COMPUTING FOURIER TRANSFORMS

Marek Matas

UPC group meeting at Decin 2.5.-3.5.2018

WHY WOULD I WANT TO DO THAT?

BASIC PROPERTIES OF FT

In particle physics, one can use FT to go from one representation to another.

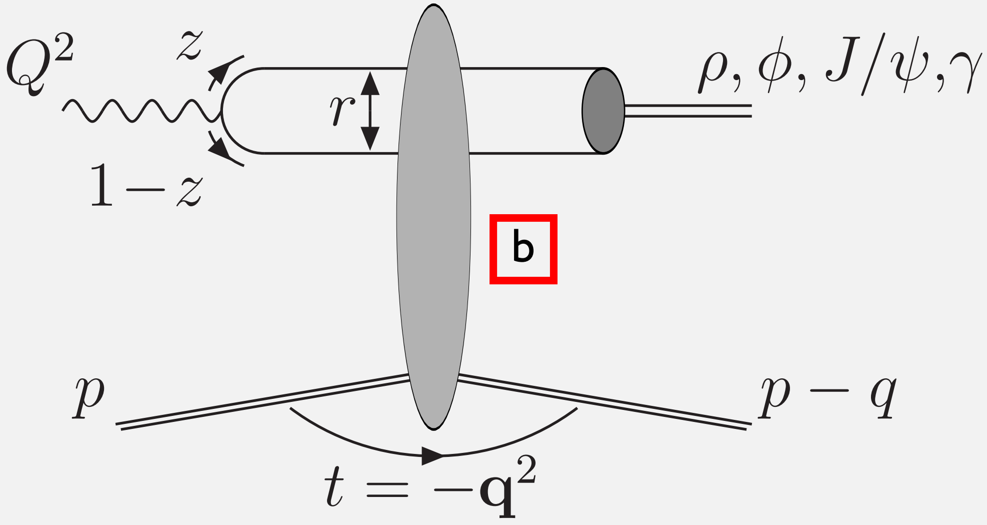
One of the possible examples is the computation of the unintegrated gluon distributions from the scattering amplitude in position space.

$$x_{tar}G^{(2)}(\boxed{k_t}, Y) = \frac{N_c k_t^2 S}{8\pi^4 \alpha_s} \int d^2r e^{-ik_t r}[1 - N(\boxed{r,} Y)]$$

BASIC PROPERTIES OF FT

In particle physics, one can use FT to go from one representation to another.

Another example is the computation of the exchanged transverse momentum dependence from the impact parameter dependent scattering amplitude.

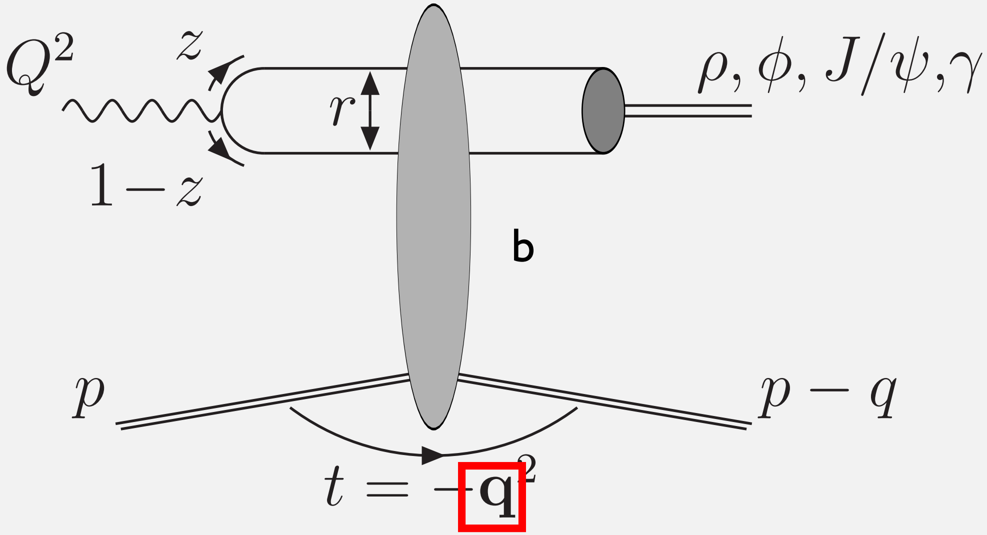


$$\tilde{T}(\mathbf{r},\mathbf{q};Y) = \int d^2b\, e^{i\mathbf{q}\cdot\mathbf{b}}\, T(\mathbf{r}\boxed{,\mathbf{b};}Y)$$

BASIC PROPERTIES OF FT

In particle physics, one can use FT to go from one representation to another.

