为了实现迭代求解，在选取时间间隔时候，并没有直接从 0 取到仿时间的最大限额，而是按照所输入的步长，按照逐步长的方式对 tspan 进行赋值，从而实现迭

代求解，在每一个步长中使用 `ode45` 来进行求解，为了使得输出的结果更加具有连续性（仿真的结果更贴合实际），将 `ode45` 中的每一个仿真点都算入仿真过程中。

#### 4.2.4 仿真参数的输入

题目要求的系统输入包括甲方、乙方的初始兵力 $x_0, y_0$ ；甲方、乙方的非战斗减员率 $\alpha, \beta > 0$ ；甲方、乙方单个士兵的射击率 $r_x, r_y$ ；甲方、乙方士兵单次射击的准确率 $p_x, p_y$ ；甲方、乙方的活动区域面积所占比例 $s_x, s_y$ 。使用 Matlab 中 GUI 中的可编辑文本实现这一功能，可以将每一个可编辑文本中的 Tag 编辑为通俗易懂的变量名，通过对相对应的变量的赋值可以实现用户通过键盘的输入完成参数的输入。

#### 4.2.5 仿真模式的选择

仿真模式有三种选择，一是正规战，二是游击战，三是混合战（甲方为游击战、乙方为正规战）。三者之间的切换使用的是 Matlab GUI 中的单选按钮完成的这一功能，将其中的 Tag 选项变为通俗易懂的标志，通过使用鼠标选择便可以将此数字赋给 Tag 的对应的参数，这样就可以通过鼠标输入完成参数的输入。再通过 `switch` 语句对于此参数进行判断，来进行不同的后面的操作。

#### 4.2.6 战争场景动态仿真

甲乙双方在战场中进行斗争，每个战争时刻甲乙双方的人数都可以通过仿真计算出来，因此，我专门留了一个坐标轴的位置用于简易的战争场景动态仿真。甲方和乙方的士兵数目借助散点图绘制出。双方士兵所占的战场活动区域面积由输入的参数决定，士兵在活动区域的具体位置则通过随机数来确定。

#### 4.2.7 仿真结果输出

在仿真的过程中，界面上通过修改对应的静态文本可以实现实时输出仿真时刻，以及该时刻下甲乙双方的剩余士兵数，在仿真结束后，可以根据甲乙方的人数情况输出作战结果。与此同时，也可以在坐标轴上绘制出战争士兵数目仿真曲

线。

## 4.3 仿真难点

### 4.3.1 数值仿真方法选取

要针对模型进行数字仿真首先要建立被仿真系统的数学模型，并将此模型转换成计算机可以接受的，与原模型等价的仿真模型，然后编制仿真程序，使得模型在计算机上运转。将连续系统的数学模型转换成计算机可以接受的等价仿真模型后，我采用了数值积分法来对所建立的仿真模型进行求解。

目前已经存在较为成熟的数值分析方法，例如欧拉法、梯形法，考虑仿真精度、速度、舍入误差等因素，此次仿真采用了龙格库塔方法进行数值仿真。龙格库塔方法是一种间接利用泰勒展开式的方法，即利用几个点上的一阶导数的函数值的线性组合来近似代替函数在某一点的各阶导数，然后用泰勒级数展开式确定线性组合中的各加权系数，这样既可以避免计算各阶导数，又可以提高数值积分的精度。

此外，在通过输入参数确定了步长之后，在每一个步长之内都采用了 `ode45`（变步长）进行数值仿真求解。将固定步长和变步长组合在一起，这也算本次仿真的一个亮点。

### 4.3.2 参数传递

在仿真的过程中，难点是如何编写每一个控件的回调函数。在回调函数之间如何传递模型的参数以及输出数值便也成了一个问题。本次仿真借助了 GUI 中的 `handles` 在不同的 `callback` 函数之间传递参数。

## 5 程序运行指南

### 5.1 程序文件介绍

此战争模型仿真系统共有 1 个 `fig` 文件：`war_model.fig` 以及其对应的 `m` 文件 `war_model.m`。还有三个数学模型所对应的函数文件：`odefun1.m`、`odefun2.m`、`odefun3.m`。

war\_model.fig: 整个仿真模型的 GUI 界面文件。

odefun1.m: 对应正规战的数学模型。

odefun2.m: 对应游击战的数学模型。

Odefun3.m: 对应混合战的数学模型。

## 5.2 运行方式介绍

运行方式一: 运行 war\_model.m 文件。

运行方式二: 在命令行窗口中输入 war\_model。

运行方式三: 在命令行窗口中输入 guide 后, 在 guide 中打开 war\_model.fig。

进入仿真运行界面后, 有仿真参数的输入以及仿真对象以及仿真模式等的选择, 设置好所有的参数后(参数一定要合理否则会报错, 而且不能为空, 比如双方的士兵初始人数不能为负数), 点击开始仿真按钮开始仿真, 默认为是在默认的系统参数中, 在迭代仿真的过程中, 系统输出双方士兵的人数, 并且有动态仿真界面用于动态表示双方军队士兵人数的变化, 也有绘制界面, 实时绘制双方士兵数目随时间变化的曲线。

## 6 仿真模型的亮点

1.在进行仿真之前, 争对所建立的数学模型进行了合理的简化从而求解出了对应的理论解, 为后面仿真模型的建立奠定了可靠的理论基础。

2.在仿真模型求解过程中, 所采用的求解方法兼顾了固定步长法和变步长法的有点, 在固定步长的基础之上借助了变步长方式的求解。从而兼顾了仿真的快速性以及准确度。

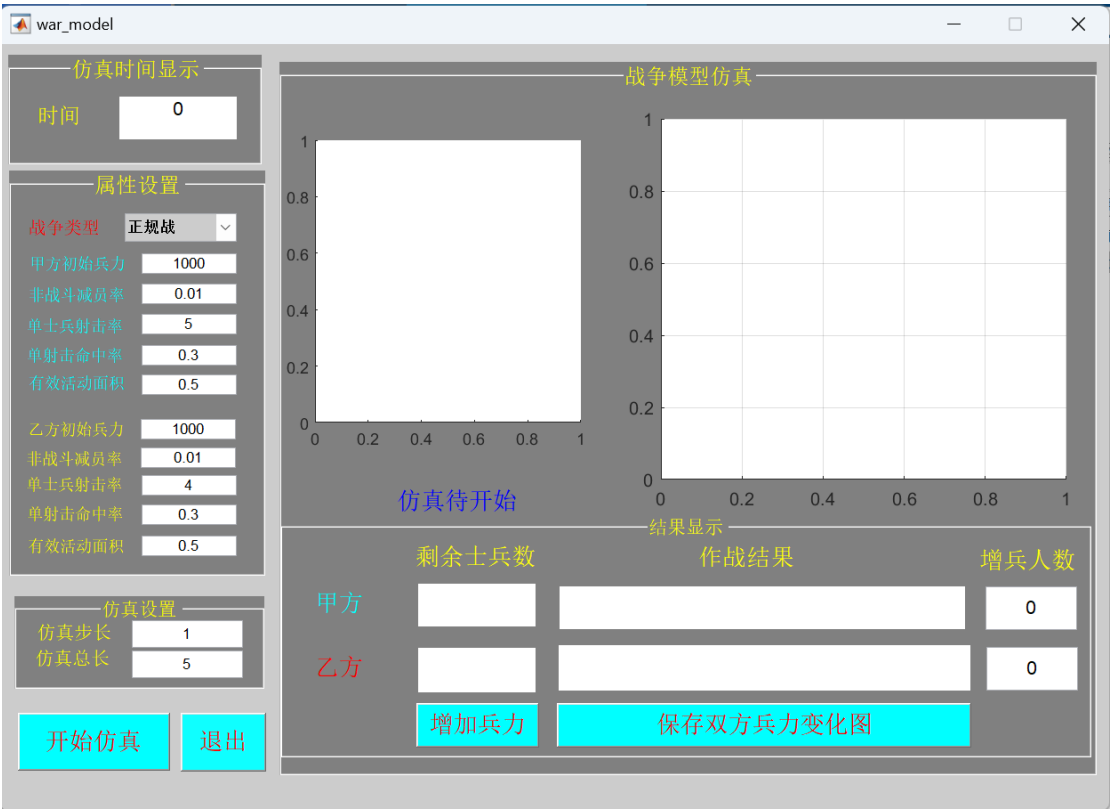
3.在仿真的过程中, 我也建立了一个动态仿真的模型, 就如同一个真正的“战场”一般, 可以事实反映对战双方士兵的剩余数目, 可以直观的反应实时的战争状态。

