SDC-91-00010

DI
DT

# SDC
# SOLENOIDAL DETECTOR NOTES

## FAST SHOWER SIMULATION BASED ON GAUSS' LAW

H.-J. TROST

March 1, 1991

The submitted manuscript has been authored by a contractor of the U. S. Government under contract No. W-31-109-ENG-38. Accordingly, the U. S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U. S. Government purposes.

ANL/HEP/TR-91/17
SDC-91-00010
March 1, 1991

# Fast shower simulation based on Gauß's law*

HANS-JOCHEN TROST

*Argonne National Laboratory*
*– High Energy Physics Division, Bldg.362 –*
*9700 S. Cass Avenue, Argonne, IL 60439–4815, U.S.A.*

## ABSTRACT

I have developed a procedure for fast shower simulation on the basis of Gauß's law, applied in analogy to charge distributions and related vector fields of electromagnetic interactions. The nature of a shower (i.e. electromagnetic or hadronic), a homogeneous medium for its development, its starting point, direction and energy content are needed to be specified. Then the contribution from the shower to the total signal in any given volume made up of the given medium can be calculated as an integral over the surface of that volume using a precalculated vector field corresponding to the shower distribution. Thus the integration problem is reduced from three to two dimensions. The quality of the shower simulation is determined by the accuracies of the vector fields and of the surface integration. This technique is shown to yield acceptable results for typical sizes of calorimeter cell structures.

*Work supported by the U.S. Department of Energy, Division of High Energy Physics, Contract W-31-109-ENG-38

## 1. Motivation

The simulation of events at high energies like e.g. at the SSC presents a major challenge. The straight method of stepping through the full development of showers initiated by energetic particles would take huge amounts of computing time even for a single event [1], about one day of VAX 780 time or more. A number of schemes for treating shower development in some approximating fashion have been developed like shower library approaches [2–4] and parametrizations [5–9]. In this note, I present a new procedure belonging to the parametrization class for approximate shower development. The basic idea is to take predetermined shower shapes, in the form of parametrizations, experimental or simulated data, and convert them into vector fields, in analogy to the vector field for a charge distribution in the theory of electromagnetic interactions. These fields allow to apply Gauß's law for transforming integrals of deposited energy over detector cell volumes into surface integrals involving the vector fields [10]. The gain expected for this approach is that it should be noticeably faster than any threedimensional integration while maintaining a comparable accuracy. A drawback will be that fluctuations have to be put in after the integration (threedimensional approaches like e.g. the one used by the H1 collaboration [9] can insert fluctuations "semi-microscopically" during the process of integration). The gain in speed is being paid for by having to calculate the vector fields ahead of time which by itself is a threedimensional integration procedure for two vector components (the third being given by rotational symmetry).

## 2. The underlying formalism

The starting point is a distribution of deposited energy

$$f(z, r; X_0, r_M, \lambda_I) \tag{1}$$

in some homogeneous material, possibly or probably an artifical, averaged one standing for a real calorimeter, characterized by a radiation length $X_0$, a Molière radius $r_M$, and an interaction length $\lambda_I$. The coordinate system is cylindrical with $z$ along the central axis of the shower distribution, $r$ the radius off that axis in the plane transverse to it, and $\phi$ the polar angle around the axis (it does not appear in the distribution (1) as rotational symmetry is assumed). I will assume by default that the

shower origin coincides with the origin of the coordinate system. Using the analogy to a distribution of electric charge, we define a vector field $\vec{F}(z_F, r_F)$ related to the distribution (1) by Gauß's law,

$$\int\int_V\int f(z,r;X_0,r_M,\lambda_I) r\, d\phi\, dr\, dz = \oint_{\delta V} \vec{F}(z_F,r_F)\cdot\vec{n}(S)\, dS \quad , \tag{2}$$

where the integral on the left side of the equation runs over a finite volume $V$, e.g. a calorimeter cell, and the integral on the right side over the surface $\delta V$ of that volume. The boundary condition that for a sufficiently large volume both integrals should give the total energy (or charge in the original picture) contained in the distribution, leads to the representation

$$\vec{F}(z_F,r_F) = \int\int\int_{all\ space} \frac{1}{4\pi} f(z,r;X_0,r_M,\lambda_I)\cdot\frac{(z_F,r_F,0)-(z,r,\phi)}{|(z_F,r_F,0)-(z,r,\phi)|^3}\, r\, d\phi\, dr\, dz \tag{3}$$

for the vector field*. As the vector field exhibits rotational symmetry like the underlying distribution, the specification of a $\phi$ coordinate fixed at zero in the integrand and omission in the symbol $\vec{F}(z_F, r_F)$ does not restrict the validity of eqs.(2, 3).

## 3. Design of the procedure

First, I have to pick distribution functions (1): The basic functional form of the distributions used for electromagnetic showers

$$f(x,t;\alpha,R) = \frac{1}{\Gamma(\alpha)} x^{\alpha-1} e^{-x} \cdot \frac{2R^2}{(t^2+R^2)^2} \tag{4}$$

is taken from the H1 collaboration [11]. It is completed as

$$f_{e.m.}(z,r) = E_{inc}\ f(\beta_{e.m.} z/X_0, r/r_M; \alpha_{e.m.}, R_{e.m.}) \tag{5}$$

where the starting point of the shower of incident energy $E_{inc}$ is the origin of the $(z,r)$ plane and the $z$ axis along the shower direction, and the three parameters remaining are set as

$$\begin{aligned}
\alpha_{e.m.} &= 2.314 + 0.549 \cdot ln(E_{inc}) \\
\beta_{e.m.} &= 0.404 + 0.015 \cdot ln(E_{inc}) \\
R_{e.m.} &= [0.099 + (0.061 - 0.0047\, ln(E_{inc})) \cdot z/X_0]^2 \quad .
\end{aligned}$$

*The component notation for the relevant space points has been used to make their relations more explicit. The vector field itself, as calculated, would have to be written as $\vec{F}(z_F, r_F) = (F_Z(z_F, r_F), F_R(z_F, r_F), 0)$ in that same notation.

Note that the radial parameter depends on the longitudinal coordinate**.

For hadronic showers, I fitted distributions

$$f_{had.}(z,r) = E_{inc}\ f(\beta_E z/\lambda_I, \beta_H z/\lambda_I, r; \alpha_E, \alpha_H, \sigma_N, \sigma_W) \tag{6}$$

to data from ZEUS test measurements [12, 13]. The functional form

$$\begin{aligned} f(x_E, x_H, r; \alpha_E, \alpha_H, \sigma_N, \sigma_W) &= \\ = \left[ w \frac{1}{\Gamma(\alpha_E)} x^{\alpha_E - 1} e^{-x_E} + (1-w) \frac{1}{\Gamma(\alpha_E)} x^{\alpha_H - 1} e^{-x_H} \right] &\cdot \\ \cdot \frac{1}{2\pi(c\sigma_N^2 + (1-c)\sigma_W^2)} &(c \cdot e^{-\frac{r^2}{2\sigma_N^2}} + (1-c) \cdot e^{-\frac{r^2}{2\sigma_W^2}}) \end{aligned} \tag{7}$$

follows Böck et al. [6] for the longitudinal shape and uses a superposition of two Gaussian functions for the radial shape (similar to CDF [7]). The radial width parameters $(\sigma_N, \sigma_W)$ do again depend on the longitudinal position. The parameters are calculated as

$$\begin{aligned} \alpha_E &= 3.8747 + 0.3893 \cdot ln(E_{inc}) \\ \beta_E &= 0.1712 + 0.0309 \cdot ln(E_{inc}) \\ \alpha_H &= 1.1941 + 0.0410 \cdot ln(E_{inc}) \\ \beta_H &= 1.1742 - 0.0933 \cdot ln(E_{inc}) \\ w &= 0.16 \\ \sigma_N &= 0.08 \cdot \lambda_I + 0.012 \cdot z \\ \sigma_W &= 0.41 \cdot \lambda_I + 0.04 \cdot z - 0.0085 \cdot \lambda_I \cdot \ln(E_{inc}) \\ c &= 0.9 - 0.09 \cdot \frac{z}{\lambda_I} + 0.0085 \cdot ln(E_{inc}) \ . \end{aligned}$$

Next I select a set of energies and a method for numerical integration and evaluate the vector field at a number of points suitable for later use (e.g. by interpolation). My current choices for energies are every decade from 0.1 $GeV$ to 10 $TeV$ for electromagnetic showers, and 1, 10, 100, 2500, and 50000 $GeV$ for hadronic showers. The

**The H1 paper [11] has an additional factor $r/r_M$ multiplied into $f_{e.m.}$ and is missing the factor of 2; I take the former to be part of the integration differential $r\,d\phi\,dr\,dz$ and carry the appropriate scale factor along, and I place the factor 2 in there to normalize $f_{e.m.}(z,r)$ to the incident energy $E_{inc}$.

integration is done in cylindrical coordinates using the trapezoidal rule. Although simple in principle, the practical layout and execution of this integration requires a lot of attention to control the effect of the numerical truncation error (mathematical, not of the number representation in the computer); a more detailed technical description of my approach to this problem and a guide to my programs are given in the appendices of this note. At this point it is sufficient to note that I calculate the vector fields (3) on all (discrete) points of a number of grids layed out to suitably cover a large region in the $(z_F, r_F)$ plane.

To allow the evaluation of the vector fields at any arbitrary point in space, I fit locally (i.e. on grids of size $4 \cdot 4$ points) quadratic polynomials for the longitudinal and radial components of the vector fields separately, the latter multiplied by an extra factor $r_F$ as rotational symmetry makes the radial component vanish for $r_F = 0$. The coefficients of the polynomials are stored on a file and read completely into memory when an application program initializes the vector field evaluation routines.

The final piece needed is a routine to perform the integral over the surface of a tower volume. For this purpose, I use two routines below, INTGRT written by L.E. Price for use in the simulation program ANLSIM [14], and a routine DPIBOX I have written for performance tests of the Gauß-law method. Both routines handle boxes of possibly nonrectangular shapes (with four corners for each of six sides expected though) by performing integrals over each of the six sides. In INTGRT, each side is split further into two triangles which are independently integrated over. A predetermined number of points, normally 8, up to 32 if the starting point of a shower is nearby, is distributed homogeneously across the triangle. The integral is then determined according to a mid-point rule. In DPIBOX, each box side is integrated directly using the trapezoidal rule* in two dimensions.

*A profitable use of any higher order rule for this purpose would require many more points for evaluation, for the same reasons as explained for the vector fields in appendix A.

## 4. Performance evaluation

The first step to evaluate the performance of the Gauß-law method is to check if the vector fields can be reasonably well calculated and interpolated with the techniques outlined above. For this test I used two variants of an exactly calculable model: a constant density inside a sphere, and a density decreasing linearily with radius from the center of a sphere to its surface. The knowledge of the analytical form of the vector field can be used to calculate the total integral of the distribution from vector field values at any single point. I have achieved relative deviations of the reconstructed values from the input of up to 0.37% for field points inside the test spheres and up to 0.08% outside, with deviations of the mean values over 164 points inside of $6.4 \cdot 10^{-5}$ and over 91 points outside of $1.7 \cdot 10^{-5}$. The field component transverse to any radius out of the center of the sphere has to vanish; parallel to the axes of the cylindrical coordinate system used in the present work on lines emerging from the center of the spheres, the absolute values for this component come out smaller than $10^{-10}$ times the maximum values of the fields anywhere (Fig. 1). As the distributions $f(\vec{r}) = 1$ and $f(\vec{r}) = (R - |\vec{r}|)/R$ are much more well-behaved than the physically relevant ones [5–9], I can conclude only that the present approach can achieve a reasonable accuracy in principle but I cannot estimate the actual errors in the physical model from the spherical models.

