starlib-EVZ

March 7, 1992

The Event Generators with Zebra
for the
STAR library at RHIC

Richard J. Morse
Nuclear Science Division
Lawrence Berkeley Laboratory

This brief note describes the use of the physics event generators with Zebra-based I/O interfaces available in the STAR library. In addition, it provides benchmark data on event topologies and distributions in rapidity and transverse momentum, for selected particle species, subject to the "standard" parameter sets for each event generator and identical cuts.

I. Event Generators

The directory evz of the STAR library contains the physics EVent generators with Zebra output [1]. The LUND-FRITIOF [2], VENUS (3.07) [3], and HIJING (2.0) [4],[5] event generators are present in the subdirectories lund, hijing, and venus under /export1/starlib/rhic/evz. The purpose of this brief note is to provide basic instructions for using these programs, and to provide some simple benchmarks of their performance at RHIC energies.

II. Sample Scripts and Parameters

The following paragraphs provide sample shell scripts which can be used in order to generate physics monte carlo data at the RHIC computing facility. Usually a command file containing an identical sequence of input lines may be employed on the CSA VAX/VMS system; the only differences will be the comment characters, definition of logical names as opposed to environment variables, and the manner of invoking the executables (all of which are operating system-dependent). It is appropriate at this point to make an observation as to the manner in which the actual ZEBRA file format is chosen, this point is also documented in /export1/starlib/rhic/egz/aaareadme.doc. If the file egzpar.par is present in the current working directory, then the ZEBRA file format which it specifies is chosen for the output file format; if egzpar.par is missing, then the file format defaults to the machine native format. If transportability is desired, then the user must explicitly select the ASCII format using an egzpar.par which looks as follows:

```
*********************************************
*
* $ZPAR
* Z_DOUBLE_OUT    = F,
* Z_BLOCKSIZE     =        3600,
* Z_FZ_OPTION_OUT = 'AO',
* Z_FZ_OPTION_IN  = 'AI',
* Z_FZLOGL        =          -2
* $END
*
*********************************************
```

Further, any other program using the EGZ interface (eg. the STAR GEANT Monte Carlo or the read_zebra_evt analysis shell) requires that the format of the data file be consistently communicated to it using the same mechanism.

The shell script used in order to generate 50 LUND-FRITIOF (version 1.7) events at 200 A·GeV in the center-of-mass system with impact parameters varying from 0 to 1 fm is presented below:

```
*********************************************
*
* #!/bin/csh
```

```
* #
* #    50 GOLD+GOLD events
* #    ===================================
* setenv Z_TAPE /scr0/rmorse/scratch_
*                /auau_200gev_50.lun
* #
* time $LUND/lund_model <<eof
* 3
* Y
* Y
* 2349785
* 0
* 0
* -200.0
* 50
* 0,1
* GOLD
* GOLD
* eof
* #
* unsetenv Z_TAPE
*
**********************************************
```

The first line of the script defines the name of the output file for the Zebra Interface package (EGZ) through the environment variable Z_TAPE. Note that under VAX/VMS it is necessary to define a logical name Z_TAPE which provides the output file name; also note that in interactive mode the package prompts for the output file name if Z_TAPE has no definition. The next line of the script invokes the executable; in the case of LUND-FRITIOF, this is $LUND/lund_model (where $LUND is defined by the rhic_defs script to be /export1/starlib/rhic/evz/lund). The manner of invoking the executable, using the *time* command, redirects all of the lines following, until the line containing *eof*, to the standard input of the child process associated with the executable. The next line defines the parameters a,b, and sigma to be used where a and b govern the fragmentation function, and sigma is the rms of the distribution of transverse momenta transferred in string-string interactions. In the case of LUND-FRITIOF the user has the option of employing either: i) the standard LUND parameters, ii) entering a user parameter set, iii) Gyulassy's default LUND parameter set [6], iv) Gyulassy's so-called F-set [6], or v) J. Costales' tuned 15 A·GeV p+A parameter set [7]. For the test case shown, Gyulassy's default LUND parameter set has been chosen. The next line indicates whether or not the user desires to include the full decay history in the output kinematics. By default, JETSET will decay some of the particles generated during the process of fragmentation into what it considers to be 'stable' secondaries; by answering 'Y' to this question, the user indicates a desire to retain all parent information in the output.

The following four lines indicate a desire to initialise the random number generator, and provides a starting seed (which should be a large positive interger) for use with the compiler-default random number generator. The beam energy (where a negative value denotes the center-of-mass system and a positive value denotes fixed-target in the lab system) and number of events desired are provided on the subsequent two lines. The final three lines provide the minium and maximum impact parameters, the projectile, and the target. Note that if the target and projectile are nuclei not known to the main driver, then the user must input 'NUCLEUS' and then provide the nucleon number and proton number on following lines. A list of the names known to the LUND driver is presented below:

```
*********************************************
*
*  ANTIPROTON             SILICON
*  PROTON                 CALCIUM
*  NEUTRON                ARGON
*  DEUTERIUM              COPPER
*  HELIUM                 SILVER
*  BORON                  XENON
*  CARBON                 GOLD
*  OXYGEN                 LEAD
*  ALUMINUM               URANIUM
*            PI+  PI-
*            K+   K-
*
*********************************************
```

The shell script used in order to generate 50 HIJING (version 2.0) events at 200 A·GeV in the center-of-mass system with impact parameters varying from 0 to 1 fm is presented below:

```
*********************************************
*
* #!/bin/csh
* #
* #
* #
* setenv Z_TAPE /scr0/rmorse/scratch_
*                   /auau_200gev_50.hij
* #
* time $HIJNG/evz/hijing/hijevt <<eof
* hij_auau.dat
* 10
* eof
* #
```

```
* unsetenv Z_TAPE
*
**********************************************
```

Note that the script is relatively simple, containing lines which: i) define the output file name, ii) invoke the executable, iii) provide the name of a HIJING parameter file, and iv) provide the number of events between event completion notification from HIJING. The actual specification of running parameters occurs through the parameter file, in this case hij_auau.dat, which is provided below; the comments following each line define the affected parameters. All of the lines up to an including the seed value must be specified by the user; the remaining values are optional and permit redefinition of the parameters contained in the arrays HIPR1, IHPR2, HINT1, AND HNT2 of the HIPARNT common. A full description of these parameters can be found in $HIJNG/hijing.txt. The format of these optional lines contains three elements:

```
'character' N VALUE
```

where character is one of the array names (HIPR1, IHPR2, HINT1, or IHNT2 as an uppercase *explicitly* quoted string), and N is the element of the array which will be set to VALUE. Note that by default HIJING: i) activates jet and minijet production, ii) activates the jet quenching mechanism, iii) activates gluon shadowing, and iv) does not decay neutral pions and kaons, D + /−, lambdas, sigma + /−, neutral cascades and omega− hyperons.

```
**********************************************
*
* Sample file with parameters to run HIJING
* Generate 50 Au+Au events at sqrt(s)=200 GeV
* One sample jet of pt=30 GeV per event
* This text header can have up to 100 lines.
* But the last line should always be at least
* four dollar signs in a row.
* $$$$
* 200.0              !  nucleon-nucleon energy
* 'CMS'              ! frame
* 'A','A'            ! proj, targ
* 197, 79,197, 79    !  A,Z proj, A, Z tgt
* 50                 !  number of events
* 0.0  1.0           !  min, max impact param
* -7.0  7.0          !  minimum, maximum eta
* 239047865          !  seed
* 'HIPR1' 10  30.0   !  sample jet pt
* 'IHPR2' 3   1      !  switch on sample jets
* 'IHPR2' 12  0      !  permit resonance decay
```

```
* 'IHPR2' 18  1      !  keep all parents
*
**********************************************
```

