MC generator KaTie[1] for modeling of hard processes at the NICA

A. Chernyshev[†,2], and V. Saleev[†,⋆]

[†]Samara University
[⋆]Joint Institute for Nuclear Research

SPD Physics&MC Meeting N41
19 June, 2024

[1]A. Van Hameren, «KaTie: For parton–level event generation with k_T–dependent initial states», Comput.Phys.Commun 224 (2018);

[2]Email: aachernyshoff@gmail.com

Outline

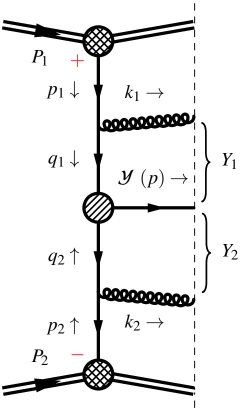
Introduction

Gluon probes at NICA SPD:

- ***Different charmonia states production:*** $\eta_c[1S]$, $\psi[1S]$ (J/ψ), $\psi[2S]$.
  - Description of hadronization of $c\bar{c}$ pair is based on *phenomenological models: CSM, NRQCD, (I)CEM*;
  - Event generators:
    - `Pythia 6.,8.` ⇠ *parton showers;*
    - `MadGraph5_aMC@NLO`[Alwall et.al. '14] ⇠ *parton level* + *matching with parton showers;*
    - ...
- ***Open charm production:*** $D^0/\bar{D}^0$.
  - Usually description of hadronization of $c \to D^0/\bar{D}^0$ is based on *fragmentation mechanism*;
  - Calculations can be included in any pQCD event generator.
- ***Prompt photons:***
  - Fully perturbative process at parton level;
  - Event generators:
    - `Pythia 6.,8.` ⇠ *parton showers;*
    - `Sherpa`[Gleisberg et.al. '09] ⇠ *parton showers*;
    - `Jetphox`[Catani et.al. '02] ⇠ *parton level*;
    - ...

- All of this generators use the ***collinear factorization approximation*** $\mu_F \sim p_T \gg \Lambda$.
- At the NICA kinematical range ***we plan to study TMD PDF's***.

Collinear parton model[3]



Initial state momenta:

$$q_{1,2} = \left(q^{\pm}_{1,2}/2\right) n_{\mp} + q_{T_{1,2}} \Longrightarrow q^2_{1,2} = -\mathbf{q}^2_{T_{1,2}}, \quad q^{\pm}_{1,2} \gg q^{\mp}_{1,2}.$$

Collinear factorization: $|\mathbf{q}_{T_i}| \ll \mu_F$:

$$d\sigma_{\rm CPM} = \left[f(x_1,\mu_F^2)\times f(x_2,\mu_F^2)\right]\otimes d\hat{\sigma}_{\rm CPM}(x_i,\mu_F,\mu_R) + \mathscr{O}\left(\frac{\Lambda^\#}{\mu_F^\#}\right),$$

where $f(x_i,\mu_F^2)$ is integrated over $|\mathbf{q}_{T_i}|$ PDF's satisfying ***DGLAP*** eq.

Advantages:

- Good for description of single–scale processes like Drell–Yan;
- There are calculation in LO, NLO, NNLO, . . .

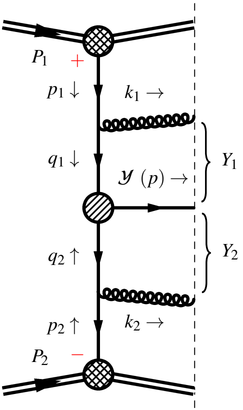
Disadvantages:

- Only applicable in high $|\mathbf{p}_T| \gg \mu_F$. In case of $\psi[1S]$ production:

$$|\mathbf{p}_T| \gg \mu_F \sim M_{T_\psi} \sim M_\psi \sim 3 \text{ GeV}.$$

[3] We use Sudakov decomposition: $p = \left(p^+ n_- + p^- n_+\right)/2 + p_T$, where light cone variables $n^\pm = (1,\mathbf{0},\mp 1)$, so that $p^\pm = (p,n^\pm)$.

TMD parton model



Initial state momenta:

$$q_{1,2} = \left(q^{\pm}_{1,2}/2\right) n_{\mp} + q_{T_{1,2}} \Longrightarrow q^2_{1,2} = -\mathbf{q}^2_{T_{1,2}}.$$

Transverse Momentum Dependent (TMD):[Collins '11] $|\mathbf{q}_{T_i}| \ll \mu_F$

$$d\sigma_{\text{TMD}} = \left[F(x_1, \mathbf{q}_{T_1}, \mu_F^2, \mu_Y^2) \times F(x_2, \mathbf{q}_{T_2}, \mu_F^2, \mu_Y^2)\right] \delta^{(2)}\left(\mathbf{q}_{T_1} + \mathbf{q}_{T_2} - \mathbf{p}_T\right)$$
$$\otimes d\hat{\sigma}_{\text{CPM}} + \mathcal{O}\left(\Lambda^{\#}/\mu_F^{\#}, \mathbf{p}_T^2/\mu_F^2\right),$$

where $F(x_i, \mathbf{q}_{T_i}, \mu^2, \mu_Y^2)$ is *TMD PDF's* satisfying ***Collins–Soper eq.***

Advantages:

- TMD PDF's include effects enhanced by

$$\ln\frac{\mu_F^2}{\Lambda^2}, \quad \ln\frac{\mu_F^2}{\mathbf{p}_T^2}, \quad \ln^2\frac{\mu_F^2}{\mathbf{p}_T^2};$$

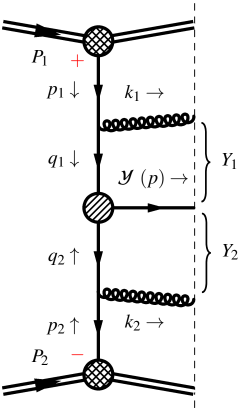
- Describe data at low $|\mathbf{p}_T| \ll \mu_F$;