To evaluate the performance on the actual physical application, I calculate longitudinal distributions "backwards" from the vector fields as follows: With each of the two integration routines, INTGRT and DPIBOX, I determine the energy deposition in a thin slab of $0.25X_0$ ($0.05\lambda_I$) thickness for electromagnetic (hadronic) showers, in $4 \cdot 4$ tiles sufficiently wide to contain the whole shower, through $30X_0$ ($6\lambda_I$) in depth and compare the result both to the input distributions and the available data (Fig. 2, Fig. 3). There are two kinds of limitations to the quality of the calculations appearing: First, the showers are very narrow with steep slopes in their front section. This will create problems with discretized evaluations like the numerical integration procedure employed at both levels where integrations are used, the calculation of the vector fields from the distribution, and the reconstruction of the distribution from the vector fields. For electromagnetic showers, integrations around the starting point, within almost one radiation length, become totally unreliable. For hadronic

showers, the ratio output over input varies in a ±20% band over a bit more than the first interaction length. Also, the transverse distributions do not reproduce as well as in the electromagnetic case because of the two-component structure with a large narrow piece at the center of the shower. The second limitation comes from the use of quadratic polynomials locally to interpolate the vector field components. These do not join continuously in neighboring regions, so the very fine slabs used to look at the longitudinal distributions show spikes across the borderlines between two polynomials. The physical applications [14] will use much larger dimensions in any direction compared to the thickness of the slabs used in this particular comparison, suppressing the effect of the discontinuities almost completely. This is exemplified in the quite good agreement of the calculations with experimental data shown for the transverse distributions in Fig. 2, 3.

*Tab. 1: Energies from the Gauß-law method for the test setups*

| | *electrons* | | *pions* | |
|---|---|---|---|---|
| | 6 *GeV* | 20 *GeV* | 10 *GeV* | 30 *GeV* |
| longitudinal | | | | |
| INTGRT | 5.467 | 18.284 | 9.291 | 28.040 |
| DPIBOX | 5.468 | 18.426 | 9.028 | 27.346 |
| lateral | | | | |
| INTGRT | 5.547 | 18.693 | 9.204 | 27.937 |
| DPIBOX | 5.446 | 18.347 | 9.129 | 27.663 |

Using either the longitudinal or lateral integrations shown in the figures one can sum up the energies in the single elements to reconstruct the total shower energy. The values come out at 7 – 10% below the input values (Tab. 1). Thus overall, the Gauß's law method produces acceptable results for typical sizes of calorimeter cell structures.

## 5. Summary and outlook

I have developed a scheme to make use of Gauß's law to reduce the integration of electromagnetic and hadronic showers over the volume of a calorimeter cell to a surface integral. The energy distributions are reasonably reproduced. The method allows to simulate full events for the SSC environment with manageable efforts and acceptable quality of the results unless the determination of the cell geometries becomes prohibitive. In practice, it has proven to deliver a reasonable performance in studies [15] for the Letter of Intent [16] of the Solenoid Detector Collaboration using this method within the detector simulation program ANLSIM [14].

To significantly improve the method over its present status, almost all technical elements may have to be revised. The choices for new methods are not completely obvious however: methods used for electrostatic and magnetic field calculations [17, 18] suffer from the assumption of locally constant fields and/or source distributions which is very hard to make for the shower distributions of concern in the present work. The output obtainable from the field determination then strongly influences the possible choices for the surface integrations. A further complication in this second step are the possibly quite irregular shapes of the surfaces to be dealt with, making already the Gaussian integration rules almost impossible to apply.

## ACKNOWLEDGEMENTS

I am pleased to thank Dr. L.E.Price for suggesting the topic and Dr. R.E.Blair for a critical reading of the manuscript. Both of them also contributed to the debugging and application of the programs, and they provided a lot of the real hard field testing of the application.

## Appendix A. The integration procedure for obtaining vector fields

As mentioned in the main text, I use the trapezoidal rule to evaluate the vector fields of eq. (3):

$$\int_a^b f(x)\,dx = \frac{h}{2}\sum_{i=1}^{n}[f(a+(i-1)h)+f(a+ih) \;-\; \frac{h^3}{12}f''(\xi_i)] \qquad (8)$$
$$h = \frac{b-a}{n}$$

with $\xi_i$ chosen out of the open interval $]a+(i-1)h, a+ih[$ suitably in accordance with the mean value theorem for integration [19]. The derivative term in eq. (8) is dropped for numerical evaluation of the integral(s) and thus describes the truncation error inferred by this method. It is apparent that the trapezoidal rule will integrate linear polynomials exactly. The Simpson rule integrates even cubic polynomials exactly, without any significant added complication to the trapezoidal rule. It turned out (by numerical experiments), however, that the Simpson rule would perform better than the trapezoidal rule only for a "sufficiently" large number of integration points, and the number of points required for the present problem would lead to excessively huge cpu time requirements. The choice of an integration formula with equidistant points over e.g. Gauß integration (i.e. "optimally" chosen points and weights with some weight function) is made for the sake of computing economy. For single integrals ("single" meaning in the present context the integral of eq. (3) for a single field point) and a given degree of polynomials that are to be integrated exactly, Gauß integration formulae require fewer points for evaluating the integrand than Newton-Cotes formulae with equidistant points of which the trapezoidal and Simpson rules are examples for first and second degree. I have found, however, that with the $1/\vec{r}^2$ pole at every field point, I still can use *one common set* of equidistant integration points to obtain all vector field values because the integration points can be placed with a minimal symmetry with respect to the poles. This specification is an empirical lesson from the tests with the analytically solvable spherical models, it is needed to cancel destructive effects of discretizing the integrand of eq. (3) with the $1/\vec{r}^2$ factor. The integral is finite, of course, because the contribution from any neighborhood of the field point actually scales not only with the $1/\vec{r}^2$ dependence of the Coulomb like "force" but also with the $\vec{r}^2$ dependence of the contained amount of energy "density" in the same neighborhood. Thereby I can perform the integral (3) simultaneously

for all field points, evaluating the energy distribution (1) only once in common for all integrals. Using a Gauß type rule would require the integration points to be adjusted for each pole which implies a recalculation of the energy distribution (1) for all integration points. The gain in the number of points per integral (50% for degree 1, 33% for degree 2 are the extremes on opposite ends) is more than offset by the multiplication of the number of evaluations of the energy distributions.

For the following, remember that the coordinates $(z, r)$ refer to the space in which the shower distribution (1) is defined while $(z_F, r_F)$ are the coordinates for the vector fields (3). Along the $z$ direction, the integration points are chosen with constant spacings of up to three different sizes which are related by one odd integer factor. The highest density is used near the origin, the others successively with increasing $z$. The odd integer factor guarantees that the field points never coincide with integration points, avoiding the $1/\vec{r}^2$ pole. In radial direction, variable step sizes are created from equidistant ones on an artificial axis $s$ according to the prescription

$$\begin{aligned} f_r(s,d) =& s^4 + d \cdot s \quad for\ s < s_c = \sqrt[3]{(1-d)/4} \\ =& s + c \qquad for\ s \geq s_c \end{aligned} \tag{9}$$

where the coefficient $d$ determines the asymptotic ratio of step sizes at the center and at large values of $s$. The location of the break point between the two functional forms is given by the requirement that the $4th$ order polynomial has the same value of the derivative

$$\frac{df_r}{ds}(s_c) = 1$$

as the linear function. The constant $c$ is then fixed by connecting the two pieces of the function continuously. The transformation to the $r$ coordinate axis is completed by scaling $f_r$ to map $s = 0$ onto $r = 0$ and the connection point $s_c$ onto $3 \cdot R$ where $R$ is e.g. the radial parameter appearing in the radial shower shape functions (5). The upper integration limit is set at a few Molière radii $r_M$ or interaction lengths $\lambda_I$ beyond the transition point $3 \cdot R$. Finally, for the angular coordinate $\phi$, a fixed number of points independent of the radial coordinate is used. For the actual calculations presented below, the numbers of integration points used are 201 along $z$, 101 along $r$, and 100 along $\phi$. This last number has to be even and $\phi = \pi/2$ has to be one of the points to produce the necessary cancellation of the radial field

components on the shower axis. (The first of those points is then actually re-used as the 101st to close the circle, i.e. the number of *different* points has to be even.)

*Tab. 2: Lattice specifications for vector fields*

| | *e.m. showers* | | *had. showers* | |
|---|---|---|---|---|
| | *points* | *range* | *points* | *range* |
| $z_F$ | 31 | $-0.525 \rightarrow +2.475\ X_0$ | 82 | $-0.0115 \rightarrow +0.0695\ \lambda_I$ |
| $r_F$ | 121 | $0 \rightarrow 2.4\ r_M$ | 31 | $0 \rightarrow 0.03\ \lambda_I$ |
| $z_F$ | 58 | $-0.525 \rightarrow +5.325\ X_0$ | 43 | $-0.0585 \rightarrow 0.3195\ \lambda_I$ |
| $r_F$ | 61 | $0 \rightarrow 4.5\ r_M$ | 31 | $0 \rightarrow 0.18\ \lambda_I$ |
| $z_F$ | 58 | $-6.1 \rightarrow +16.7\ X_0$ | 49 | $-0.29 \rightarrow +1.63\ \lambda_I$ |
| $r_F$ | 61 | $0 \rightarrow 6\ r_M$ | 31 | $0 \rightarrow 18\ \lambda_I$ |
| $z_F$ | 79 | $-26.5 \rightarrow +51.5\ X_0$ | 73 | $-5.95 \rightarrow +15.65\ \lambda_I$ |
| $r_F$ | 46 | $0 \rightarrow 15\ r_M$ | 49 | $0 \rightarrow 9.6\ \lambda_I$ |

The field points are chosen on rectangular lattices in the $(z_F, r_F)$ space (Tab. 2). The integration points are placed between field points such that they are covering the neighborhood of every field point nearly symmetrically, for the reasons explained earlier. The odd factor relating the different step sizes in $z$ guarantees that the field points fall always into the center of an integration interval* for all step sizes if they do for one of them. The lattices are chosen such that a transition point in the integration step sizes will never occur in the range where a lattice will later be used, i.e. the transition point is either "hidden" under a smaller lattice or is located outside the lattice by a few steps (more than the value of the step size factor). The purpose of the variable step size strategies is to cope with steep and rapidly changing slopes in the energy distributions (1). An example for the resulting components of the vector fields is shownin Fig. 4.

*in this z coordinate; the radial pattern is different and has to have nearly identical point locations in successive radial rows near the center of the shower only, leaving just enough room for the "symmetry-breaking" effect of the variable step size technique of eq. (3).

