

FeynCalc Tutorial 1

(Dated: September 8, 2015)

In this tutorial, we will learn to use Mathematica and FeynCalc to contract Lorentz indices. After some practice to get familiar with FeynCalc, we will do a complete Feynman diagram calculation for two massive scalar annihilation into two photons.

You are encouraged to work in groups of two or three, and I assume at least one person in each group has used Mathematica before. For the tutorial today, if you can do simple calculation like $\sin(\pi/2)$ with Mathematica, and know how to use the Mathematica help document, you have enough knowledge to be the group leader.

Throughout the tutorial, I will stress some good coding habits. They may seem to be unnecessary for the simple calculation currently, but they are very beneficial in your future research. I will also introduce some tricks which are convenient when dealing with complicated calculations.

Let's first test whether the FeynCalc is properly installed in your computer. Run the following command in your Mathematica notebook. If you get the same output, you can start from section. II.

```
In[1]:= << FeynCalc`
```

```
FeynCalc 9.0.0. For help, type ?FeynCalc, use the  
help browser or visit www.feyncalc.org.
```

I. Installation

FeynCalc package can be downloaded on feyncalc.org. You can follow the instructions to install it in your computer after the class. In class, we have a version not requiring installation. Copy the folder “FeynCalc” to your local disk and start with “Tutorial.1.nb”.

II. Lorentz indices contraction

Let's start with a simple example

- Example 1: $g_{\mu\nu}p^\mu p^\nu$ with $p^2 = m^2$.

Think about how we calculate it by hand. We know that $\mu, \nu = 0, 1, 2, 3$ and repeated indices are summed over implicitly, so we can get $g_{\mu\nu}p^\mu p^\nu = p^2$. We then use the relation $p^2 = m^2$ and replace p^2 by m^2 .

So there are two things we need to tell FeynCalc. First, we need to type in the expression $g_{\mu\nu}p^\mu p^\nu$ in the language FeynCalc can understand. Second, we need to tell it $p^2 = m^2$. After these two steps, we can call FeynCalc functions to evaluate the expression. In Mathematica, these are realized as

```

In[2]:= ScalarProduct [p, p] = m2;

In[3]:= MT [μ, ν] FV [p, μ] FV [p, ν]
Contract [%]

Out[3]=  $p^\mu p^\nu g^{\mu\nu}$ 

Out[4]=  $m^2$ 

```

In the first row, `ScalarProduct[p,p] = m2` tells FeynCalc that $p^2 = m^2$. The second row is the expression we want to evaluate, in the language of FeynCalc. The translation rule for vector, metric tensor and Levi-Civita symbol is in the table below. Note that Mathematica is case sensitive. In the third row, Mathematica will replace % by the last expression it evaluated, which is the second row in the example. The command “Contract” ask FeynCalc to contract repeated Lorentz indices, giving us the result m^2 .

	symbol	in FeynCalc
Four vector	p^μ, p_μ	FourVector[p,μ] or FV[p,μ]
Metric tensor	$g^{\mu\nu}, g_{\mu\nu}$	MetricTensor[μ, ν] or MT[μ, ν]
Levi-Civita	$\epsilon^{\alpha\beta\mu\nu}$	LeviCivita[α, β, μ, ν] or LC[α, β, μ, ν]
Scalar Product	$p \cdot k$	ScalarProduct[p, k] or SP[p, k]

Several things need to be explained in this example. First, FeynCalc does not specify whether a Lorentz index is upper or lower. As long as all Lorentz indices are contracted and the result is a scalar, it is not a problem (why?). However, you must be very careful when the final result is not a scalar, i.e. the result includes leftover Lorentz indices. Second, FeynCalc assumes the summation over repeated Lorentz indices. So there is no need to write the sum over μ and ν explicitly in the example. Third, the relation $p^2 = m^2$ does not automatically disappear after the calculation. You have to manually erase it from the memory.

The second row and the third row of the above example can be combined into one sentence

```
Contract [MT [μ, ν] FV [p, μ] FV [p, ν]]
```

Try it and compare the output with the output of the example. The two-line expression in the example has the virtue that we can check the correctness of our input easily. If you are familiar with the usage of % in Mathematica, you will know it always causes confusions and mistakes. The usage of % in this example is almost the only place it is encouraged.

Although not necessary, it is encouraged to use p, k and q for momentum, especially, $p1, p2 \dots$ for incoming particles, $k1, k2 \dots$ for outgoing particles and $q1, q2 \dots$ for loop momentum. It is also encouraged to use Greek letters for Lorentz indices. The Greek letters can be typed easily by hitting ESC key, typing some letter(s), and press ESC again. For example, μ can be typed by ESC, letter m, and ESC. Try different letters to see what you can get, and remember the key combination for six Greek letters you prefer to use a lot in the future. Don't use π , since it is already be assigned value 3.14... in Mathematica.