Disadvantages:

- Only applicable in low $|\mathbf{p}_T| \ll \mu_F$. In case of $\psi[1S]$ production:

$$|\mathbf{p}_T| \ll \mu_F \sim M_{T_\psi} \sim M_\psi \sim 3 \text{ GeV}.$$

High energy factorization ($\neq$ TMD)



Initial state momenta:

$$q_{1,2} = \left(q^{\pm}_{1,2}/2\right) n_{\mp} + q_{T_{1,2}} \Longrightarrow q^2_{1,2} = -\mathbf{q}^2_{T_{1,2}}, \quad q^{\pm}_{1,2} \gg q^{\mp}_{1,2}.$$

High energy factorization a.k.a k_T–factorization: [Gribov et.al.'83; Catani et.al.'91]
$|\mathbf{q}_{T_i}| \sim \mu_F$ and $Y_i \gg 1$

$$d\sigma_{\rm HEF} = \left[\Phi(x_1, \mathbf{q}_{T_1}, \mu^2) \times \Phi(x_2, \mathbf{q}_{T_2}, \mu^2)\right] \otimes d\hat{\sigma}_{\rm HEF} + \mathcal{O}\left(\frac{\Lambda^\#}{\mu_F^\#}, \frac{\mu_F^2}{s}\right),$$

where $\Phi(x_i, \mathbf{q}_{T_i}, \mu^2)$ is *unintegrated PDF's (uPDF's).*

Advantages:

- Reggeized amplitudes are gauge–invariant;
- uPDF's include DGLAP evolution and small x effects:

$$\ln \frac{\mu_F^2}{\Lambda^2}, \quad \ln^2 \frac{\mathbf{q}_T^2}{\mu_F^2}, \quad \ln \frac{1}{x}$$

- Describe region between TMD $|\mathbf{p}_T| \ll \mu_F$ and CPM $|\mathbf{p}_T| \gg \mu_F$;

Disadvantage (at NICA energies):

- The main effects relate to the «small» x, at NICA: $x \sim 10^{-2} - 1$.

uPDF's

The uPDF's must include DGLAP evolution and small x effects:

PRA = Reggeized amplitudes + mKMRW uPDF's

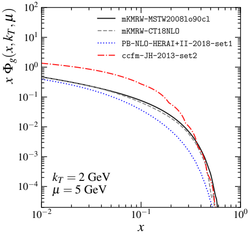
- We use uPDF's calculated in **modified Kimber–Martin–Ryskin–Watt model**[Nefedov, Saleev '20; KMR '01; MRW '03]:
  - `mKMRW-MSTW2008lo90cl` ←-- LO collinear input;
  - `mKMRW-CT18NLO` ←-- NLO collinear input;
  - Normalization condition holds exactly:

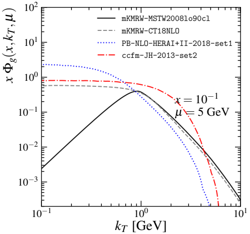
$$\int^{\mu^2} d\mathbf{q}_T^2 \, \Phi(x, \mathbf{q}_T, \mu^2) = x f(x, \mu^2), \quad \forall x, |\mathbf{q}_T|$$

  - In the region $|\mathbf{p}_T| \ll \mu_F$:

$$\Phi(x, \mathbf{q}_T, \mu^2) \simeq F(x, \mathbf{q}_T, \mu_F^2, \mu_Y^2 = \mu_F^2) \to \mathbf{PRA} \simeq \mathbf{TMD} + \mathcal{O}\left(\frac{\mathbf{p}_T^2}{\mu_F^2}\right)$$

- A large number (~ 30) of different uPDF's are collected in `TMDlib 2.x`[Jung et.al. '21]:
  - `PB-NLO-HERAI+II-2018-set1` ←-- Particle Branching method;
  - `ccfm-JH-2013-set2` ←-- Monte–Carlo CCFM equation solution.





KaTie

The main aspects of KaTie: **(see manual for details)**

- **Parton level** event generator;
- *Fully numerical method for calculating gauge invariant amplitudes with off–shell initial states* based on spinor amplitudes formalism and recurrence relations of the Britto–Cachazo–Feng–Witten (BCFW) type;
- Order of diagrams up to:

$$\mathcal{O}(e^n g^m), \qquad n+m \leq 4;$$

- **Tree–level CPM** calculations with collinear PDF sets from LHAPDF;
- **Tree–level HEF** calculations with uPDF's from:
  - TMDlib and from user grid files;
  - **user grids with format**:

$$\ln x \quad \ln|\mathbf{q}_T|^2 \quad x\Phi(x,|\mathbf{q}_T|) \qquad \text{or} \qquad \ln x \quad \ln|\mathbf{q}_T|^2 \quad \ln\mu^2 \quad x\Phi(x,|\mathbf{q}_T|,\mu);$$

  - At $|\mathbf{p}_T| \ll \mu_F$ KaTie may be used for TMD calculations with TMD PDF's;
- Output files in custom format and in the LHEF --→ *connection with multipurpose generatros like* Pythia;

and many more. . .

KaTie installation

Download repositories from

https://bitbucket.org/hameren/katie/downloads

https://bitbucket.org/hameren/avhlib/downloads

and unzip files. Edit `settings.py` script inside the KaTie directory:

```
# Path to the AVHLIB directory
AVHLIBpath = '/path/to/AVHLIB'
# Path to the directory where libLHAPDF.so is
LHAPDFpath = '/path/to/libLHAPDF.so'
# Fortran compiler with flags
FC = 'gfortran -fcheck=bounds'
```

If you want to use TMDlib:

```
# Path to the directory where TMDlib-config is (for tmdlib-2.0.x or
# older you can still put the path where libTMDlib.so is)
TMDLIBpath = '/path/to/libTMDlib.so'
```

Then, inside the KaTie–directory execute:

```
$ ./config.py lib
```

If there are no errors, then the library is built.