Finally, the shell script used in order to generate 50 VENUS (version 3.07) events at 200 A·GeV in the center-of-mass system with impact parameters varying from 0 to 1 fm is presented below:

```
**********************************************
*
* #!/bin/csh
* #
* #    50 GOLD GOLD events
* #    ===================================
* setenv Z_TAPE /scr0/rmorse/scratch
*               /auau_200gev_50.ven
* #
* echo "'had'">optns.dat
* echo "50 10  -200.  79 197 79 197 6">>_
*       optns.dat
* echo "'istmax' 1">>optns.dat
* echo "'irescl' 0">>optns.dat
* echo "'radiac' 0">>optns.dat
* echo "'ndecay' 0">>optns.dat
* echo "'bmaxim' 1.00">>optns.dat
* echo "'seedi' 1340975">>optns.dat
* echo "'stp'">>optns.dat
* #
* time $VENUS/venus307 <<eof
* eof
* #
* # clean up
* #
* rm optns.dat check.dat histo.dat
* unsetenv Z_TAPE
* #
*
**********************************************
```

This script actually creates the parameter file, optns.dat, which VENUS reads in order to set running conditions. The first line of this file indicates the type of interactions, hadron-hadron or hadron-nucleus, or nucleus-nucleus, to be modelled. The second line provides: i) the number of events to be generated, ii) the number of events between notification, iii) the beam energy (where a negative value denotes center-of-mass frame), iv) the projectile and target proton and nucleon numbers, and v) the number of parameter cards following. A full description of the parameters which can be set using the optns.dat cards, and their defaults, can be found in the ASCII header of the VENUS subroutine ainitl in the file $VENUS/venus307.car; for most purposes, the set of optns.dat cards illustrated above should be adequate. The 'istmax' card indicates that parents of stable secondaries will be kept in the output kinematics. The 'irescl' card activates so-called *momentum rescaling* which is necessary in order to achieve finite event generation times for 200 A·GeV center-of-mass beam energies. The 'radiac' card indicates the radius over which final-state rescattering will occur; setting this parameter equal to zero effectively deactivates rescattering. The 'ndecay' card is used in order to flag resonances and particles which will be treated as unstable, and hence decayed, by VENUS; setting ndecay equal to zero causes VENUS to treat all resonances, K shorts, lambdas, sigmas, cascades, omegas, and neutral pions as unstable. The 'bmaxim' card sets the maximum impact parameter for the generated events. The 'seedi' card provides the initial seed for the random number generator. Finally, the 'stp' card is an obligatory card indicating the end of parameter input. The script then invokes the executable, and then at the end of the run deletes the optns.dat file along with two other diagnostic files written by VENUS.

III. Analysis

In order to provide users with a mechanism for analysing the output of the event generators, a simple shell program has been provided. This shell calls three user-defined subroutines, aninit, anuser, and anfin, which perform the tasks associated with initialisation, event analysis, and analysis summary output. Note that initialisation can include the booking of HBOOK histograms, and summary output can include the output of HBOOK histograms. The event analysis entry has access to the full event kinematics, and associated information, through FORTRAN common blocks. A summary of the event kinematics variables available to anuser is provided below:

```
*********************************************
* c
* c maxptl > maximum number of particles per
* c           event.
* c nptls  > number of particles in event
* c nnproj > # neutron spectators (in proj)
* c npproj > # proton  spectators (in proj)
* c naproj > # spectators (in proj)
* c nptarg > # proton  spectators (in target)
* c natarg > # spectators (in target)
* c ntry   > # of tries to get an interaction
* c            since last call
* c            This number is needed to get a
* c          cross section from the number
* c          of attempts + the impact
```

```
* c             parameter range.
* c px > 4 momentum of each particle
* c py >
* c pz >
* c energy>
* c idptl > ID for particle (as given by
* c                            event generator)
* c ioptl > Parent sequence number
* c                           ( 0 if no parent)
* c
* c pt    > perpendicular momentum
* c                          (sqrt(px**2+py**2)
* c y     > rapidity
* c pt    > transverse momentum (GeV/c)
* c mass  > mass in GeV of particle
* c theta > polar angle theta in radians
* c phi   > azimuthal angle in radians
* c gtype > GEANT particle ID.
* c
**********************************************
```

The shell program is $EGZ_SRC/read_zebra_evt.f; a sample set of user subroutines can be found in $EGZ_EXA/dump.f. The user subroutines can be linked into a complete program using the script link_read found in $EGZ_COM (or the analogous command file found on the CSA cluster in DISK$RNC0:[RHIC.EGZ.COM]). For example, in order to link the user subroutines dump.f into the executable dump the user simply issues the command $EGZ_COM/link_read dump.f.

IV. Sample Output

In order to compare the gross features of the output of each of the three event generators in terms of their performance, Monte Carlo data sets consisting of fifty events were created for each of the generators. The three tables that follow present the event-averaged yields, into all phase space, of pions, kaons, protons, and anti-protons, in addition to the integral over all charged particles (where this implies adding the yield of electrons, positrons, muons, and anti-muons to the sum of the former species).

```
**********************************************
*                                            *
*           FRITIOF Averages                 *
*                                            *
* pi(+)>    2550.120   pi(-)>       2605.540 *
* k (+)>    268.7200   k (-)>       247.8800 *
* proton>   355.5800   anti-proton> 171.9000 *
* charged>  6274.860                         *
```

```
*                                             *
***********************************************


***********************************************
*                                             *
*           HIJING Averages                   *
*                                             *
* pi(+)>    2333.880   pi(-)>       2380.640 *
* k (+)>    252.9400   k (-)>       236.7400 *
* proton>   365.1400   anti-proton> 176.7400 *
* charged>  5814.600                          *
*                                             *
***********************************************


***********************************************
*                                             *
*           VENUS Averages                    *
*                                             *
* pi(+)>    2897.360   pi(-)>       2953.080 *
* k (-)>    273.3800   k (-)>       254.2800 *
* proton>   489.4200   anti-proton> 159.0600 *
* charged>  7094.460                          *
*                                             *
***********************************************
```

Finally, the appended figures illustrate $\frac{1}{N_{ev}}\frac{dN}{dy}$ and $\frac{1}{N_{ev}}\frac{dN}{p_t dp_t}$ for a sample of 50 events from each of the three event generators.

Notes

[1] F. Videbaek and T. Throwe, Fourth Workshop on Experiments and Detectors for a Relativistic Heavy Ion Collider, BNL 52262, p. 283.

[2] B. Andersson, G. Gustafson, and B. Nilsson-Almqvist, Nucl. Phys. B281, p. 289 (1987); B. Nilsson-Almqvist and E. Stenlund, Comput. Phys. Comm. 43, p. 387 (1987).

[3] K. Werner, Phys. Rev. D34, p. 780 (1989); K. Werner and P. Koch, Phys. Lett. B242, p. 251 (1990).

[4] X. N. Wang and M. Gyulassy, Phys. Rev. D44, p. 3501 (1991).

[5] A big thanks goes to Matt Bloomer for his valuable remarks concerning the description of the HIJING event generator.

[6] M. Gyulassy, *ATTILA...*, unpublished preprint available as CERN-TH.4794 (1987).

[7] J. B. Costales, *LUND Model Results at or Near AGS Energies*, unpublished E-802 memo available as E-802-Mem-8, (1988).