Another example is the computation of the exchanged transverse momentum dependence from the impact parameter dependent scattering amplitude.



$$\tilde{T}(\mathbf{r}, \boxed{\mathbf{q};} Y) = \int d^2b \; e^{i\mathbf{q}\cdot\mathbf{b}} \; T(\mathbf{r}, \mathbf{b}; Y)$$

HOW DO I DO IT?

COMPUTING FT

Let’s take the dipole gluon distribution as an example.

$$x_{tar}G^{(2)}(k_t, Y) = \frac{N_c k_t^2 S}{8\pi^4 \alpha_s} \int d^2 r e^{-ik_t r}[1 - N(r, Y)]$$

How do we compute this complex integral?

We move to polar coordinates.

COMPUTING FT

Let’s take the dipole gluon distribution as an example.

$$x_{tar}G^{(2)}(k_t,Y)=\frac{N_c k_t^2 S}{8\pi^4\alpha_s}\int d^2r e^{-ik_t r}[1-N(r,Y)]$$

$\searrow$

rdrdφ

COMPUTING FT

Let’s take the dipole gluon distribution as an example.

cos(**φ**)

$$x_{tar}G^{(2)}(k_t, Y) = \frac{N_c k_t^2 S}{8\pi^4 \alpha_s} \int d^2r e^{-ik_t r}[1 - N(r, Y)]$$

rdrd**φ**

COMPUTING FT

Let’s take the dipole gluon distribution as an example.

cos(**φ**)

$$x_{tar}G^{(2)}(k_t, Y) = \frac{N_c k_t^2 S}{8\pi^4 \alpha_s} \int d^2 r e^{-ik_t r} \boxed{[1 - N(r, Y)]}$$

rdrd**φ**

The integrand doesn’t depend on **φ**, only on r.
What do we do? Factorize!

COMPUTING FT

Let's take the dipole gluon distribution as an example.

cos(**φ**)

$$x_{tar}G^{(2)}(k_t, Y) = \frac{N_c k_t^2 S}{8\pi^4 \alpha_s} \int d^2 r e^{-ik_t r} \boxed{[1 - N(r, Y)]}$$

Bessel function definition

rdrd**φ**

$$\int d\boldsymbol{\varphi} e^{-ik_t r cos(\boldsymbol{\varphi})} = 2\pi J_0(k_t r)$$

The integrand doesn't depend on **φ**, only on r.
What do we do? Factorize!

COMPUTING FT

Let's take the dipole gluon distribution as an example.

cos(**φ**)

$$x_{tar}G^{(2)}(k_t, Y) = \frac{N_c k_t^2 S}{8\pi^4 \alpha_s} \int d^2r e^{-ik_t r} \boxed{[1 - N(r, Y)]}$$

rdrd**φ**

Bessel function definition

$$\int d\boldsymbol{\varphi} e^{-ik_t r cos(\boldsymbol{\varphi})} = 2\pi J_0(k_t r)$$

The integrand doesn't depend on **φ**, only on r.
What do we do? Factorize!

$$x_{tar}G^{(2)}(k_t, Y) = \frac{N_c k_t^2 S}{8\pi^4 \alpha_s} \int dr J_0(k_t r)(1 - N(r, Y))2\pi r$$

COMPUTING FT

Now we have a real integrand!

$$x_{tar}G^{(2)}(k_t,Y) = \frac{N_c k_t^2 S}{8\pi^4 \alpha_s} \int dr J_0(k_t r)(1 - N(r,Y))2\pi r$$

COMPUTING FT

Now we have a real integrand!

$$x_{tar}G^{(2)}(k_t, Y) = \boxed{\frac{N_c k_t^2 S}{8\pi^4 \alpha_s} \int dr J_0(k_t r)(1 - N(r, Y))2\pi r}$$

We can use the usual Simpson method to integrate it.

