b-jet shape analysis using Monte-Carlo methods

Luke Pomfrey

Monday 9th March, 2009

Introduction : Project Aims and Objectives

- Investigate *b*-jet shapes using Monte-Carlo methods.
- Compare the output of several generators and tunes to actual CDF-II data.
- Attempt to tune a generator to better match the CDF data.

Introduction : Project Aims and Objectives

- Investigate b-jet shapes using Monte-Carlo methods.
- Compare the output of several generators and tunes to actual CDF-II data.
- Attempt to tune a generator to better match the CDF data.

Introduction : Project Aims and Objectives

- Investigate b-jet shapes using Monte-Carlo methods.
- Compare the output of several generators and tunes to actual CDF-II data.
- Attempt to tune a generator to better match the CDF data.

Introduction : A quick introduction to Monte-Carlo

Figure: Monte Carlo, Monaco, the source of the Monte Carlo methods' name.

- A way to model complex physical situations that contain a degree of improbability using statistical sampling.
- General procedure:
  1. Define a domain of possible inputs.
  2. Randomly generate inputs and perform a deterministic computation on them.
  3. Aggregate the results of the inputs into a final result.
- Free parameters $\longrightarrow$ Tuning.

Introduction : A quick introduction to Monte-Carlo

Figure: Monte Carlo, Monaco, the source of the Monte Carlo methods' name.

- A way to model complex physical situations that contain a degree of improbability using statistical sampling.
- General procedure:
  1. Define a domain of possible inputs.
  2. Randomly generate inputs and perform a deterministic computation on them.
  3. Aggregate the results of the inputs into a final result.
- Free parameters $\longrightarrow$ Tuning.

Introduction : A quick introduction to Monte-Carlo

Figure: Monte Carlo, Monaco, the source of the Monte Carlo methods' name.

- A way to model complex physical situations that contain a degree of improbability using statistical sampling.
- General procedure:
  1. Define a domain of possible inputs.
  2. Randomly generate inputs and perform a deterministic computation on them.
  3. Aggregate the results of the inputs into a final result.
- Free parameters → Tuning.

Introduction : A quick introduction to Monte-Carlo

Figure: Monte Carlo, Monaco, the source of the Monte Carlo methods' name.

- A way to model complex physical situations that contain a degree of improbability using statistical sampling.
- General procedure:
  1. Define a domain of possible inputs.
  2. Randomly generate inputs and perform a deterministic computation on them.
  3. Aggregate the results of the inputs into a final result.
- Free parameters $\longrightarrow$ Tuning.

Introduction : A quick introduction to Monte-Carlo

Figure: Monte Carlo, Monaco, the source of the Monte Carlo methods' name.

- A way to model complex physical situations that contain a degree of improbability using statistical sampling.
- General procedure:
  1. Define a domain of possible inputs.
  2. Randomly generate inputs and perform a deterministic computation on them.
  3. Aggregate the results of the inputs into a final result.
- Free parameters → Tuning.

Introduction : A quick introduction to Monte-Carlo

Figure: Monte Carlo, Monaco, the source of the Monte Carlo methods' name.

- A way to model complex physical situations that contain a degree of improbability using statistical sampling.
- General procedure:
  1. Define a domain of possible inputs.
  2. Randomly generate inputs and perform a deterministic computation on them.
  3. Aggregate the results of the inputs into a final result.
- Free parameters $\longrightarrow$ Tuning.

Introduction : *b*-jets

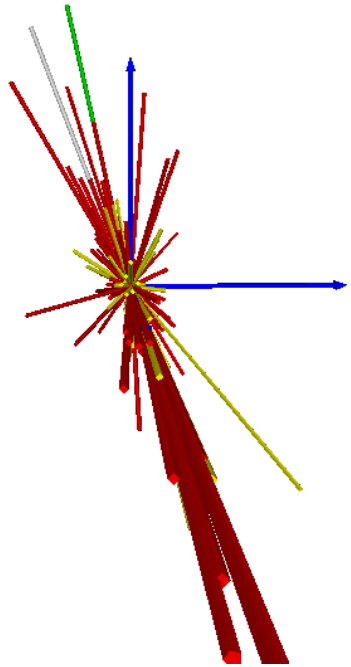


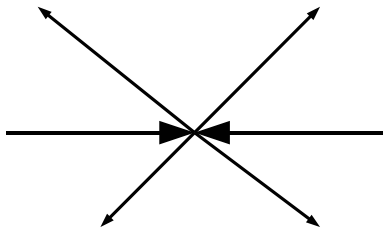
Figure: A *b*-jet event rendered using `hepmcview`.

Introduction : Detecting b-jets

- The displaced vertex method (simplified).

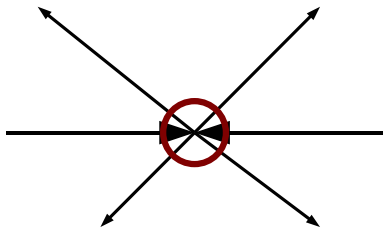
Introduction : Detecting b-jets

- The displaced vertex method (simplified).



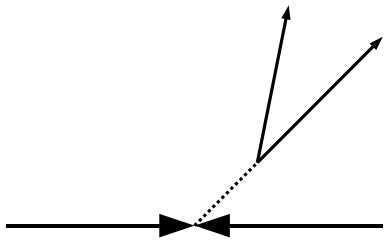
Introduction : Detecting b-jets

- The displaced vertex method (simplified).



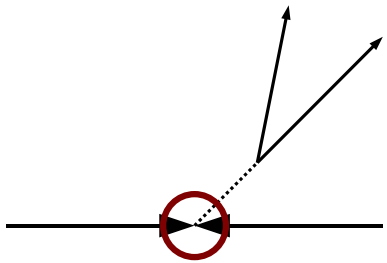
Introduction : Detecting b-jets

- The displaced vertex method (simplified).



