

Grassmann Numbers and Clifford Algebras

The elementary particles of Physics are classified according to the behavior of the multi-particle states under exchange of identical particles: bosonic states are symmetric while fermionic This manifests itself also in the commutation properties of the respective creation operators: bosonic creation operators commute while fermionic ones anticommute.

The function `fn_c` represents the function `fn_contract` which is introduced in the MPS calculations, and the function `fn_f` are the function in spin theories which is used to generate the s^+ and s^- presentations.

```
fn_f=@(j,m) sqrt((j+m).*(j-m+1));  
fn_c=@(x1,x2,x3,x4,x5,x6) fn_contract(x1,x2,x3,x4,x5,x6);
```

the total spin $J=1/2$; and `Num_J` is the dimension of the representation for each site.

```
J=1/2;    Num_J=2*J+1;
```

then we generate the Z operator:

```
v_J=(Num_J-1)/2:-1:-(Num_J-1)/2; mJ_z=diag(v_J);
```

and then we can introduce the J^+ , J^- , J_x and J_y operator

```
v_m=J:-1:-J+1;  
mJ_p=diag(fn_f(J,v_m),1); mJ_m=diag(fn_f(J,v_m),-1);  
mJ_x=(mJ_p+mJ_m)/2; mJ_y=(mJ_p-mJ_m)/(2j);
```

Finally we get the identity operator I :

```
mJ_id=eye(Num_J);
```

The delta matrix

The delta matrices are closely related to the spin operator, we wish we could have:

$$[\sigma_i, \sigma_j] = 2\varepsilon_{ijk} \sigma_k$$

```
m_dtx=mJ_x*2;  
m_dty=mJ_y*2;  
m_dtz=mJ_z*2;  
m_dti=mJ_id;
```

Show the above result:

`mJ_p` is:

```
disp('mJ_p=');disp(mJ_p);
```

```
mJ_p=  
  0      1  
  0      0
```

mJ_m is:

```
disp('mJ_m=');disp(mJ_m);
```

```
mJ_m=
     0     0
     1     0
```

identity operator mJ_id is:

```
disp('mJ_id');disp(mJ_id);
```

```
mJ_id
     1     0
     0     1
```

The delta matrices are:

```
format compact;
disp('m_dtx');disp(m_dtx);disp('m_dty');disp(m_dty);
```

```
m_dtx
     0     1
     1     0
m_dty
0.0000 + 0.0000i    0.0000 - 1.0000i
0.0000 + 1.0000i    0.0000 + 0.0000i
```

```
disp('m_dtz');disp(m_dtz);disp('m_dti');disp(m_dti);
```

```
m_dtz
     1     0
     0    -1
m_dti
     1     0
     0     1
```

The Gamam matrix generators

Next we deal with the gamma matrix. In this example we generate 6 gamma matrix named m_gamma1, ..., m_gamma6 which are reshaped from the T_gamma1,...,T_gamma6 represent the tensor form.

```
Num=Num_J; % dimension of the spin vector space
T_gamma1=fn_c(m_dtx,3,3,reshape(m_dti,[1,Num,Num]),3,1);
f1=T_gamma1;
T_gamma1=fn_c(T_gamma1,5,5,reshape(m_dti,[1,Num,Num]),3,1);

T_gamma2=fn_c(m_dtz,3,3,reshape(m_dtx,[1,Num,Num]),3,1);
f2=T_gamma2;
T_gamma2=fn_c(T_gamma2,5,5,reshape(m_dtx,[1,Num,Num]),3,1);

T_gamma3=fn_c(m_dtz,3,3,reshape(m_dtz,[1,Num,Num]),3,1);
T_gamma3=fn_c(T_gamma3,5,5,reshape(m_dtx,[1,Num,Num]),3,1);

T_gamma4=fn_c(m_dty,3,3,reshape(m_dti,[1,Num,Num]),3,1);
f3=T_gamma4;
```

```

T_gamma4=fn_c(T_gamma4,5,5,reshape(m_dti,[1,Num,Num]),3,1);

T_gamma5=fn_c(m_dtz,3,3,reshape(m_dty,[1,Num,Num]),3,1);
f4=T_gamma5;
T_gamma5=fn_c(T_gamma5,5,5,reshape(m_dti,[1,Num,Num]),3,1);

T_gamma6=fn_c(m_dtz,3,3,reshape(m_dtz,[1,Num,Num]),3,1);
T_gamma6=fn_c(T_gamma6,5,5,reshape(m_dty,[1,Num,Num]),3,1);

```

Then we permute the tensor and reshape them, after all we get the six gamma matrix. (notice f1,...,f4 are the four gamma matrix).

```

T_gamma1=permute(T_gamma1,[1,3,5,2,4,6]);
m_gamma1=reshape(T_gamma1,Num^3,Num^3);
T_gamma2=permute(T_gamma2,[1,3,5,2,4,6]);
m_gamma2=reshape(T_gamma2,Num^3,Num^3);
T_gamma3=permute(T_gamma3,[1,3,5,2,4,6]);
m_gamma3=reshape(T_gamma3,Num^3,Num^3);
T_gamma4=permute(T_gamma4,[1,3,5,2,4,6]);
m_gamma4=reshape(T_gamma4,Num^3,Num^3);
T_gamma5=permute(T_gamma5,[1,3,5,2,4,6]);
m_gamma5=reshape(T_gamma5,Num^3,Num^3);
T_gamma6=permute(T_gamma6,[1,3,5,2,4,6]);
m_gamma6=reshape(T_gamma6,Num^3,Num^3);

f1=permute(f1,[1,3,2,4]);
f2=permute(f2,[1,3,2,4]);
f3=permute(f3,[1,3,2,4]);
f4=permute(f4,[1,3,2,4]);
f1=reshape(f1,Num^2,Num^2);
f2=reshape(f2,Num^2,Num^2);
f3=reshape(f3,Num^2,Num^2);
f4=reshape(f4,Num^2,Num^2);