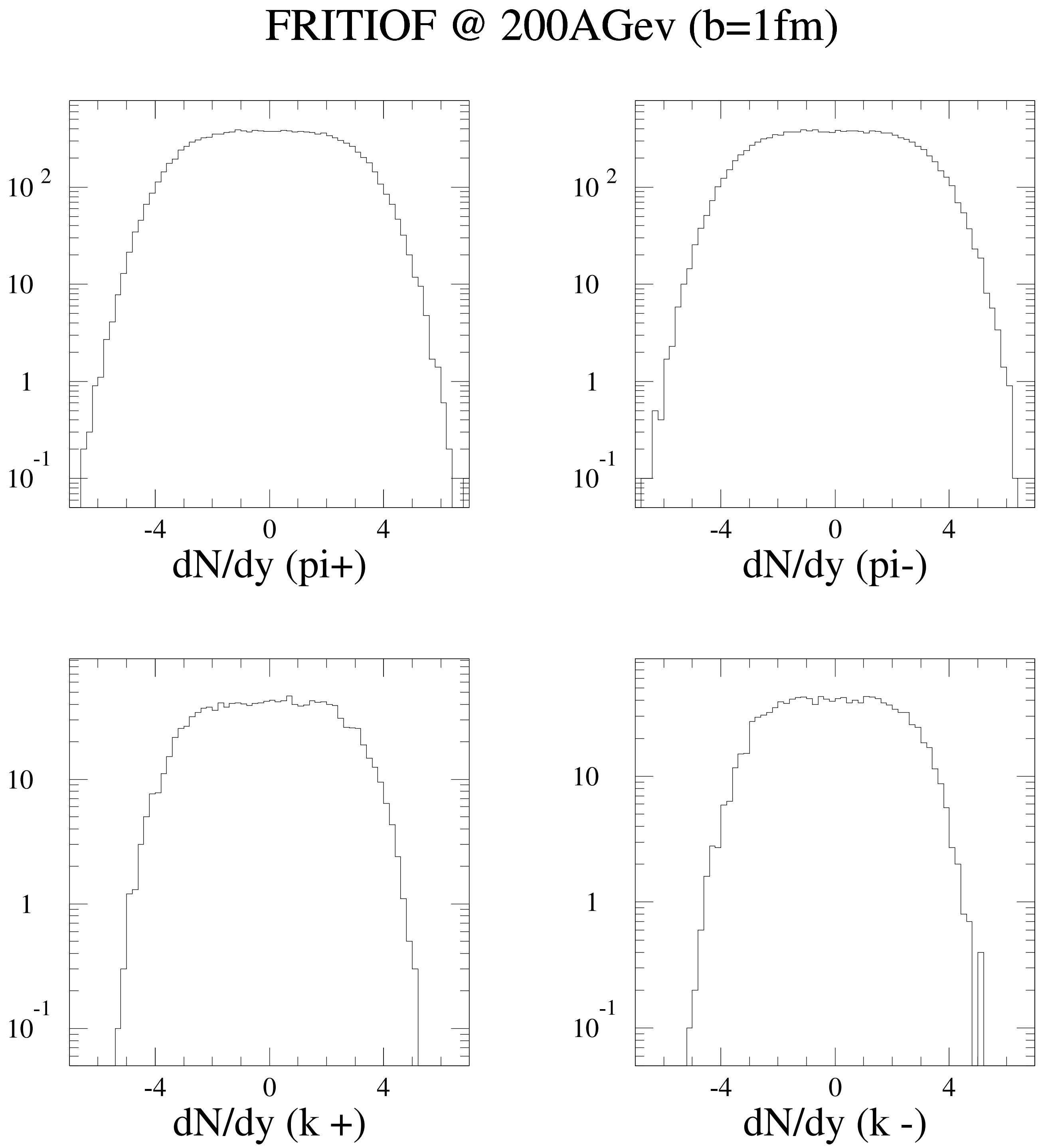

Fig. 1. FRITIOF rapidity distributions for pions and kaons.

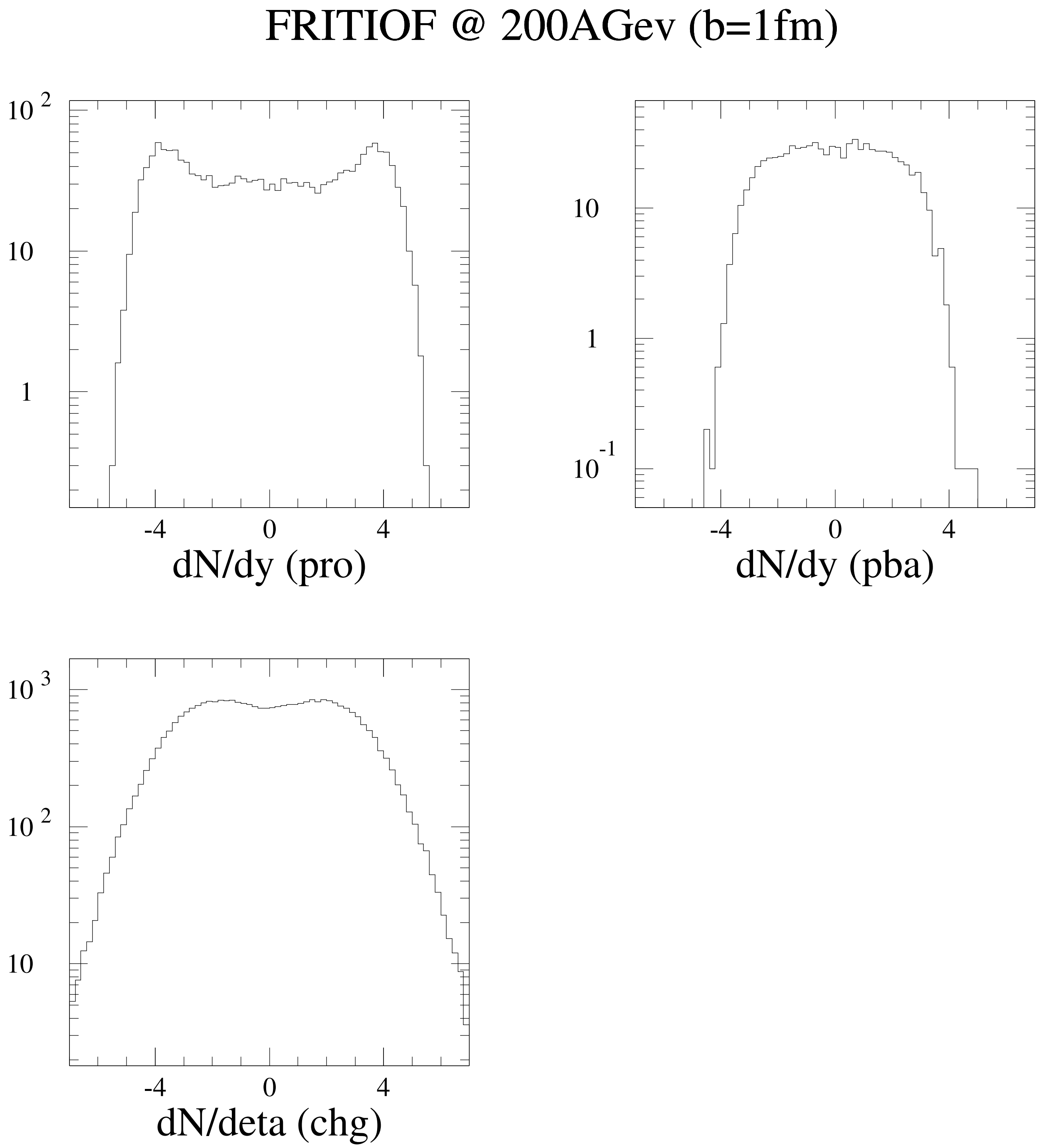

Fig. 2. FRITIOF rapidity distributions for protons, antiprotons, and all charged particles