As mentioned before, the simplicity of this integration technique allows to do the integrals of eq.(3) simultaneously for all points of a given lattice. For the specifications in Tab. 2, there are between 1333 and 3751 parallel integrations to be done. This piece is by far the most cpu-intensive part of generating the tools for the shower simulation: One set of electromagnetic and hadronic vector fields for a single material takes 44 $cpu - hours$ on the CRAY X-MP/14 at Argonne wherein vectorization has gained more than a factor of 10 in speed for program execution. I have measured the execution speed to be about 137 $MFLOPS$, making the vector field determination one of the highest performing prorams currently in use on that installation.

## Appendix B. Guide to generating vector fields

Below I list a plain ASCII file containing - for users on the Argonne site - instructions for use of my vector field generation programs. For users elsewhere, some fairly trivial translations will be needed (logical names, CERNLIB setup, mathematical libraries). The file is separately accessible on ANLHEP in

```
$9$DUB10:[SSC.TROST.TEX]DEPFIELD.DOC
```

```
                                        H.-J.Trost, ANL HEP, 01 March 1991

Running instructions for creating and fitting vector fields:
============================================================

The following directories are used:

$9$DRA1:[TROST.SSC]          logical name HJTSSC
$9$DRA1:[TROST.UTIL]         logical name HJTUTIL (possibly not used)
ANLHEP$DUA10:[SSC.TROST]     logical names HJTSCR, SCR (equivalent),
                                 used for work files and log files

Rules for adapting to your environment (on ANLHEP):

1) define a symbol
                    YPATCHY := "RUN CERN$EXE:YPATCHY"

2) in any .COM file, change the lines

                    $ YPATCHY - fortranfile cradle ...

   to two lines (note the leading blank in the second line)

                    $ YPATCHY
```

- fortranfile cradle ...

Creating vector fields on VAX (far too slow for production):
---

| | | |
|---|---|---|
| Job to run: | HJTSSC:DEPFIELD.COM | |
| Input | HJTSSC:DEPFIELD.CRA | for code production by YPATCHY |
| | HJTSSC:DEPFIELD.DAT | data input for program run |

The source code is in HJTSSC:DEPFIELD.CAR. This file is structured by cards

```
+PATCH,patchname
```

and substructured by cards

```
+DECK,deckname
```

Lines of the form

```
+SEQ,sequencename.
```

are calling in pieces of code defined further up with header lines

```
+KEEP,sequencename.
```

The VAXese equivalences are the statement

```
INCLUDE 'sequencename.INC/LIST'
```

for +SEQ and the file

```
sequencename.INC
```

for the piece of code headed by a "+KEEP" line and terminated by any other "+xxxxx" line where "xxxxx" is anything but blank or SEQ.

Lines of the form

```
+SELF,IF=xxxxx.
. . .
+SELF<,IF=yyyyy>.
```

enclose code that is used only when an instruction

```
+USE,xxxxx.
```

appears in the .CRA file. Rules for the "xxxxx" labels are:

like all names used by YPATCHY, up to 8 characters are allowed (exceeding characters are ignored silently)

a minus sign means "not", i.e. there is NO "+USE,xxxxx." in the cradle

OR is accomplished as "+SELF,IF=xxxxx,yyyyy."

AND is accomplished as "+SELF,IF=xxxxx,IF=yyyyy." i.e. by repeating the "IF=".

The cradle file (DEPFIELD.CRA in this case), contains lines

```
+USE,patchname,T=EXE.
```

which cause the corresponding patch to be written into the FORTRAN code file with all conditional clauses and sequence calls resolved. In the form

```
+USE,patchname.
```

a patch is made available for e.g. pulling a +KEEP sequence out. (Side remark: A patch name can then also be "ab"used as a logical flag e.g. in +SELF lines.)

What to look for in DEPFIELD.CAR: There is a lot of stuff kept together under one roof. You will need (cf. DEPFIELD.CRA):

the patch DEPCOMS to give you all +KEEP sequences
the patch DEPUTIL to give you all "utility" routines
the patch FIELDVOL to give you the vector field generation code.

The code is pretty extensively documented inline. To use the e.m. distributions of H1 (deck DPDENE in patch DEPUTIL) you to "+USE" the flags DEPSETS, H1 and EMSET. For the hadronic distributions (deck DPDENH in DEPUTIL) "+USE" DEPSETS, H1 and HADSET. To create your own routines DPDENE or DPDENH, proceed as follows:

a) put lines

```
+USE,P=DEPUTIL,D=DPDENE,T=INHIBIT.
+USE,P=DEPUTIL,D=DPDENH,T=INHIBIT.
+USE,MYCODE,T=EXE,
```

in after the "+USE,CRA*,T=EXE." and

b) put a section

```
+PATCH,MYCODE.
+DECK,DPDENE.
      SUBROUTINE DPDENE(XR,N,F)
          your version
+DECK,DPDENH.
      SUBROUTINE DPDENH(XR,N,F)
          your version
```

in between the "+PAM,LUN=11,..." and "+QUIT." lines in DEPFIELD.CRA.
NB: If you use substantially different functional forms, make sure that the main routine FLDVOL does not get far off track in adjusting its radial step sizes. For this purpose, it "knows" something about the radial form and governing parametric dependences.

c) don't forget to use the common block /DENPRC/ through the sequence named DEPDENSP in those routines. You need not normalize the densities yourself

but keep a factor DENORM in calculating them; the normalization will be done numerically by the main program FLDVOL if you include the line "+USE,FINDNORM." in your cradle.

Lines in the cradle beginning with a "C" are comments and will be ignored.

The data input attempts to be selfexplanatory by having always one comment line followed by a data line. Data lines are always in fixed format. If you do not "+USE" either of the flags VARYRPOINT, DEPTEST, DEPTESTO you will get 151 points each on the z and r axes used for the density, i.e. 150 intervals. Similarly, when you specify the numbers of points for the vector fields as "nz" and "nr" then you will have nz-1 and nr-1 intervals. Make sure that the z points of the density and the vector fields do never coincide. Also, nr*nz is limited to INZR=101**2 (if one is larger, the other must be small enough to meet the limit for their product). Either of the two, nr and nz, should be chosen as 3*m+1 so all points are used in the quadratic fits (which will be done on 4*4 grids). The flag VARYRPOINT offers a variation on handling the radial integrations, and under the master flag DEPTEST, the flag DEPTESTO allows to run quick and dirty tests that can live with 21 integration points each in z and r, 10 points in phi, and nz*nr field points no more that 21**2.

Fitting vector fields on VAX:
---

DEPFIELD will leave a file FOR021.DAT in the HJTSCR scratch directory containing the discretized vector fields.

The job FITFIELD.COM with a cradle FITFIELD.CRA (both in HJTSSC) will do the fitting using MINUIT. (Here, HJTUTIL can come into play.) Note that MINUIT will fill your log file heavily with a message on every fit of which there will be a few thousand per run if you were doing a production run. Some steering is taken from FIELDFIT.INP (!). The vector fields are taken from the directory HJTSSC file H1EMFIELD.DAT unless you specify /PAR with the SUBMIT command. Thereby you can request the file "HJTSSC:yourname.DAT" via "/PAR=yourname". Perhaps your best choice is changing the handling of this file in the job. The resulting fitted field parameters will appear on unit 22 in the file

HJTSCR:yourname.PAR

given a field input file "yourname.DAT". (Our DELIVERYxxx.POL files are compilations of these .PAR files.)
The file FITFIELD.INP you probably take best the way it is now.

The source code is again in DEPFIELD.CAR. The common and utility patches are used, and the deck FLDFIT plus some more of the patch FIELDFIT. The flag QUADFORM turns on the local polynomial ansatz; else a global function ansatz will be used which currently may not be in working order. Forget all other

decks that contain main programs in the patch FIELDFIT. (You need the FUNCTIONs and SUBROUTINEs, of course.)

Running on a CRAY:
---

DEPFIELD is installed on the CRAY X/MP-14 at Argonne and the CRAY Y/MP-432 at SCRI/Florida State University. The master source is not available there, so you have to run PATCHY ahead of time on ANLHEP to obtain the FORTRAN sources and ship those over.

To get your programs onto a CRAY, use these command files:

a) for DEPFIELD (produces DEPFELD.FOR for the CRAY): HJTSSC:DEPCRAY.COM, along with an adjusted cradle HJTSSC:DEPCRAY.CRA ;

b) for FITFIELD (produces FITFELD.FOR for the CRAY): HJTSSC:FITCRAY.COM, along with an adjusted cradle HJTSSC:FITCRAY.CRA ;
this job will access a file HJTSSC:ZXSSQ10.CAR in addition to DEPFIELD.CAR which takes care of hooking up some IMSL fitting routines.

Both command files make use of HJTSSC:CRAYCODE.COM to do the actual work. (Check that one to make sure that its use of YPATCHY conforms with your definition.)

The shipping to (and from later on) the CRAY's is easiest done with FTP in ASCII mode.

On the CRAY's you go into one of the areas

| | |
|---|---|
| /u/trost/vector | in the case of the SCRI YMP |
| /n2/trost/vector | in the case of the ANL XMP |

and steal e.g. the files

| | |
|---|---|
| emfeld.job | equivalent to DEPFIELD.COM plus FITFIELD.COM |
| emfeld.inp | equivalent to DEPFIELD.DAT |
| fitemfl.inp | equivalent to FITFIELD.INP |
| deptran.for | for binary to ASCII conversion |

(feel free to look at the other files, in particular "hadfeld,xxx" and "fithdfl.xxx" also) where you change in the ".job" files all the references to my directories appropriately to yours. The vector fields appear as "emfeld.dat" (usually only in the work area but you can go and save them; they are LARGE) and the fitted fields as "emfeld.pol" (in binary form) and "emfeld.crd" (in ASCII form for shipping). Additionally, some histograms appear in "xxxx.hcd" files (HBOOK3 formatted by HWRITE) and printed outputs in files "xxxx.out". Concatenate the card image files (extension .CRD) and transform them back using "depback.job" with "depback.for" or copy them to e.g. ANLHEP where you would obtain the module (prepared with HJTSSC:DEPBACK.COM)

```
ANLHEP$DUA10:[SSC.TROST]DEPBACK.EXE
```

assign your card image file to unit 19, define unit 18 as output, and run DEPBACK:

```
$ ASSIGN yourfields.CRD FOR019
$ ASSIGN yourfields.POL FOR018
$ RUN/NODEB DEPBACK
$ DEASSIGN FOR018
$ DEASSIGN FOR019
```

The raw vector fields from DEPFIELD cannot be converted (unless you write a program to do that); they are usually so large that this does not make any sense anyway.

On the CRAY, the fitting uses IMSL routines instead of MINUIT. For this, keep the fourth number in the first line in "fitemfl.inp" set to 3.

Miscellaneous remarks:
---

Investigate in particular "fitemfl.out" if there is any occasion of 9 or more (!) error messages about number of iterations exceeded before an additional message that finally a code different from 3 (usually 0, 1, or 2) has been obtained. If that happens (i.e. 9 or more), then there is a local area where the fit may have remained screwed up.