KaTie installation

Download repositories from

```
https://bitbucket.org/hameren/katie/downloads
https://bitbucket.org/hameren/avhlib/downloads
```

and unzip files. Edit `settings.py` script inside the KaTie directory:

```
# Path to the AVHLIB directory
AVHLIBpath = '/path/to/AVHLIB'
# Path to the directory where libLHAPDF.so is
LHAPDFpath = '/path/to/libLHAPDF.so'
# Fortran compiler with flags
FC = 'gfortran -fcheck=bounds'
```

If you want to use TMDlib:

```
# Path to the directory where TMDlib-config is (for tmdlib-2.0.x or
# older you can still put the path where libTMDlib.so is)
TMDLIBpath = '/path/to/libTMDlib.so'
```

Then, inside the KaTie–directory execute:

```
$ ./config.py lib
```

If there are no errors, then the library is built.

KaTie installation

Download repositories from

https://bitbucket.org/hameren/katie/downloads

https://bitbucket.org/hameren/avhlib/downloads

and unzip files. Edit `settings.py` script inside the KaTie directory:

```
# Path to the AVHLIB directory
AVHLIBpath = '/path/to/AVHLIB'
# Path to the directory where libLHAPDF.so is
LHAPDFpath = '/path/to/libLHAPDF.so'
# Fortran compiler with flags
FC = 'gfortran -fcheck=bounds'
```

If you want to use TMDlib:

```
# Path to the directory where TMDlib-config is (for tmdlib-2.0.x or
# older you can still put the path where libTMDlib.so is)
TMDLIBpath = '/path/to/libTMDlib.so'
```

Then, inside the `KaTie`–directory execute:

```
$ ./config.py lib
```

If there are no errors, then the library is built.

KaTie working principle

input $\xrightarrow{\texttt{run.sh}}$

directory
- proc001
  (for each process)
- proc002
- ⋮
- create_eventfile.f90
- create_eventfile.sh
- extra_cuts.h90
- extra_weights.h90
- input
- optimize.sh $\xrightarrow{\texttt{optimize.sh}}$ main.out → raw123.dat (Custom format)
- recompile.sh

↓ run.sh

eventfile.dat (LHEF)

Input file: processes

The user must explicitly list all desired parton–level processes:

```
Nfinst = 2
process = g g  -> c c~    factor = 1
process = q q~ -> c c~    factor = 1
Nflavors = 4
partlumi = combined
```

and set the number of non–QCD vertices:

```
pNonQCD = 0 0 0
          EW Hg HA
```

It is relevant, f.e., for process $q\bar{q} \to \mu\bar{\mu}jj$.

Interactions can be switched on/off with

```
switch = withQCD yes
switch = withQED no
switch = withWeak no
switch = withHiggs no
switch = withHG no
switch = withHA no
```

The list of all possible particles are as follows:

ve	ve~	e-	e+	u	u~	d
vmu	vmu~	mu-	mu+	c	c~	s
vtau	vtau~	tau-	tau+	t	t~	b
g	H	A	Z	W-	W+	

PDF sums:

q q~: $(u_1\bar{u}_2 + d_1\bar{d}_2 + \ldots) + (1 \leftrightarrow 2)$

Input file: processes

The user must explicitly list all desired parton–level processes:

```
Nfinst = 2
process = g g  -> c c~    factor = 1
process = q q~ -> c c~    factor = 1
Nflavors = 4
partlumi = combined
```

and set the number of non–QCD vertices:

```
pNonQCD = 0 0 0
```

EW Hg HA

It is relevant, f.e., for process $q\bar{q} \to \mu\bar{\mu} jj$.

Interactions can be switched on/off with

```
switch = withQCD yes
switch = withQED no
switch = withWeak no
switch = withHiggs no
switch = withHG no
switch = withHA no
```

The list of all possible particles are as follows:

ve	ve~	e-	e+	u	u~	d
vmu	vmu~	mu-	mu+	c	c~	s
vtau	vtau~	tau-	tau+	t	t~	b
g	H	A	Z	W-	W+	

PDF sums:

q q~: $(u_1\bar{u}_2 + d_1\bar{d}_2 + \ldots) + (1\leftrightarrow 2)$

Input file: PDF's

The user must set the PDF set from LHAPDF (always necessary for α_S calculation)

```
lhaSet = MSTW2008lo90cl
```

and indicate the type of initial states (on/off mass–shell)

```
offshell = 1 1          1 1:  g* g* -> ...
                        1 0:  g* g  -> ...   or   0 1:  g  g* -> ...
                        0 0:  g  g  -> ...
```

The user can use uPDF/TMD PDF set from TMDlib by specifying the set id

```
TMDlibSet = 102100
```

From TMDlib v.2.0.0

Alternatively, the user can provide TMD PDF's as grids directly:

```
tmdTableDir = /path/to/PDFs/
```

Actual grid file must be indicated for each parton separately:

```
tmdpf = g  gluon.dat
tmdpf = u  u.dat
tmdpf = u~ uBar.dat
⋮
```

Input file: PDF's

The user must set the PDF set from LHAPDF (always necessary for α_S calculation)

```
lhaSet = MSTW2008lo90cl
```

and indicate the type of initial states (on/off mass–shell)

```
offshell = 1 1          1 1:  g* g* -> ...
                        1 0:  g* g  -> ...   or   0 1:  g  g* -> ...
                        0 0:  g  g  -> ...
```

The user can use uPDF/TMD PDF set from TMDlib by specifying the set id

```
TMDlibSet = 102100      From TMDlib v.2.0.0
```

Alternatively, the user can provide TMD PDF's as grids directly:

```
tmdTableDir = /path/to/PDFs/
```

Actual grid file must be indicated for each parton separately:

```
tmdpf = g   gluon.dat
tmdpf = u   u.dat
tmdpf = u~  uBar.dat
⋮
```