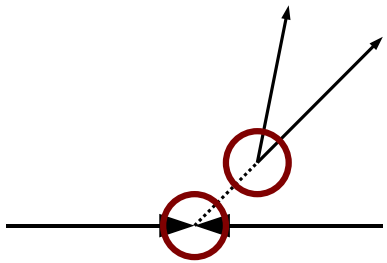
Introduction : Detecting *b*-jets

- The displaced vertex method (simplified).

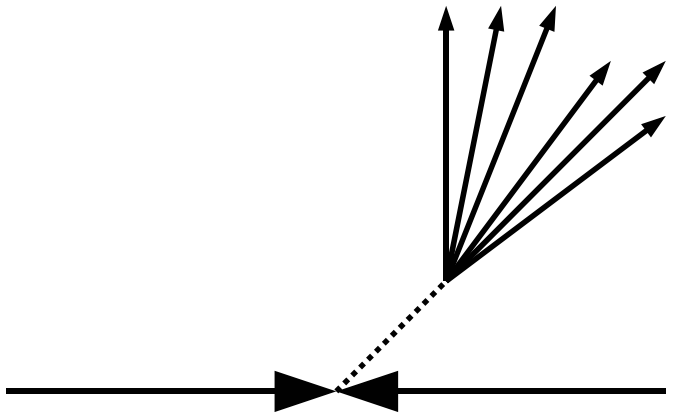


Introduction : Detecting b-jets

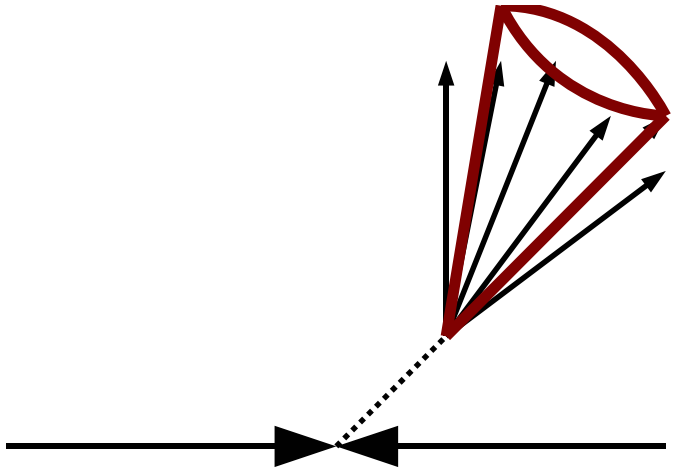
- The displaced vertex method (simplified).



Introduction : Jet shapes



Introduction : Jet shapes



Comparison of generators and tunes : The generators and tunes used

- Pythia6
  - Tune A
  - LEP Tune (with old and new shower methods)
  - S0-Pro Tune (with old and new shower methods)
- Herwig
- HerwigJimmy
- Does explicitly stating that only b-jet containing events be generated make a difference?

Comparison of generators and tunes : The generators and tunes used

- Pythia6
  - Tune A
  - LEP Tune (with old and new shower methods)
  - S0-Pro Tune (with old and new shower methods)
- Herwig
- HerwigJimmy
- Does explicitly stating that only b-jet containing events be generated make a difference?

Comparison of generators and tunes : The generators and tunes used

- Pythia6
  - Tune A
  - LEP Tune (with old and new shower methods)
  - S0-Pro Tune (with old and new shower methods)
- Herwig
- HerwigJimmy
- Does explicitly stating that only b-jet containing events be generated make a difference?

Comparison of generators and tunes : The generators and tunes used

- Pythia6
  - Tune A
  - LEP Tune (with old and new shower methods)
  - S0-Pro Tune (with old and new shower methods)
- Herwig
- HerwigJimmy
- Does explicitly stating that only b-jet containing events be generated make a difference?

Comparison of generators and tunes : The generators and tunes used

- Pythia6
  - Tune A
  - LEP Tune (with old and new shower methods)
  - S0-Pro Tune (with old and new shower methods)
- Herwig
- HerwigJimmy
- Does explicitly stating that only b-jet containing events be generated make a difference?

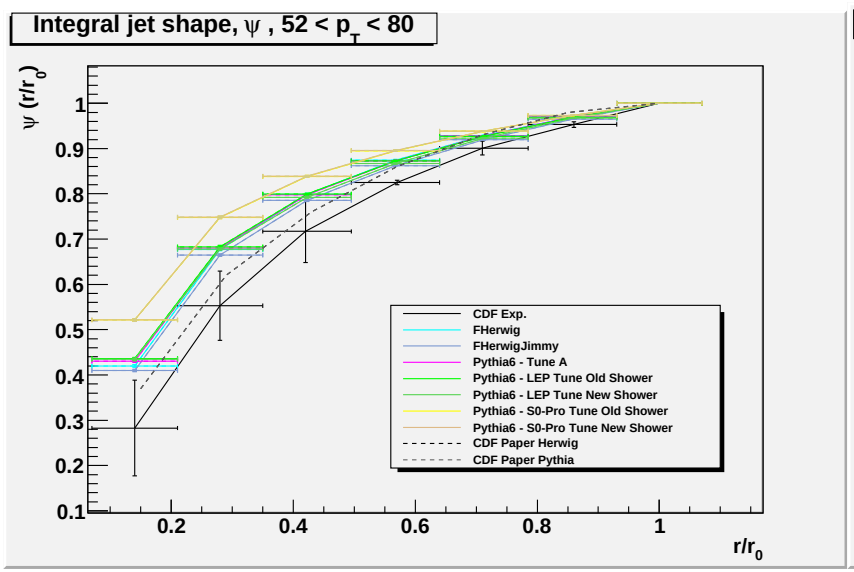
Comparison of generators and tunes : The generators and tunes used