A test program that produces plot of the longitudinal and transverse distributions like those used in the note describing the method is available with the files TESTBEAM.COM, .CRA and .INP, and their counterparts for the CRAY's named "emtest.xxx"; the code shoveling is prepared using CRAYBEAM.xxx. Combined plots including experimental data when available are done by postprocessing histogram files (HSTORE and HWRITE files are both accepted) from the test program with the procedure HJTSSC:BEAMPLOT.COM and associated input files (BEAMPLOT.INP, BEAMSCREEN.INP, etc in HJTSSC) on ANLHEP only.

If you are suspecting problems with the numerical stability of the field determination (you have to experiment a bit and try to get a feeling for how to design the lattices), the job DISFIELD ("disfeld" on the XMP, not exiting on the YMP; either with the corresponding extensions for the individual files) can be used to look at the raw vector field data in axis-parallel cuts. The field components should not show any kind of oscillatory pattern in a region where they are used. In the plots from TESTBEAM, in particular the longitudinal reconstructed distributions, you may find locally huge (or negative) values of the distribution over a limited interval. Check if that interval is in (one of) the last, highest z radial rows of polynomial grids (4*4 grids used

for fitting). If so, you can cure the problem by going into the .CRD file and changing the number of rows, i.e. reduce it, to actually be considered. This will work as long as there is a larger grid available. For the largest grid, you may have to change more numbers in the header.

A glance at the physical content:
---

The integration job DEPFIELD first normalizes the density to one and then integrates for the vector field at all requested field points simultaneously. This keeps the longest loop as the innermost one; it is fully vectorized by the CRAY compiler thus making the best use of that machine. Almost all the cpu time is spent in the integration loop nest, and even in the vectorized version that is a lot (measure it in CRAY cpu-hours rather than minutes in production mode). The coordinate system used is cylindrical; only for putting the vector field components together the cartesian components are calculated intermediately. The integrations are all done using the trapezoidal rule as described in the note. The integration directions on the radial and longitudinal coordinates are taken outside-inward to first accumulate the mostly very small terms which otherwise get lost in the rounding errors and finite precision of number representation. The densities are calculated for whole arrays for the sake of speed (that vectorizes well if you are careful). The density calculation does not take much time in the current setup, though.

The fits are done in a least-squares ansatz. The program can try up to three different strategies to obtain starting values. (That's what the fourth number in FITF(I)ELD.INP allows; a value larger than 3 does not make any sense.)

Using the vector fields in an application:
---

The TESTBEAM job shows a good example of how to use the vector fields. You have to call VFLGET (for electromagnetic showers) and HFLGET (for hadronic showers) once at the beginning of your program. For each new shower, you need to call once VFLINI (or HFLINI). Then the vector fields are obtained from calls to VFIELD and HFIELD (these are subroutines !). Inspect the code of the TESTBEAM program and the utility routines in DEPFIELD.CAR for more details. ANLSIM is another example of using the vector fields.

Good luck.

## REFERENCES


[1] S.Linn et al., Practical Shower Algorithms, Proc. Workshop on Detector Simulation for the SSC, August 24-28, 1987, ed. L.E.Price, ANL-HEP-CP-88-51, Argonne National Laboratory, Argonne, Illinois, USA 1988, p.270

[2] E.Longo and L.Luminari, Nucl.Instr.Meth. A239 (1985) 506

[3] D.Ward, OPAL internal note, CERN, Geneva 1987, unpublished

[4] R.Raja, in: Proc. Workshop on Detector Simulation for the SSC, August 24-28, 1987, ed. L.E.Price, ANL-HEP-CP-88-51, Argonne National Laboratory, Argonne, Illinois, USA 1988, p.146

[5] E.Longo and I.Sestili, Nucl.Instr.Meth. 128 (1975) 283

[6] R. Böck, T.Hansl-Kozanecka, and T.P.Shah, Nucl.Instr.Meth. 186 (1981) 533

[7] J.Freeman and A.Beretvas, Proc. DPF Summer Study Snowmass 1986, p.482

[8] H.-J.Shih and A.Kanovsky, Proc. DPF Summer Study Snowmass 1986, p.493

[9] G.Grindhammer, M.Rudowicz and S.Peters, Proc. Workshop on Calorimetry for the Supercollider, March 13-17, 1989, Tuscaloosa, Alabama, U.S.A., eds. R.Donaldson and M.D.G.Gilchriese, World Scientific Publishing Co. Pte. Ltd., Singapore, 1990, p.151

[10] proposed at ANL by Larry Price which triggered my work; I have heard claims since then that the idea had been proposed elsewhere at different times by different people, without any resulting active work on it though.

[11] G.Grindhammer, R.Mundt, and M.Rudowicz, internal note of the H1 collaboration H1-09/87-71, unpublished

[12] M.G.Catanesi et al., Nucl.Instr.Meth. A260 (1987) 43

[13] The ZEUS calorimeter group: F.Barreiro et al., DESY 89-171, DESY, Hamburg, Germany 1989, and FTUAM-EP-89-08, Universidad Autonóma de Madrid, Madrid, Spain 1989

[14] R.E.Blair, L.E.Price and H.-J.Trost, program ANLSIM, Argonne National Laboratory, Argonne, Illinois, USA 1989, version 1.04, 1990 (unpublished); H.-J.Trost, talk given at: Workshop on the Future Development of GEANT, SSC Laboratory, Dallas, Texas, Jan.10-13, 1990, unpublished; R.E.Blair, in: Workshop on Physics and Detector Simulation for SSC Experiments, SSC Laboratory, Dallas, Texas, Jan.9-19, 1990, vol.1, p.69; H.-J.Trost, ibid., vol.2, p.265

[15] R.E.Blair, L.E.Price, and H.-J.Trost, internal report ANL-HEP-TR-91-13, Argonne National Laboratory, Argonne, Illinois, USA, and SDC internal note SDC-90-00150, December 1990 (unpublished)

[16] Solenoid Detector Collaboration, G.H.Trilling et al., SDC internal note SDC-90-00151, and SSC-LOI-1, SSC Laboratory, Dallas, Texas, November 30, 1990

[17] R.F.Holsinger and Ch.Iselin, The CERN-POISSON Program Package, CERN, Geneva, Switzerland 1983 (unpublished)

[18] F.Krawczyk, Ph.D. thesis, internal report DESY-M-90-13, Deutsches Elektronen-Synchrotron, Hamburg, Germany 1990 (unpublished)

[19] see e.g.: E.Isaacson and H.B.Keller, Analysis of Numerical Methods, John Wiley and Sons, New York, New York, USA 1966, pp. 312 ff; I.N.Bronshtein and K.A.Semendyayev, Handbook of Mathematics, 3rd English edition, Van Nostrand Reinhold Company, New York, New York, USA 1985, section 7.1.2.7

## FIGURE CAPTIONS

1. Vector field components for a spherical distribution $f(\vec{r}) = (R - |\vec{r}|)/R$ centered at $(z, r) = (10, 0)$ on the $z$ and $r$ axes

2. Longitudinal and radial distributions for electrons of 6 $GeV$ and 20 $GeV$ incident energies: crosses are data (from simulation), dash-dotted lines for INTGRT, dashed lines for DPIBOX, dotted lines for input distributions (in the lateral cases integrated to better than 2 % accuracy); also shown are the ratios of the longitudinal distributions from INTGRT ("SRF") and DPIBOX ("HJT") over the input distributions.

3. Longitudinal and radial distributions for negative pions of 10 $GeV$ and 30 $GeV$ incident energies: crosses are data [12, 13], dash-dotted lines for INTGRT, dashed lines for DPIBOX, dotted lines for input distributions (in the lateral cases integrated to better than 2 % accuracy); also shown are the ratios of the longitudinal distributions from INTGRT ("SRF") and DPIBOX ("HJT") over the input distributions.

4. Longitudinal (top) and radial (bottom) vector field components for electromagnetic showers of 1 $GeV$, on the fourth lattice of Tab. 2. The vertical scales are linear but not normalized to each other, the offset of the values in the top plot from the base is the largest negative value of the longitudinal component calculated. The plane of asymptotic values marks the zero of the vertical axis.

$$f(\vec{r}) = 10 \cdot \frac{R - |\vec{r}|}{R} \quad , \quad R = 5.0 \text{ cm}$$