COMPUTING FT

Now we have a real integrand!

$$x_{tar}G^{(2)}(k_t, Y) = \boxed{\frac{N_c k_t^2 S}{8\pi^4 \alpha_s} \int dr J_0(k_t r)(1 - N(r, Y))2\pi r}$$

We can use the usual Simpson method to integrate it.

Our work here is done then, this is easy. Should we go to conclusions?

There is still one problem.

COMPUTING FT

Now we have a real integrand!

$$x_{tar}G^{(2)}(k_t, Y) = \frac{N_c k_t^2 S}{8\pi^4 \alpha_s} \int dr \boxed{J_0(k_t r)} (1 - N(r, Y)) 2\pi r$$

We can use the usual Simpson method to integrate it.

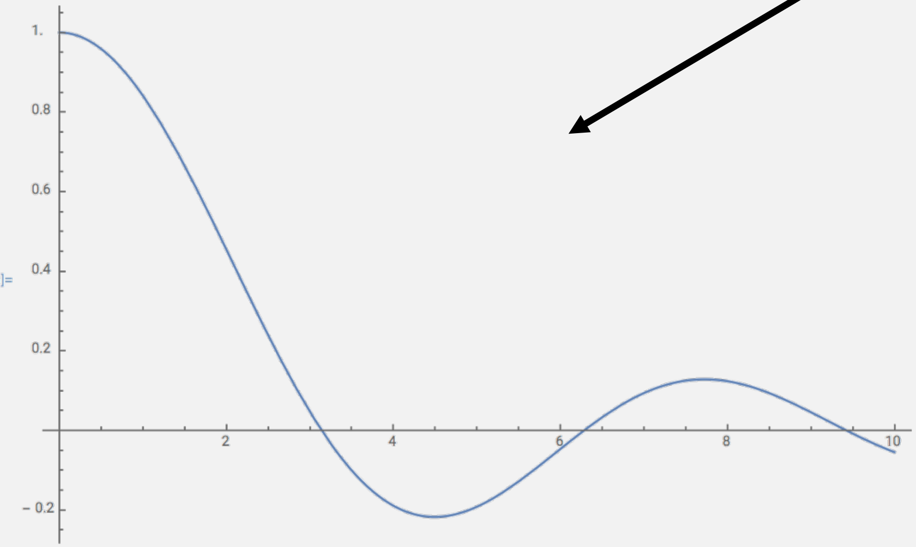
Our work here is done then, this is easy. Should we go to conclusions?

There is still one problem.

The Bessel function oscillates and makes the convergence of the integral slow.

COMPUTING FT

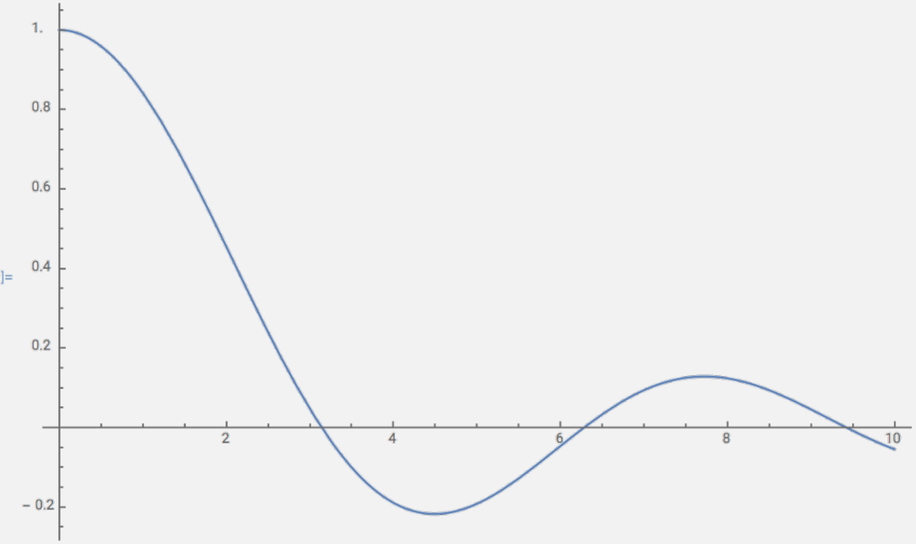
$$x_{tar}G^{(2)}(k_t, Y) = \frac{N_c k_t^2 S}{8\pi^4 \alpha_s} \int dr \boxed{J_0(k_t r)} (1 - N(r, Y)) 2\pi r$$