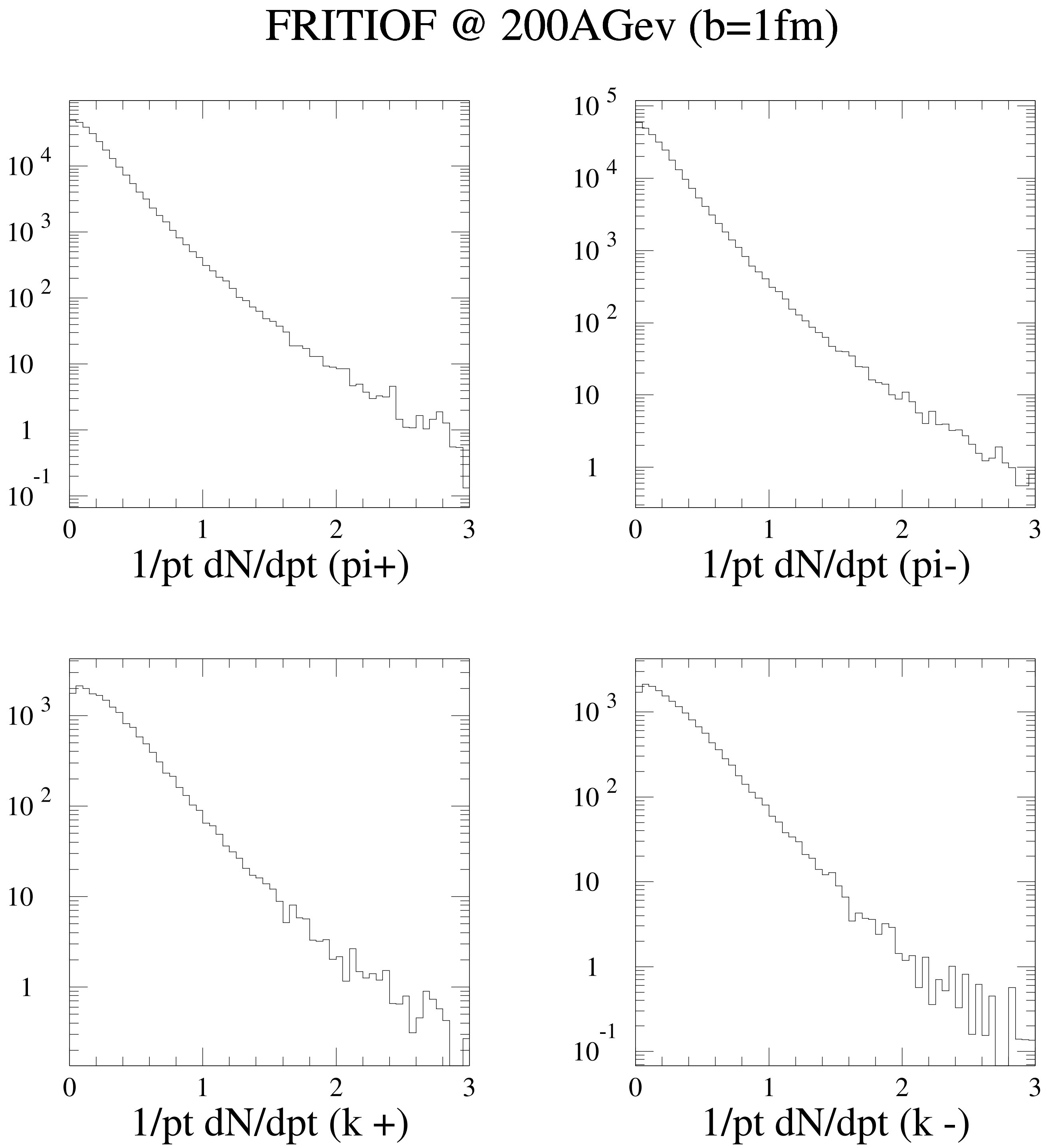

Fig. 3. FRITIOF p_t distributions for pions and kaons.

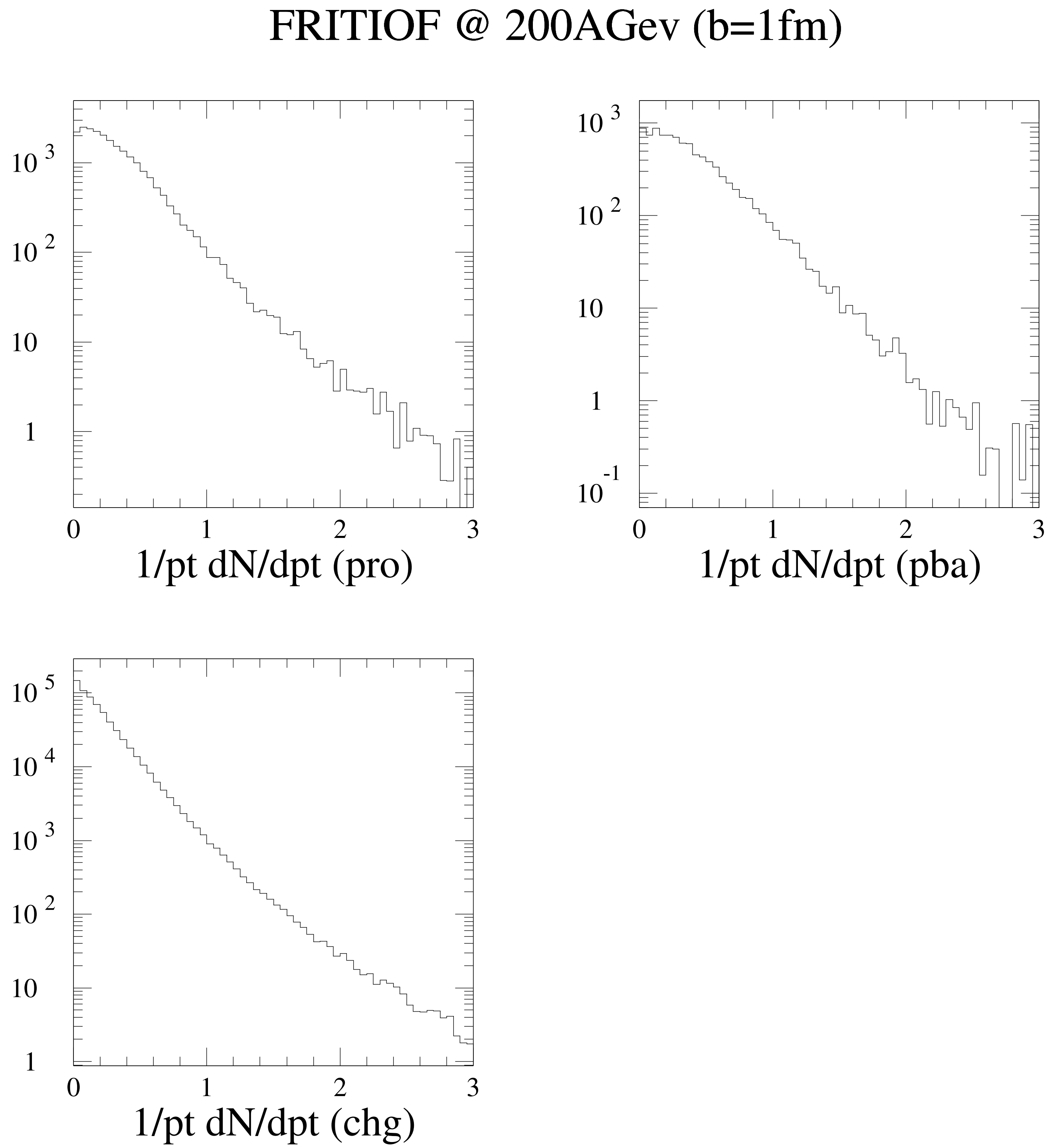

Fig. 4. FRITIOF p_t distributions for protons, antiprotons, and all charged particles