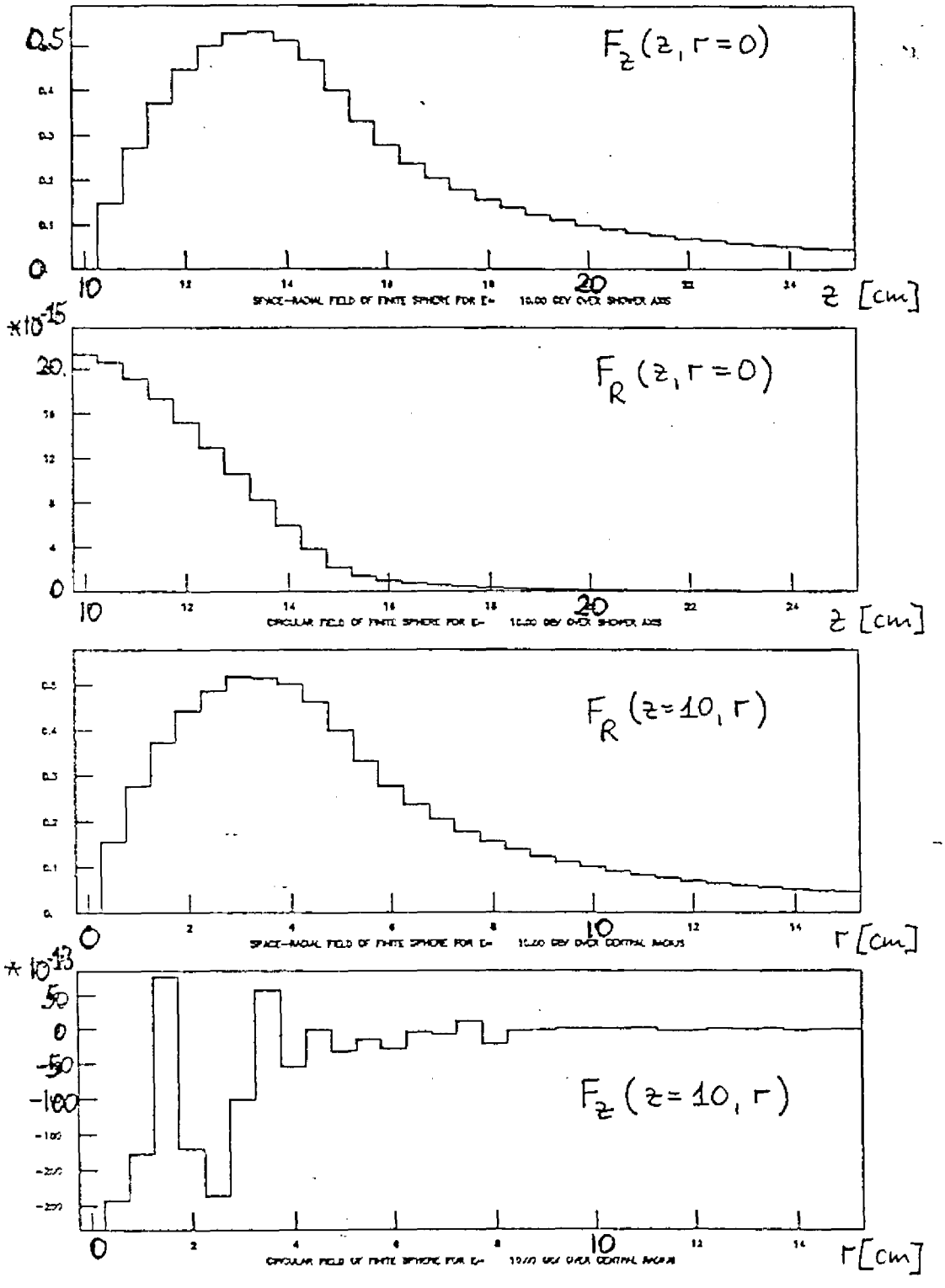



Fig. 1

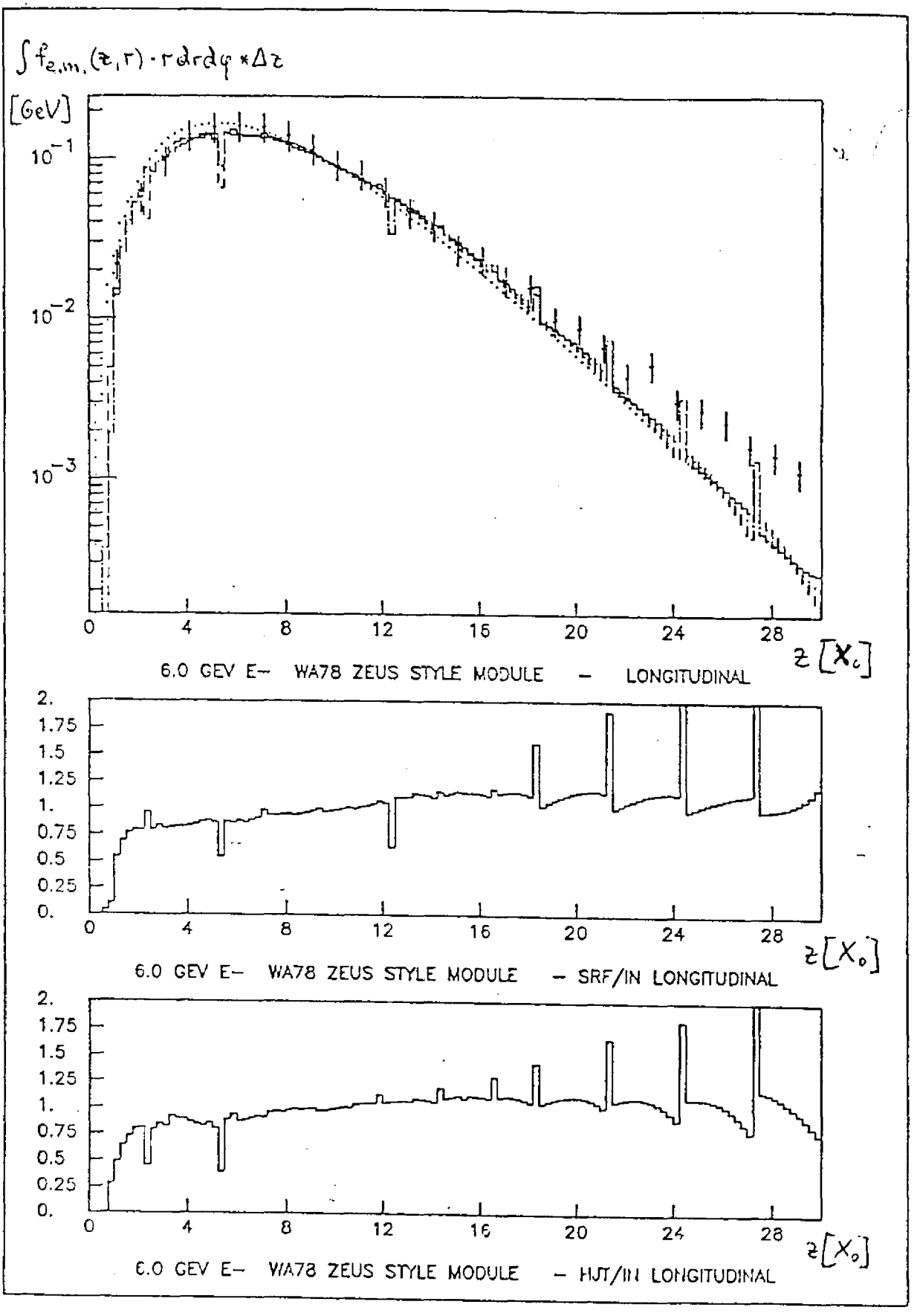

$\int f_{e.m.}(z,r) \cdot r\,dr\,d\varphi \times \Delta z$
[GeV]
$10^{-1}$
$10^{-2}$
$10^{-3}$
0 4 8 12 16 20 24 28
$z\,[X_0]$
6.0 GEV E− WA78 ZEUS STYLE MODULE − LONGITUDINAL
2. 1.75 1.5 1.25 1. 0.75 0.5 0.25 0.
0 4 8 12 16 20 24 28
$z\,[X_0]$
6.0 GEV E− WA78 ZEUS STYLE MODULE − SRF/IN LONGITUDINAL
2. 1.75 1.5 1.25 1. 0.75 0.5 0.25 0.
0 4 8 12 16 20 24 28
$z\,[X_0]$
6.0 GEV E− WA78 ZEUS STYLE MODULE − HJT/IN LONGITUDINAL

Fig. 2 (1 of 7)

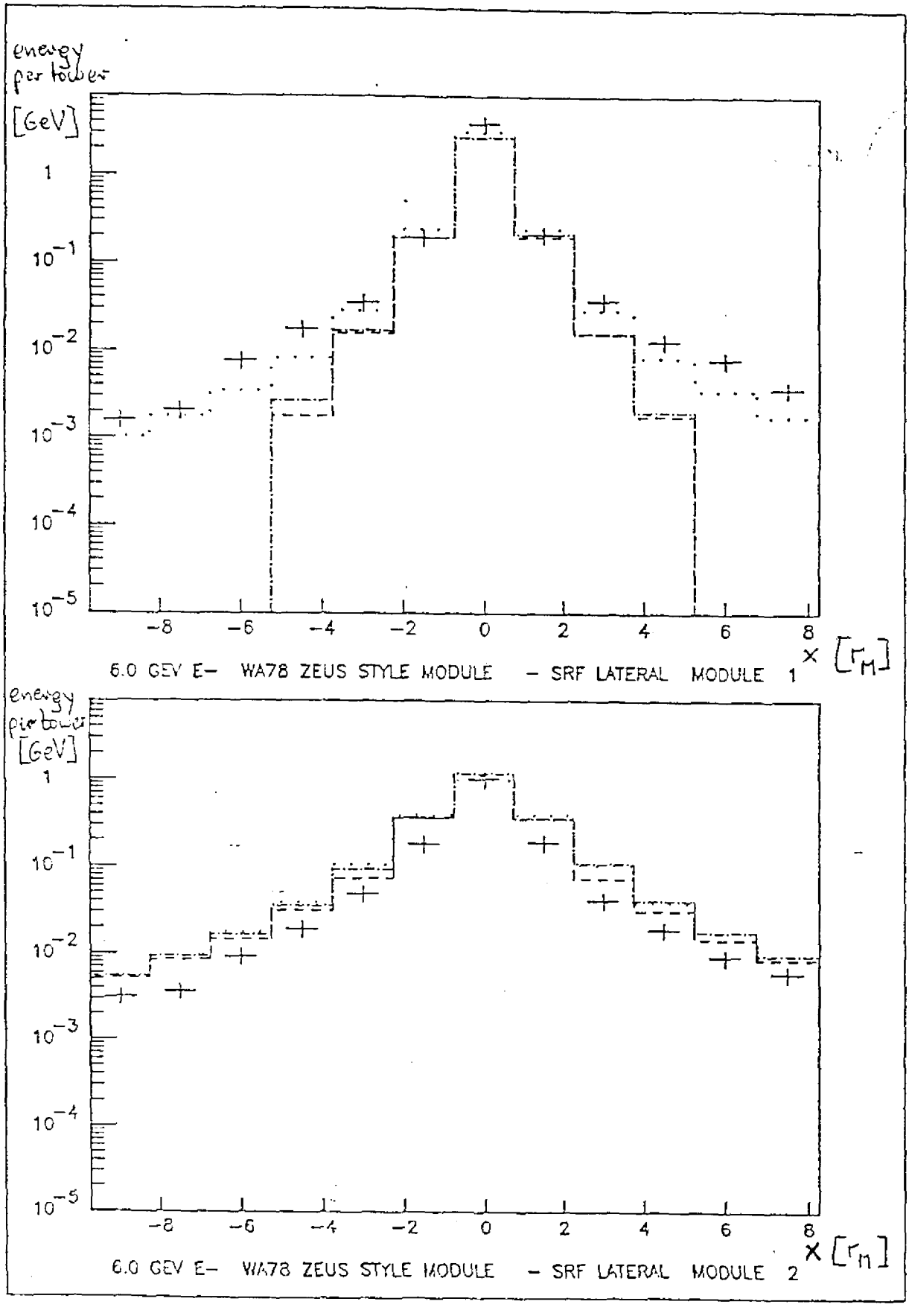

energy per tower [GeV]
1
$10^{-1}$
$10^{-2}$
$10^{-3}$
$10^{-4}$
$10^{-5}$
−8 −6 −4 −2 0 2 4 6 8
x [$r_M$]
6.0 GEV E− WA78 ZEUS STYLE MODULE − SRF LATERAL MODULE 1
energy per tower [GeV]
1
$10^{-1}$
$10^{-2}$
$10^{-3}$
$10^{-4}$
$10^{-5}$
−8 −6 −4 −2 0 2 4 6 8
x [$r_M$]
6.0 GEV E− WA78 ZEUS STYLE MODULE − SRF LATERAL MODULE 2

Fig. 2 (2 of 7)