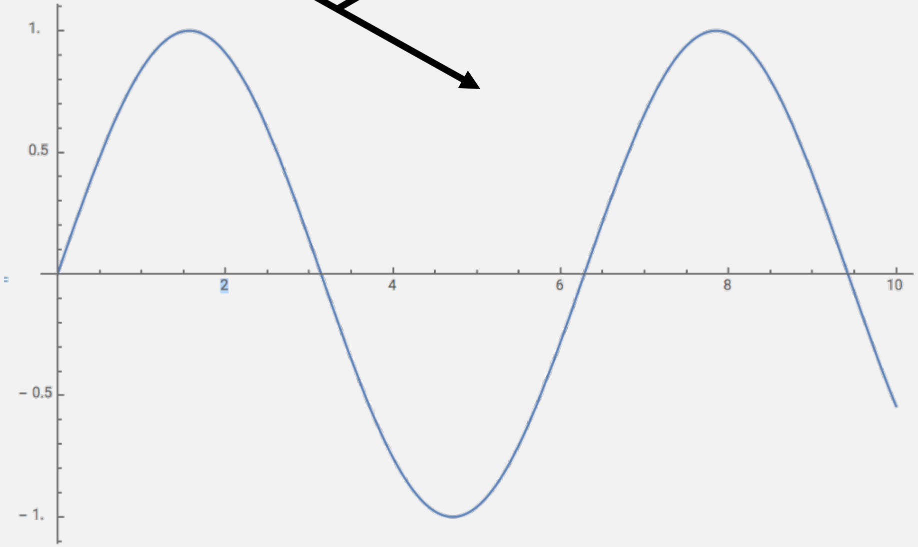


J$_0$(x) is an oscillatory function, that slowly converges to 0. Did we forget about something?

COMPUTING FT

$$x_{tar}G^{(2)}(k_t, Y) = \frac{N_c k_t^2 S}{8\pi^4 \alpha_s} \int dr\, \boxed{J_0(k_t r)}(1 - N(r, Y))2\pi \boxed{r}$$





When we include *r*, Bessel function does not converge at all!
Do we have to integrate to infinity?
In finite time?