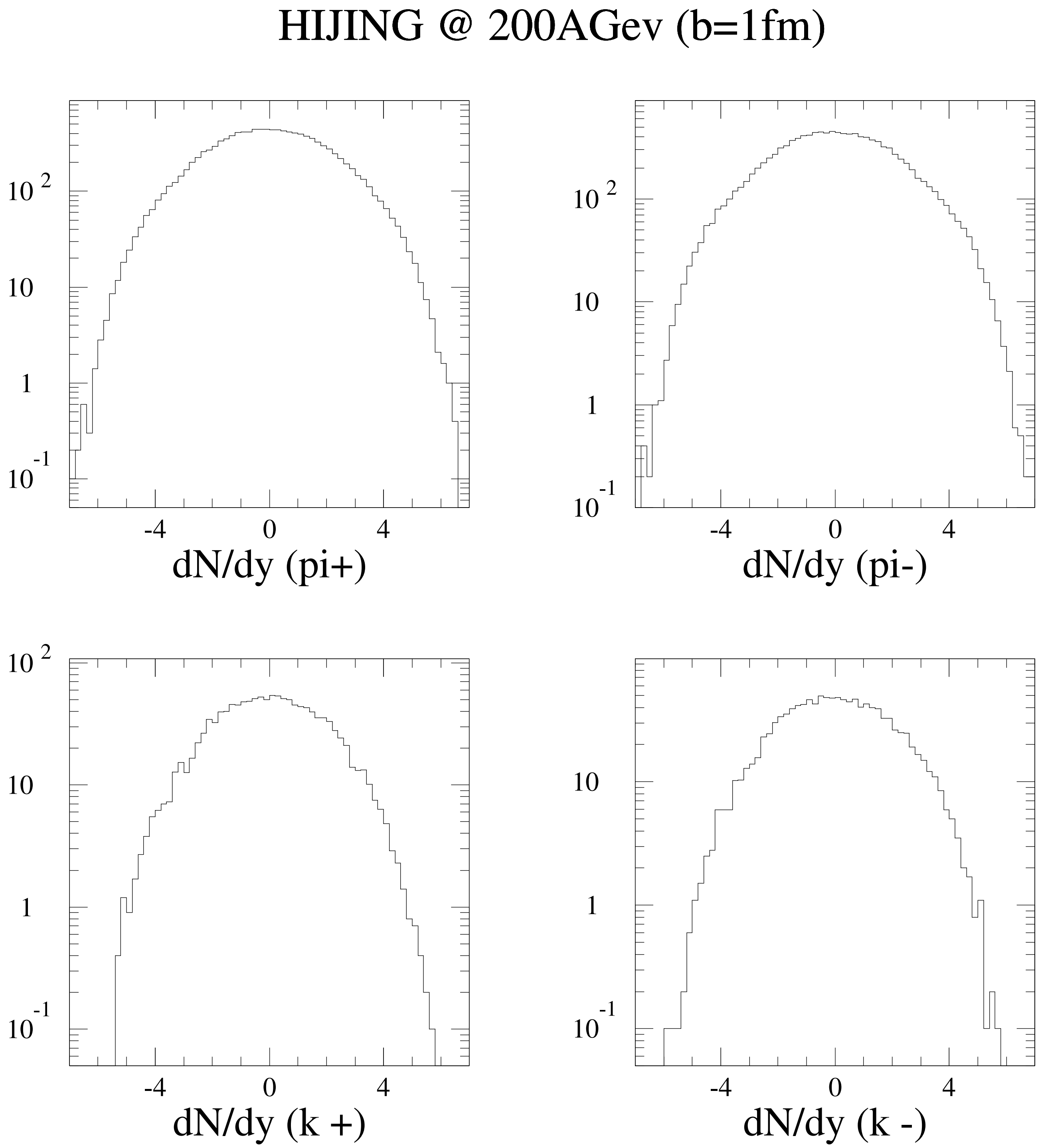

Fig. 5. HIJING rapidity distributions for pions and kaons.

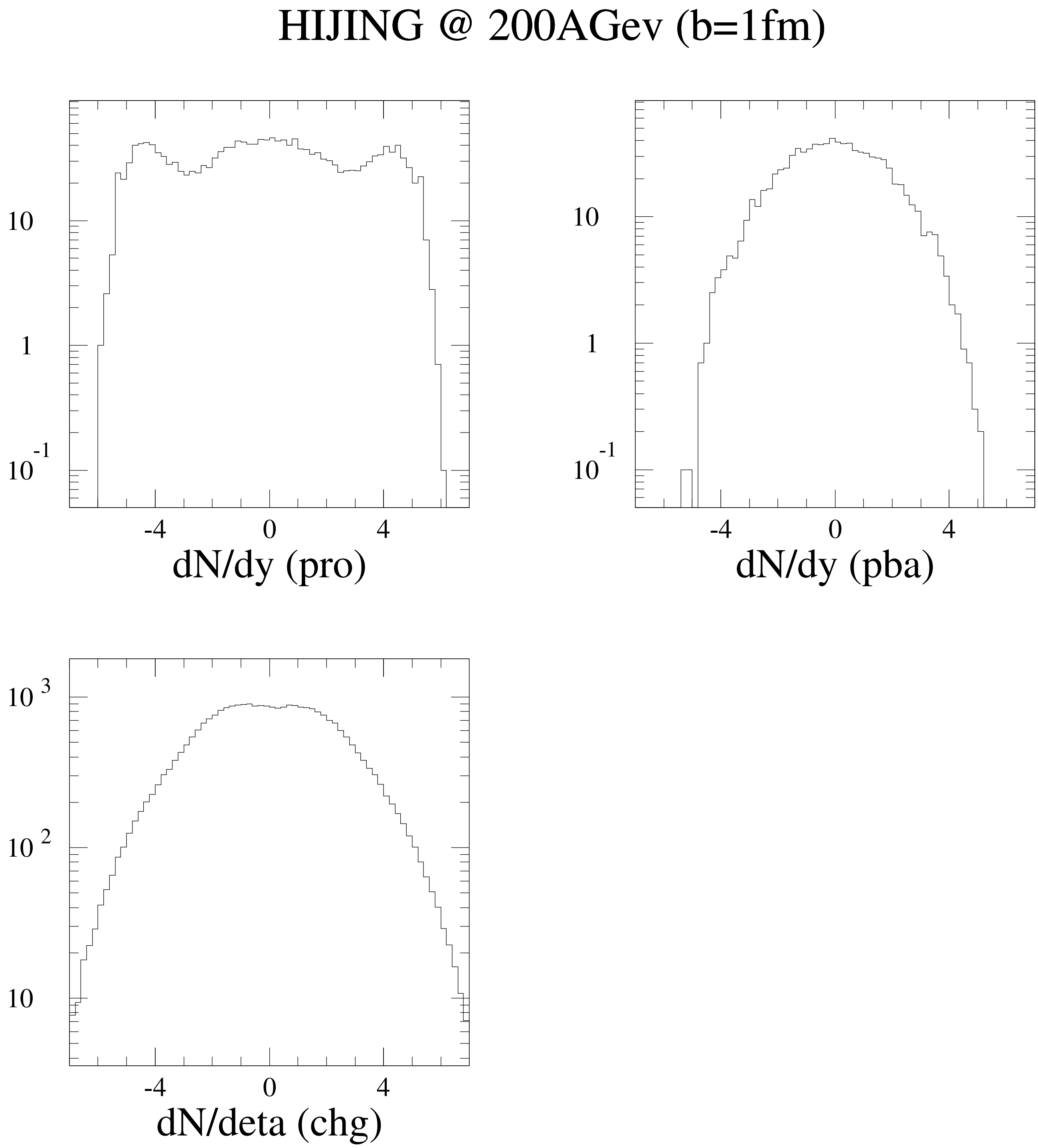

Fig. 6. HIJING rapidity distributions for protons, antiprotons, and all charged particles

