

```

function result=VGM_1Nri(r)
% clc,clear all;
format long g;

disp([' 一种新结构的多阶多变量灰色预测模型，VGM(1,N,ri)']);
disp(' ')
disp(['————— 【数据输入】 —————
—————']);

Xs=[9.45 8.93 8.51 7.52 7.48 7.26 7.13 6.71 7.24 6.64 6.49 5.38
    1.2690 0.9766 0.7905 0.5879 0.7782 0.7037 0.5780 0.6689 0.4675 0.5048 0.4254
    0.3591
    0.0064 0.0075 0.0086 0.0033 0.0032 0.0049 0.0088 0.0028 0.0059 0.0090 0.0122
    0.0107
    0.0098 0.0087 0.0116 0.0142 0.0094 0.0099 0.0103 0.0126 0.0097 0.0093 0.0137
    0.0197
    0.0097 0.0045 0.0106 0.0189 0.0121 0.0245 0.0159 0.0000 0.0167 0.0069 0.0035
    0.0036];

[Row,Col]=size(Xs);
Sim_len=Col-0;
Pre_len=0;
Steps=0;

for i=1:Row
    if i==1
        disp(['因 变 量： ', Tools(Xs(i,:))]);
    else
        disp(['自变量[' num2str(i-1) ']:', Tools(Xs(i,:))]);
    end
end

disp(' ')
disp(['————— 【数据信息与功能划分】 —————
—————']);

disp(['变量及数据个数：输入数据中共[' num2str(Row) ']个变量，每个变量中共[' num2str(Col)
']个数据']);
disp(['建模数据子矩阵：矩阵前[' num2str(Sim_len) ']组数据']);
disp(['预测误差子矩阵：矩阵第[' num2str(Sim_len+1) ']到第[' num2str(Sim_len+Pre_len) ']
之间的数据']);
disp(['数据预测子矩阵：矩阵后[' num2str(Steps) ']组数据，即矩阵第[' num2str(Col-Steps+1)
']到第[' num2str(Col) ']之间的数据']);

disp(' ')

```

disp(['—————【参数计算】—————
—————']);

```
Xsr_AGO=[];  
Xsr_AGO = zeros(Row,Col);  
for i=1:Row    %行 第 i 个变量  
    for k=1:Col    %列 第 i 个变量的第 k 个数据  
        ago=0;  
        for u=1:k  
            %ago=ago+gamma(r1+k-u)*Xs(1,u)/(gamma(k-u+1)*gamma(r1));  
            ago=ago+gamma(r(i)+k-u)*Xs(i,u)/(gamma(k-u+1)*gamma(r(i)));  
        end  
        Xsr_AGO(i,k)=ago;  
    end  
end  
Xsr_AGO
```

```
X1r_1=[];%因变量的 r-1 阶累加生成  
for k=2:Sim_len  
    tmp=0;  
    tmp=Xsr_AGO(1,k)-Xsr_AGO(1,k-1);  
    X1r_1=[X1r_1,tmp];  
end;  
X1r_1
```

```
Y=X1r_1;  
Y=Y';
```

```
B=[];
```

```
B1=Xsr_AGO(2:Row,2:Sim_len)';
```

```
Z1=[];F=[];M=[];C=[];  
X1=Xsr_AGO(1:1,1:Sim_len);  
for k=2:Sim_len  
    Z1=[Z1,-0.5*(X1(k)+X1(k-1))];  
    alpha=r(Row+1);  
    F=[F,k^alpha];  
    M=[M,k];  
    C=[C,1];  
end
```

```

B2=Z1';B3=F';B4=M';B5=C';
B=[B,B1];B=[B,B2];B=[B,B3];B=[B,B4];B=[B,B5];
B
Y
Z1

```

```

fprintf(['VGM(1,N,ri)模型参数 b2,b3,...,bn,a,h1,h2,h3:']);
%if Sim_len>Row+3
    Ps=(B*B)^(-1)*B'*Y
%elseif Sim_len=Row+3
%    Ps=B^(-1)*Y
%else
%    Ps=B'*(B*B')^(-1)*Y
%end

```

```

len=length(Ps);
miu_1=1/(1+0.5*Ps(len-3))
miu_2=(1-0.5*Ps(len-3))/(1+0.5*Ps(len-3))
miu_3=Ps(len-2)/(1+0.5*Ps(len-3))
miu_4=Ps(len-1)/(1+0.5*Ps(len-3))
miu_5=Ps(len)/(1+0.5*Ps(len-3))