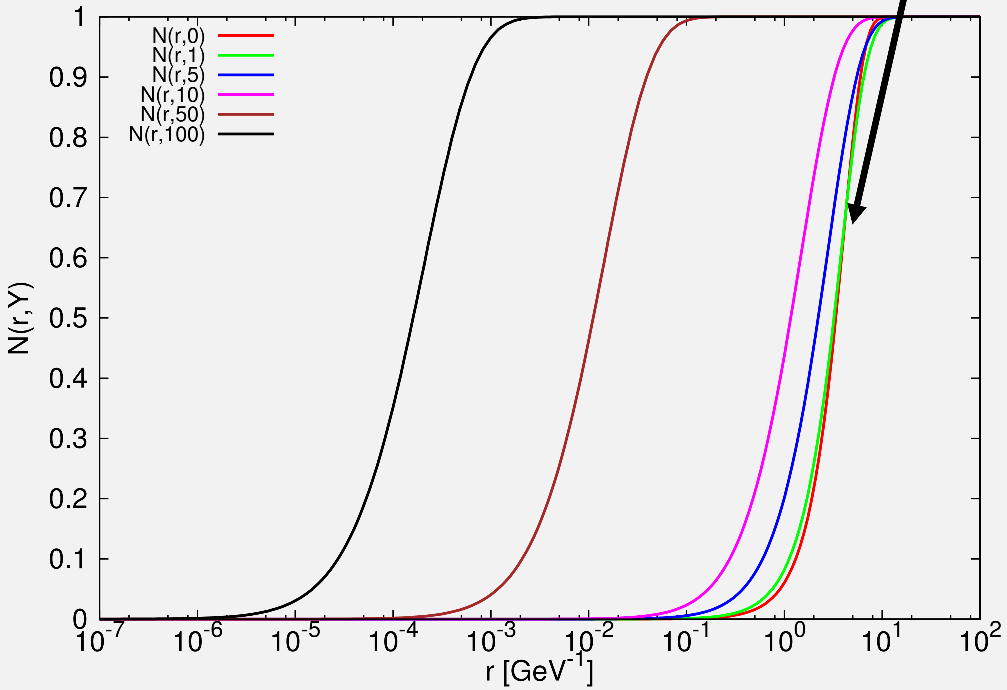
COMPUTING FT

$$x_{tar}G^{(2)}(k_t, Y) = \frac{N_c k_t^2 S}{8\pi^4 \alpha_s} \int dr\, J_0(k_t r)(1 - \boxed{N(r, Y)})2\pi r$$

When N(r,Y) reaches 1, the term (1 – N(r,Y)) kills the integrand.

COMPUTING FT

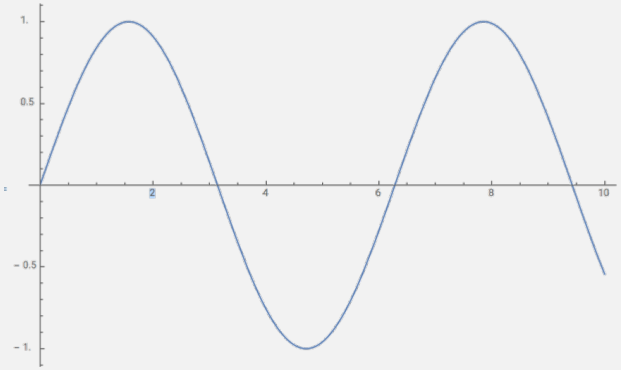
$$x_{tar}G^{(2)}(k_t, Y) = \frac{N_c k_t^2 S}{8\pi^4 \alpha_s} \int dr\, J_0(k_t r)(1 - \boxed{N(r,Y)})2\pi r$$



This happens at r ~10 GeV^{-1}, why do we worry then?

COMPUTING FT

$$x_{tar}G^{(2)}(k_t,Y) = \frac{N_c k_t^2 S}{8\pi^4\alpha_s}\int dr\, J_0(k_t r)(1 - N(r,Y))2\pi r$$



Up to $J_0(10)$, we have ~ 1.5 wavelengths.

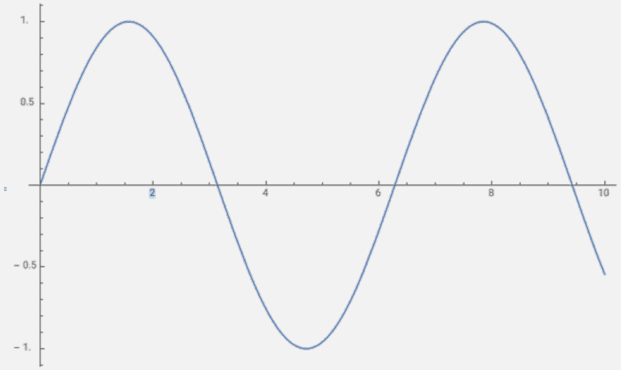
We can still use Simpson. What is the problem?

COMPUTING FT

$$x_{tar}G^{(2)}(k_t, Y) = \frac{N_c k_t^2 S}{8\pi^4 \alpha_s} \int dr\, J_0\boxed{(k_t r)}(1 - N(r, Y))2\pi r$$

Up to $J_0(10)$, we have ~ 1.5 wavelengths.

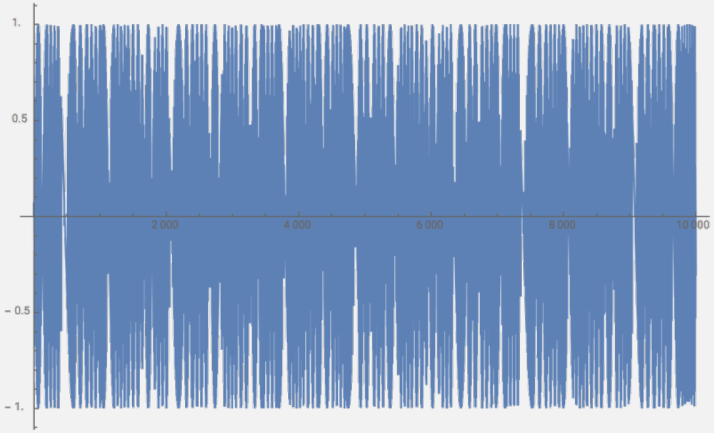
We can still use Simpson. What is the problem?