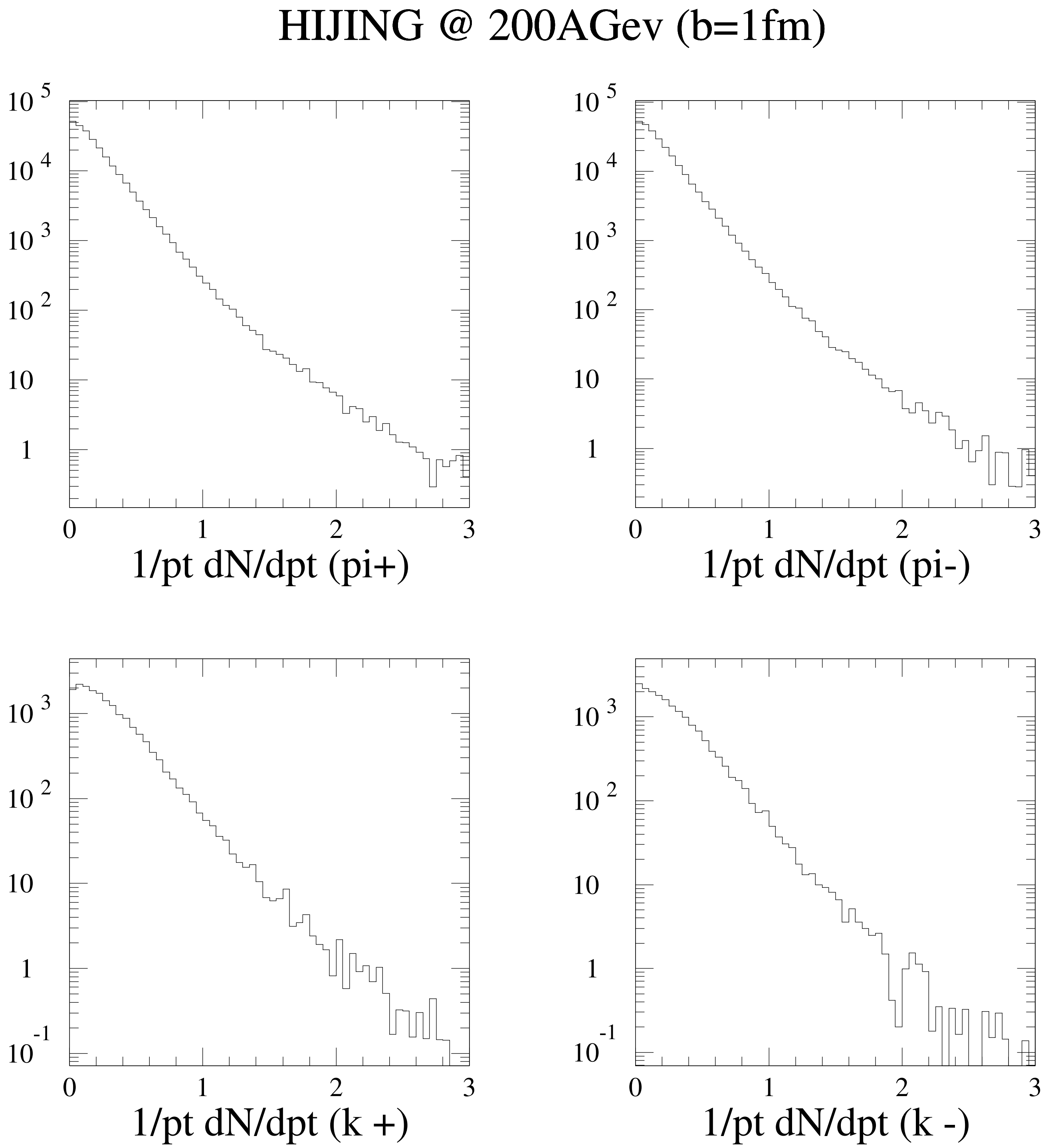

Fig. 7. HIJING p_t distributions for pions and kaons.

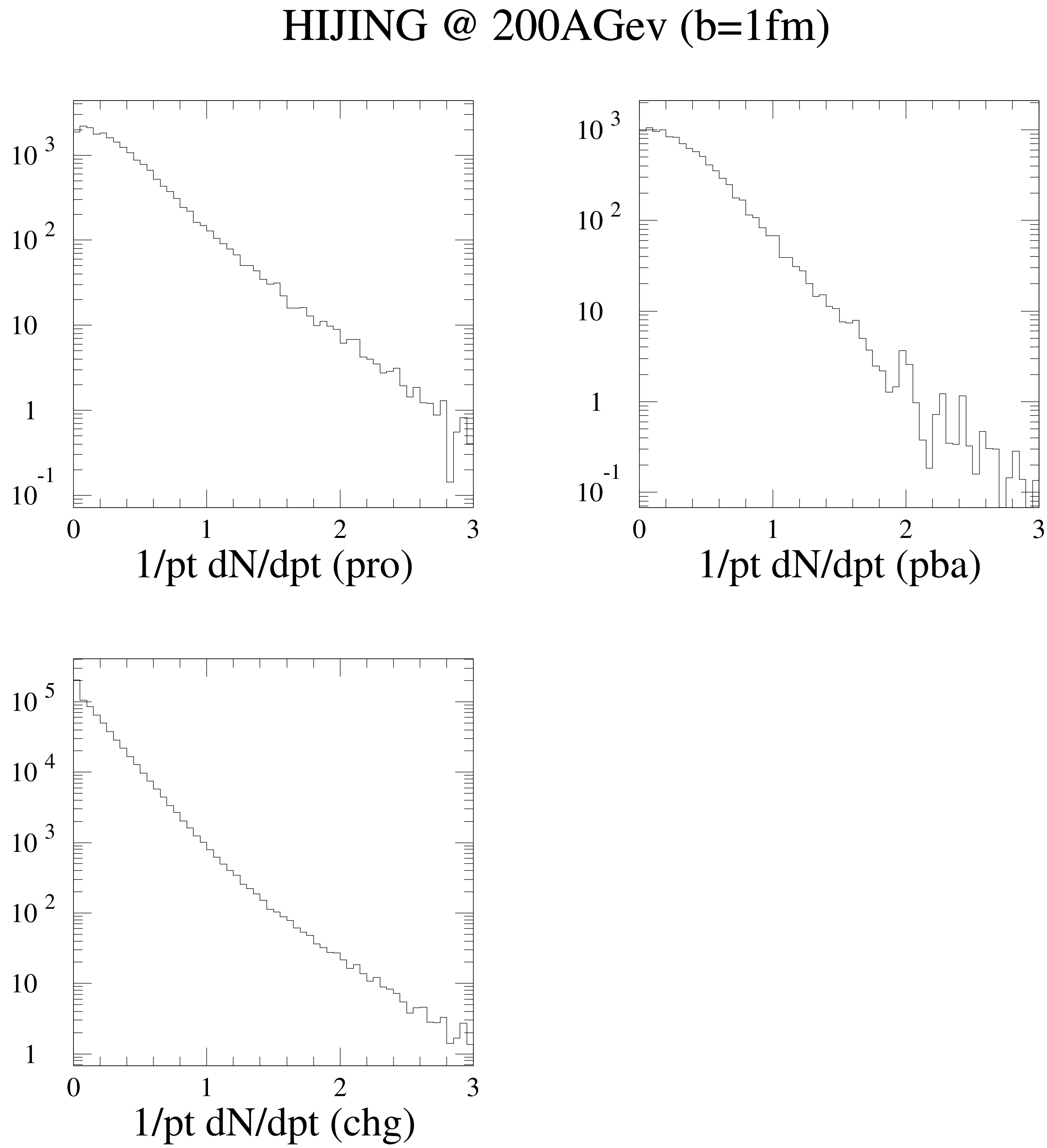

Fig. 8. HIJING p_t distributions for protons, antiprotons, and all charged particles