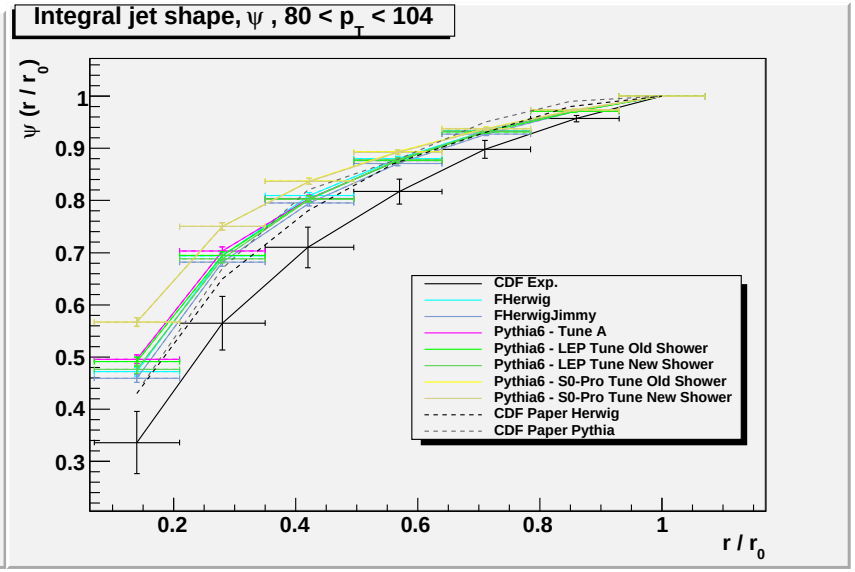
- Pythia6
  - Tune A
  - LEP Tune (with old and new shower methods)
  - S0-Pro Tune (with old and new shower methods)
- Herwig
- HerwigJimmy
- Does explicitly stating that only b-jet containing events be generated make a difference?

Comparison of generators and tunes : The generators and tunes used

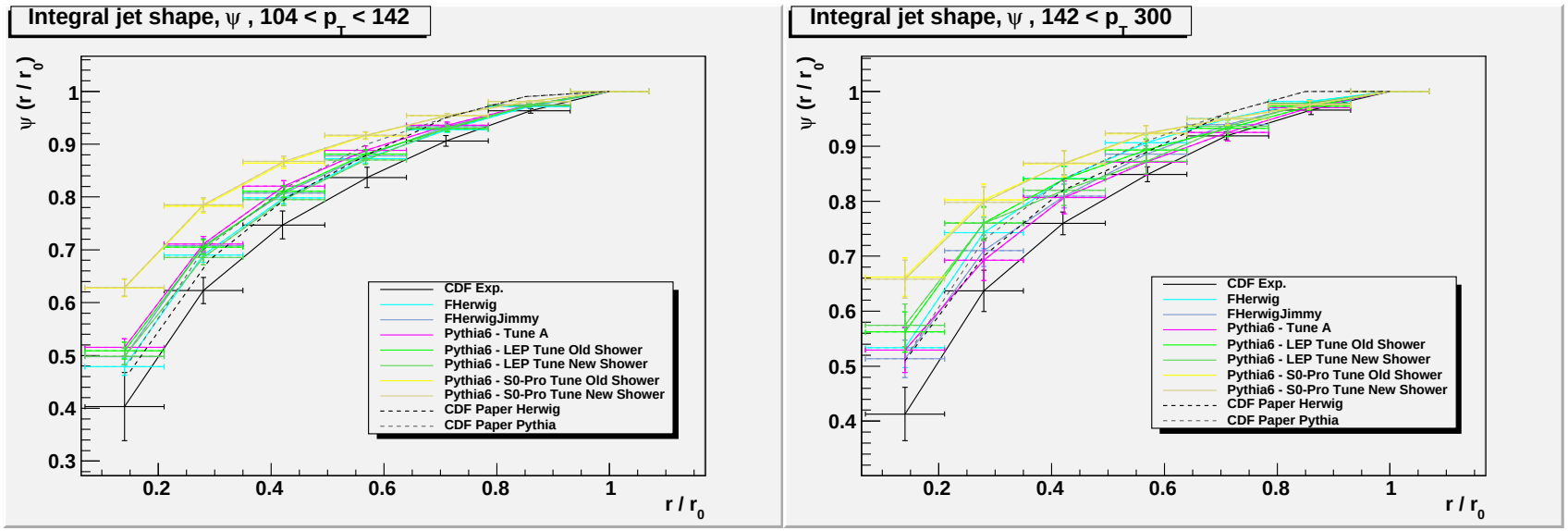
- Pythia6
  - Tune A
  - LEP Tune (with old and new shower methods)
  - S0-Pro Tune (with old and new shower methods)
- Herwig
- HerwigJimmy
- Does explicitly stating that only b-jet containing events be generated make a difference?