If k_t is very big (say 1000 GeV),

J_0 doesn't get suppressed until $J_0(10\,000)$.

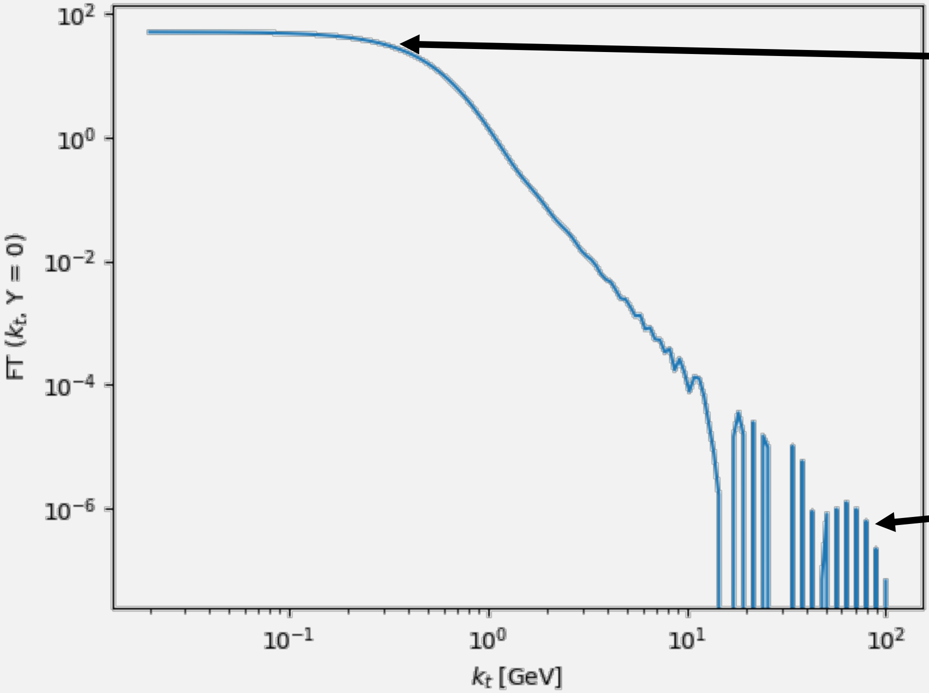
Try Simpsonning that..