Enough for the first example, let's move on to the next.

- Example 2: $(p - k)^\mu(p - k)_\mu - (p - k)^2 - p^2$

```
In[5]:= a1 = Contract [FV[p - k, μ] FV[p - k, μ]]
a2 = SP[p - k] + SP[p]
a1 - a2
ExpandScalarProduct[a1 - a2]
```

Out[5]= $-2(\bar{k} \cdot \bar{p}) + \bar{k}^2 + m^2$

Out[6]= $(\bar{p} - \bar{k})^2 + m^2$

Out[7]= $-(\bar{p} - \bar{k})^2 - 2(\bar{k} \cdot \bar{p}) + \bar{k}^2$

Out[8]= $-m^2$

In the first row, $FV[p - k, \mu]$ is the same as $FV[p, \mu] - FV[k, \mu]$. In the second row, $SP[p - k]$ is the same as $SP[p - k, p - k]$, similarly for $SP[p]$. In the third row, we expect $(p - k)^\mu(p - k)_\mu$ will cancel $(p - k)^2$. But it didn't happen since FeynCalc does not automatically expand scalar product such as $SP[p - k]$. In the last row, the command "ExpandScalarProduct" expands $SP[p - k]$ and does the cancellation.

The last row can also be written as

```
a1 - a2 // ExpandScalarProduct
```

The symbol "//" means to apply the function after it to the expression in front. This usage of function is actually more common in research.

Notice that in this example, FeynCalc still remembers $p^2 = m^2$.

The third example is about Levi-Civita symbol

- Example 3: $\epsilon^{\alpha\beta\mu\nu} g_{\alpha\beta}$

```
In[9]:= LC[α, β, μ, ν] MT[α, β]
Contract[%]
```

Out[9]= $\bar{g}^{\alpha\beta} \epsilon^{\alpha\beta\mu\nu}$

Out[10]= 0

It gives Zero! (why?)