## 7 程序运行实例分析

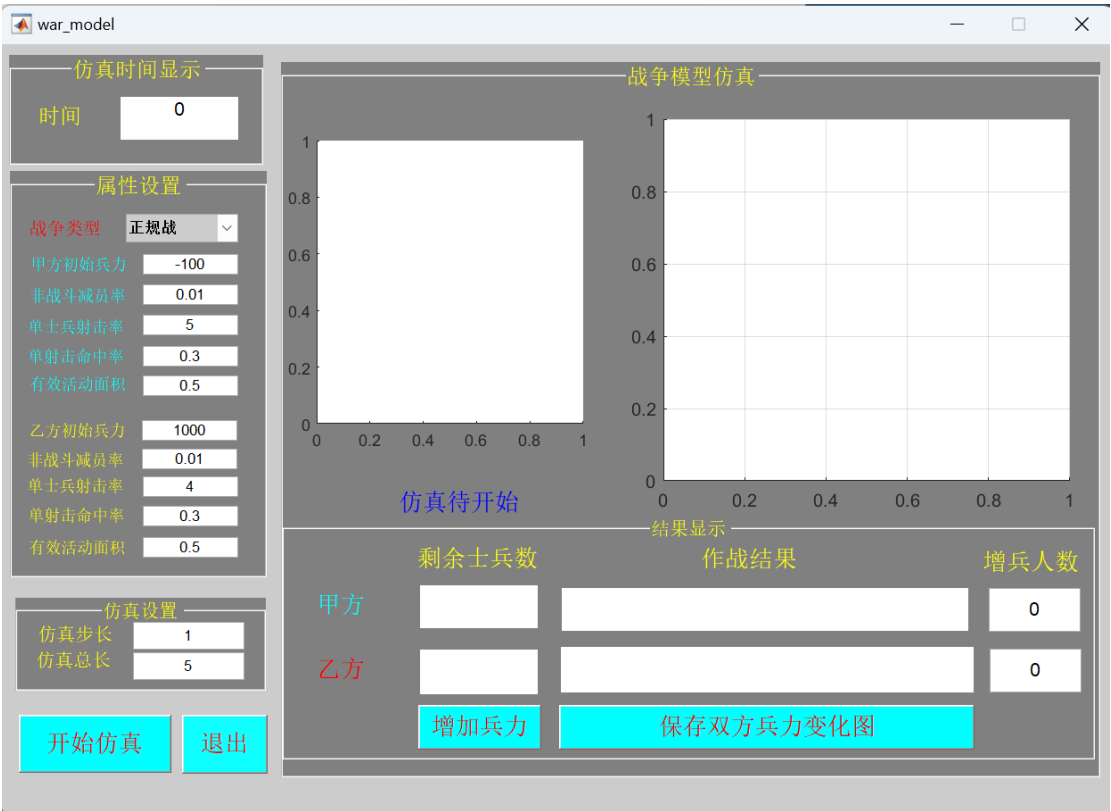
下面介绍仿真程序运行过程, 并截取仿真过程中的部分图片。



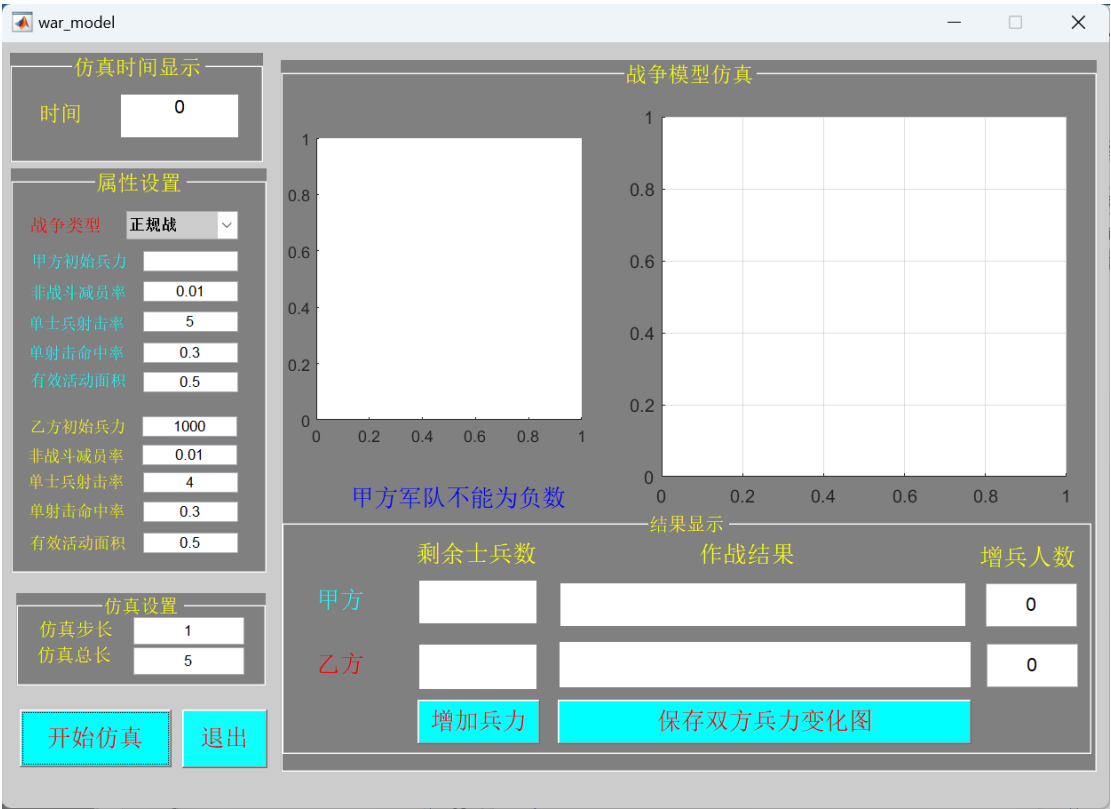
在命令行窗口输入 war\_model 开始运行程序，进入系统仿真界面。



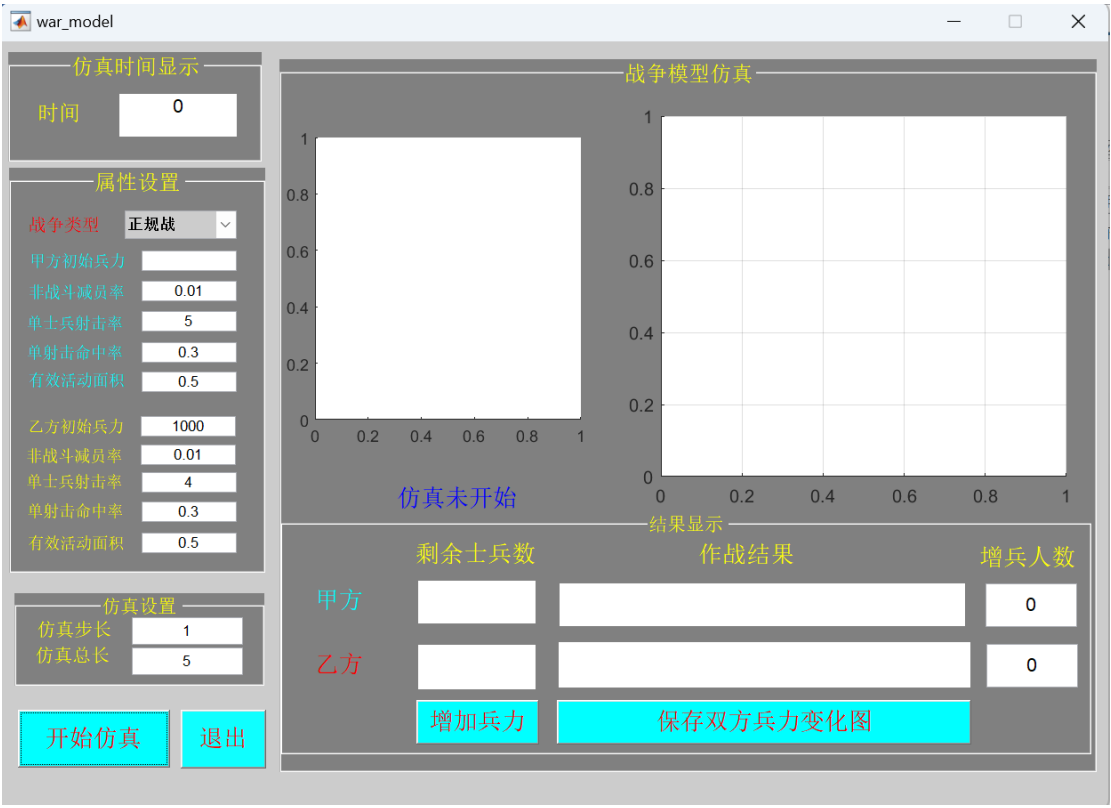
首先，用户应该设置好对应的参数，参数的设置必须合理，否则会提醒用户输入的参数不合理，例如，输入甲方的初始士兵数为负数的时候：