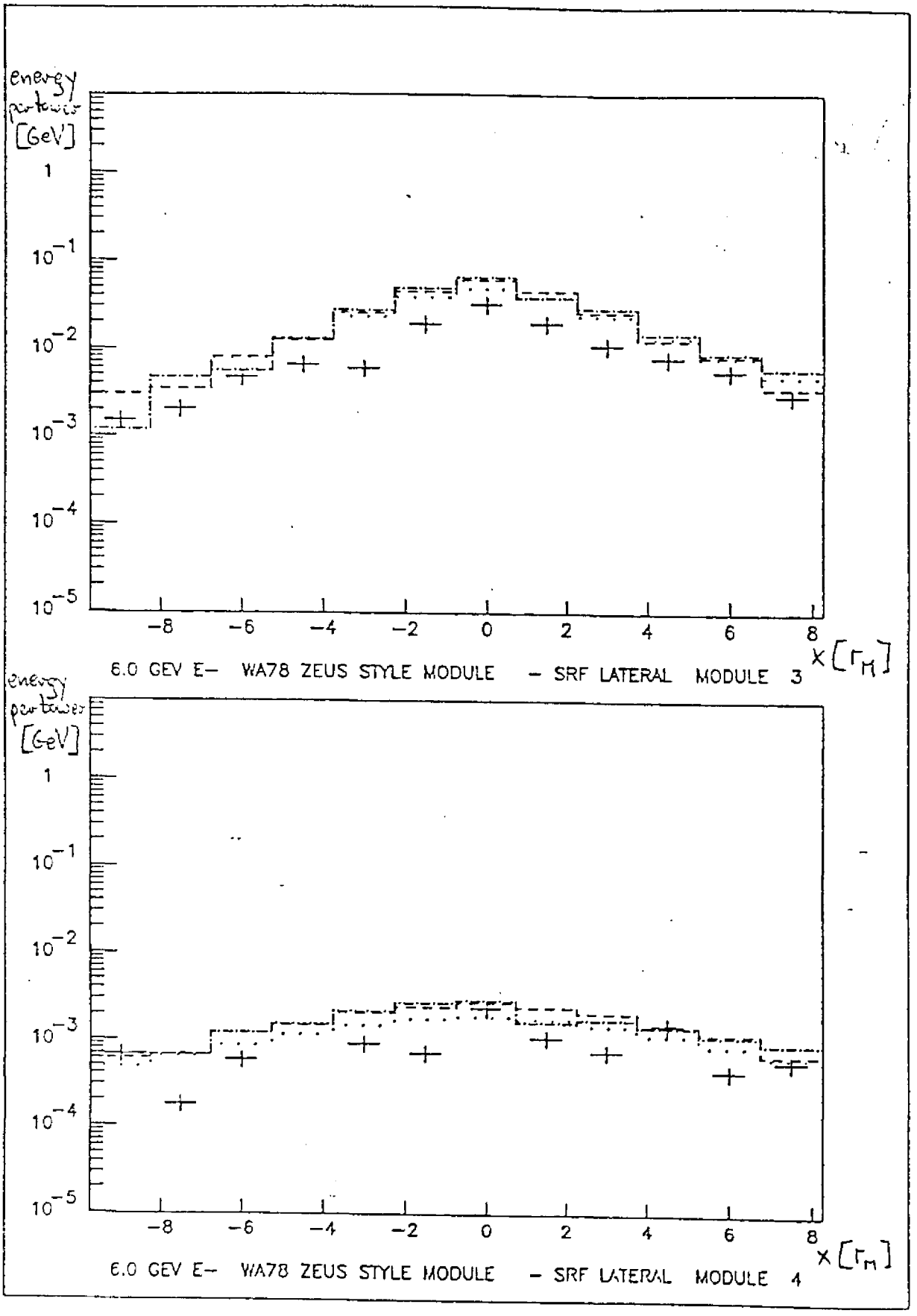

energy per tower [GeV]
1
$10^{-1}$
$10^{-2}$
$10^{-3}$
$10^{-4}$
$10^{-5}$
−8 −6 −4 −2 0 2 4 6 8
x [$r_M$]
6.0 GEV E− WA78 ZEUS STYLE MODULE − SRF LATERAL MODULE 3
energy per tower [GeV]
1
$10^{-1}$
$10^{-2}$
$10^{-3}$
$10^{-4}$
$10^{-5}$
−8 −6 −4 −2 0 2 4 6 8
x [$r_M$]
6.0 GEV E− WA78 ZEUS STYLE MODULE − SRF LATERAL MODULE 4

Fig. 2 (3 of 7)

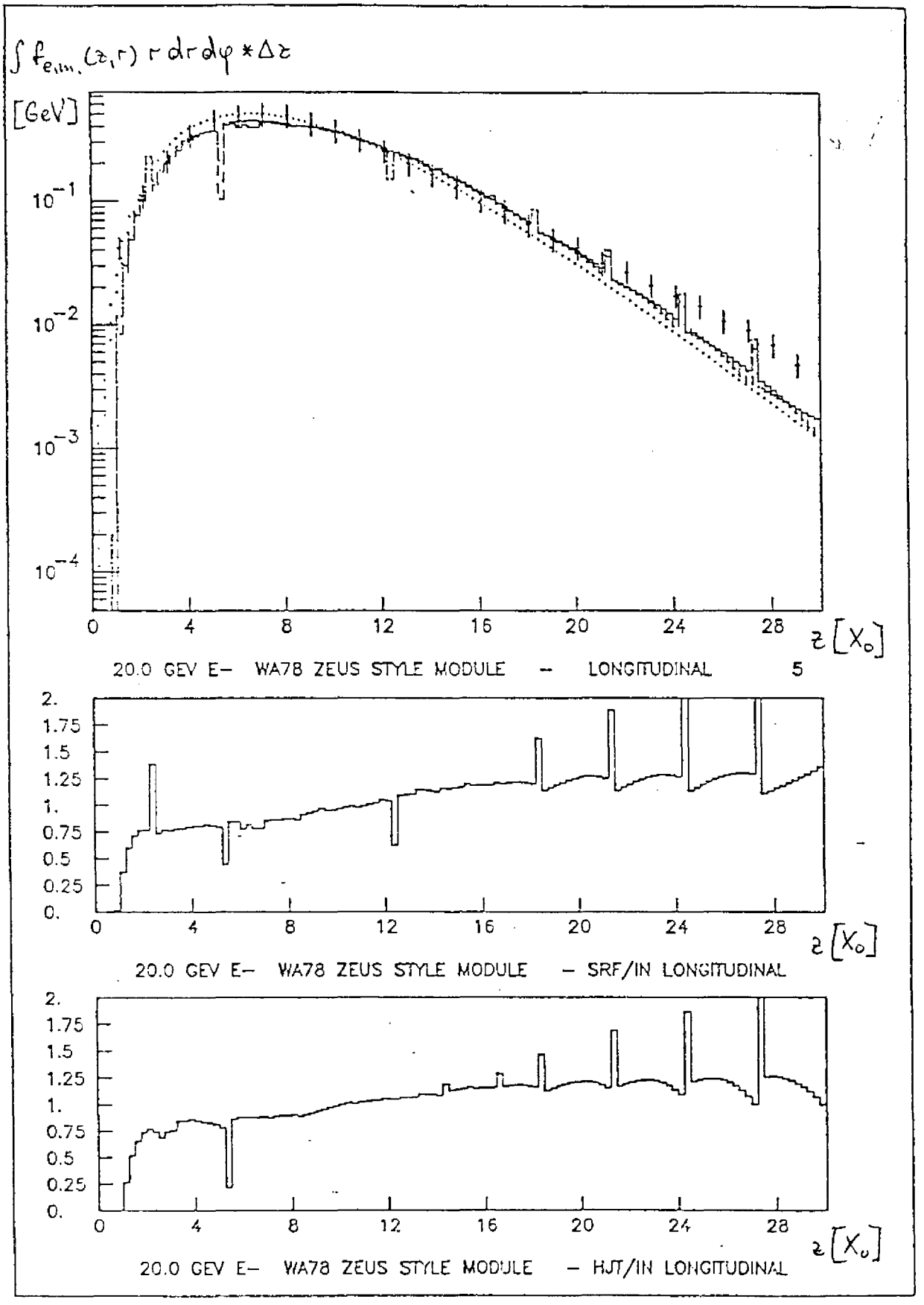
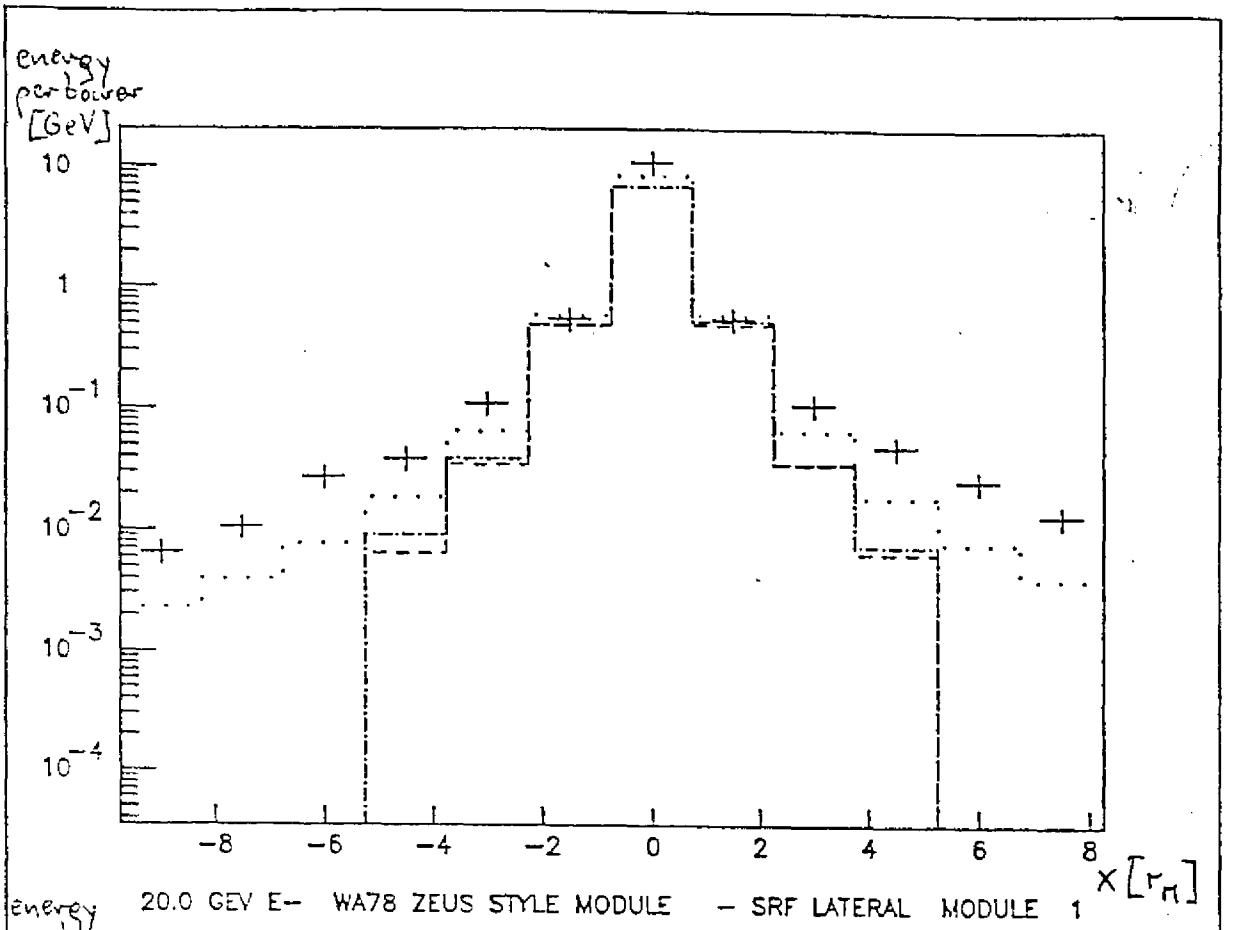