Comparison of generators and tunes : The generator comparison results

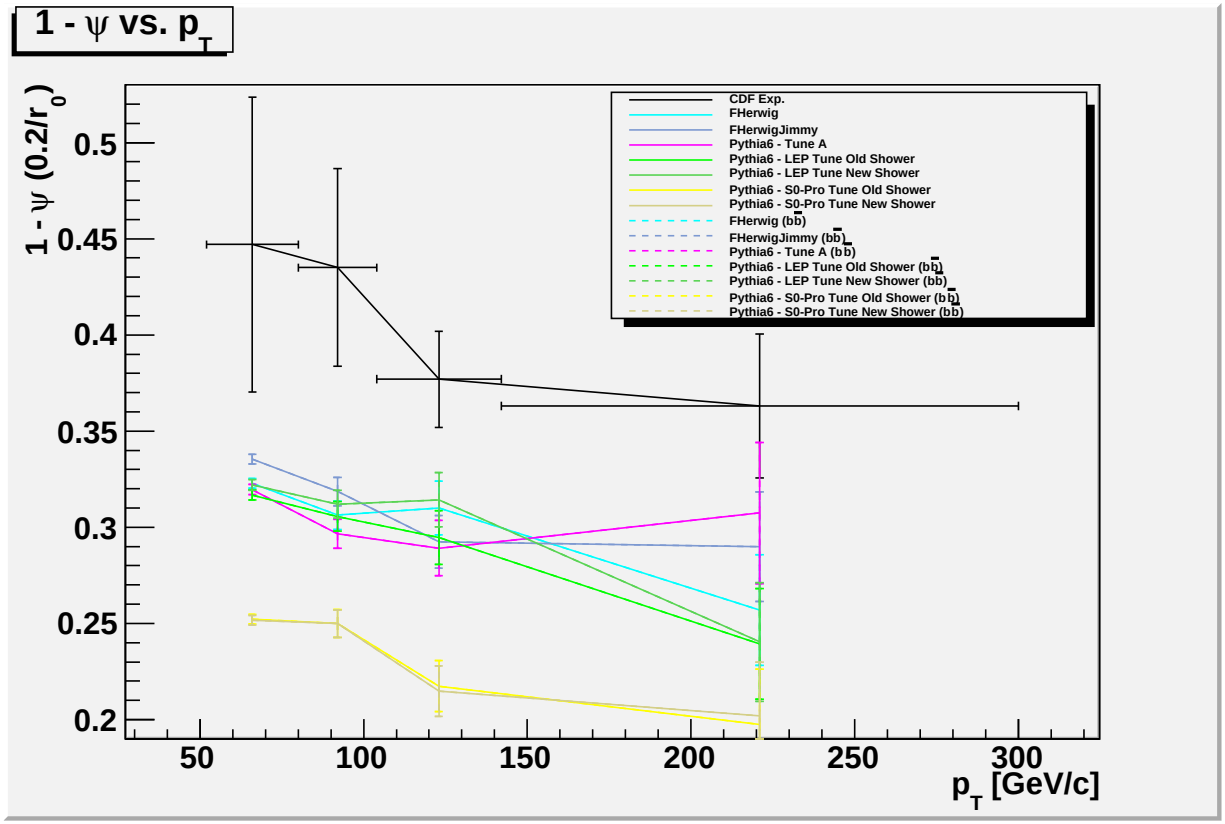




Comparison of generators and tunes : The generator comparison results



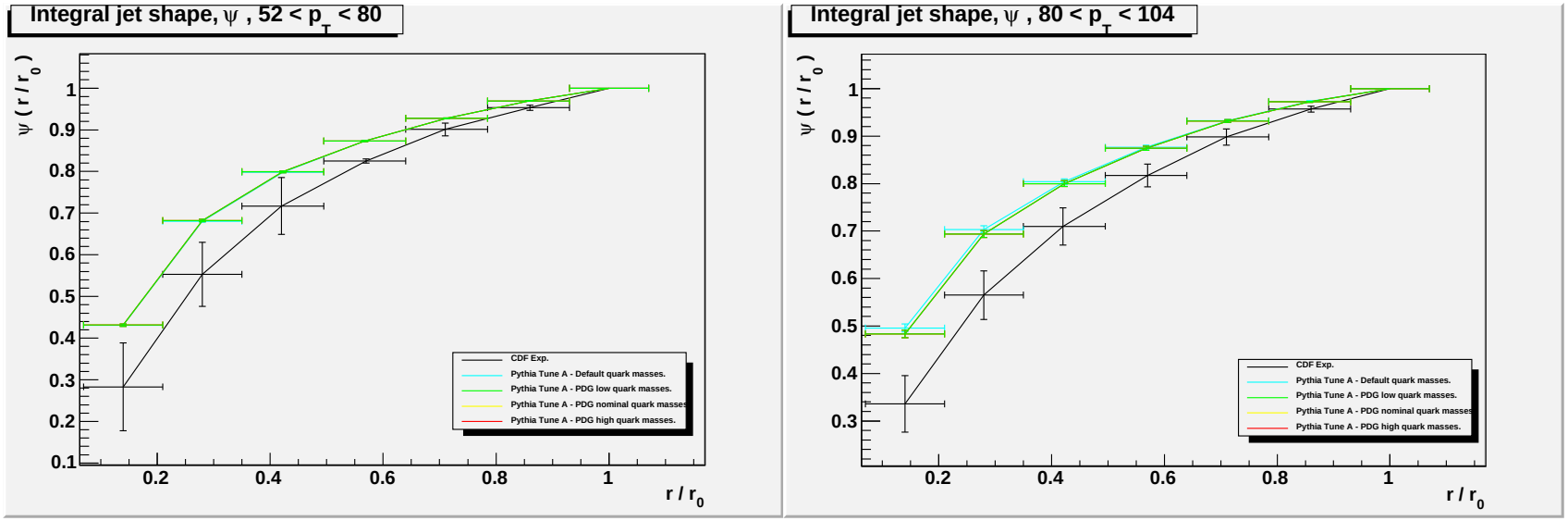
Comparison of generators and tunes : The generator comparison results