会有对应的提醒--“甲方士兵不能为负数”，并且紧跟着清除对应的输入参数。



清楚对应的输入参数：

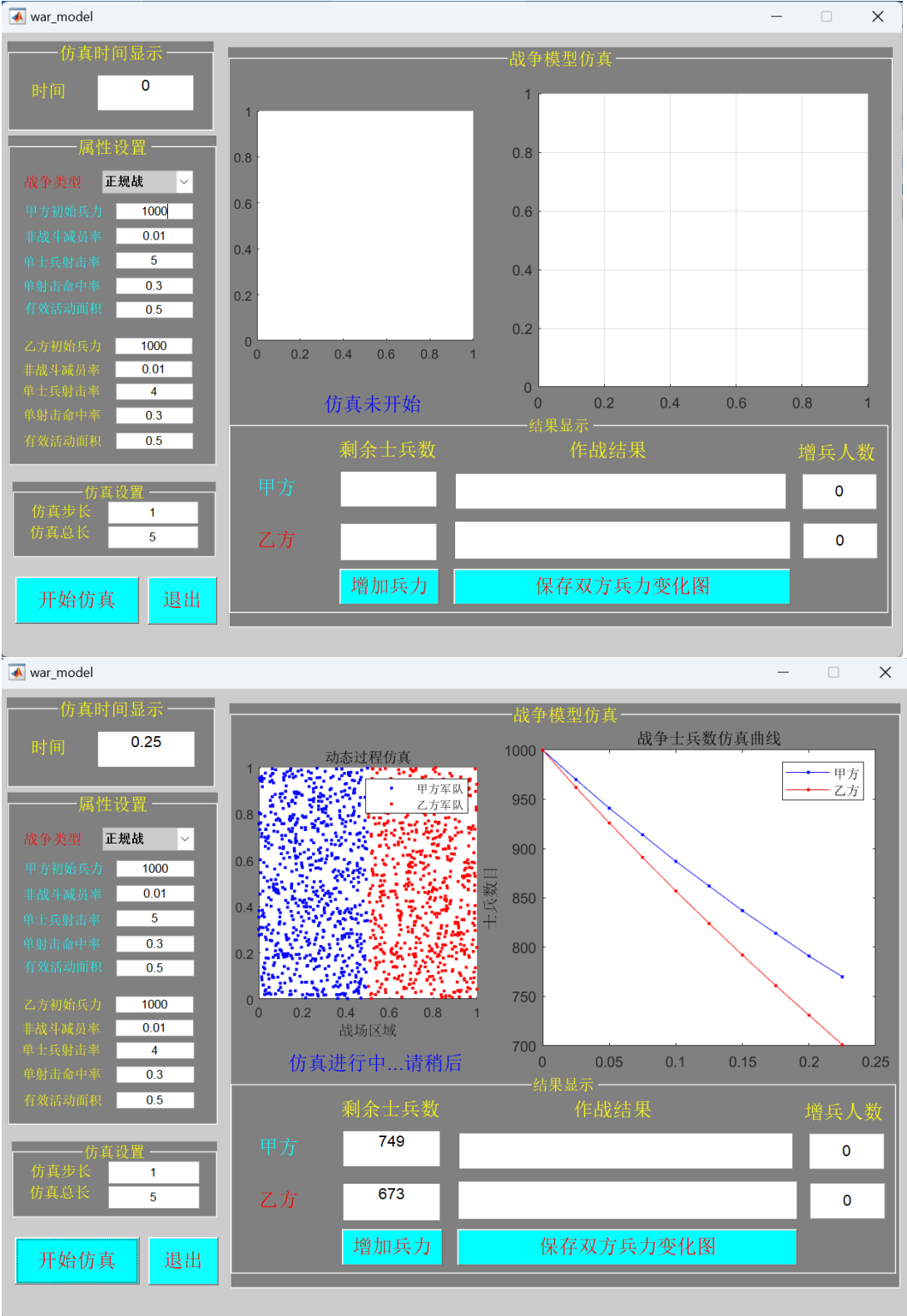


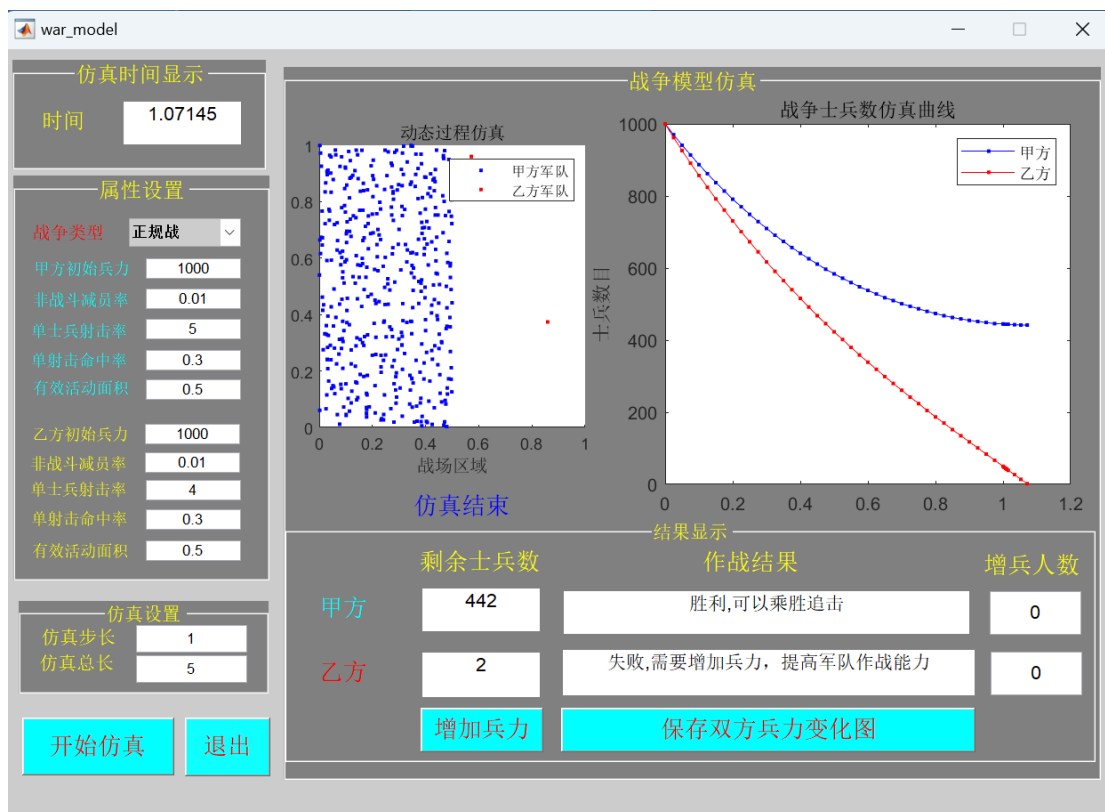
设置好对应的参数后，点击开始仿真按钮，即可以按照对应的战争类型对双方的

交战情况进行仿真。

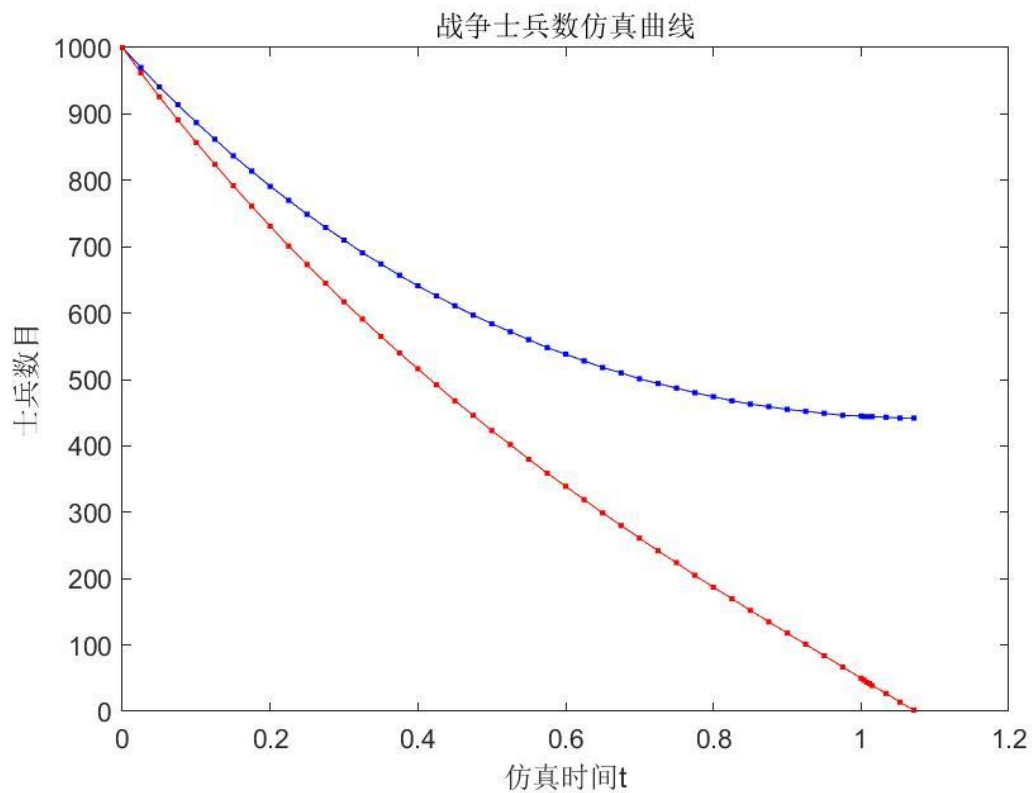
## 7.1 正规战实例

(1) 当输入参数为默认参数时候，正规战开始仿真的结果如下：





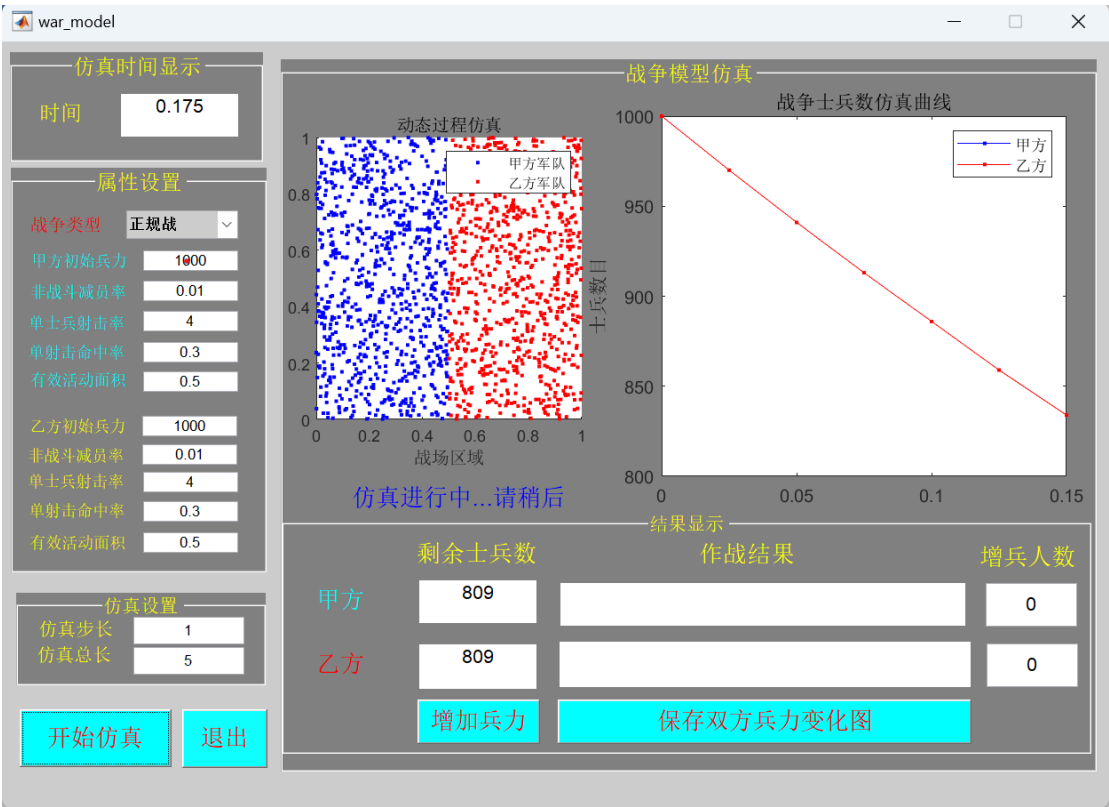
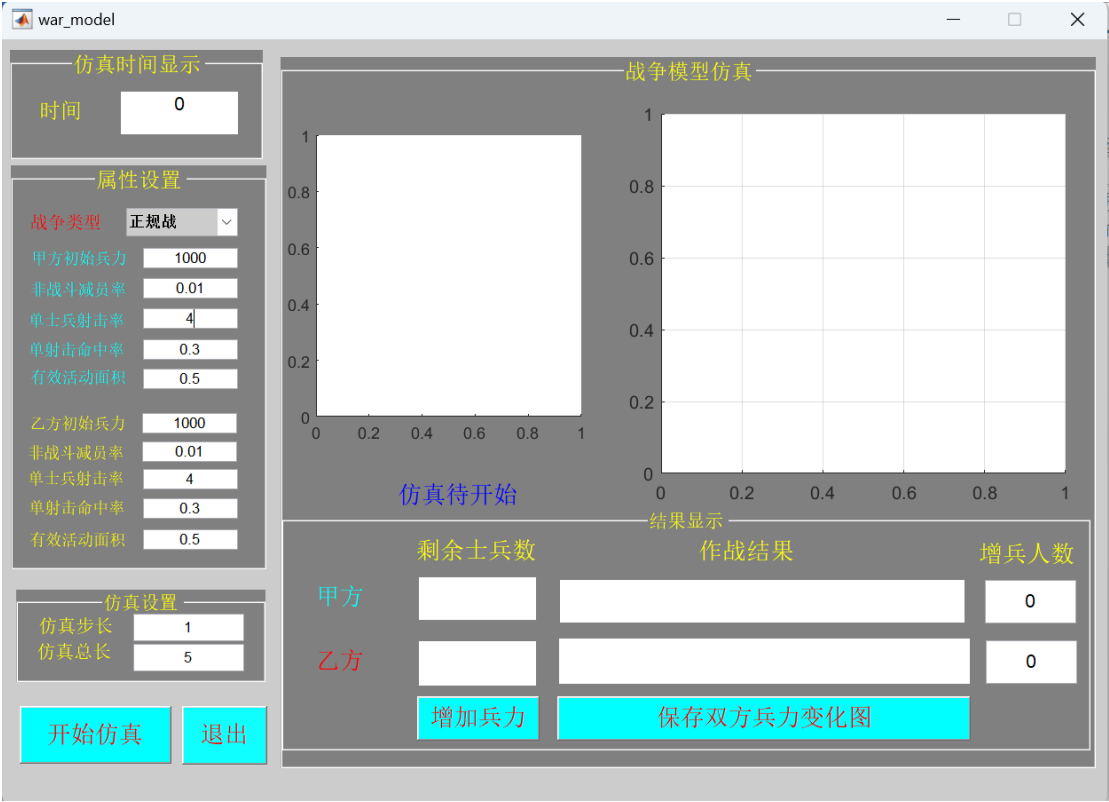
最终的仿真结果曲线为:

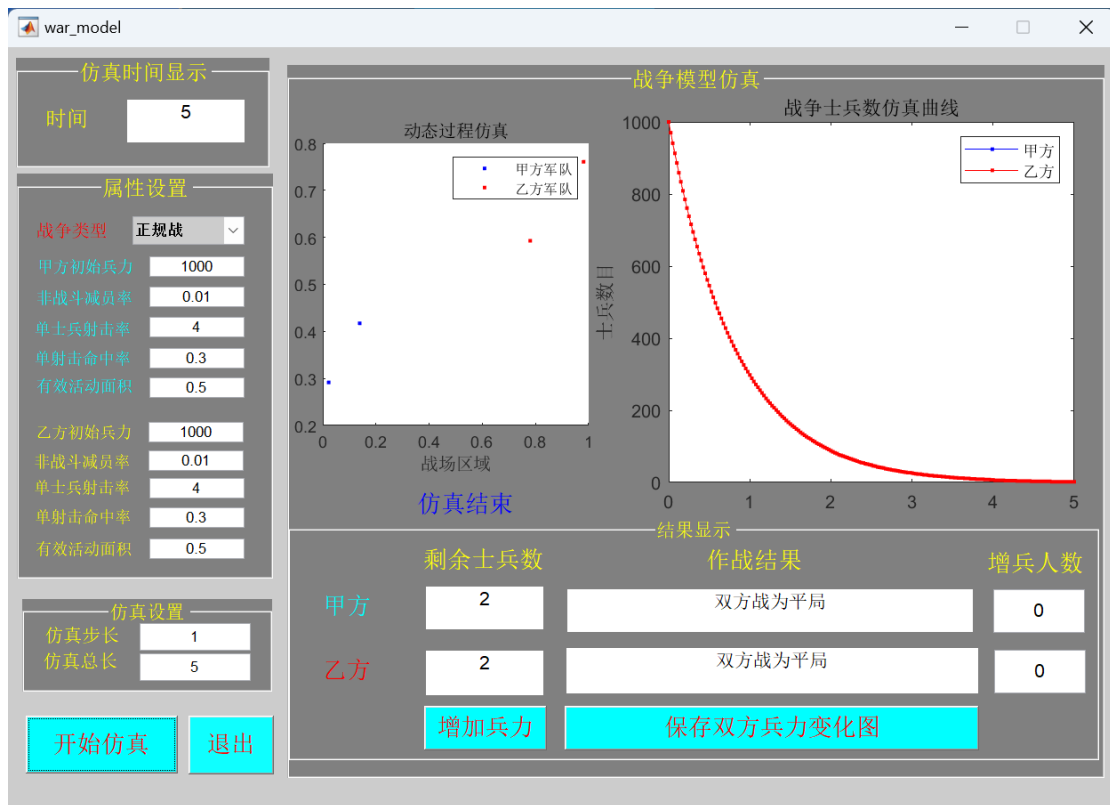


不难看到, 在双方初始兵力相同, 且非战争减员率相等的情况下, 由于甲方单

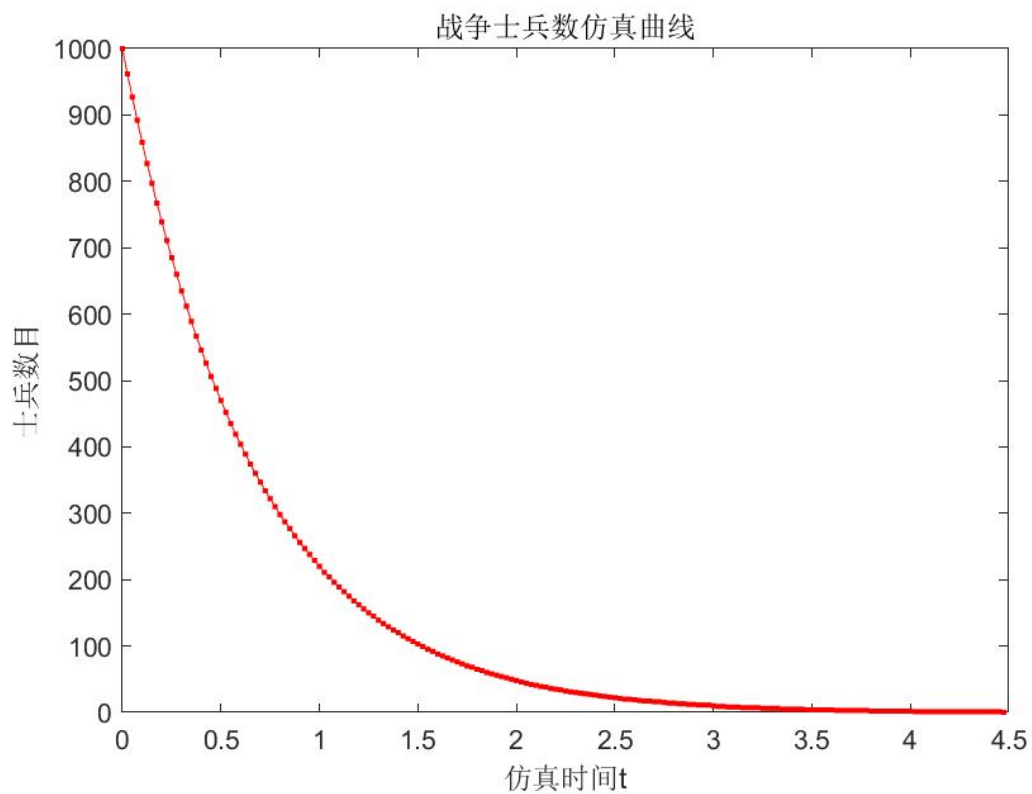
士兵的射击率很高，因此，甲方取得了最终的胜利。

(2) 当双方初始兵力以及军队作战能力（相关参数）完全相等时候，仿真过程如下：





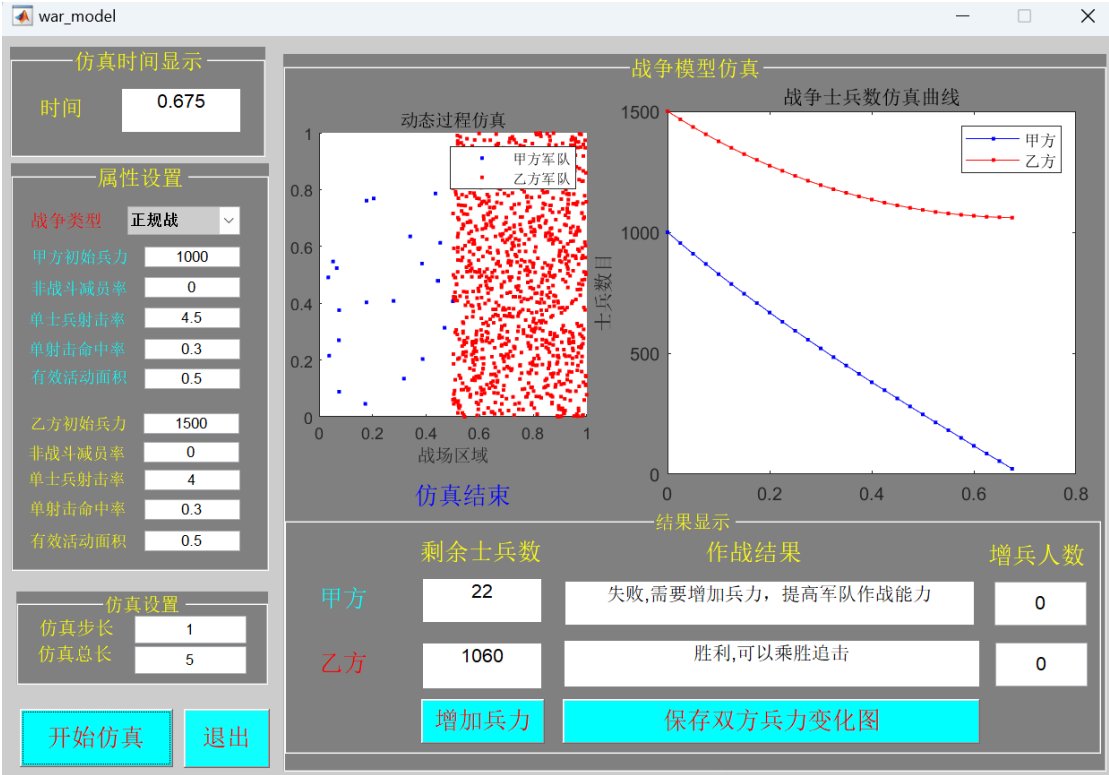
最终的仿真曲线为：



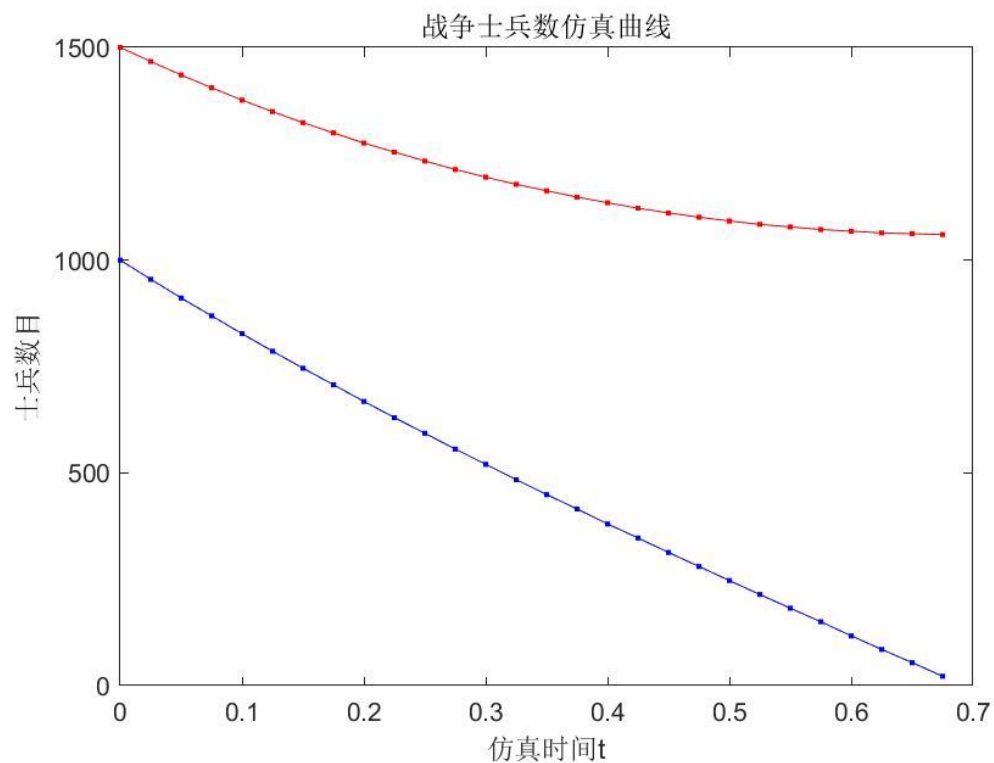
最终双方战为平局，也说明了仿真模型建立的正确性。

接下来简化仿真模型，即不考虑非战斗减员率的影响，根据 3.1 所求解的数学模型可以得到，当 $k = (r_y * p_y)y_0^2 - (r_x * p_x)x_0^2 > 0$ 时， $y > 0, x = 0$ ，则乙方获得最终的胜利，当 $k = (r_y * p_y)y_0^2 - (r_x * p_x)x_0^2 < 0$ 时， $x > 0, y = 0$ ，则甲方获得最终的胜利。当 $k = (r_y * p_y)y_0^2 - (r_x * p_x)x_0^2 = 0$ 时， $x = 0, y = 0$ ，则双方取得平局。

(3) 设置参数使其满足  $k > 0$ ，其仿真结果如下图：

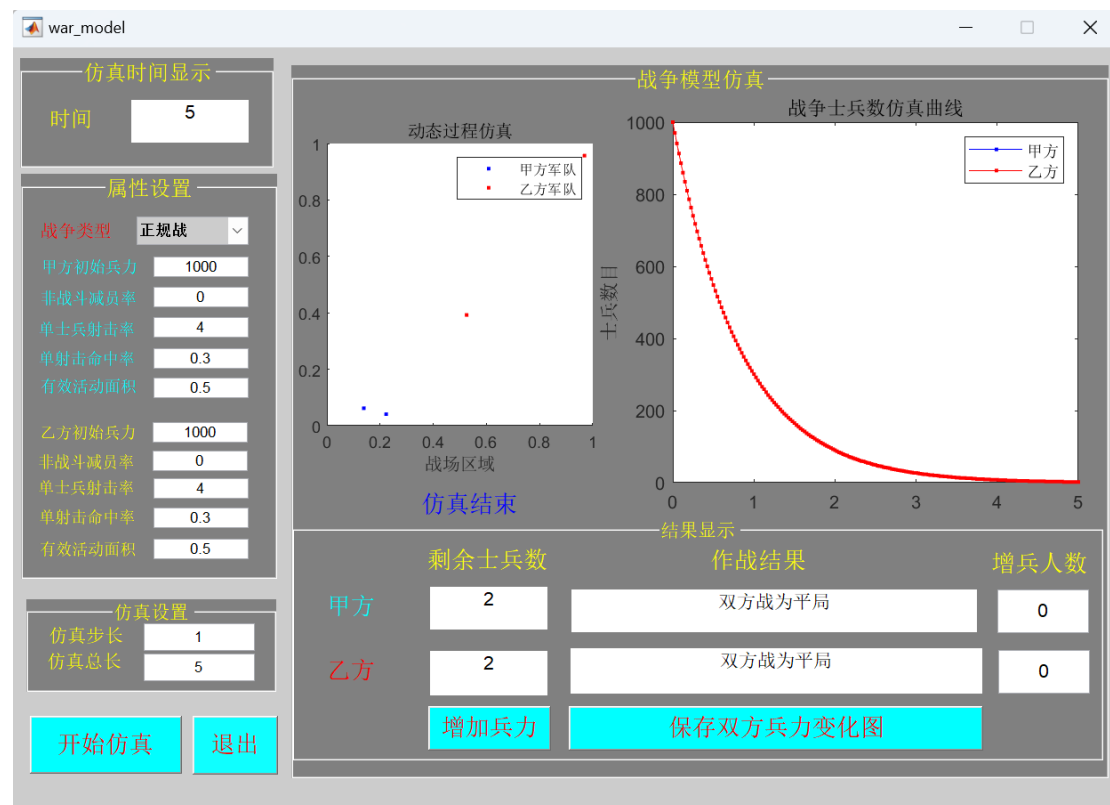


其最终的仿真曲线为：



可以看出，乙方胜利，甲方失败，符合理论求解的结果。

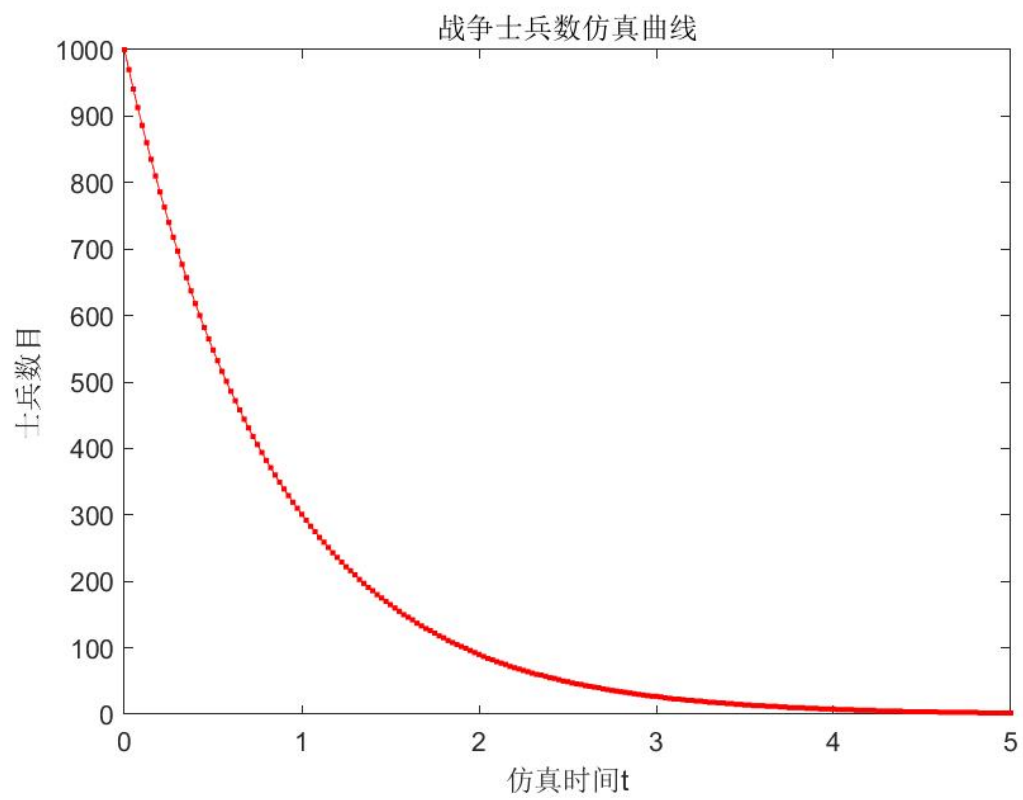
(4) 修改参数使得  $k=0$ :