$\int f_{em.}(z,r)\, r\, dr\, d\varphi * \Delta z$
[GeV]
$10^{-1}$
$10^{-2}$
$10^{-3}$
$10^{-4}$
0 4 8 12 16 20 24 28
$z\,[X_0]$
20.0 GEV E− WA78 ZEUS STYLE MODULE − LONGITUDINAL 5
2. 1.75 1.5 1.25 1. 0.75 0.5 0.25 0.
0 4 8 12 16 20 24 28
$z\,[X_0]$
20.0 GEV E− WA78 ZEUS STYLE MODULE − SRF/IN LONGITUDINAL
2. 1.75 1.5 1.25 1. 0.75 0.5 0.25 0.
0 4 8 12 16 20 24 28
$z\,[X_0]$
20.0 GEV E− WA78 ZEUS STYLE MODULE − HJT/IN LONGITUDINAL

Fig. 2 (4 of 7)


energy per tower [GeV]
10
1
$10^{-1}$
$10^{-2}$
$10^{-3}$
$10^{-4}$
−8 −6 −4 −2 0 2 4 6 8
x [$r_M$]
20.0 GEV E− WA78 ZEUS STYLE MODULE − SRF LATERAL MODULE 1

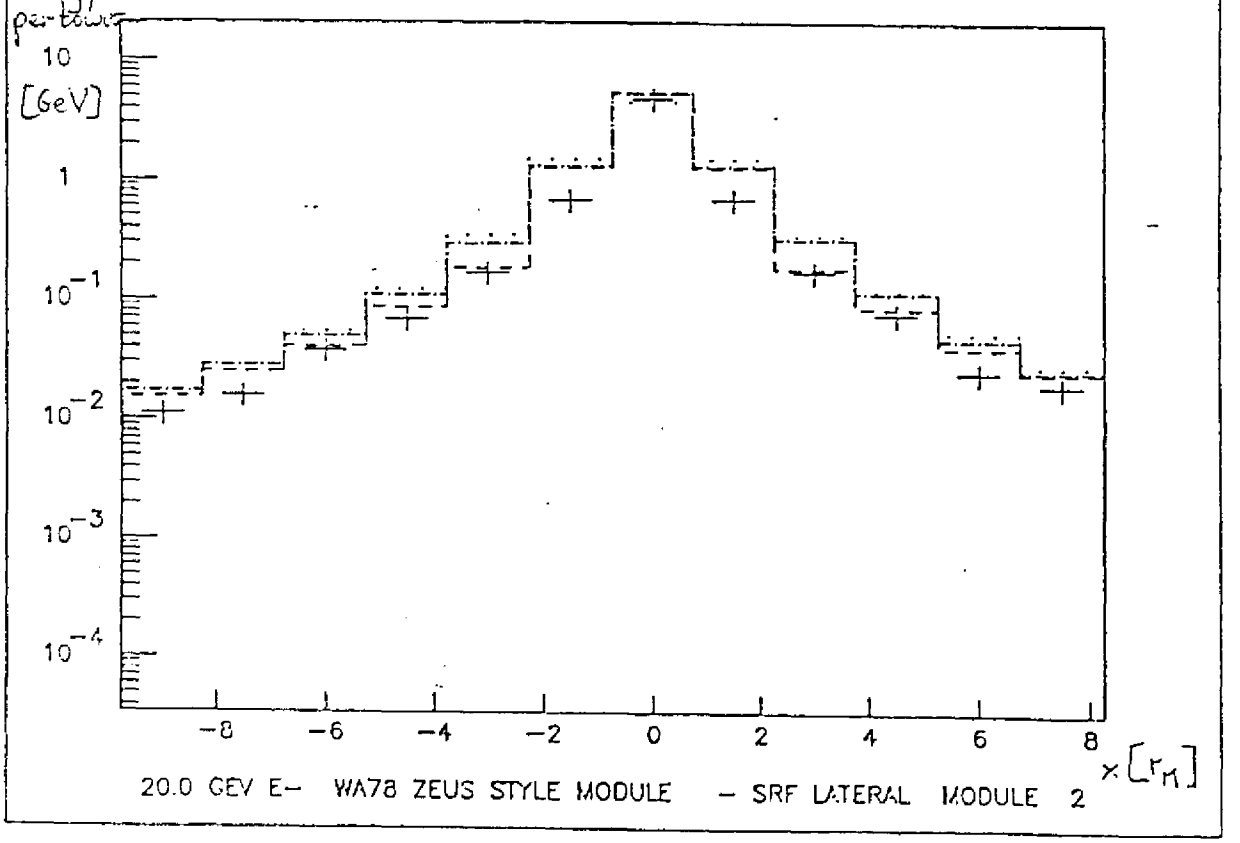

energy per tower [GeV]
10
1
$10^{-1}$
$10^{-2}$
$10^{-3}$
$10^{-4}$
−8 −6 −4 −2 0 2 4 6 8
x [$r_M$]
20.0 GEV E− WA78 ZEUS STYLE MODULE − SRF LATERAL MODULE 2

Fig. 2 (5 of 7)

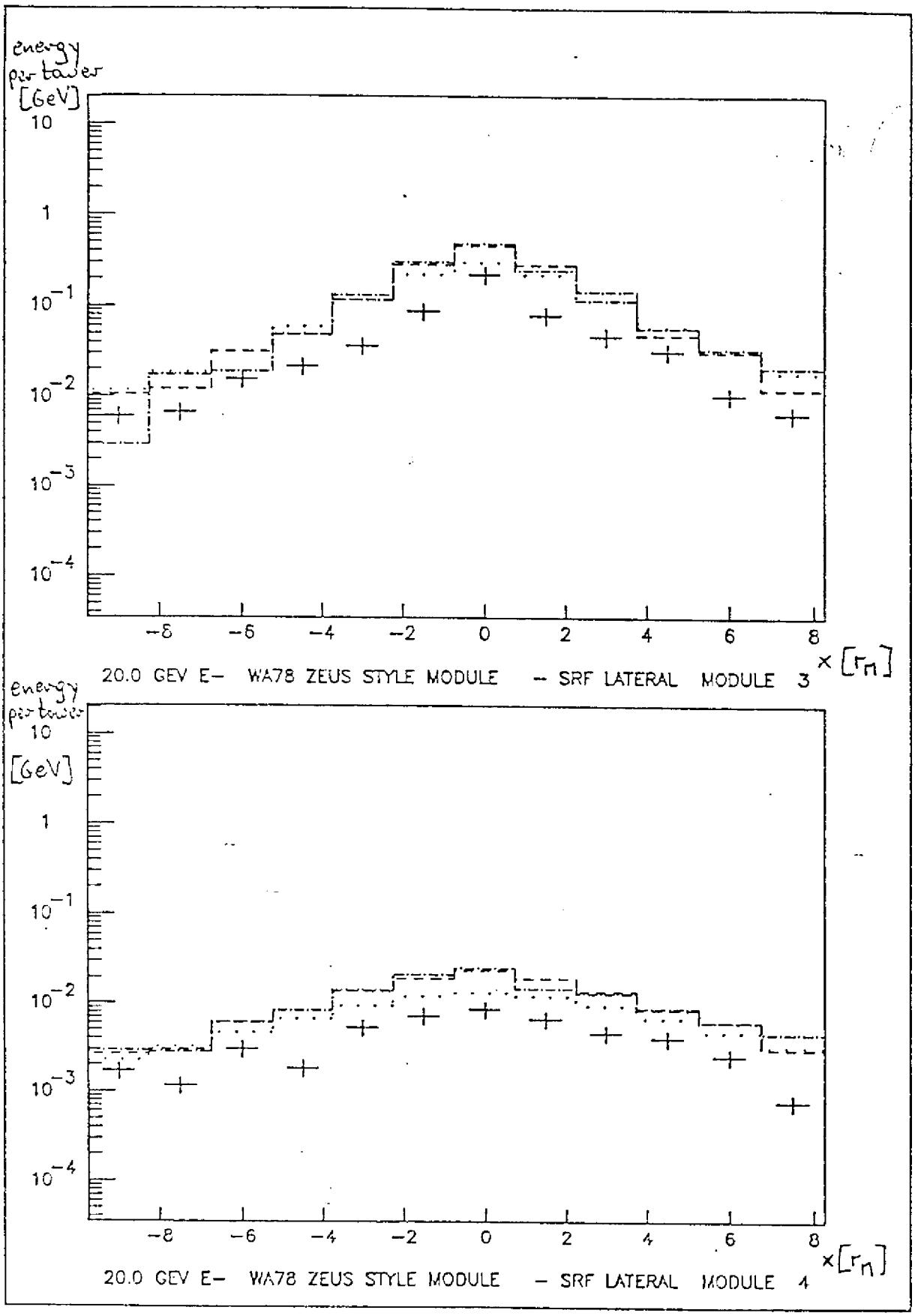

energy per tower [GeV]
10
1
$10^{-1}$
$10^{-2}$
$10^{-3}$
$10^{-4}$
−8 −6 −4 −2 0 2 4 6 8
x [$r_M$]
20.0 GEV E− WA78 ZEUS STYLE MODULE − SRF LATERAL MODULE 3
energy per tower [GeV]
10
1
$10^{-1}$
$10^{-2}$
$10^{-3}$
$10^{-4}$
−8 −6 −4 −2 0 2 4 6 8
x [$r_M$]
20.0 GEV E− WA78 ZEUS STYLE MODULE − SRF LATERAL MODULE 4

Fig. 2 (6 of 7)