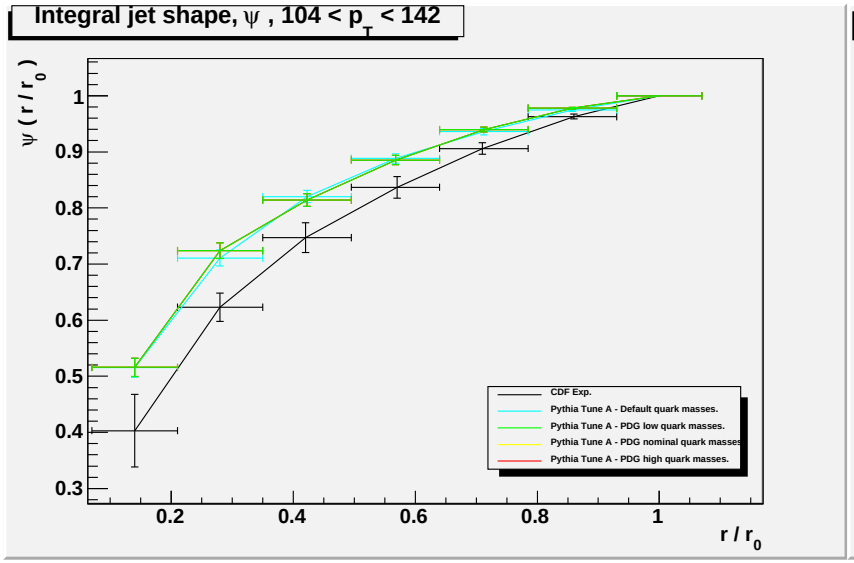
Tuning the quark masses in Pythia6 : Pythia6 default quark masses

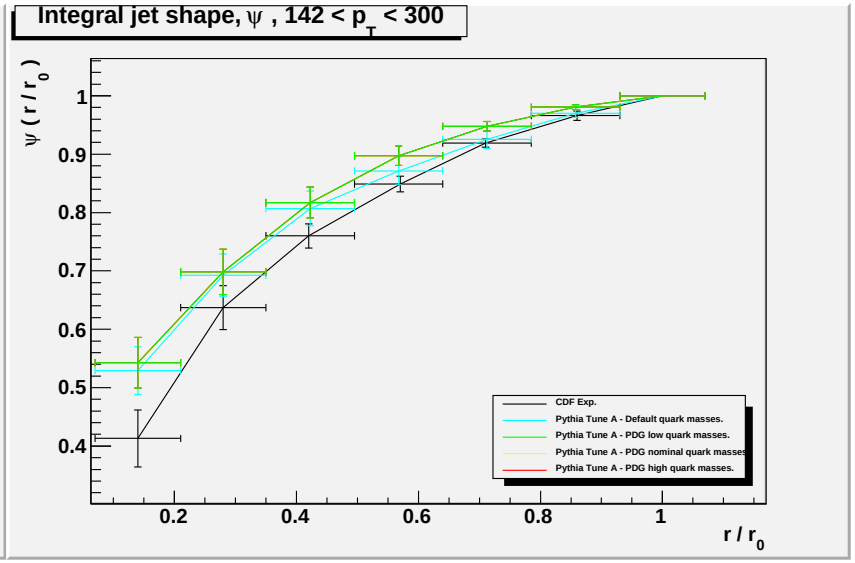
Quark	PDG mass	Pythia6 mass
d	$5.04^{+0.96}_{-1.54}$ MeV	9.9 MeV
u	$2.55^{+0.75}_{-1.05}$ MeV	5.6 MeV
s	104^{+26}_{-34} MeV	199 MeV
c	$1.27^{+0.07}_{-0.11}$ GeV	1.23 GeV
b	$4.20^{+0.17}_{-0.07}$ GeV	4.17 GeV
t	171.2 ± 2.1 GeV	165 GeV