Input file: PDF's

The user must set the PDF set from LHAPDF (always necessary for α_S calculation)

```
lhaSet = MSTW2008lo90cl
```

and indicate the type of initial states (on/off mass–shell)

```
offshell = 1 1          1 1:  g* g* -> ...
                        1 0:  g* g  -> ...   or   0 1:  g  g* -> ...
                        0 0:  g  g  -> ...
```

The user can use uPDF/TMD PDF set from TMDlib by specifying the set id

```
TMDlibSet = 102100
```

From TMDlib v.2.0.0

Alternatively, the user can provide TMD PDF's as grids directly:

```
tmdTableDir = /path/to/PDFs/
```

Actual grid file must be indicated for each parton separately:

```
tmdpf = g  gluon.dat
tmdpf = u  u.dat
tmdpf = u~ uBar.dat
⋮
```

Input file: PDF's

The user must set the PDF set from LHAPDF (always necessary for α_S calculation)

```
lhaSet = MSTW2008lo90cl
```

and indicate the type of initial states (on/off mass–shell)

```
offshell = 1 1          1 1:  g* g* -> ...
                        1 0:  g* g  -> ...   or   0 1:  g  g* -> ...
                        0 0:  g  g  -> ...
```

The user can use uPDF/TMD PDF set from TMDlib by specifying the set id

```
TMDlibSet = 102100      From TMDlib v.2.0.0
```

Alternatively, the user can provide TMD PDF's as grids directly:

```
tmdTableDir = /path/to/PDFs/
```

Actual grid file must be indicated for each parton separately:

```
tmdpf = g  gluon.dat
tmdpf = u  u.dat
tmdpf = u~ uBar.dat
⋮
```

Input file: more than one TMD set

For the applications of nuclear scattering studies, the user can set two different TMD sets via

```
TMDlibSet A = 400002
TMDlibSet B = 102200
```

The symbol `A` refers to the positive–rapidity initial states, and the symbol `B` to the negative one, e.g. `B + A -> 1 2 3 4`

If the user provides their own grids

```
tmdTableDir = /path/to/PDFsA/
tmdpf A = g  gluon.dat
tmdpf A = u  uQuark.dat
tmdpf A = u~ uBar.dat
⋮

tmdTableDir = /path/to/PDFsB/
tmdpf B = g  gluon.dat
tmdpf B = u  uQuark.dat
tmdpf B = u~ uBar.dat
⋮
```

`Input file`: more than one TMD set

For the applications of nuclear scattering studies, the user can set two different TMD sets via

```
TMDlibSet A = 400002
TMDlibSet B = 102200
```

The symbol `A` refers to the positive–rapidity initial states, and the symbol `B` to the negative one, e.g. `B + A -> 1 2 3 4`

If the user provides their own grids

```
tmdTableDir = /path/to/PDFsA/
tmdpf A = g  gluon.dat
tmdpf A = u  uQuark.dat
tmdpf A = u~ uBar.dat
⋮

tmdTableDir = /path/to/PDFsB/
tmdpf B = g  gluon.dat
tmdpf B = u  uQuark.dat
tmdpf B = u~ uBar.dat
⋮
```

`Input file`: kinematics

The center–of–mass energy $\sqrt{s}$ in GeV:

`Ecm = 27.0` The user can also set beam energies separetly with `EbeamA` and `EbeamB`

Typical value of softest scale in GeV

`Esoft = 3`

Number of nonzero-weight phase space points to be spent on optimization

`Noptim = 100,000`

Kinematical cuts can be set with

```
cut = {pT|1|} > 3.0
cut = {rapidity|1|} <  3.0
cut = {rapidity|1|} > -3.0

cut = {pT|2|} > 3.0
cut = {rapidity|2|} <  3.0
cut = {rapidity|2|} > -3.0

cut = {deltaR|1,2|} > 0.4
```

Cone distance $\Delta R = \sqrt{\Delta\phi^2 + \Delta y^2}$

`Input file`: kinematics

The center–of–mass energy $\sqrt{s}$ in GeV:

`Ecm = 27.0` The user can also set beam energies separetly with `EbeamA` and `EbeamB`

Typical value of softest scale in GeV

`Esoft = 3`

Number of nonzero-weight phase space points to be spent on optimization

`Noptim = 100,000`

Kinematical cuts can be set with

```
cut = {pT|1|} > 3.0
cut = {rapidity|1|} <  3.0
cut = {rapidity|1|} > -3.0

cut = {pT|2|} > 3.0
cut = {rapidity|2|} <  3.0
cut = {rapidity|2|} > -3.0

cut = {deltaR|1,2|} > 0.4
```

Cone distance $\Delta R = \sqrt{\Delta\phi^2 + \Delta y^2}$

Input file: other possible kinematical cuts

Other possible variables:

{pseudoRap|i|}	Pseudo rapidity
{ET|i|}	Transverse energy
{theta|i|}	Polar angle
{deltaPhi|i,j|}	Angle between i and j

Some variables can take arguments that consists of sums:

{mass|i+j+...|}, {pT|i+j+...|}, {rapidity|i+j+...|}, ...

The user can set complicated cuts:

cut = {pT|i|j,k,...} > 10.0 set the minimum pT of i–th final state in the pT–ordered list of final–states j,k,...

cut = {pT|i|rapidity|j,k,...} > 10.0 set the minimum pT of i–th final state in the rapidity–ordered list of final–states j,k,...

For even more complicated cuts, the user can provide blocks of FORTRAN pseudo source code like

```
cut source = if (ABS({rapidity|1|}).gt.3.D0) REJECT
cut source = if ({rapidity|1|}.gt.{rapidity|2|}) then
cut source = if ({pT|2|1,2,3|}.lt.30.D0) REJECT
cut source = endif
```

For blocks of FORTRAN source code the user can change extra_cuts.h90 file.