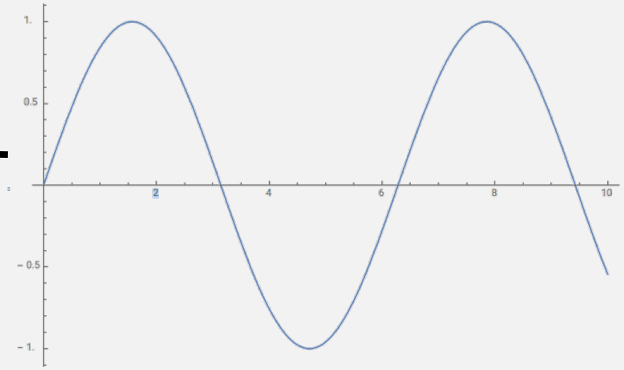


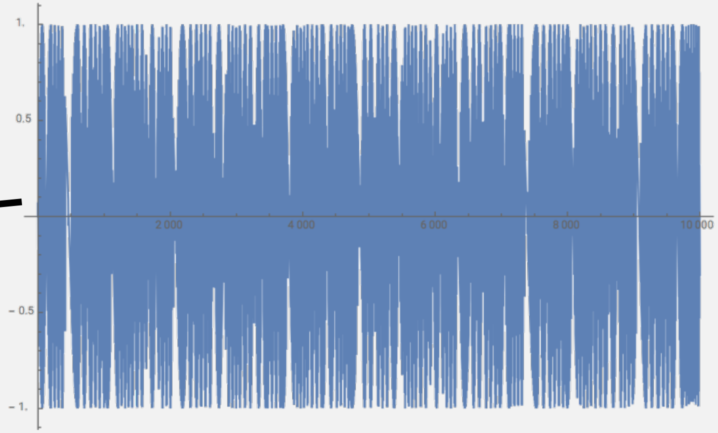
COMPUTING FT

$$x_{tar} G^{(2)}(k_t, Y) = \frac{N_c k_t^2 S}{8\pi^4 \alpha_s} \int dr\, J_0 \boxed{(k_t r)} (1 - N(r, Y)) 2\pi r$$

If we try anyway..







COMPUTING FT

How do we tackle that? Here are some key ideas:

1. Find the table of zeros of the Bessel function, integrate separately all positive, and negative parts and then subtract them.

2. Use pre-implemented FFT routines in some libraries, that have already taken care of these problems for us.

3. Cutting and extrapolating the regions of high k_t.

4. Use the old methods, but increase precision as much as we can.

COMPUTING FT

How do we tackle that? Here are some key ideas:

1. ~~Find the table of zeros of the Bessel function, integrate separately all positive, and negative parts and then subtract them.~~

2. Use pre-implemented FFT routines in some libraries, that have already taken care of these problems for us.

3. Cutting and extrapolating the regions of high k_t.

4. Use the old methods, but increase precision as much as we can.

The error of the computation does not come from the subtractions, but rather from imprecise integration of the sub-integrals. This will not fix this problem.

COMPUTING FT

How do we tackle that? Here are some key ideas:

1. ~~Find the table of zeros of the Bessel function, integrate separately all positive, and negative parts and then subtract them.~~

2. ~~Use pre-implemented FFT routines in some libraries, that have already taken care of these problems for us.~~

3. Cutting and extrapolating the regions of high k_t.

4. Use the old methods, but increase precision as much as we can.

After spending tens of hours trying to understand the implementation and making it work, I gave up. In case someone knows how to do these, please let me know.

COMPUTING FT

How do we tackle that? Here are some key ideas:

1. ~~Find the table of zeros of the Bessel function, integrate separately all positive, and negative parts and then subtract them.~~
2. ~~Use pre-implemented FFT routines in some libraries, that have already taken care of these problems for us.~~
3. ~~Cutting and extrapolating the regions of high k_t.~~
4. Use the old methods, but increase precision as much as we can.

Cutting and extrapolating will work if we are then done with the distribution. If not, it creates a bump in its derivation, which spoils convolution with other distributions.

COMPUTING FT

How do we tackle that? Here are some key ideas:

1. ~~Find the table of zeros of the Bessel function, integrate separately all positive, and negative parts and then subtract them.~~

2. ~~Use pre-implemented FFT routines in some libraries, that have already taken care of these problems for us.~~

3. ~~Cutting and extrapolating the regions of high k_t.~~

4. Use the old methods, but increase precision as much as we can.

Surprisingly enough, this is the way!

COMPUTING FT

How do we increase the precision?

1. Increase the amount of steps in the Simpson method.
   1. We can make sure, that we have at least 1000 points per oscillation of the Bessel function. This alone will not fix our troubles though.

$$x_{tar}G^{(2)}(k_t, Y) = \frac{N_c k_t^2 S}{8\pi^4 \alpha_s} \int \boxed{dr} J_0(k_t r)(1 - N(r, Y))2\pi r$$

COMPUTING FT

How do we increase the precision?

1. Increase the amount of steps in the Simpson method.
    1. We can make sure, that we have at least 1000 points per oscillation of the Bessel function. This alone will not fix our troubles though.

2. Increase the precision of *N(r,Y)*.
    1. This is crucial, since the transition point from 0 to 1 in N is the region most responsible for the final value of the Fourier transform.