Tuning the quark masses in Pythia6 : Quark mass tuning results

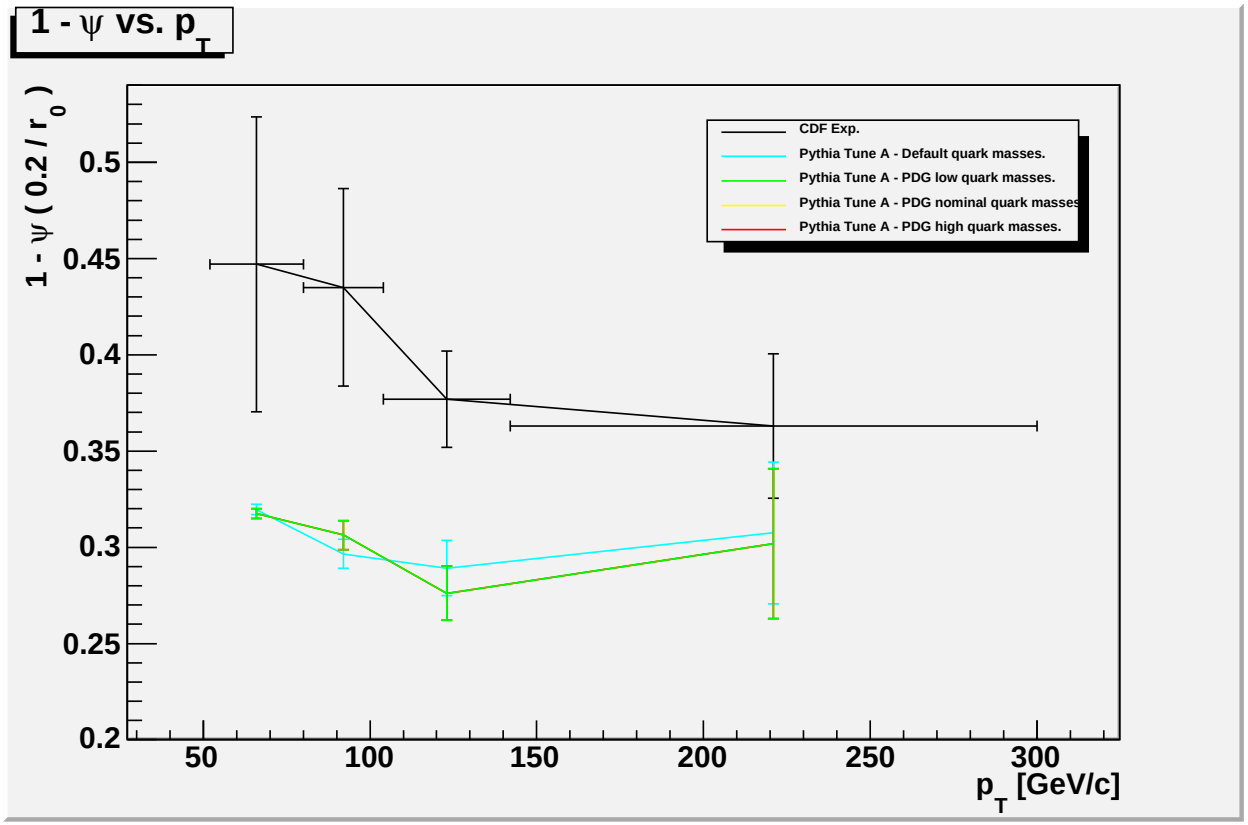


Tuning the quark masses in Pythia6 : Quark mass tuning results





Tuning the quark masses in Pythia6 : Quark mass tuning results



PARP(9x) tuning : The PARP(9x) parameters

- PARP(9x) control energy partitioning in hadron or resolved-photon remnant.
- PARP(91), PARP(94), PARP(95), PARP(96), PARP(97), PARP(98).
- Specify the k and χ factors of the energy partitioning function $(k+1)(1-\chi)^k$ in various situations.

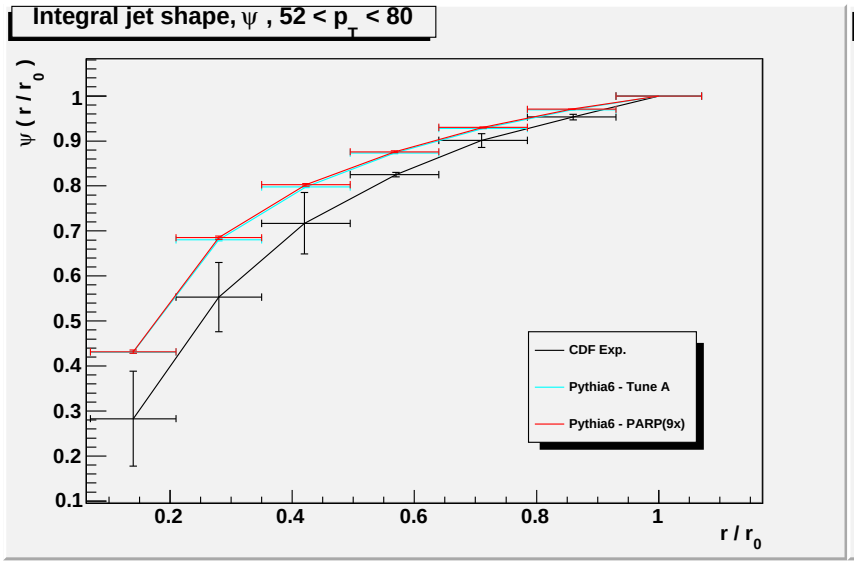
PARP(9x) tuning : The PARP(9x) parameters

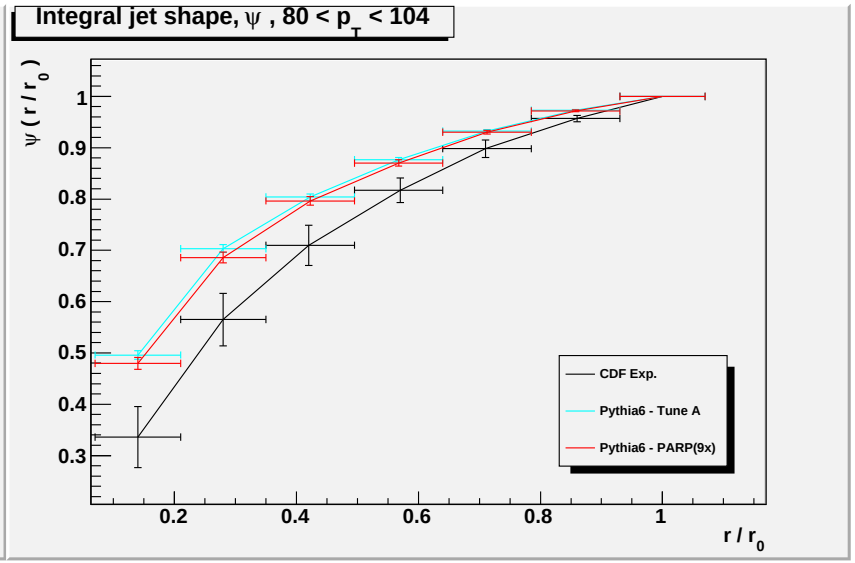
- PARP(9x) control energy partitioning in hadron or resolved-photon remnant.
- PARP(91), PARP(94), PARP(95), PARP(96), PARP(97), PARP(98).
- Specify the k and χ factors of the energy partitioning function $(k+1)(1-\chi)^k$ in various situations.