Input file: other possible kinematical cuts

Other possible variables:

{pseudoRap|i|}	Pseudo rapidity
{ET|i|}	Transverse energy
{theta|i|}	Polar angle
{deltaPhi|i,j|}	Angle between i and j

Some variables can take arguments that consists of sums:

{mass|i+j+...|}, {pT|i+j+...|}, {rapidity|i+j+...|}, ...

The user can set complicated cuts:

cut = {pT|i|j,k,...} > 10.0 set the minimum pT of i–th final state in the pT–ordered list of final–states j,k,...

cut = {pT|i|rapidity|j,k,...} > 10.0 set the minimum pT of i–th final state in the rapidity–ordered list of final–states j,k,...

For even more complicated cuts, the user can provide blocks of FORTRAN pseudo source code like

```
cut source = if (ABS({rapidity|1|}).gt.3.D0) REJECT
cut source = if ({rapidity|1|}.gt.{rapidity|2|}) then
cut source = if ({pT|2|1,2,3|}.lt.30.D0) REJECT
cut source = endif
```

For blocks of FORTRAN source code the user can change extra_cuts.h90 file.

Input file: other possible kinematical cuts

Other possible variables:

`{pseudoRap|i|}`	Pseudo rapidity
`{ET|i|}`	Transverse energy
`{theta|i|}`	Polar angle
`{deltaPhi|i,j|}`	Angle between `i` and `j`

Some variables can take arguments that consists of sums:
`{mass|i+j+...|}`, `{pT|i+j+...|}`, `{rapidity|i+j+...|}`, ...

The user can set complicated cuts:
`cut = {pT|i|j,k,...} > 10.0` set the minimum `pT` of `i`–th final state in the `pT`–ordered list of final–states `j,k,...`
`cut = {pT|i|rapidity|j,k,...} > 10.0` set the minimum `pT` of `i`–th final state in the `rapidity`–ordered list of final–states `j,k,...`

For even more complicated cuts, the user can provide blocks of `FORTRAN` pseudo source code like

```
cut source = if (ABS({rapidity|1|}).gt.3.D0) REJECT
cut source = if ({rapidity|1|}.gt.{rapidity|2|}) then
cut source = if ({pT|2|1,2,3|}.lt.30.D0) REJECT
cut source = endif
```

For blocks of `FORTRAN` source code the user can change `extra_cuts.h90` file.

`Input file`: other possible kinematical cuts

Other possible variables:

`{pseudoRap|i|}`	Pseudo rapidity
`{ET|i|}`	Transverse energy
`{theta|i|}`	Polar angle
`{deltaPhi|i,j|}`	Angle between `i` and `j`

Some variables can take arguments that consists of sums:

`{mass|i+j+...|}`, `{pT|i+j+...|}`, `{rapidity|i+j+...|}`, ...

The user can set complicated cuts:

`cut = {pT|i|j,k,...} > 10.0` set the minimum `pT` of `i`–th final state in the `pT`–ordered list of final–states `j,k,...`

`cut = {pT|i|rapidity|j,k,...} > 10.0` set the minimum `pT` of `i`–th final state in the `rapidity`–ordered list of final–states `j,k,...`

For even more complicated cuts, the user can provide blocks of `FORTRAN` pseudo source code like

```
cut source = if (ABS({rapidity|1|}).gt.3.D0) REJECT
cut source = if ({rapidity|1|}.gt.{rapidity|2|}) then
cut source = if ({pT|2|1,2,3|}.lt.30.D0) REJECT
cut source = endif
```

For blocks of `FORTRAN` source code the user can change `extra_cuts.h90` file.

Input file: scales

By default, it is assumed that the factorization scale and the renormalization scale are equal

```
scale = ({pT|1|})
```

If the user wants set a different renormalization scale, it can be set with

```
renormalization scale = ({ET|1|})
```

The user can also set different scales for different PDF's with

```
scaleA = {pT|1|}
scaleB = {ET|1|}
```

Input file: scales

By default, it is assumed that the factorization scale and the renormalization scale are equal

```
scale = ({pT|1|})
```

If the user wants set a different renormalization scale, it can be set with

```
renormalization scale = ({ET|1|})
```

The user can also set different scales for different PDF's with

```
scaleA = {pT|1|}
scaleB = {ET|1|}
```

Input file: scales

By default, it is assumed that the factorization scale and the renormalization scale are equal

```
scale = ({pT|1|})
```

If the user wants set a different renormalization scale, it can be set with

```
renormalization scale = ({ET|1|})
```

The user can also set different scales for different PDF's with

```
scaleA = {pT|1|}
scaleB = {ET|1|}
```

Input file: model parameters

Masses and widths:

```
mass = Z   91.1882   2.4952
mass = W   80.419   2.21
mass = H   125.0   0.00429
mass = c   1.31
mass = b   4.75
mass = t   173.5

coupling = alphaEW 0.00794
coupling = Gfermi 1.16639D-5
```

KaTie usage

After writing the input file, you need to create a calculation directory with:

```
$ ./run.sh prepare input directory
```

Complete Toolkit of /.run.sh script:

```
$ run.sh lib
$ run.sh prepare <filename> <dirname>
$ run.sh compile <sourcefile>
$ run.sh compile,run <sourcefile>
$ run.sh compile,run <sourcefile> <datafile>
$ run.sh merge raw1.dat raw2.dat raw3.dat
$ run.sh merge raw*
$ run.sh lhef raw1.dat raw2.dat raw3.dat
$ run.sh lhef raw*
$ run.sh help compile
$ run.sh katamp
```

KaTie usage

At this stage the user can edit

`extra_cuts.h90` and `extra_weights.h90`

and use `recompile.sh` to confirm changes.