Let's conclude this section by clearing the memory of FeynCalc

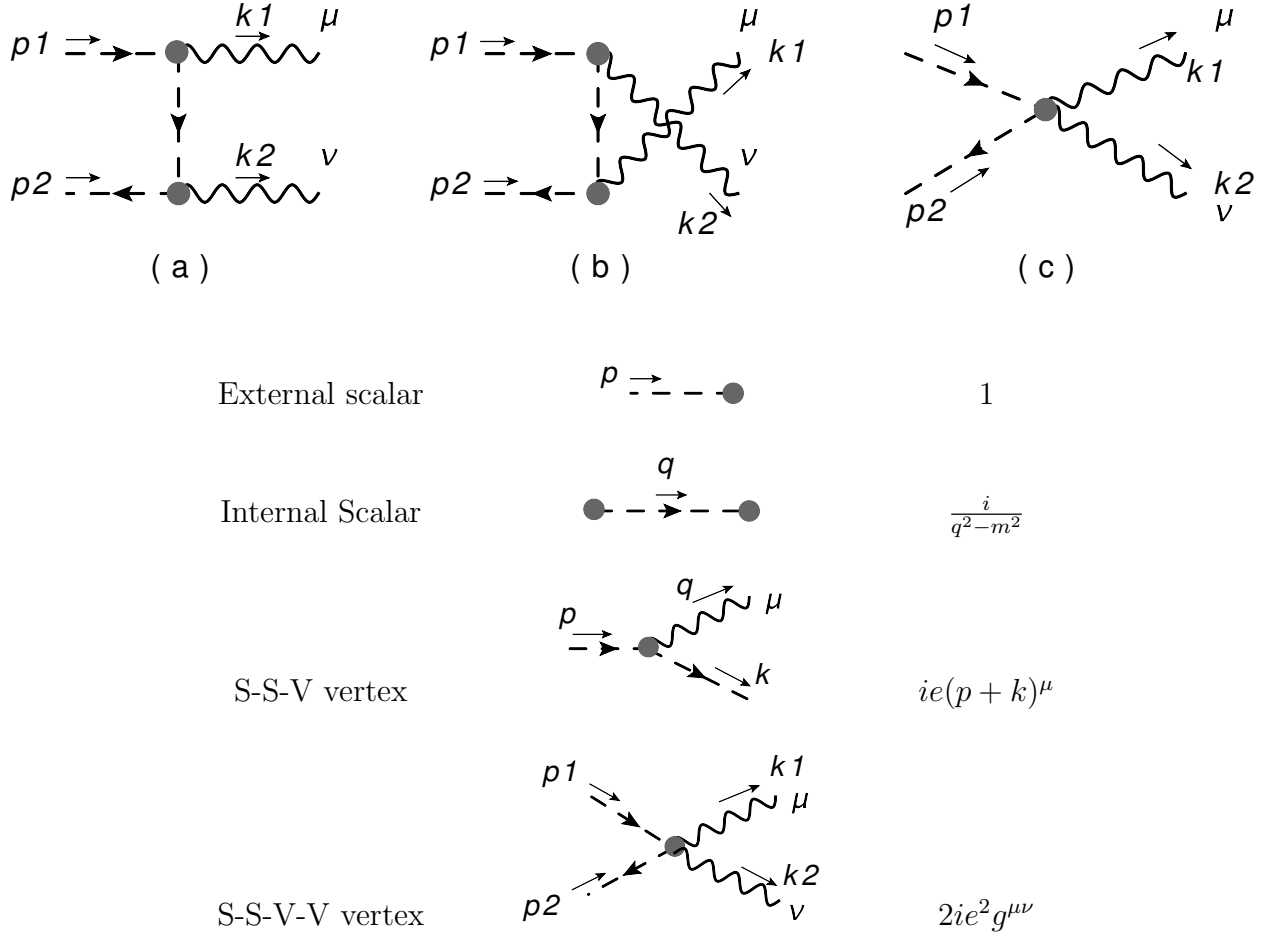
```
In[11]:= ClearScalarProducts
```

Now evaluate $SP[p]$ again to see whether it still gives m^2 .

III. Project: Massive scalar annihilation to two photons

The standard procedure to calculate the scattering matrix for a process is (1) draw all the Feynman diagrams, (2) write down the expression for each Feynman diagram and (3) do the algebra. You will learn how to draw the Feynman diagrams and derive the Feynman rules later in this QFT course. At this stage, we do all this hard work for you. However, you need to figure out how to use the Feynman rules to write down the expression for each diagram, and calculate the scattering matrix element.

There are three Feynman diagrams at tree level, shown in the figure below. The needed Feynman rules are also showed.



In the Feynman rules, m and e are the mass and the electric charge of the scalar. Notice the Feynman rule of the S-S-V vertex depends on the relative direction of the momentum and the charge current (indicated by the arrow on the dashed line).

Since each diagram for this process has two indices: μ and ν . The expressions are Lorentz tensors. Let's call them $M_a^{\mu\nu}$, $M_b^{\mu\nu}$ and $M_c^{\mu\nu}$. The matrix element is defined by

$$|M|^2 = (M_a^{\mu\nu} + M_b^{\mu\nu} + M_c^{\mu\nu}) \times (M_a^{\mu'\nu'} + M_b^{\mu'\nu'} + M_c^{\mu'\nu'})^* \times \Sigma(k_1)_{\mu\mu'} \Sigma(k_2)_{\nu\nu'}, \quad (1)$$

where

$$\Sigma(k_1)_{\mu\nu} = \Sigma(k_2)_{\mu\nu} = -g_{\mu\nu} + \frac{k_1^\mu k_2^\nu + k_1^\nu k_2^\mu}{k_1 \cdot k_2} \quad (2)$$

is from the summation of photon polarization states.

In the following, we show how to calculate

$$(M_a^{\mu\nu}) \times (M_a^{\mu'\nu'})^* \times \Sigma(k_1)_{\mu\mu'} \Sigma(k_2)_{\nu\nu'} \quad (3)$$

step by step. You need to follow these steps to calculate $|M|^2$.

From the Feynman rule, we can get

$$M_a^{\mu\nu} = [i e (p_1 - k_1 + p_1)^\mu] \times \frac{i}{(p_1 - k_1)^2 - m^2} \times [i e (p_1 - k_1 + (-p_2))^\nu] \quad (4)$$

Now lets pass the calculation to FeynCalc. First we need to tell FeynCalc the values for scalar products

```
In[12]:= ScalarProduct [p1, p1] = m2;
ScalarProduct [p2, p2] = m2;
ScalarProduct [k1, k1] = 0;
ScalarProduct [k2, k2] = 0;

In[16]:= ScalarProduct [p1, p2] =  $\frac{s}{2} - m^2$ ;
ScalarProduct [p1, k1] =  $-\frac{t - m^2}{2}$ ;
ScalarProduct [p2, k1] =  $-\frac{u - m^2}{2}$ ;
```

where we also use the Mandelstam variables $s = (p_1 + p_2)^2$, $t = (p_1 - k_1)^2$ and $u = (p_2 - k_1)^2$ to work out the last three scalar products.