PARP(9x) tuning : The PARP(9x) parameters

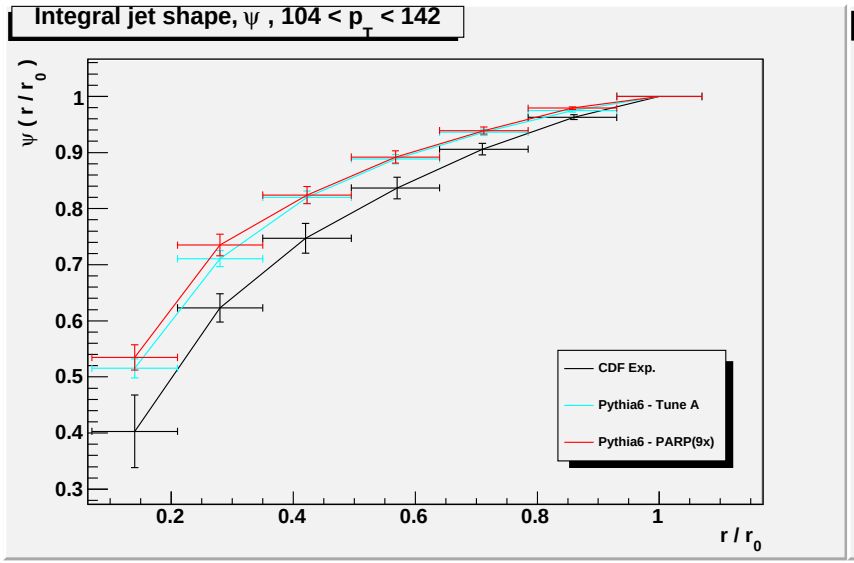
- PARP(9x) control energy partitioning in hadron or resolved-photon remnant.
- PARP(91), PARP(94), PARP(95), PARP(96), PARP(97), PARP(98).
- Specify the k and χ factors of the energy partitioning function $(k+1)(1-\chi)^k$ in various situations.

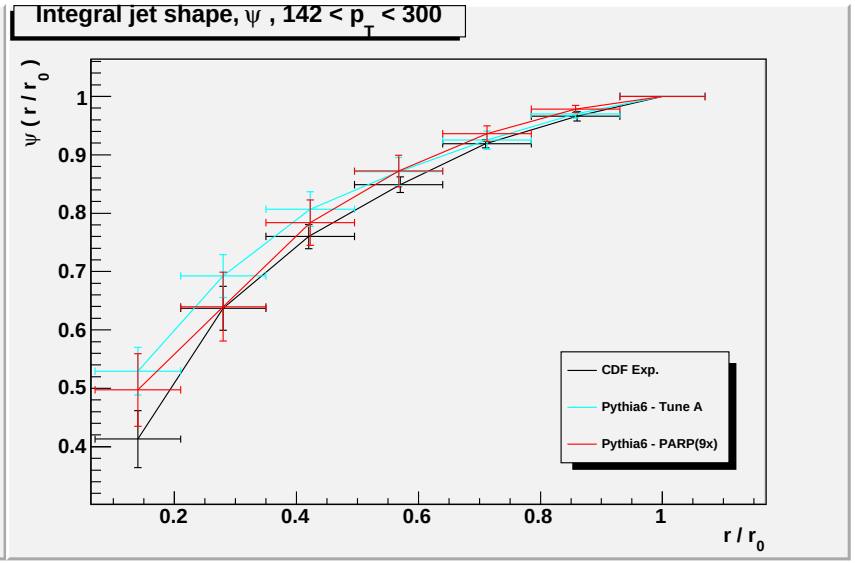
PARP(9x) tuning : PARP(9x) tuning results