Then phase space can be optimized

```
$ ./optimize.sh
```

Complete Toolkit of `./optimize.sh` script:

```
To run 4 optimization processes at a time:
$ ./optimize.sh Ncpu=4
Exactly the same is achieved with Nparallel=4
To run optimization process 3:
$ ./optimize.sh proc=3
To run optimization process 3 and 12:
$ ./optimize.sh proc=3,12
You should now understand the following:
$ ./optimize.sh proc=3,12,4,11,2 Ncpu=4
You can monitor the progress with
$ tail -f proc*/output
You can kill all processes with
$ pkill -f main
```

KaTie usage

After optimization stage, you can run your calculations:

```
$ ./main.out
```

Complete Toolkit of `./main.out` script:

```
Execute as, for example:
$ ./main.out seed=12345
or
$ nohup ./main.out seed=12345 > output12345 &
or
$ nohup ./main.out seed=12345 dir=R001/ > R001/output &
```

Upon completion of calculations, `raw` file with the following structure will be created:

```
# Information from input
EVENT WEIGHT: ...
1
E   px   py   pz   E**2-px**2-py**2-pz**2   color   anti-color   helicity
⋮
```

One line for each particle. Initial states have `E < 0`.

`matrix element` `parton luminosity` $(x_1 f_1 x_2 f_2)$ α_S μ_R

```
pdfB   xB   kTB   muB
pdfA   xA   kTA   muA
```

i. Fragmentation production

Fragmentation approach $i \to A$:

$$\frac{d\sigma_A}{dp_T\, dy} = \sum_i \mathcal{D}_{i/A}(z) \otimes \frac{d\sigma_i}{dq_T\, d\eta}$$

- $D^{0/+}$ **mesons production:** $gg, q\bar{q} \to c\ (\to D) + \bar{c}$.
  Non–perturbative fragmentation function (FF):

$$\mathcal{D}_{c/D}(z) = \mathcal{N}\frac{z(1-z)^2}{[(1-z)^2+\varepsilon z]^2}, \quad \varepsilon = 0.06,$$

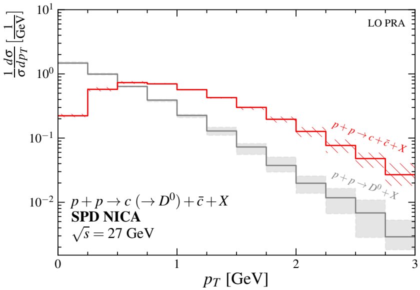
  where $z = (p^0 + |\mathbf{p}|)/(q^0 + |\mathbf{q}|)$.

- $\gamma^{\rm frg}$ **production (at LO):** $gg, q\bar{q} \to q'\ (\to \gamma^{\rm frg}) + \bar{q}'$.
  Perturbatively calculated FF:

$$\mathcal{D}_{q/\gamma}(z,\mu_F) = \mathcal{D}_{\bar{q}/\gamma}(z) = \frac{\alpha}{2\pi}\frac{1+(1-z)^2}{z}\ln\frac{\mu_F^2}{\Lambda^2},$$

  where $z = p^0/q^0$.

Predictions for D^0 production at the SPD NICA



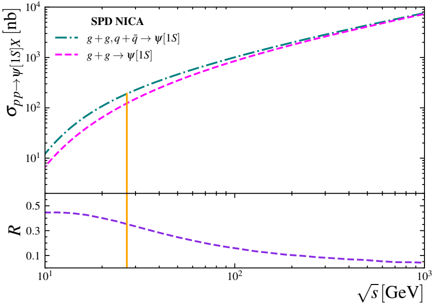
$$\frac{d\sigma_D}{dp_T^D} = \mathcal{D}(z) \otimes \frac{d\sigma_{c\bar{c}}}{dp_T^c}$$

ii. Charmonia production

Improved color evaporation model (ICEM)[Ma, Vogt '16]**:**

$$\frac{d\sigma_\psi}{d^3p} = \mathcal{F}^\psi \times \int_{M_\psi}^{2M_D} dM\, d^3\mathbf{p}'\; \delta^{(3)}\left(\mathbf{p} - \frac{M_\psi}{M}\mathbf{p}'\right)\; \frac{d\sigma_{c\bar{c}}}{dM\, d^3p'} + \mathcal{O}\left(\frac{\lambda^2}{m_q^2}\right)$$

Each $\mathcal{F}^\psi$ *factor for each* $\psi = \eta_c[1S], \psi[1S], \psi[2S], \ldots$



At NICA energies we obtained[A.C., Saleev '22]:

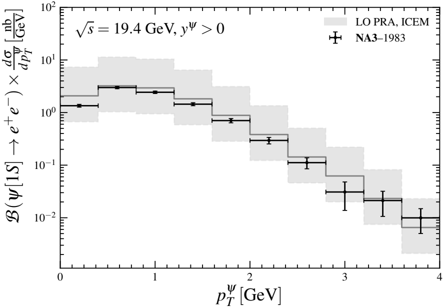
$$R = \frac{\sigma_{q\bar{q}\to\psi[1S]X}}{\sigma_{gg\to\psi[1S]X} + \sigma_{q\bar{q}\to\psi[1S]X}} \simeq 30\%$$

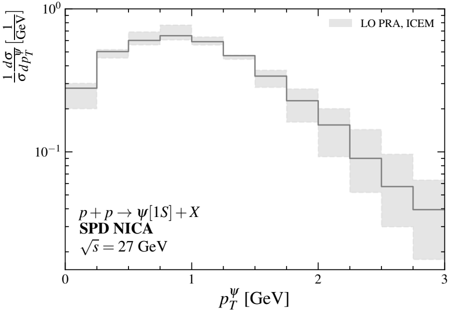
KaTie scheme:

```
cut source = {mass|1+2|} < 3.74D0
cut source = {mass|1+2|} > 3.10D0

cut source = if
((3.10D0/{mass|1+2|}*{pT|1+2|}).gt.4.D0)
REJECT
```