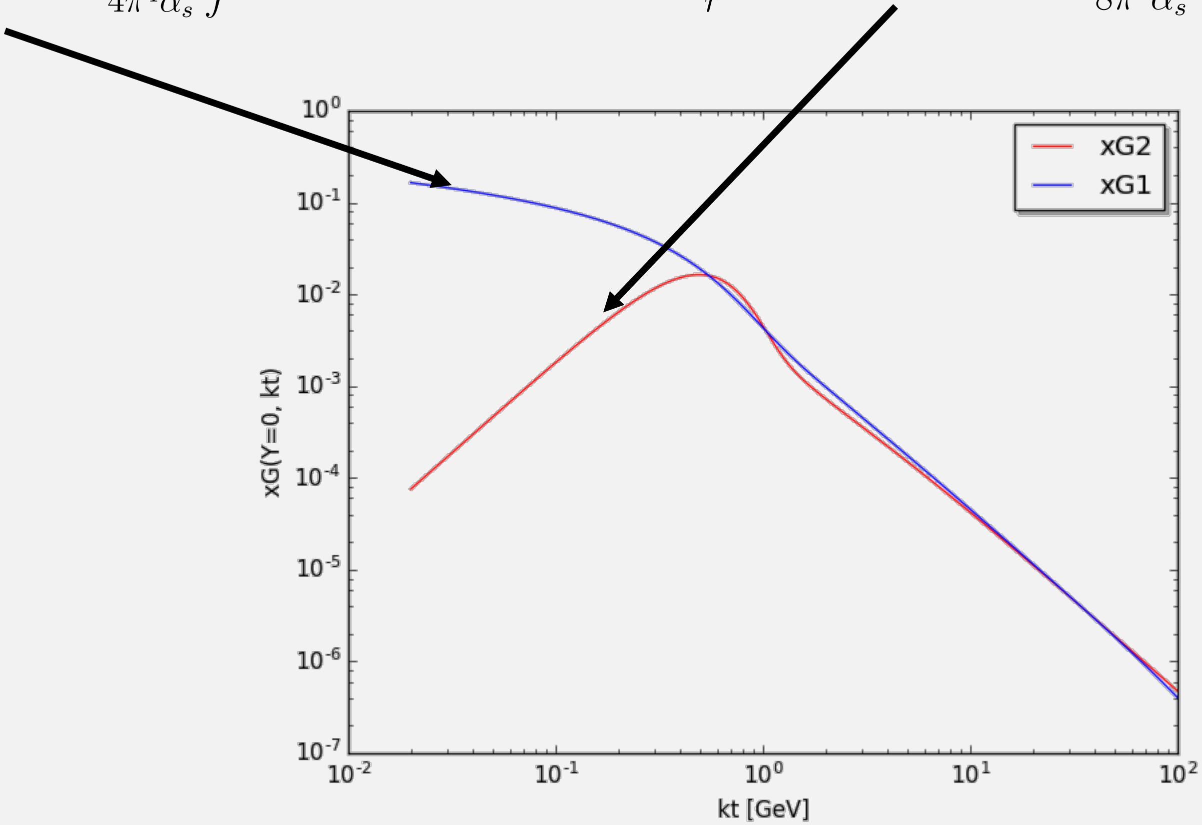
$$x_{tar}G^{(2)}(k_t, Y) = \frac{N_c k_t^2 S}{8\pi^4 \alpha_s} \int dr\, J_0(k_t r)(1 - \boxed{N(r, Y)})2\pi r$$

COMPUTING FT

If we do this:

$$x_{tar}G^{(1)}(k_t,Y) = \frac{N_c S}{4\pi^4\alpha_s}\int dr J_0(k_t r)(1-(1-N(r,Y))^2)\frac{2\pi}{r}$$

$$x_{tar}G^{(2)}(k_t,Y) = \frac{N_c k_t^2 S}{8\pi^4\alpha_s}\int dr J_0(k_t r)(1-N(r,Y))2\pi r$$



CONCLUSIONS

- Fourier transform is an irreplaceable tool in particle physics.

- Computing it numerically can prove to be a problematic task if we are dealing with an oscillatory integrand with many oscillations.

- Both precision of the integrand and of the integration method is the key to success.

- This computation can be rather CPU-time demanding. It is viable to use clusters if you need high precision.

THANK YOU FOR YOUR ATTENTION

No matter what, don't lose hope. We are all bombastic.

- Dan Nekonečný