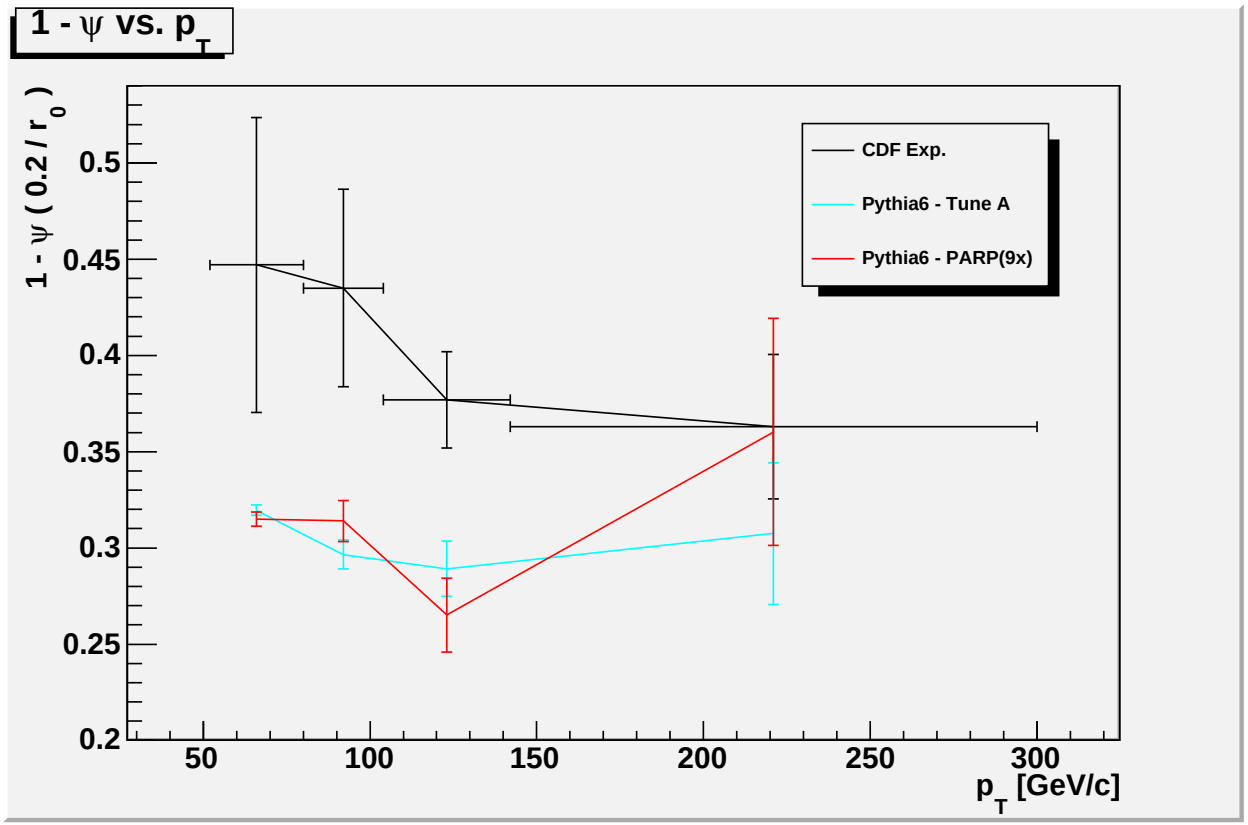


PARP(9x) tuning : PARP(9x) tuning results





PARP(9x) tuning : PARP(9x) tuning results



Conclusion : Results

- In all cases the jets produced in Monte-Carlo are too narrow.
- Explicitly selecting the jet flavours to be simulated doesn't affect the results.
- Tunes aren't necessarily good for different analyses.
- The latest tune (S0-Pro) is the worst for this analysis.
- As expected, the Jimmy multi-parton interaction add-on for Herwig produces better results than Herwig alone.
- For this analysis, the old and new shower methods in Pythia6 produce largely similar results.
- Quark masses don't seem to affect the jet width in simulations.

Conclusion : Results

- In all cases the jets produced in Monte-Carlo are too narrow.
- Explicitly selecting the jet flavours to be simulated doesn't affect the results.
- Tunes aren't necessarily good for different analyses.
- The latest tune (S0-Pro) is the worst for this analysis.
- As expected, the Jimmy multi-parton interaction add-on for Herwig produces better results than Herwig alone.
- For this analysis, the old and new shower methods in Pythia6 produce largely similar results.
- Quark masses don't seem to affect the jet width in simulations.

Conclusion : Results

- In all cases the jets produced in Monte-Carlo are too narrow.
- Explicitly selecting the jet flavours to be simulated doesn't affect the results.
- Tunes aren't necessarily good for different analyses.
- The latest tune (S0-Pro) is the worst for this analysis.
- As expected, the Jimmy multi-parton interaction add-on for Herwig produces better results than Herwig alone.
- For this analysis, the old and new shower methods in Pythia6 produce largely similar results.
- Quark masses don't seem to affect the jet width in simulations.

Conclusion : Results

- In all cases the jets produced in Monte-Carlo are too narrow.
- Explicitly selecting the jet flavours to be simulated doesn't affect the results.
- Tunes aren't necessarily good for different analyses.
- The latest tune (S0-Pro) is the worst for this analysis.
- As expected, the Jimmy multi-parton interaction add-on for Herwig produces better results than Herwig alone.
- For this analysis, the old and new shower methods in Pythia6 produce largely similar results.
- Quark masses don't seem to affect the jet width in simulations.

UCL

Conclusion : Results

- In all cases the jets produced in Monte-Carlo are too narrow.
- Explicitly selecting the jet flavours to be simulated doesn't affect the results.
- Tunes aren't necessarily good for different analyses.
- The latest tune (S0-Pro) is the worst for this analysis.
- As expected, the Jimmy multi-parton interaction add-on for Herwig produces better results than Herwig alone.
- For this analysis, the old and new shower methods in Pythia6 produce largely similar results.
- Quark masses don't seem to affect the jet width in simulations.

UCL

Conclusion : Results

- In all cases the jets produced in Monte-Carlo are too narrow.
- Explicitly selecting the jet flavours to be simulated doesn't affect the results.
- Tunes aren't necessarily good for different analyses.
- The latest tune (S0-Pro) is the worst for this analysis.
- As expected, the Jimmy multi-parton interaction add-on for Herwig produces better results than Herwig alone.
- For this analysis, the old and new shower methods in Pythia6 produce largely similar results.
- Quark masses don't seem to affect the jet width in simulations.

Conclusion : Results

- In all cases the jets produced in Monte-Carlo are too narrow.
- Explicitly selecting the jet flavours to be simulated doesn't affect the results.
- Tunes aren't necessarily good for different analyses.
- The latest tune (S0-Pro) is the worst for this analysis.
- As expected, the Jimmy multi-parton interaction add-on for Herwig produces better results than Herwig alone.
- For this analysis, the old and new shower methods in Pythia6 produce largely similar results.
- Quark masses don't seem to affect the jet width in simulations.

Conclusion : Further Work

- Need more statistics for the quark mass investigation and the PARP(9x) tune.
- Errors scale $\propto \frac{1}{\sqrt{N}}$.
- Need to check that PARP(9x) tune doesn't only affect this analysis, *i.e.* check against CDF-II min-bias data.
- Beyond the scope of this project, investigate Higgs mass via b-jet analysis.
- Compare with newer C++ generators Pythia8 and Herwig++.

Conclusion : Further Work

- Need more statistics for the quark mass investigation and the PARP(9x) tune.
- Errors scale $\propto \frac{1}{\sqrt{N}}$.
- Need to check that PARP(9x) tune doesn't only affect this analysis, *i.e.* check against CDF-II min-bias data.
- Beyond the scope of this project, investigate Higgs mass via b-jet analysis.
- Compare with newer C++ generators Pythia8 and Herwig++.

Conclusion : Further Work

- Need more statistics for the quark mass investigation and the PARP(9x) tune.
- Errors scale $\propto \frac{1}{\sqrt{N}}$.
- Need to check that PARP(9x) tune doesn't only affect this analysis, *i.e.* check against CDF-II min-bias data.
- Beyond the scope of this project, investigate Higgs mass via b-jet analysis.
- Compare with newer C++ generators Pythia8 and Herwig++.

Conclusion : Further Work

- Need more statistics for the quark mass investigation and the PARP(9x) tune.
- Errors scale $\propto \frac{1}{\sqrt{N}}$.
- Need to check that PARP(9x) tune doesn't only affect this analysis, *i.e.* check against CDF-II min-bias data.
- Beyond the scope of this project, investigate Higgs mass via b-jet analysis.
- Compare with newer C++ generators Pythia8 and Herwig++.