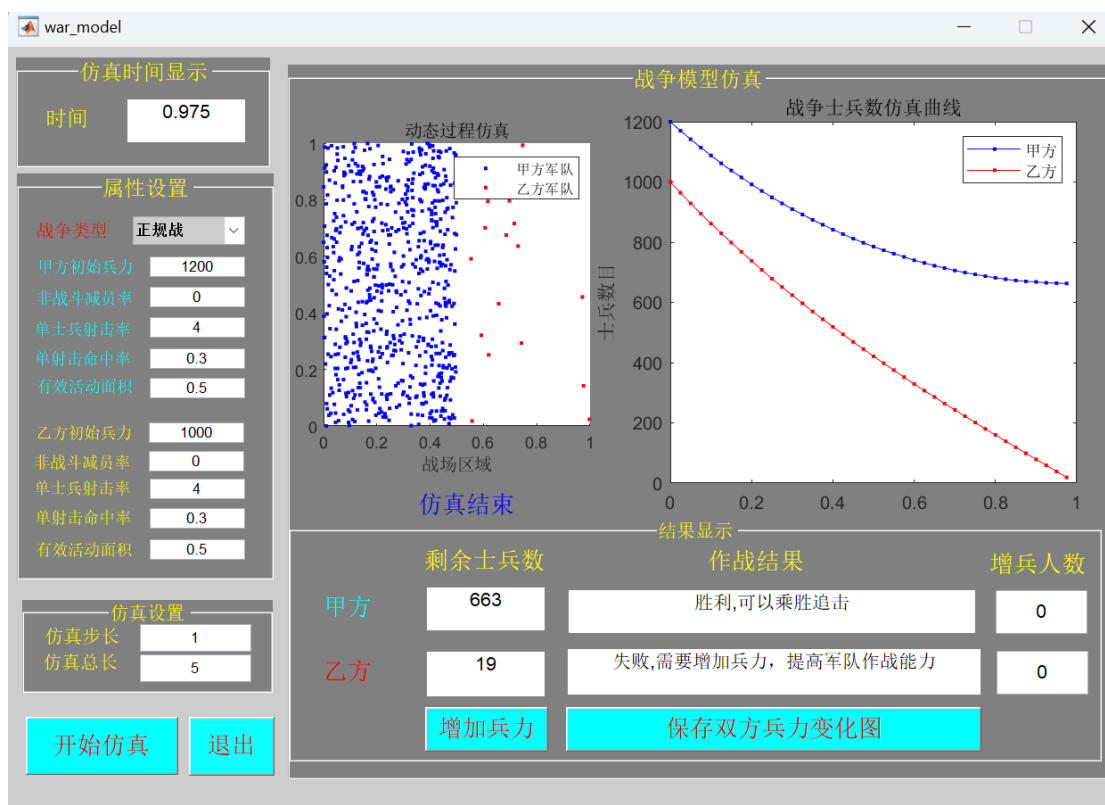
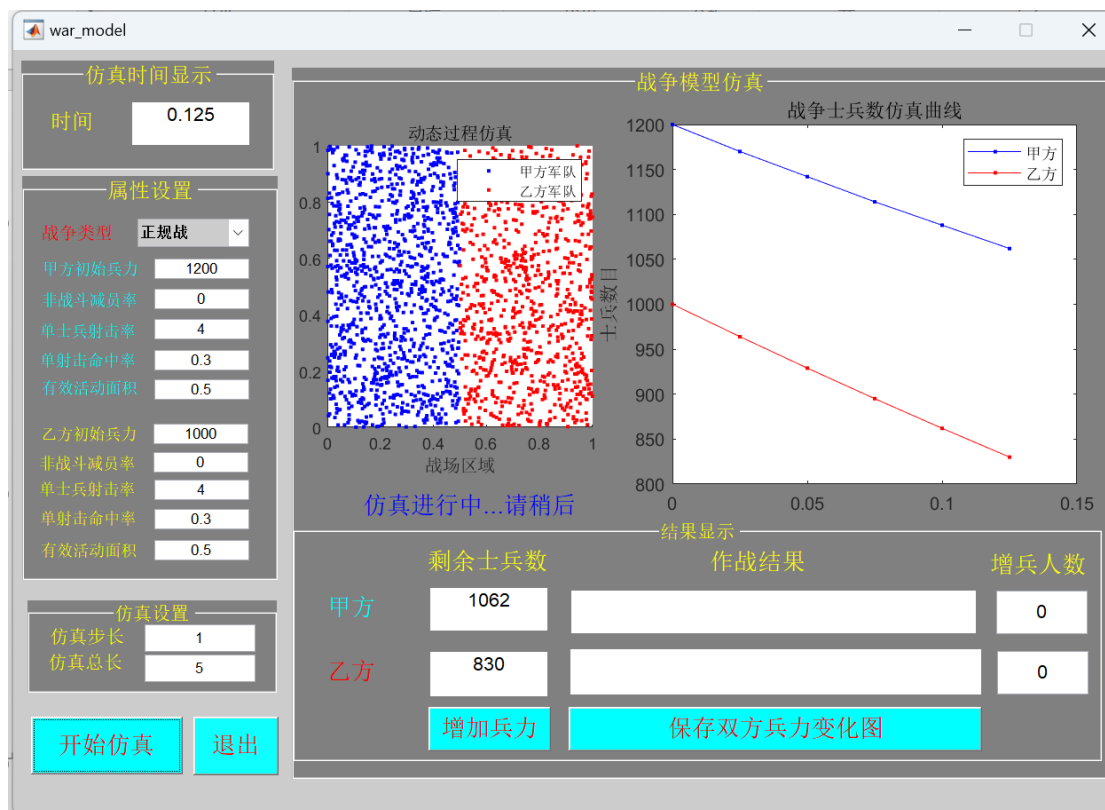


仿真所得到的结果为，双方战平，符合理论求解的结果。

最终的仿真曲线如下图：

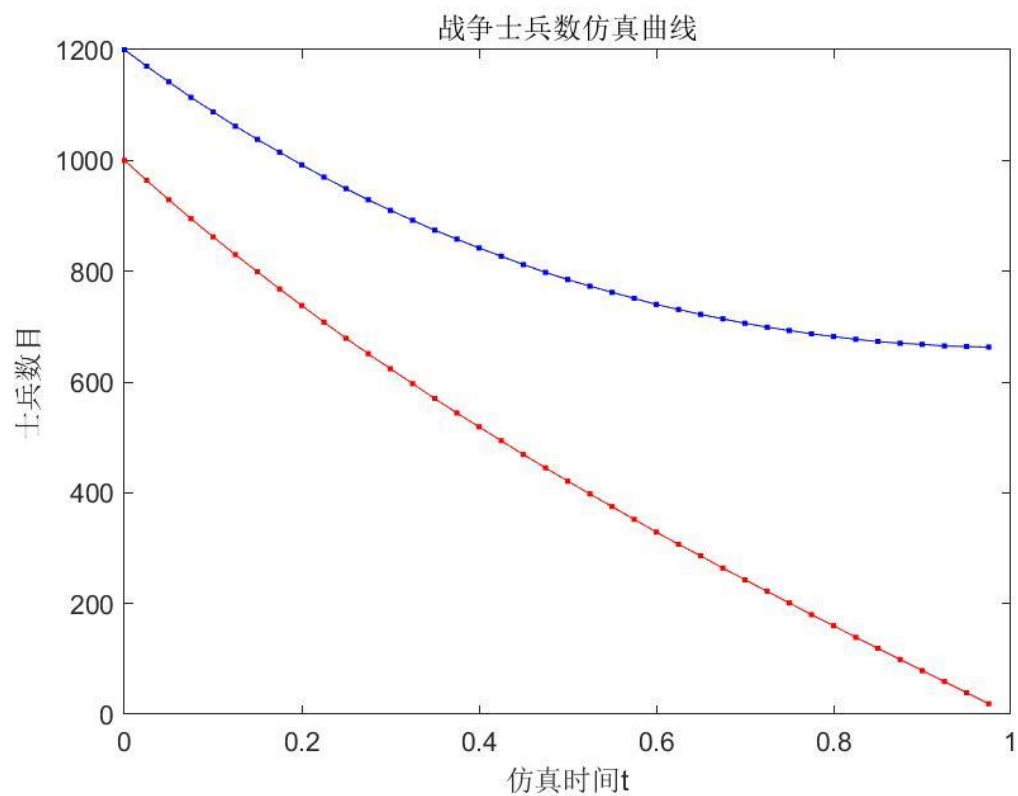


(5) 修改参数使得  $k < 0$ :