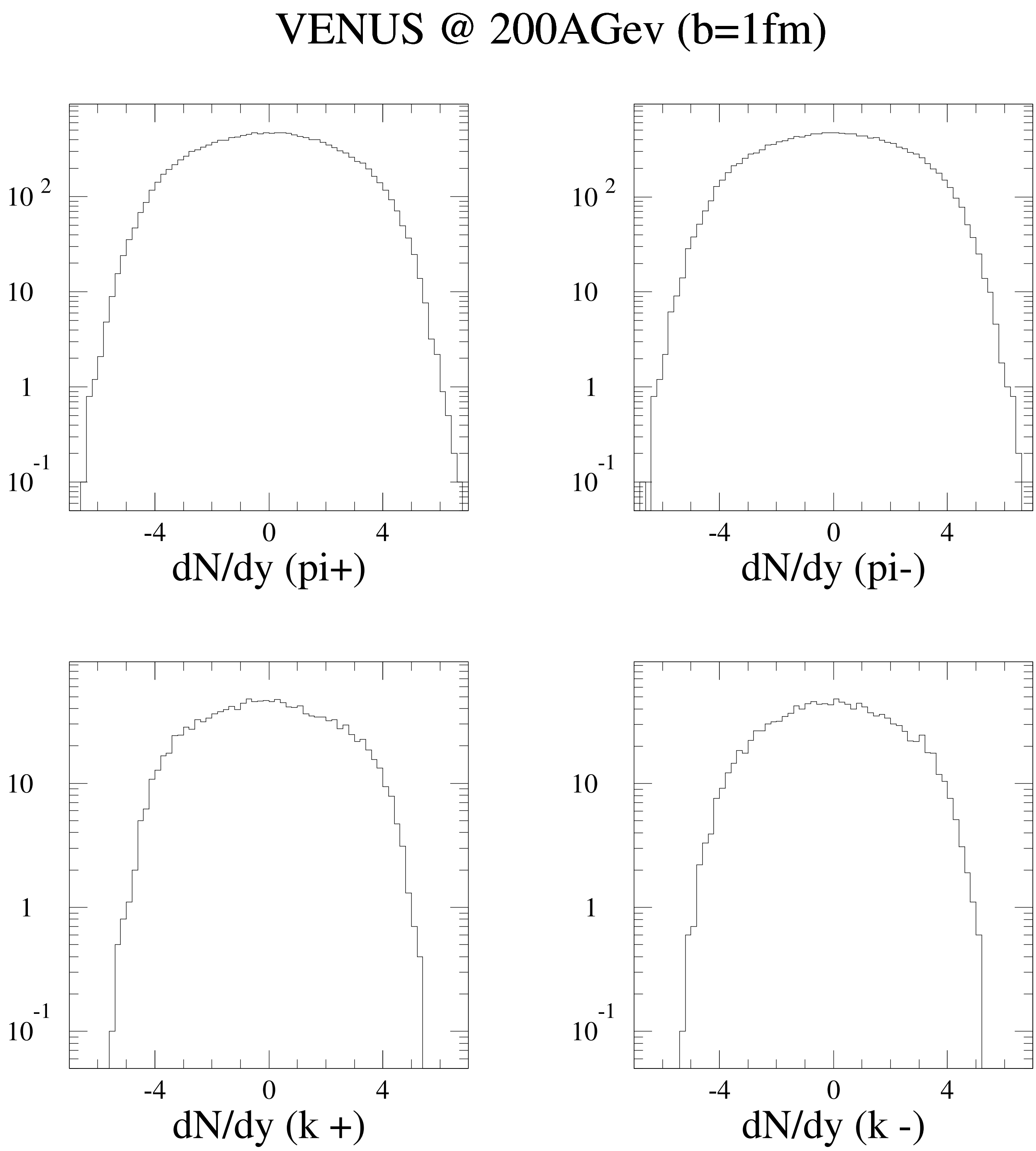

Fig. 9. VENUS rapidity distributions for pions and kaons.

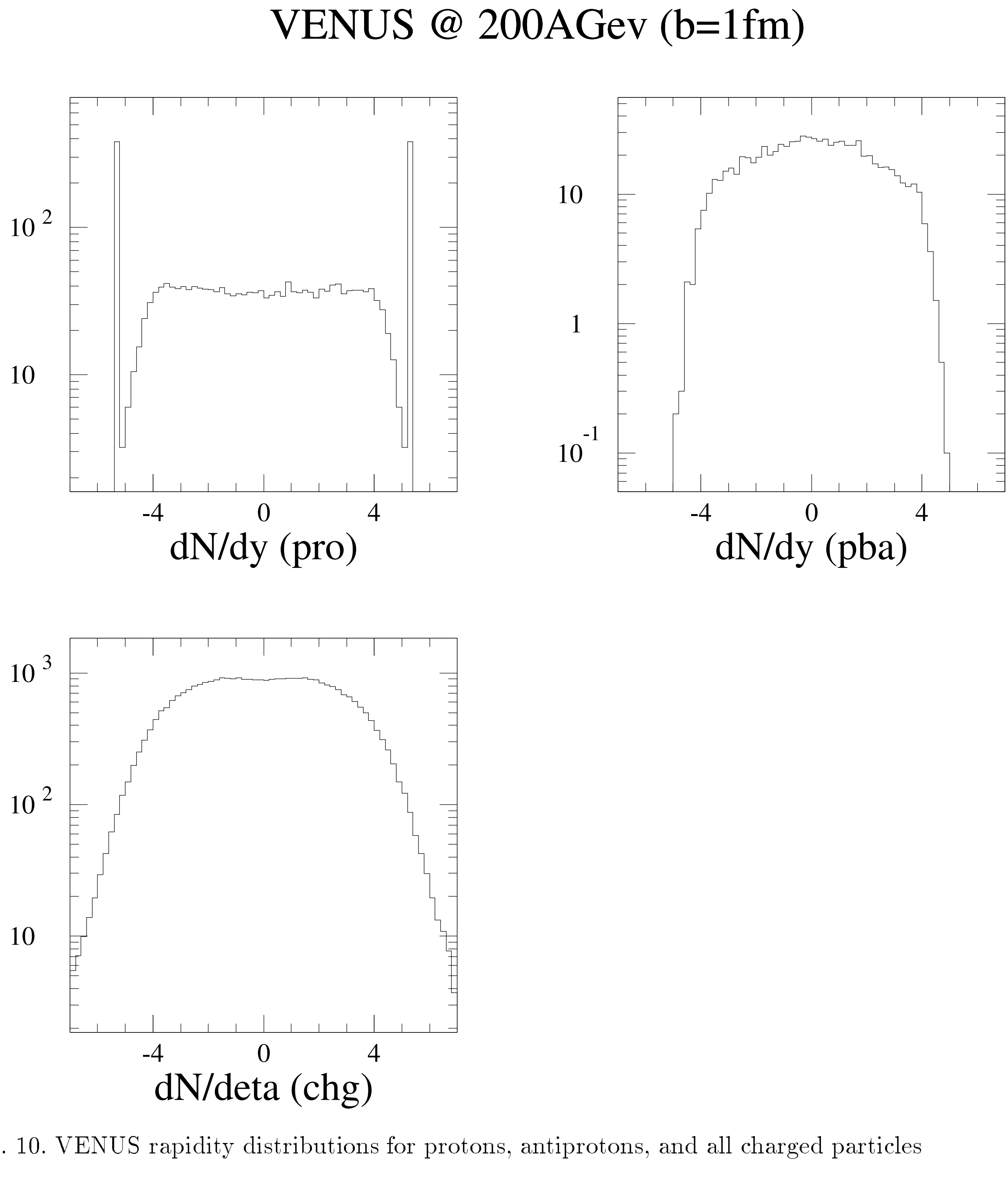

Fig. 10. VENUS rapidity distributions for protons, antiprotons, and all charged particles