Then we tell FeynCalc the expression of $M_a^{\mu\nu}$

```
In[19]:= Ma =  $\frac{\mathbf{I}}{\mathbf{SP}[\mathbf{p1} - \mathbf{k1}] - m^2} (\mathbf{I} \mathbf{e} \mathbf{FV}[\mathbf{2} \mathbf{p1} - \mathbf{k1}, \mu]) (\mathbf{I} \mathbf{e} \mathbf{FV}[\mathbf{p1} - \mathbf{k1} - \mathbf{p2}, \nu])$ 

Out[19]=  $-\frac{i e^2 (2 \bar{p1} - \bar{k1})^\mu (-\bar{k1} + \bar{p1} - \bar{p2})^\nu}{(\bar{p1} - \bar{k1})^2 - m^2}$ 
```

With $M_a^{\mu\nu}$, it is trivial to define $(M_a^{\mu'\nu'})^*$

```
In[20]:= MaC = ComplexConjugate [Ma] /. {μ => μC, ν => νC}

Out[20]=  $\frac{i e^2 (2 \bar{p1} - \bar{k1})^{\mu C} (-\bar{k1} + \bar{p1} - \bar{p2})^{\nu C}}{(\bar{p1} - \bar{k1})^2 - m^2}$ 
```

where “ComplexConjugate” is a FeynCalc function, giving the complex conjugate of its argument. The funny symbol “/.” is a short-hand notation for Mathematica function

“ReplaceAll”. It replaces μ and ν in the expression before “/.” by μC and νC . ReplaceAll is one of the most useful functions in practice. Make sure you understand its usage.

Now we can ask Mathematica to contract the Lorentz indices

```
In[21]:= Ma2 =
  Ma * MaC * ( -MT[μ, μC] +  $\frac{\text{FV}[\mathbf{k1}, \mu] \text{FV}[\mathbf{k2}, \mu\text{C}] + \text{FV}[\mathbf{k1}, \mu\text{C}] \text{FV}[\mathbf{k2}, \mu]}{\text{SP}[\mathbf{k1}, \mathbf{k2}]}$  )
  ( -MT[ν, νC] +  $\frac{\text{FV}[\mathbf{k1}, \nu] \text{FV}[\mathbf{k2}, \nu\text{C}] + \text{FV}[\mathbf{k1}, \nu\text{C}] \text{FV}[\mathbf{k2}, \nu]}{\text{SP}[\mathbf{k1}, \mathbf{k2}]}$  ) // Contract // Simplify
Out[21]=  $\frac{4 e^4 (m^2 (\mathbf{k1} \cdot \mathbf{k2}) + (t - m^2) (\mathbf{k2} \cdot \mathbf{p1})) ((4 m^2 - s) (\mathbf{k1} \cdot \mathbf{k2}) + (t - u) (\mathbf{k2} \cdot \mathbf{p1} - \mathbf{k2} \cdot \mathbf{p2}))}{(m^2 - t)^2 (\mathbf{k1} \cdot \mathbf{k2})^2}$ 
```

This is from Eqs. (2) and (3). We ask Mathematica to first contract the Lorentz indices, then simplify the result, and save the result in variable “Ma2”, meaning “Ma”-squared.

```
In[22]:= Ma2 /. k2 -> p1 + p2 - k1 // ExpandScalarProduct // Simplify
res1 = % /. u -> 2 m^2 - s - t // Simplify
Out[22]=  $\frac{4 e^4 (3 m^4 - m^2 (s + 3 t + u) + t (s + t)) (8 m^4 - 2 m^2 (s + 2 (t + u)) + s (t + u) + (t - u)^2)}{(m^2 - t)^2 (-2 m^2 + t + u)^2}$ 
Out[23]=  $\frac{16 e^4 (m^4 - 2 m^2 t + t (s + t))^2}{s^2 (m^2 - t)^2}$ 
```

In the first row we use momentum conservation. In the second row we use the relation $s + t + u = 2m^2$ to further simplify the result, and save the result in variable “res1”.

Now its your turn to write down the expression for $M_b^{\mu\nu}$ and $M_c^{\mu\nu}$ with the Feynman rules. (First you need to understand how we get $M_a^{\mu\nu}$ in Eq. (4).) Then make proper changes to the codes above to calculate $|M|^2$ in Eq. (1).

Now repeat the above calculation of $|M|^2$, but this time use

$$\Sigma(k_1)_{\mu\nu} = \Sigma(k_2)_{\mu\nu} = -g_{\mu\nu}. \quad (5)$$

instead of Eq. (2). Do you get the same result? If you do, congratulations, you have showed the famous Ward identity, also known as gauge invariance, of scalar QED at tree level.