Predictions for $\psi[1S]$ production at the SPD NICA





NLO* CPM calculations with KaTie

LO CPM $2 \to 2$: processes of order $\mathcal{O}(\alpha_S^2)$ are finite:

$$g+g \to c+\bar{c},$$
$$q+\bar{q} \to c+\bar{c}.$$

NLO* CPM $2 \to 3$: first α_S real correction of order $\mathcal{O}(\alpha_S^3)$:

$$\left.\begin{array}{l} g+g \to c+\bar{c}+g\ (k'), \\ q+\bar{q} \to c+\bar{c}+g\ (k'), \\ g+q \to c+\bar{c}+q\ (k') \end{array}\right\} \quad \text{infrared diverge } |\mathbf{k}'_T| \to 0$$

Phenomenological cutoff at the lower limit and suppresion function:

$$\sigma_{ij\to c\bar{c}g}(\lambda) \sim \int_0^\infty d|\mathbf{k}'_T|\, F_{\text{sup}}(|\mathbf{k}'_T|;\lambda) \times \ldots, \qquad F_{\text{sup}}(|\mathbf{k}'_T|;\lambda) = \frac{|\mathbf{k}'_T|^4}{(|\mathbf{k}'_T|^2+\lambda^2)^2}$$

- ***Suitable for describing data on charmonia production***[Cheung, Vogt '21]***;***
- ***Also can be applied to*** D ***mesons production***[Maciula, Szczurek '19]**.**

DLSA

Double Longitudinal–Spin Asymmetry:

$$A_{LL} = \frac{d\sigma^{++} - d\sigma^{+-}}{d\sigma^{++} + d\sigma^{+-}} = \frac{d\Delta\sigma}{d\sigma},$$

$d\Delta\sigma$–polarized cross section:

$$d\Delta\sigma \simeq \sum_{i,\bar{j}} \left[\Delta f_i(x_1,\mu_F^2) \times \Delta f_j(x_2,\mu_F^2)\right] \otimes d\Delta\sigma_{ij}(x_1,x_2,\mu_F,\mu_R),$$

$d\sigma$–unpolarized cross section.

For each event we know:

`E px py pz E**2-px**2-py**2-pz**2 color anti-color helicity`

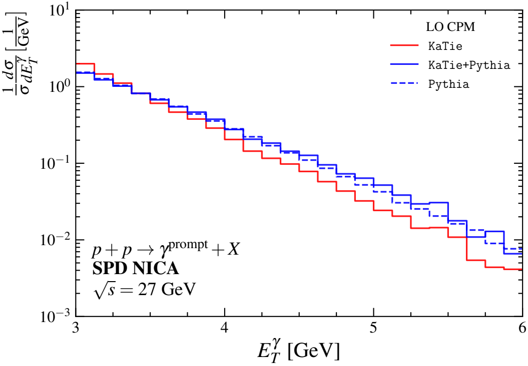
$\Longrightarrow$ *we can sample events with a fixed helicity value.*

LHAPDF polarized PDF sets:

- NNPDFpol10_100;
- NNPDFpol11_100.

KaTie with parton showers from Pythia 8

In collaboration with L. Alimov.

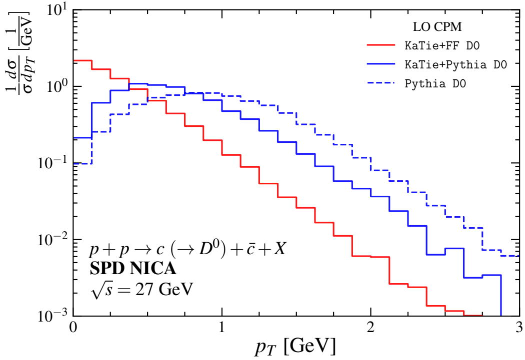


Pythia settings:

```
PartonLevel:ISR = on
PartonLevel:FSR = on
HadronLevel:Hadronize = on
HadronLevel:Decay = on

BeamRemnants:primordialKT = off
```

KaTie with parton showers from Pythia 8



Pythia settings:

```
PartonLevel:ISR = on
PartonLevel:FSR = on
HadronLevel:Hadronize = on
HadronLevel:Decay = on

BeamRemnants:primordialKT = off
```

Pair charmonia studies at SPD

ICEM also can be applied to pair charmonia production [A.C., Saleev '22–24]***:***

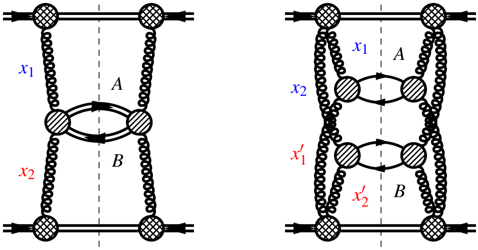
- Single parton scattering contribution (SPS):

$$\frac{d\hat{\sigma}^{\rm SPS}_{\psi\psi'}(x_1,x_2)}{d^3p_1\,d^3p_2} \simeq \mathcal{F}^{\psi\psi'} \times \int_{M_\psi}^{2M_H} dM_1 \int_{M_{\psi'}}^{2M_{H'}} dM_2\, \frac{d\hat{\sigma}^{\rm SPS}_{c\bar{c}c\bar{c}}(x_1,x_2)}{dM_1\,d^3p'_1\,dM_2\,d^3p'_2}$$

Following Pauli principle: $\mathcal{F}^{\psi\psi'} = \mathcal{F}^{\psi} \times \mathcal{F}^{\psi'}$ *only in case* $\psi \neq \psi'$.