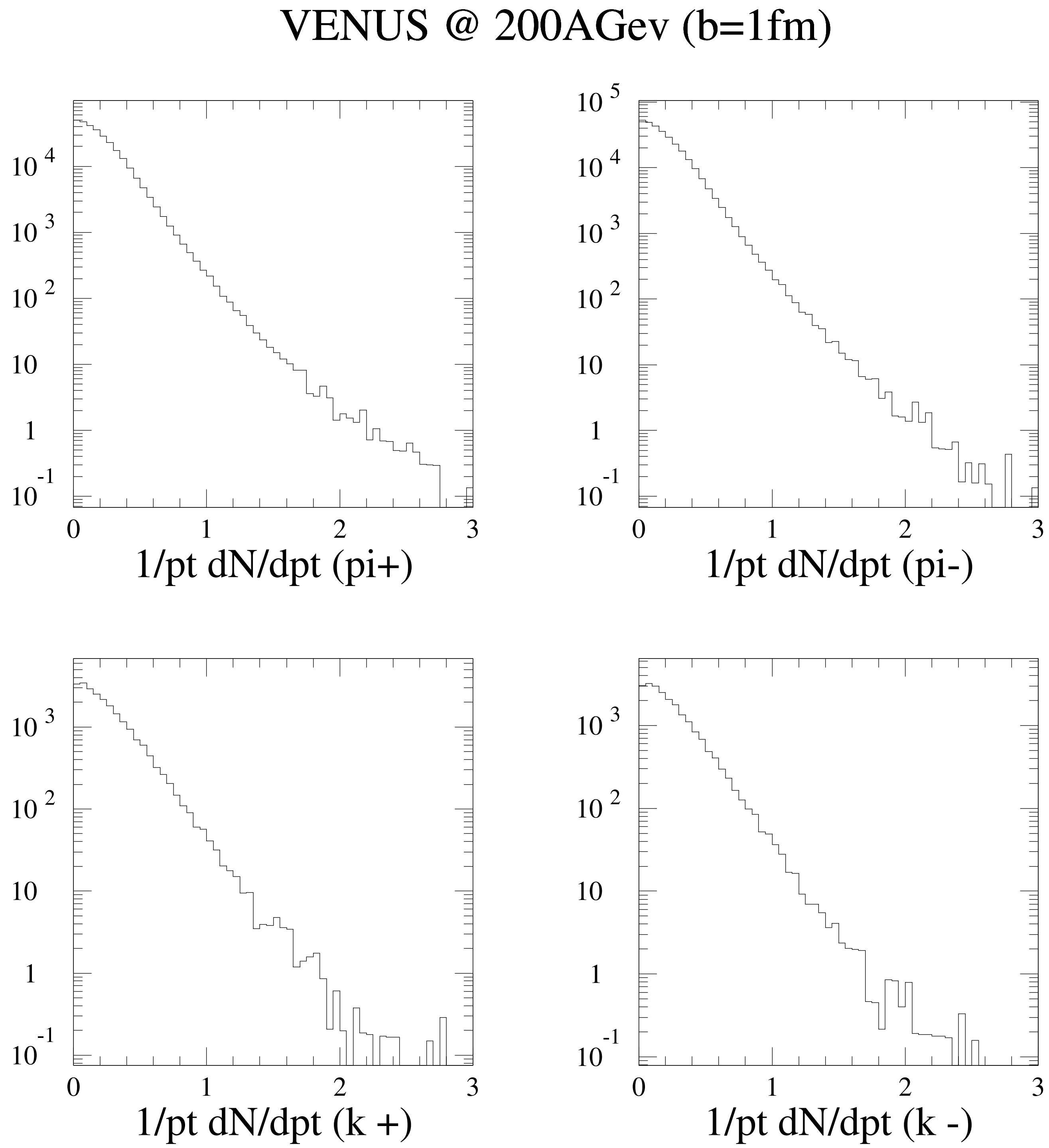

Fig. 11. VENUS p_t distributions for pions and kaons.

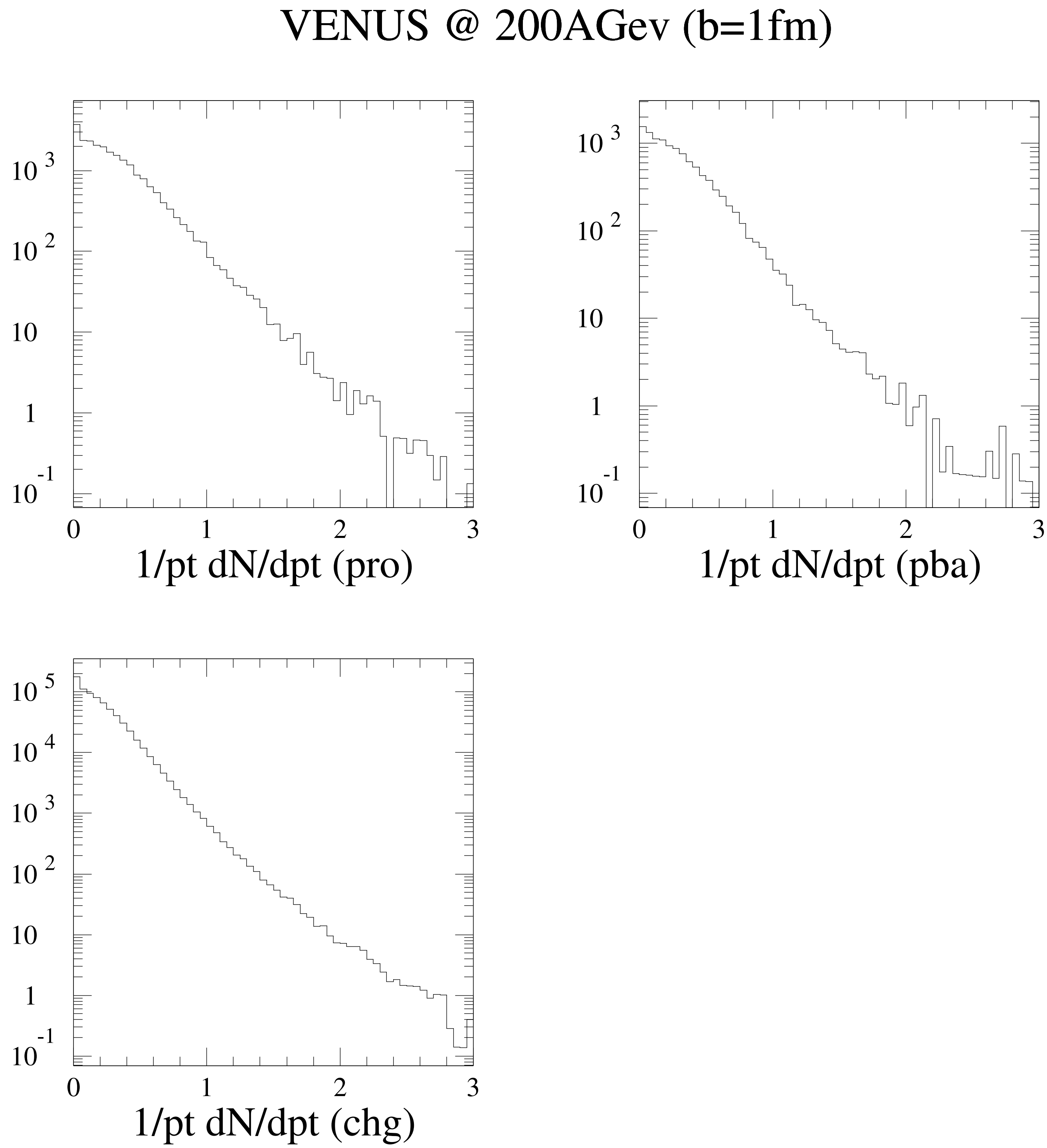

Fig. 12. VENUS p_t distributions for protons, antiprotons, and all charged particles