```

```

Simu_Xs_r=[];
Simu_Xs_r(1,1)=Xs(1,1);
for k=2:Sim_len+Pre_len+Steps
    tmp1=0;
    for t=1:k-1
        for i=2:Row
            tmp1=tmp1+(miu_2^(t-1)*Ps(i-1)*Xsr_AGO(i,k-t+1))*miu_1;
        end
    end

    tmp2=0;
    for j=0:k-2
        tmp2=tmp2+miu_2^j*((k-j)*miu_4+miu_5);
    end

    tmp3=0;
    for v=2:k
        tmp3=tmp3+miu_2^(k-v)*miu_3*v^alpha;
    end
end

```

```

tmp4=miu_2^(k-1)*Xs(1,1);

tmp=tmp1+tmp2+tmp3+tmp4;
Simu_Xs_r=[Simu_Xs_r,tmp];
end

All_cal_data=[];
All_cal_data(1)=Xs(1,1);
for k=2:Sim_len+Pre_len+Steps
    data=0;
    for i=0:k-1

        data=data+(-1)^i*gamma(r(1)+1)*Simu_Xs_r(k-i)/(gamma(i+1)*gamma(r(1)-i+1));
    end
    All_cal_data=[All_cal_data,data];
end

disp(' ')
fprintf(['—————【计算模拟值、残差及相对误差】—————
—————']);
SNO=[];
Raw_data=[];
Sim_data=[];
Residual_sim_error=[];
Relative_sim_error=[];
Sum_sim_error=0;

for k=2:Sim_len
    canCha=All_cal_data(k)-Xs(1,k);
    xdError=abs(canCha*100/Xs(1,k));
    Sum_sim_error=Sum_sim_error+xdError;
    SNO=[SNO,k];
    Raw_data=[Raw_data,Xs(1,k)];
    Sim_data=[Sim_data,All_cal_data(k)];
    Residual_sim_error=[Residual_sim_error,canCha];
    Relative_sim_error=[Relative_sim_error,xdError];
end

Col_name={'No','Raw_data','Simulated_data','Residual_error','Percentage_error'};
Simulation=table(SNO,'Raw_data',Sim_data,'Residual_sim_error','Relative_sim_error','VariableNames',Col_name)
Mean_sim_error=Sum_sim_error/(Sim_len-1);
str=['平均相对模拟百分误差 =' num2str(Mean_sim_error)];

```

```

disp(str);

Sum_pre_error=0;
if Pre_len>0
    disp(' ')
    fprintf(['—————【计算预测值、残差及相对误差】—————
—————']);
    SNO_pre=[];
    Pre_data=[];
    Raw_data=[];
    Residual_pre_error=[];
    Relative_pre_error=[];
    for k=Sim_len+1:Sim_len+Pre_len
        canCha=All_cal_data(k)-Xs(1,k);
        xdError=abs(canCha*100/Xs(1,k));
        Sum_pre_error=Sum_pre_error+xdError;
        SNO_pre=[SNO_pre,k];
        Raw_data=[Raw_data,Xs(1,k)];
        Pre_data=[Pre_data,All_cal_data(k)];
        Residual_pre_error=[Residual_pre_error,canCha];
        Relative_pre_error=[Relative_pre_error,xdError];
    end

    Col_name={'No','Raw_data','Predicted_data','Residual_error','Percentage_error'};

    Prediction=table(SNO_pre,Raw_data,Pre_data,Residual_pre_error,Relative_pre_error,
    'VariableNames',Col_name)
    Mean_pre_error=Sum_pre_error/Pre_len;
    str=['平均相对预测百分误差 =' num2str(Mean_pre_error)];
    disp(str);
end

disp(['—————
—————']);

MAPE=(Sum_sim_error+Sum_pre_error)/(Sim_len+Pre_len-1);
MAPE=roundn(MAPE,-4);
str2=['模拟及预测的综合百分误差 =' num2str(MAPE)];
disp(str2);

if Steps>0
    disp(' ')
    fprintf(['—————【应用 NSGM(1,N,r)模型进行预测】—————
—————']);

```

```

SNO=[];
Pre_steps_data=[];
for k=Sim_len+Pre_len+1:Sim_len+Pre_len+Steps
    pre=All_cal_data(k);
    SNO=[SNO,k];
    Pre_steps_data=[Pre_steps_data,pre];
end

Col_name={'No','Predicted_data'};
Prediction_steps=table(SNO,Pre_steps_data,'VariableNames',Col_name)
end
disp(['—————【建模结束】—————
—————']);
result=MAPE;

```