```

Finally we show the results:

```

disp('m_gamma1='); disp(num2str(m_gamma1));

```

```

m_gamma1=
0 1 0 0 0 0 0 0
1 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0
0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0
0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 1
0 0 0 0 0 0 1 0

```

```

disp('m_gamma2='); disp(num2str(m_gamma2));

```

```

m_gamma2=
0 0 1 0 0 0 0 0
0 0 0 -1 0 0 0 0
1 0 0 0 0 0 0 0
0 -1 0 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 -1
0 0 0 0 1 0 0 0
0 0 0 0 0 -1 0 0

```

```
disp('m_gamma3='); disp(num2str(m_gamma3));
```

```
m_gamma3=
0  0  0  0  1  0  0  0
0  0  0  0  0 -1  0  0
0  0  0  0  0  0 -1  0
0  0  0  0  0  0  0  1
1  0  0  0  0  0  0  0
0 -1  0  0  0  0  0  0
0  0 -1  0  0  0  0  0
0  0  0  1  0  0  0  0
```

```
disp('m_gamma4='); disp(num2str(m_gamma4));
```

```
m_gamma4=
0+0i  0-1i  0+0i  0+0i  0+0i  0+0i  0+0i  0+0i
0+1i  0+0i  0+0i  0+0i  0+0i  0+0i  0+0i  0+0i
0+0i  0+0i  0+0i  0-1i  0+0i  0+0i  0+0i  0+0i
0+0i  0+0i  0+1i  0+0i  0+0i  0+0i  0+0i  0+0i
0+0i  0+0i  0+0i  0+0i  0+0i  0-1i  0+0i  0+0i
0+0i  0+0i  0+0i  0+0i  0+1i  0+0i  0+0i  0+0i
0+0i  0+0i  0+0i  0+0i  0+0i  0+0i  0+0i  0-1i
0+0i  0+0i  0+0i  0+0i  0+0i  0+0i  0+1i  0+0i
```

```
disp('m_gamma5='); disp(num2str(m_gamma5));
```

```
m_gamma5=
0+0i  0+0i  0-1i  0+0i  0+0i  0+0i  0+0i  0+0i
0+0i  0+0i  0+0i  0+1i  0+0i  0+0i  0+0i  0+0i
0+1i  0+0i  0+0i  0+0i  0+0i  0+0i  0+0i  0+0i
0+0i  0-1i  0+0i  0+0i  0+0i  0+0i  0+0i  0+0i
0+0i  0+0i  0+0i  0+0i  0+0i  0+0i  0-1i  0+0i
0+0i  0+0i  0+0i  0+0i  0+0i  0+0i  0+0i  0+1i
0+0i  0+0i  0+0i  0+0i  0+1i  0+0i  0+0i  0+0i
0+0i  0+0i  0+0i  0+0i  0+0i  0-1i  0+0i  0+0i
```

```
disp('m_gamma6='); disp(num2str(m_gamma6));
```

```
m_gamma6=
0+0i  0+0i  0+0i  0+0i  0-1i  0+0i  0+0i  0+0i
0+0i  0+0i  0+0i  0+0i  0+0i  0+1i  0+0i  0+0i
0+0i  0+0i  0+0i  0+0i  0+0i  0+0i  0+1i  0+0i
0+0i  0+0i  0+0i  0+0i  0+0i  0+0i  0+0i  0-1i
0+1i  0+0i  0+0i  0+0i  0+0i  0+0i  0+0i  0+0i
0+0i  0-1i  0+0i  0+0i  0+0i  0+0i  0+0i  0+0i
0+0i  0+0i  0-1i  0+0i  0+0i  0+0i  0+0i  0+0i
0+0i  0+0i  0+0i  0+1i  0+0i  0+0i  0+0i  0+0i
```

```
disp('f1='); disp(num2str(f1));
```

```
f1=
0  1  0  0
1  0  0  0
0  0  0  1
0  0  1  0
```

```
disp('f2='); disp(num2str(f2));
```

```
f2=
0  0  1  0
0  0  0 -1
1  0  0  0
```

0 -1 0 0

```
disp('f3='); disp(num2str(f3));
```

```
f3=
0+0i    0-1i    0+0i    0+0i
0+1i    0+0i    0+0i    0+0i
0+0i    0+0i    0+0i    0-1i
0+0i    0+0i    0+1i    0+0i
```

```
disp('f4='); disp(num2str(f4));
```

```
f4=
0+0i    0+0i    0-1i    0+0i
0+0i    0+0i    0+0i    0+1i
0+1i    0+0i    0+0i    0+0i
0+0i    0-1i    0+0i    0+0i
```