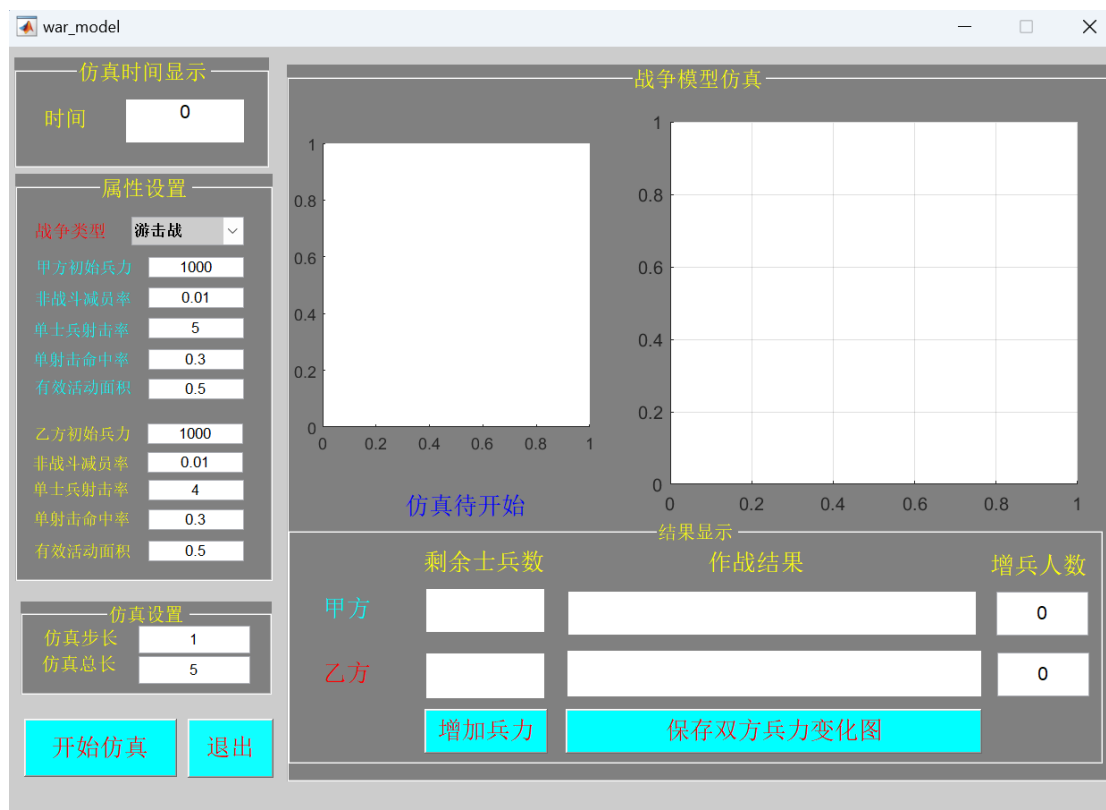
则最终的仿真结果为甲方胜利、乙方失败，符合理论求解的结果。

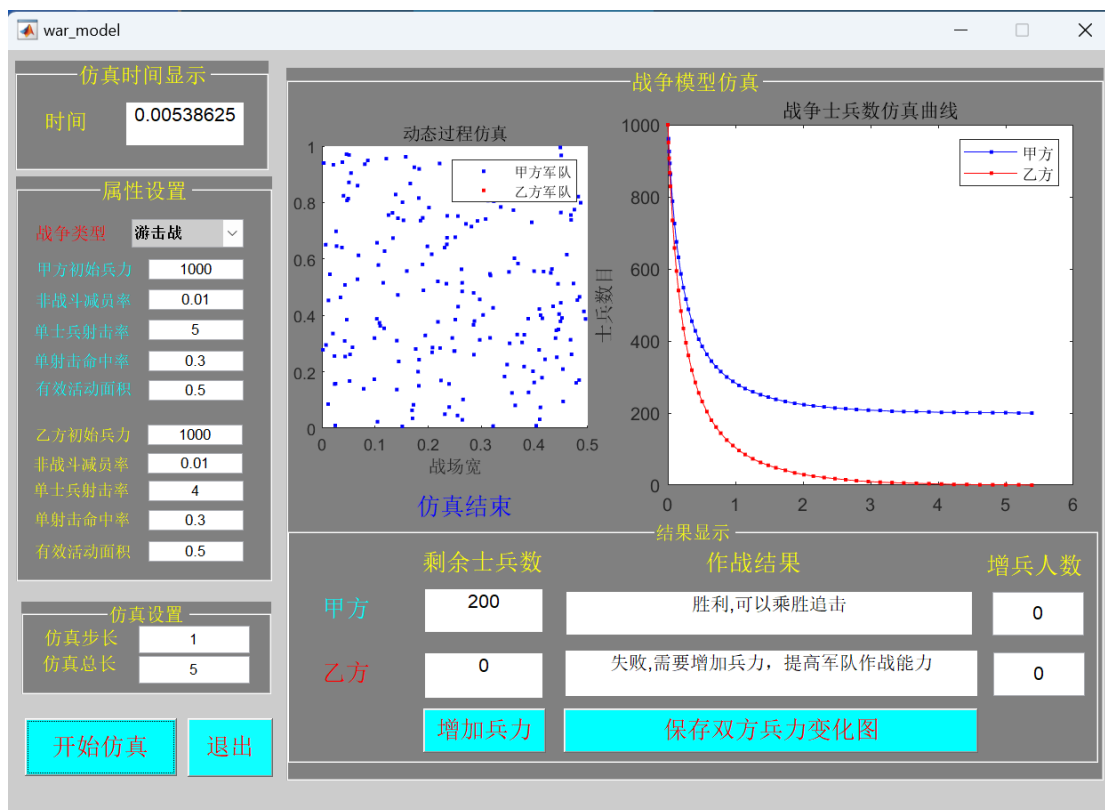
其最终的仿真曲线为：



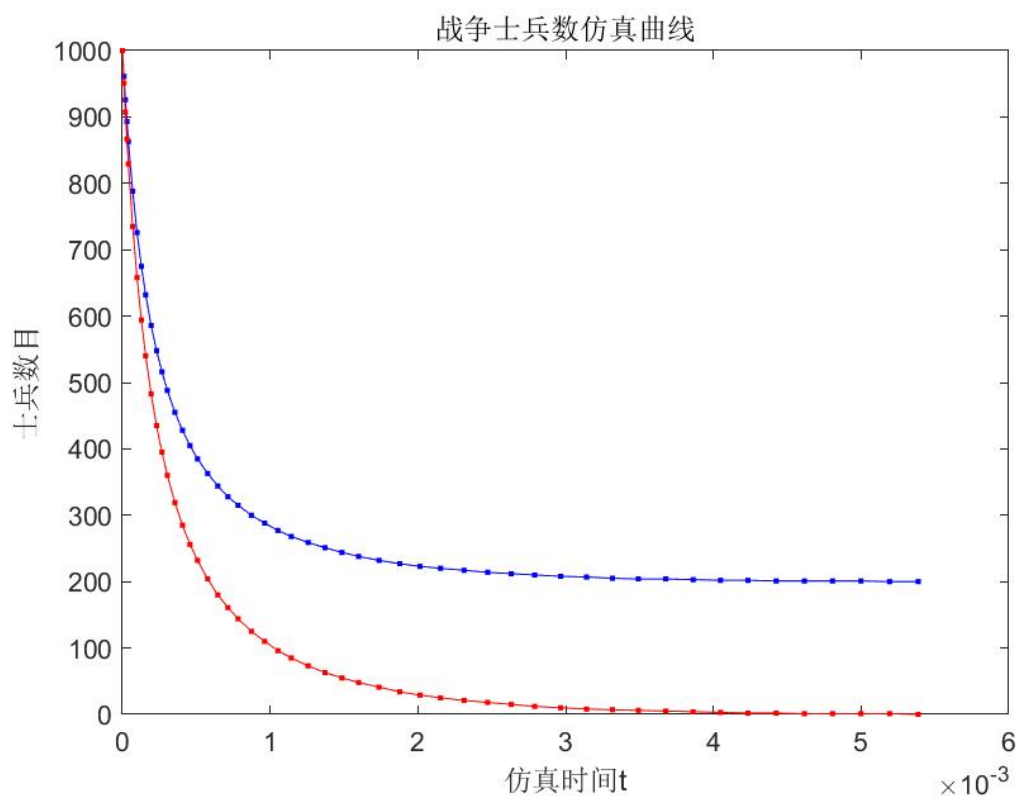
## 7.2 游击战实例

(1) 使用默认参数进行仿真：

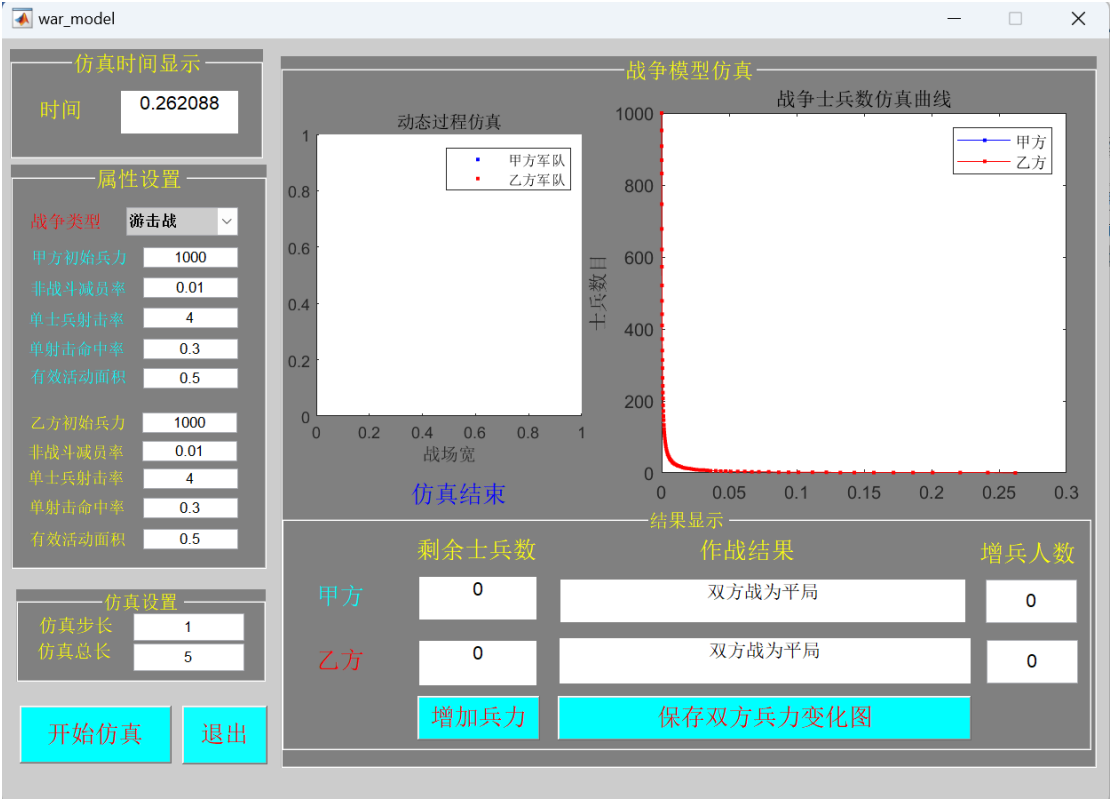
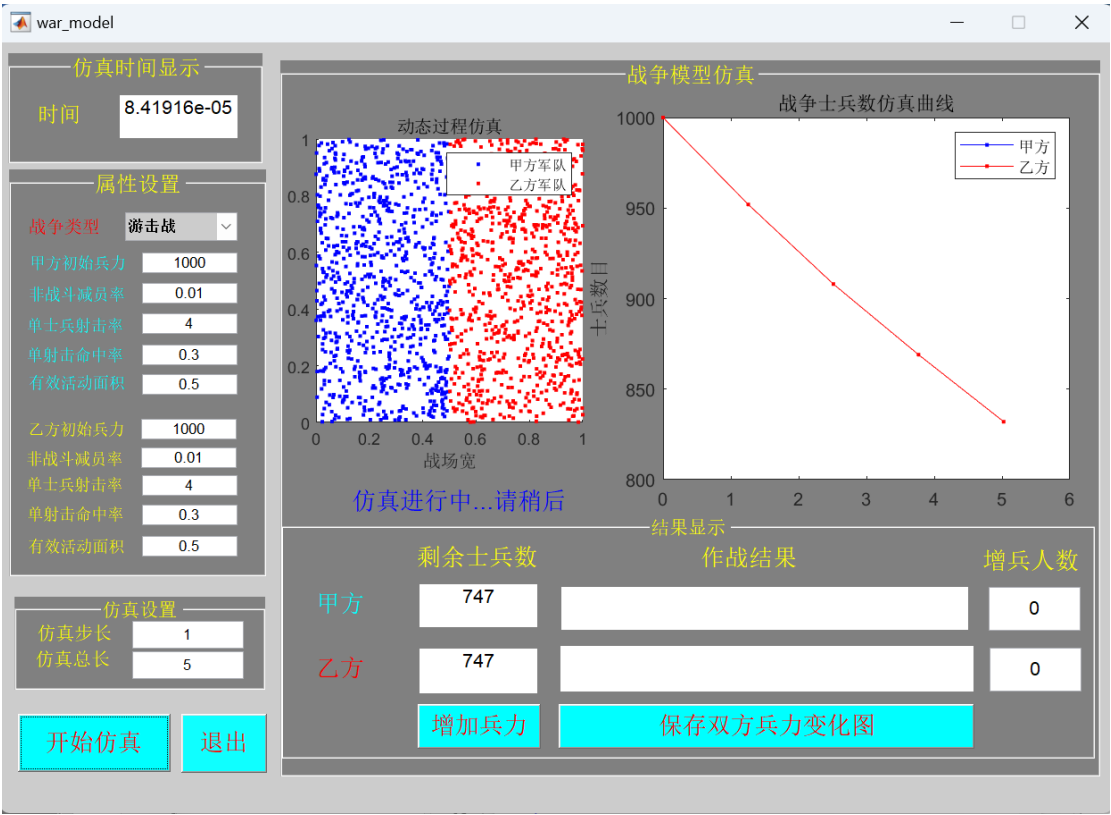




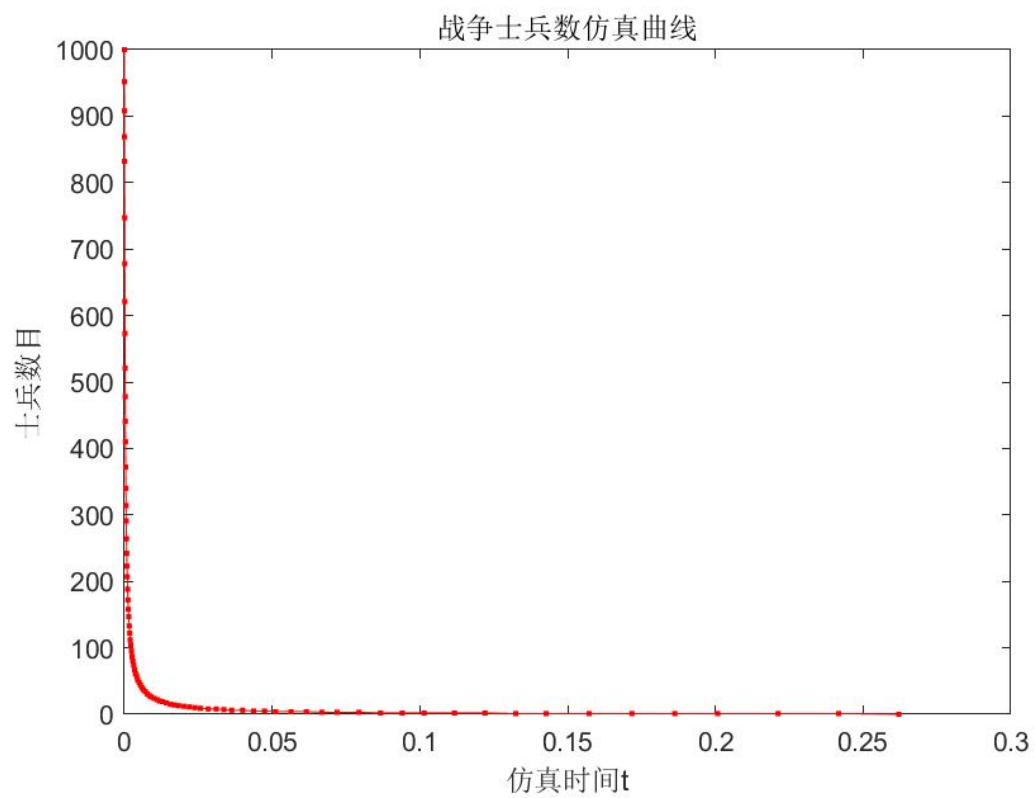
其最终得到的仿真曲线为:



(2) 若调整双方军队人数使得双方军队的初始人数相等且双方的作战能力相同，则有：



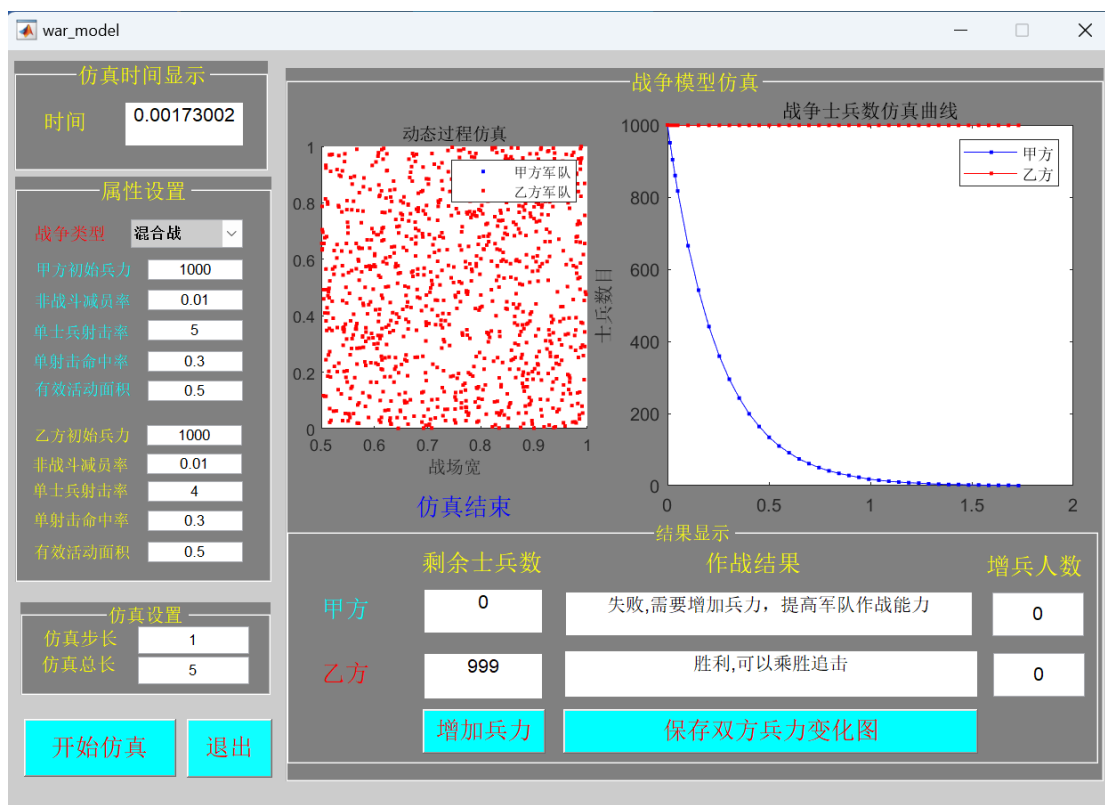
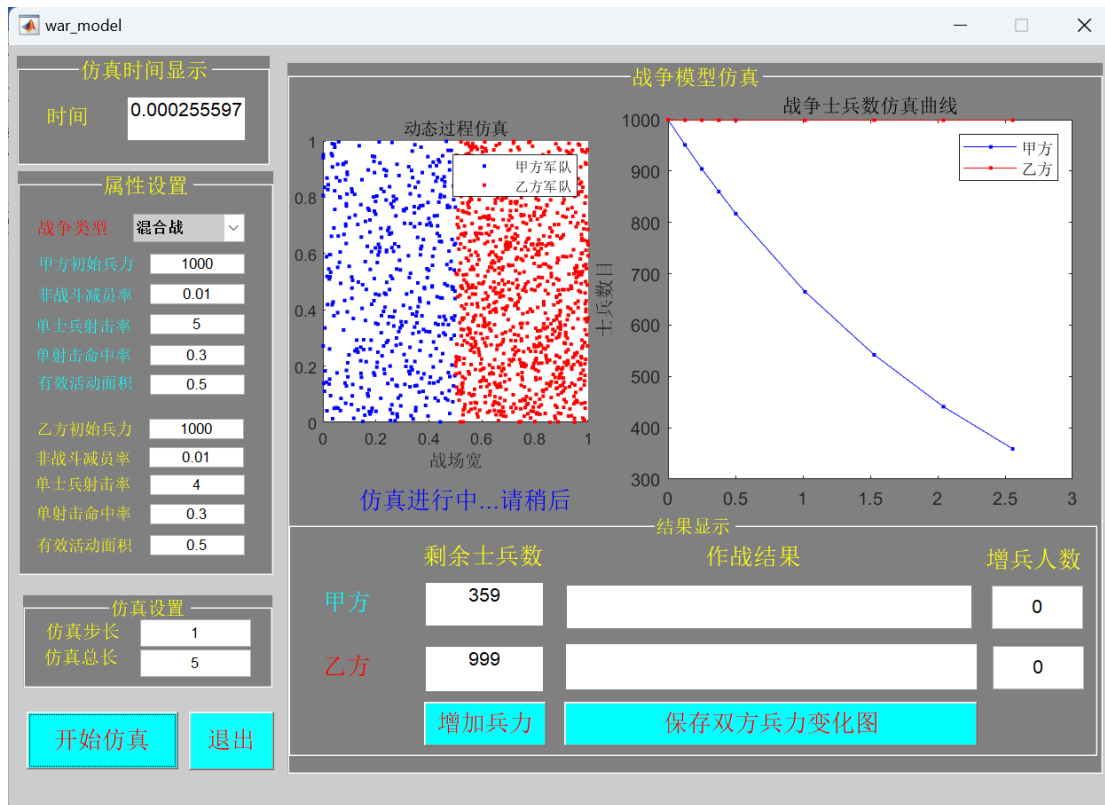
最终双方的仿真曲线如下图：



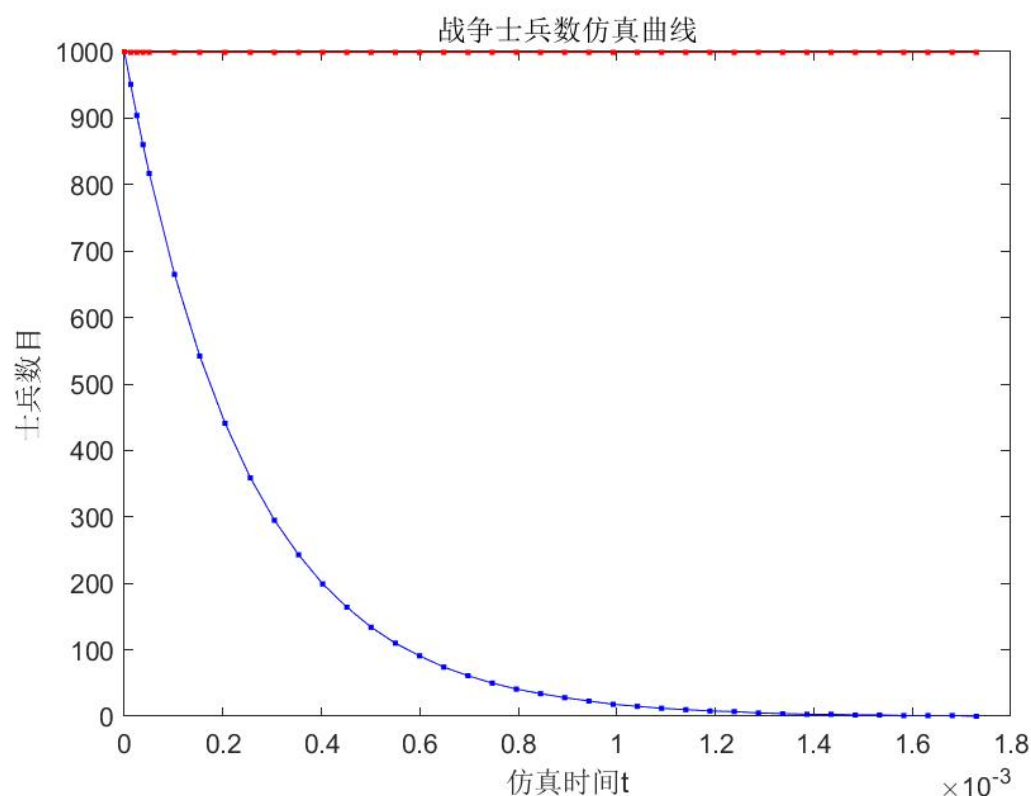
此时，双方战为平局，符合理论。

### 7.3 混合战实例

初始值情况下，双方的人数变化曲线如下图所示：



最终的战争士兵数仿真曲线如下：



由于甲方采取游击战形式，甲方人数的衰减速率与甲方的人数成正比，远远大于乙方，战争的结果也是乙方完胜甲方，符合理论。由此可见，当双方采取战争策略不同时，战争的最终结果绝大部分取决于所采取战争策略。相较于采取游击战的军队，采取正规战的军队更加占有优势。

## 8 心得体会

之前对于数学建模、仿真建模一直处于略知皮毛的阶段，通过论文的阅读对这个题目有过一定的了解之后，根据上课的时候也跟着老师学习了系统建模以及 MATLAB 的知识，这个题目做起来还是比较快的。其中在搞清楚如何处理函数之间的调用关系花费了较多的时间，最终使用 matlab 中的 handles 在各回调函数之间传递参数。

尽管之前接触过 matlab，但是对于其中的具体的句柄等概念都不太清晰，通过课程的学习以及本次实验建模，我也进一步的对 matlab 这个实用工具有了进一步的了解。本课程的本质是让我们利用 matlab 对现实中的问题进行建模，从而能够解决实际问题。在程序编写的过程中我也遇到了很多的问题，绝大部分的问题都是通过网络搜索解决。



通过本次课程设计，锻炼了自己 matlab 图形化界面的设计和分析问题解决问题的能力，较为熟练的掌握了使用 matlab 的 GUI 功能，从而搭建基于图形化界面的系统。

最后也在此感谢教授我知识、带领我入门的王老师。

## 参考文献

- [1]高瑜,高艺,王伟.常微分方程在数学建模中的应用之战争模型[J].知识文库,2019,(17)84+86.
- [2]陈庆文.例说数学建模在战争定量评估中的应用[J].才智,2015(22)327.
- [3]黑志华.混合型战争微分方程模型研究[J].数学学习与研究,2011(05)92+94.
- [4]刘星,田德生.正规战争的随机微分方程模型[J].数学的实践与认识,2017,47(09)149-158.
- [5] 刘杨,赖笑.动力系统模型仿真在军事战争建模中的应用[J].四川兵工学报,2010,31(04)112-114.

## 部分代码附录

```
%% 参数初始设置
set(handles.now,'string','0')%初始时间为 0
set(handles.type,'value',1);%默认设置为正规战
handles.wartype = 1;
set(handles.x0,'string',1000);%甲方初始人数为 100 人
handles.x_0 = 1000;
set(handles.alpha,'string',0.01);%甲方非战斗减员率
handles.a = 0.01;
set(handles.rx,'string',5);%甲方单个士兵射击率
handles.r_x = 5;
set(handles.px,'string',0.3);%甲方单射击命中率
handles.p_x = 0.3;
set(handles.sx,'string',0.5);%甲方有效活动面积所占比例
handles.s_x = 0.5;
set(handles.y0,'string',1000);%乙方初始人数为 100 人
handles.y_0 = 1000;
set(handles.beta,'string',0.01);%乙方非战斗减员率
handles.b = 0.01;
set(handles.ry,'string',4);%乙方单个士兵射击率
handles.r_y = 4;
```

```

set(handles.py, 'string', 0.3); %乙方单射击命中率
handles.p_y = 0.3;
set(handles.sy, 'string', 0.5); %乙方有效活动面积所占比例
handles.s_y = 0.5;
set(handles.dt, 'string', 1); %等间隔仿真步长初始化为 1
handles.step_dt = 1;
set(handles.time, 'string', 5); %设置总仿真时长限制
handles.timeall = 5;
set(handles.x_add, 'string', 0); %设置甲方增加兵力人数
handles.xadd = 0;
set(handles.y_add, 'string', 0); %设置乙方增加兵力人数
handles.yadd = 0;
handles.flag_add = 0; %是否增兵标志 如果为 1 则为增兵 否则为不增兵
guidata(hObject, handles);
%% 坐标轴初始化设置
%axes1 为结果绘制窗口，初始化为显示刻度，坐标轴设置为 0 到无穷
set(handles.axes1, 'XGrid', 'on'); set(handles.axes1, 'XLim', [0 inf]);
set(handles.axes1, 'YGrid', 'on'); set(handles.axes1, 'YLim', [0 inf]);
% UIWAIT makes war_model wait for user response (see UIRESUME)
% uiwait(handles.figure1);
end

function begin_Callback(hObject, eventdata, handles)
% hObject    handle to begin (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%% 开始仿真
%% 仿真参数初始化
%将 handles 中存储的参数读出
now = 0; dt = handles.step_dt; %初始仿真时间、时间间隔
t_list = []; %用于存储记录时间序列，方便后续绘图
time = handles.timeall; %仿真总时长
wartype = handles.wartype; %战争类型，决定仿真过程中的具体模型建立
x0 = handles.x_0; a = handles.a; %读出甲军队的相关参数
if(x0 <= 0)
    set(handles.flag, 'string', '甲方军队不能为负数'); %报错信息
    pause(1);
    set(handles.x0, 'string', ''); %删除不合法参数 重新输入
    set(handles.flag, 'string', '仿真未开始'); %修改仿真状态为仿真未开始
end
x = x0; %甲军队的最初数目 初始化为 x0
x_list = []; %用于存储甲军队兵力数目
rx = handles.r_x; %甲方军队的单士兵射击率