- Double parton scattering contribution (DPS):

$$\frac{d\hat{\sigma}^{\rm DPS}_{\psi\psi'}(x_1,x_2,x'_1,x'_2)}{d^3p_1\,d^3p_2} \simeq \frac{\mathcal{F}^{\psi}\times\mathcal{F}^{\psi'}}{(1+\delta_{\psi\psi'})\,\sigma_{\rm eff}} \times \int_{M_\psi}^{2M_H} dM_1\, \frac{d\hat{\sigma}^{\rm SPS}_{c\bar{c}}(x_1,x'_1)}{dM_1\,d^3p'_1} \int_{M_{\psi'}}^{2M_{H'}} dM_2\, \frac{d\hat{\sigma}^{\rm SPS}_{c\bar{c}}(x_2,x'_2)}{dM_2\,d^3p'_2}$$



We can perform DPS calculations using `KaTie`.

Results: $\psi[1S] + \psi[1S]$

LO PRA + NRQCD $\quad\vee\quad$ NLO* CPM + CSM $\quad\vee\quad$ LO PRA + ICEM

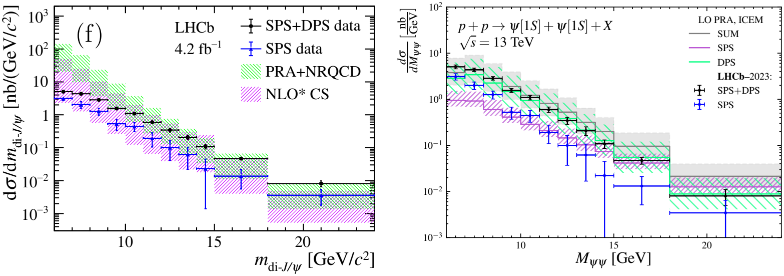
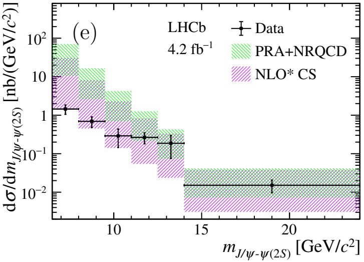


Figure 1: The left plot is from[Aaij et.al. '23].

Predictions in LO PRA+NRQCD and CSM+NLO* CPM are performed only taking into account the SPS contribution!

Results: $\psi[1S]+\psi[2S]$

$$\text{LO PRA}+\text{NRQCD} \quad \vee \quad \text{NLO}^\star\ \text{CPM}+\text{CSM} \quad \vee \quad \text{LO PRA}+\text{ICEM}$$



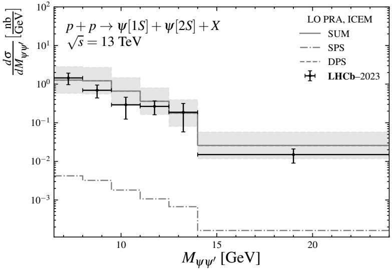


Figure 2: The left plot is from [Aaij et.al. '23].

$$\sigma^{\text{SPS+DPS}}_{\psi[1S]\psi[2S]}/\sigma^{\text{SPS+DPS}}_{\psi[1S]\psi[1S]} = 0.274\pm0.044\pm0.08$$

Predictions in LO PRA+NRQCD and CSM+NLO* CPM are performed only taking into account the SPS contribution!

Pair charmonia studies at SPD: motivation

A	N_{ev}	Exp. $\pm$ (stat.) $\pm$ (syst.) [pb]	ICEM [pb]	SPS [pb]	DPS [pb]
NA3 '85, pA, $\sqrt{s} = 27$ GeV					
Pt	15 ± 4	27.0 ± 10.0	$5.0^{+38.1}_{-4.4}$	$3.1^{+20.0}_{-2.6}$	$1.9^{+18.1}_{-1.8}$
COMPASS[4] '22, $\pi^- A$, $\sqrt{s} = 23$ GeV					
NH_3	25 ± 1	$10.7 \pm 2.3 \pm 3.2$	$1.3^{+3.8}_{-1.0}$	$0.9^{+2.3}_{-0.6}$	$0.3^{+1.5}_{-0.2}$
Al	1	$3.6 \pm 8.2 \pm 1.4$	$1.2^{+3.7}_{-0.8}$	$0.9^{+2.2}_{-0.6}$	$0.3^{+1.4}_{-0.2}$
W	5	$3.3 \pm 3.0 \pm 1.8$	$1.2^{+3.5}_{-0.8}$	$0.9^{+2.1}_{-0.6}$	$0.3^{+1.4}_{-0.2}$

- Data at low energies have not yet been described (MODELS/EXPERIMENT PROBLEMS?);
- ICEM predicts non–negligible DPS contribution (MPI STUDIES AT NICA?);
- Need more measurements with more statistics;
- Pair charmonia production processes are more dependent on the hadronization model than inclusive one (TEST ICEM/CSM/NRQCD);

Any chance to observe $\psi[1S] + \psi[1S, 2S]$***?***

[4] For details see Talk at the 3rd COMPASS «Analysis Phase» mini–workshop, 19 April '23 by V. Saleev.

Conclusions

- We have made a brief review of `KaTie` event generator;
- We have developed a scheme for calculating heavy quarkonia and D mesons production using `KaTie`;
- At the $|\mathbf{p}_T| \ll \mu$ KaTie may be used for calculations in the TMD factorization;
- For the intermediate region $|\mathbf{p}_T| \sim \mu$ we may use the PRA, which takes into account power corrections $\mathcal{O}\left(\mathbf{p}_T^2/\mu^2\right)$;
- *Preliminary:* `KaTie` may be applied for NLO* CPM calculations and polarizations studies;
- `KaTie` can be connected with `Pythia`;
- **`KaTie` can be a powerful tool for calculating hard processes even at NICA energies**.

`KaTie` can be found at `Bitbucket/hameren/katie`

The efficiency of `KaTie` for calculating different hard processes at high energies was demonstrated in[A. van. Hameren et.al. '18–23] and some of our works[A. Chernyshev and V. Saleev '22–24].

A. Chernyshev and V. Saleev would like to thank A. van Hameren for helpful discussions on `KaTie` *program and H. Jung for help in* `TMDlib 2.x` *installation.*

Thank you for your attention!