```

```

px = handles.p_x;%甲方军队的单射击命中率
sx = handles.s_x;%甲方有效活动面积
y0 = handles.y_0;b = handles.b;%读出乙军队的相关参数
y = y0;%乙军队的最初数目 初始化为 y0
if(y0<=0)
    set(handles.flag, 'string','乙方军队不能为负数');%报错信息
    pause(1);
    set(handles.y0,'string','');%删除不合法参数 重新输入
    set(handles.flag,'string','仿真未开始');%修改仿真状态为仿真未开始
end
y_list = [];%用于存储乙军队的兵力数目
ry = handles.r_y;%乙方军队的单士兵射击率
py = handles.p_y;%乙方军队的单射击命中率
sy = handles.s_y;%乙方有效活动面积
if(x0>0&y0>0)%验证初始人数是否符合要求 甲方乙方的初始兵力不能为负数
%% 开始仿真
switch wartype
    %% 正规战
    case 1
        %进行迭代仿真
        set(handles.xwinflag,'string',' ');%清除甲方的士兵数目
        set(handles.ywinflag,'string',' ');%清除乙方的士兵数目
        for now = 0 : dt : time-dt %根据仿真步长和仿真时间总限制进行循环
            set(handles.flag,'string','仿真进行中...请稍后');%修改仿真状态为仿真进
            行时
            set(handles.now,'string',now);%修改仿真时间显示
            tspan=[now now+dt];%ode 求解时间间隔 利用 ode 进行迭代求解
            [ttemp,ytemp] = ode45(@(t,y)
            odefun1(t,y,rx,px,ry,py,a,b),tspan,[x;y]);%在等间隔的时间范围时间点 tspan 之间
            进行 ode 求解（变长求解）
            for j = 1:1: length(ttemp)%变长求解的每一个时间点都记录下来
                now = ttemp(j);%变长求解的时间点
                x = floor(ytemp(j,1)); y = floor(ytemp(j,2));%甲方和乙方人数取整
                （向下取整）
                if (x>=0 & y>=0)%人数不能为负数，出现有一方人数为 0 则退出仿真
                    t_list = [t_list,now];%记录仿真的时间点，方便后续作图
                    x_list = [x_list,x];%记录甲方人数
                    y_list = [y_list,y];%记录乙方人数
                    set(handles.now,'string',now);%显示仿真的时间点
                    set(handles.xnum,'string',x);%显示甲方的人数
                    set(handles.ynum,'string',y);%显示乙方的人数
                    axes(handles.axes1);%设置作图的句柄 绘制人数-时间变化图
                    plot(t_list,x_list,'-b.',t_list,y_list,'-r.');
```

制

```

        legend('甲方','乙方');%标记线
        xlabel('仿真时间 t');ylabel('士兵数目');%标记图
        title('战争士兵数仿真曲线')
        axes(handles.axes3);%设置作图的句柄 动态仿真图
        cla reset;%清空上次的动态仿真图
        box on;set(handles.axes3,'xtick',[]);
        set(handles.axes3,'ytick',[]);
        x_idx = zeros([x,2]);y_idx = zeros([y,2]);%随机产生甲乙俩军队
        的士兵坐标
        x_idx(:,1) =0 +(sx -0).*rand(x ,1);x_idx(:,2) = 0 +(1 -
        0).*rand(x ,1);
        y_idx(:,1) =sx +(sy-0).*rand(y ,1); y_idx(:,2) =0 +(1-
        0).*rand(y ,1);
        scatter(x_idx(:,1),x_idx(:,2),'.b');%绘制散点图，甲方军队
        hold on;
        scatter(y_idx(:,1),y_idx(:,2),'.r');%绘制散点图，乙方军队
        legend('甲方军队','乙方军队');
        xlabel('战场区域');
        title('动态过程仿真');
        if(x*y==0)%如果出现某方士兵数目为 0 退出仿真
            break;
        end
        elseif(x*y<=0)%如果出现某方士兵数目为负数 退出仿真
            break;
        end
    end
    if(x*y<=0)%如果出现某方士兵数目为负数 退出仿真
        break;
    end
end
if(x>y)%甲方胜利 乙方失败
    set(handles.xwinflag,'string','胜利,可以乘胜追击');
    set(handles.ywinflag,'string','失败,需要增加兵力,提高军队作战能力');
elseif(x<y)%甲方失败 乙方胜利
    set(handles.ywinflag,'string','胜利,可以乘胜追击');
    set(handles.xwinflag,'string','失败,需要增加兵力,提高军队作战能力');
elseif(x==y)%双方战平
    set(handles.ywinflag,'string','双方战为平局');
    set(handles.xwinflag,'string','双方战为平局');
end
set(handles.flag,'string','仿真结束');%修改仿真状态 结束仿真

%% 游击战
case 2

```

```

    %进行迭代仿真
    set(handles.xwinflag,'string',' ');%清除甲方的士兵数目
    set(handles.ywinflag,'string',' ');%清除乙方的士兵数目
    for now = 0 : dt : time-dt %根据仿真步长和仿真时间总限制进行循环
        set(handles.flag,'string','仿真进行中...请稍后');%修改仿真状态为仿真进
行时
        set(handles.now,'string',now);%修改仿真时间显示
        tspan=[now now+dt];%ode 求解时间间隔 利用 ode 进行迭代求解
        [ttemp,ytemp] = ode45(@(t,y)
odefun2(t,y,rx,sx,ry,sy,a,b),tspan,[x;y]);%在等间隔的时间范围时间点 tspan 之间
进行 ode 求解（变长求解）
        for j = 1:1: length(ttemp)%变长求解的每一个时间点都记录下来
            now = ttemp(j);%变长求解的时间点
            x = floor(ytemp(j,1)); y = floor(ytemp(j,2));%甲方和乙方人数取整
（向下取整）
            if (x>=0 & y>=0)%人数不能为负数，出现有一方人数为 0 则退出仿真
                t_list = [t_list,now];%记录仿真的时间点，方便后续作图
                x_list = [x_list,x];%记录甲方人数
                y_list = [y_list,y];%记录乙方人数
                set(handles.now,'string',now);%显示仿真的时间点
                set(handles.xnum,'string',x);%显示甲方的人数
                set(handles.ynum,'string',y);%显示乙方的人数
                axes(handles.axes1);%设置作图的句柄 绘制人数-时间变化图
                plot(t_list,x_list,'-b.',t_list,y_list,'-r.');
```

制

```

                %进行人数图的绘

                legend('甲方','乙方');%标记线
                xlabel('仿真时间 t');ylabel('士兵数目');
                title('战争士兵数仿真曲线')
                axes(handles.axes3);%设置作图的句柄 动态仿真图
                cla reset;%清空上次的动态仿真图
                box on;set(handles.axes3,'xtick',[]);
                set(handles.axes3,'ytick',[]);
                x_idx = zeros([x,2]);y_idx = zeros([y,2]);%随机产生甲乙俩军队
的士兵坐标
                x_idx(:,1) =0 +(sx -0).*rand(x ,1);x_idx(:,2) = 0 +(1 -
0).*rand(x ,1);
                y_idx(:,1) =sx +(sy-0).*rand(y ,1); y_idx(:,2) =0 +(1-
0).*rand(y ,1);
                scatter(x_idx(:,1),x_idx(:,2),'b');%绘制散点图，甲方军队
                hold on;
                scatter(y_idx(:,1),y_idx(:,2),'r');%绘制散点图，乙方军队
                legend('甲方军队','乙方军队');
                xlabel('战场宽');ylabel('战场高');
                title('动态过程仿真');
```

```

        if(x*y==0)%如果出现某方士兵数目为 0 退出仿真
            break;
        end
        elseif(x*y<=0)%如果出现某方士兵数目为负数 退出仿真
            break;
        end
    end
    if(x*y<=0)%如果出现某方士兵数目为负数 退出仿真
        break;
    end
end
if(x>y)%甲方胜利 乙方失败
    set(handles.xwinflag,'string','胜利,可以乘胜追击');
    set(handles.ywinflag,'string','失败,需要增加兵力,提高军队作战能力');
elseif(x<y)%甲方失败 乙方胜利
    set(handles.ywinflag,'string','胜利,可以乘胜追击');
    set(handles.xwinflag,'string','失败,需要增加兵力,提高军队作战能力');
elseif(x==y)%双方战平
    set(handles.ywinflag,'string','双方战为平局');
    set(handles.xwinflag,'string','双方战为平局');
end
set(handles.flag,'string','仿真结束');%修改仿真状态 结束仿真

%% 混合战
case 3
    %进行迭代仿真
    set(handles.xwinflag,'string',' ');%清除甲方的士兵数目
    set(handles.ywinflag,'string',' ');%清除乙方的士兵数目
    for now = 0 : dt : time-dt %根据仿真步长和仿真时间总限制进行循环
        set(handles.flag,'string','仿真进行中...请稍后');%修改仿真状态为仿真进
行时
        set(handles.now,'string',now);%修改仿真时间显示
        tspan=[now now+dt];%ode 求解时间间隔 利用 ode 进行迭代求解
        [ttemp,ytemp] = ode45(@(t,y)
odefun3(t,y,rx,sx,ry,sy,px,a,b),tspan,[x;y]);%在等间隔的时间范围时间点 tspan
之间进行 ode 求解 (变长求解)
        for j = 1:1: length(ttemp)%变长求解的每一个时间点都记录下来
            now = ttemp(j);%变长求解的时间点
            x = floor(ytemp(j,1)); y = floor(ytemp(j,2));%甲方和乙方人数取整
(向下取整)
            if (x>=0 & y>=0)%人数不能为负数, 出现有一方人数为 0 则退出仿真
                t_list = [t_list,now];%记录仿真的时间点, 方便后续作图
                x_list = [x_list,x];%记录甲方人数
                y_list = [y_list,y];%记录乙方人数
            end
        end
    end
end

```

制

的士兵坐标

```
set(handles.now, 'string', now); %显示仿真的时间点
set(handles.xnum, 'string', x); %显示甲方的人数
set(handles.ynum, 'string', y); %显示乙方的人数
axes(handles.axes1); %设置作图的句柄 绘制人数-时间变化图
plot(t_list, x_list, '-b.', t_list, y_list, '-r. '); %进行人数图的绘制

legend('甲方', '乙方'); %标记线
xlabel('仿真时间 t'); ylabel('士兵数目');
title('战争士兵数仿真曲线')
axes(handles.axes3); %设置作图的句柄 动态仿真图
cla reset; %清空上次的动态仿真图
box on; set(handles.axes3, 'xtick', []);
set(handles.axes3, 'ytick', []);
x_idx = zeros([x, 2]); y_idx = zeros([y, 2]); %随机产生甲乙俩军队的士兵坐标

x_idx(:, 1) = 0 + (sx - 0) .* rand(x, 1); x_idx(:, 2) = 0 + (1 - 0) .* rand(x, 1);
y_idx(:, 1) = sx + (sy - 0) .* rand(y, 1); y_idx(:, 2) = 0 + (1 - 0) .* rand(y, 1);

scatter(x_idx(:, 1), x_idx(:, 2), '.b'); %绘制散点图, 甲方军队
hold on;
scatter(y_idx(:, 1), y_idx(:, 2), '.r'); %绘制散点图, 乙方军队
legend('甲方军队', '乙方军队');
xlabel('战场宽'); ylabel('战场高');
title('动态过程仿真');
if(x*y==0) %如果出现某方士兵数目为0 退出仿真
    break;
end
elseif(x*y<=0) %如果出现某方士兵数目为负数 退出仿真
    break;
end
end
if(x*y<=0) %如果出现某方士兵数目为负数 退出仿真
    break;
end
end
if(x>y) %甲方胜利 乙方失败
    set(handles.xwinflag, 'string', '胜利,可以乘胜追击');
    set(handles.ywinflag, 'string', '失败,需要增加兵力,提高军队作战能力');
elseif(x<y) %甲方失败 乙方胜利
    set(handles.ywinflag, 'string', '胜利,可以乘胜追击');
    set(handles.xwinflag, 'string', '失败,需要增加兵力,提高军队作战能力');
elseif(x==y) %双方战平
    set(handles.ywinflag, 'string', '双方战为平局');
```

```
        set(handles.xwinflag,'string','双方战为平局');  
    end  
    set(handles.flag,'string','仿真结束');%修改仿真状态 结束仿真  
end  
end  
end
```