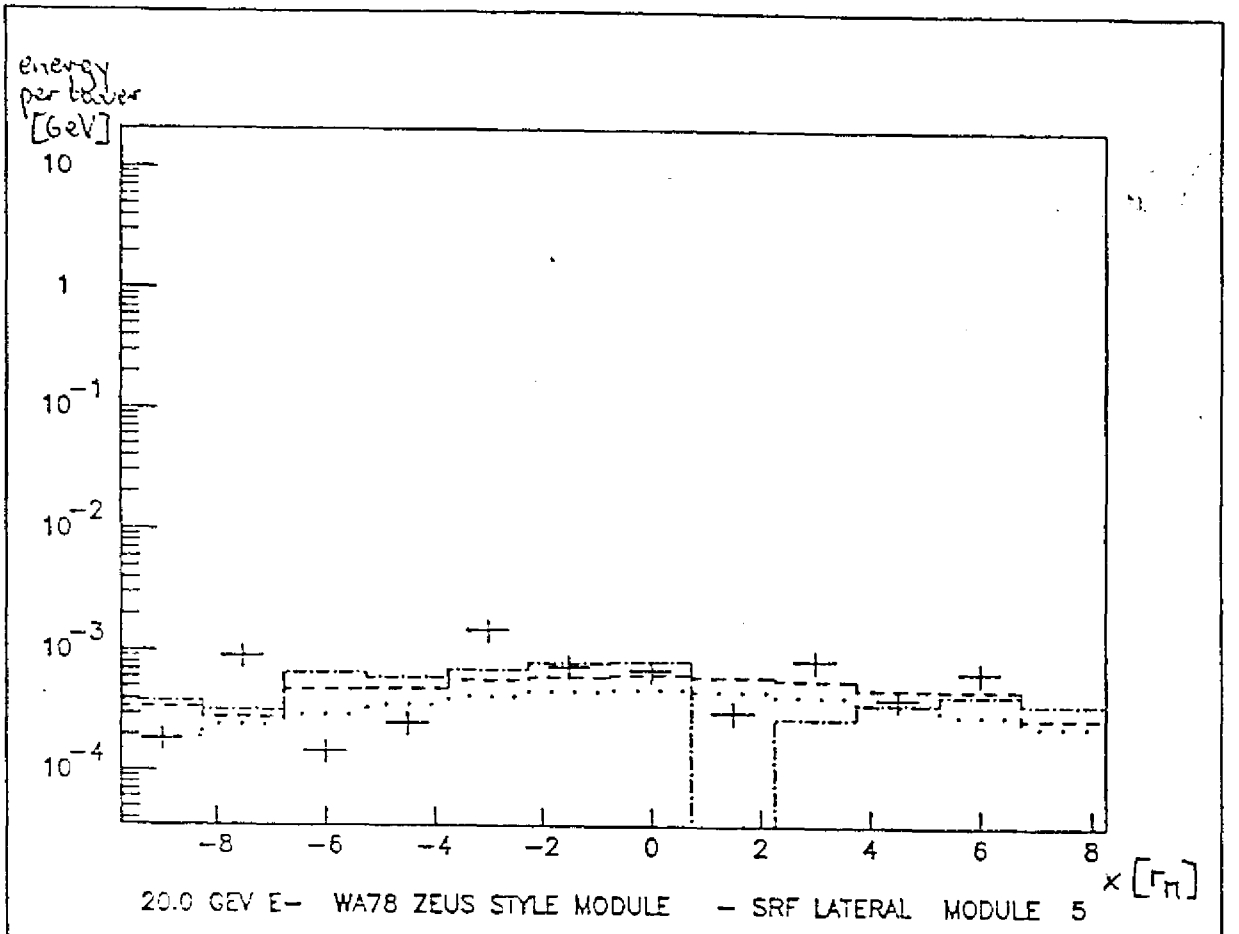

energy per layer [GeV]
10
1
$10^{-1}$
$10^{-2}$
$10^{-3}$
$10^{-4}$
−8
−6
−4
−2
0
2
4
6
8
x [$r_M$]
20.0 GEV E− WA78 ZEUS STYLE MODULE − SRF LATERAL MODULE 5

Fig. 2 (7 of 7)

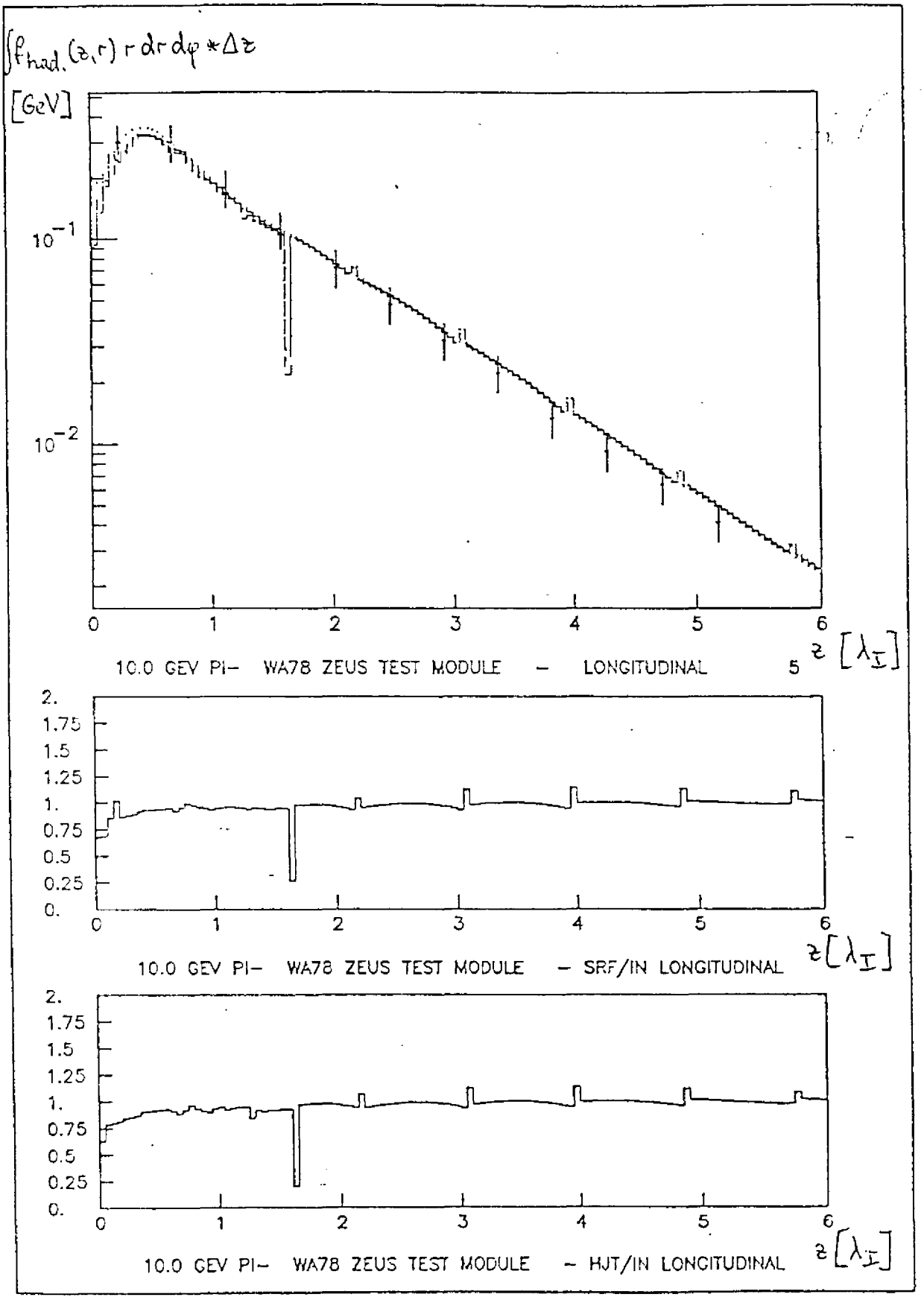

$\int f_{had.}(z,r)\, r\, dr\, d\varphi * \Delta z$
[GeV]
$10^{-1}$
$10^{-2}$
0 1 2 3 4 5 6
$z\ [\lambda_I]$
10.0 GEV PI- WA78 ZEUS TEST MODULE - LONGITUDINAL
2. 1.75 1.5 1.25 1. 0.75 0.5 0.25 0.
0 1 2 3 4 5 6
$z\ [\lambda_I]$
10.0 GEV PI- WA78 ZEUS TEST MODULE - SRF/IN LONGITUDINAL
2. 1.75 1.5 1.25 1. 0.75 0.5 0.25 0.
0 1 2 3 4 5 6
$z\ [\lambda_I]$
10.0 GEV PI- WA78 ZEUS TEST MODULE - HJT/IN LONGITUDINAL

Fig. 3 (1 of 6)

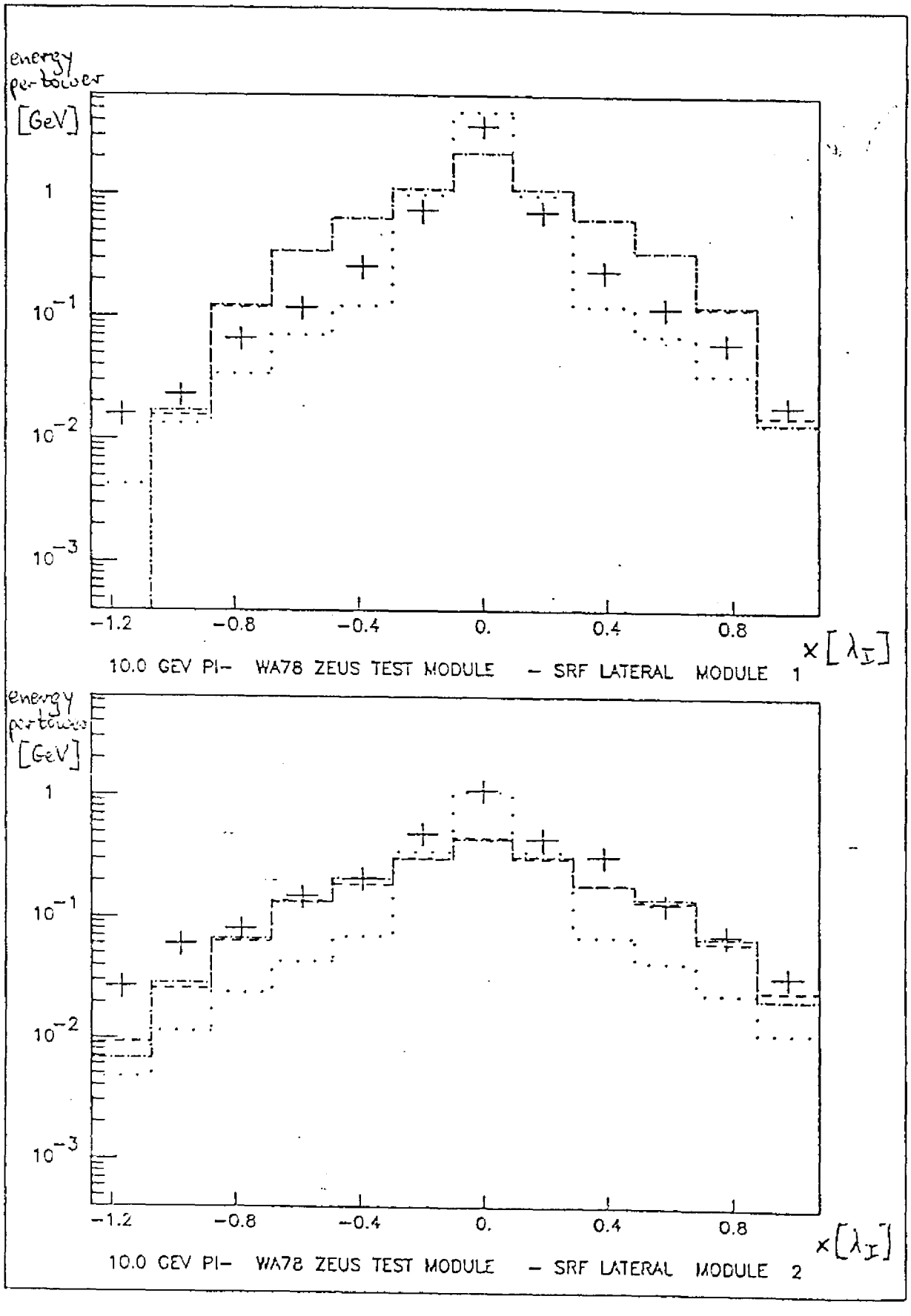

energy per tower [GeV]
1
$10^{-1}$
$10^{-2}$
$10^{-3}$
−1.2
−0.8
−0.4
0.
0.4
0.8
x [$\lambda_I$]
10.0 GEV PI− WA78 ZEUS TEST MODULE − SRF LATERAL MODULE 1
energy per tower [GeV]
1
$10^{-1}$
$10^{-2}$
$10^{-3}$
−1.2
−0.8
−0.4
0.
0.4
0.8
x [$\lambda_I$]
10.0 GEV PI− WA78 ZEUS TEST MODULE − SRF LATERAL MODULE 2

Fig. 3 (2 of 6)

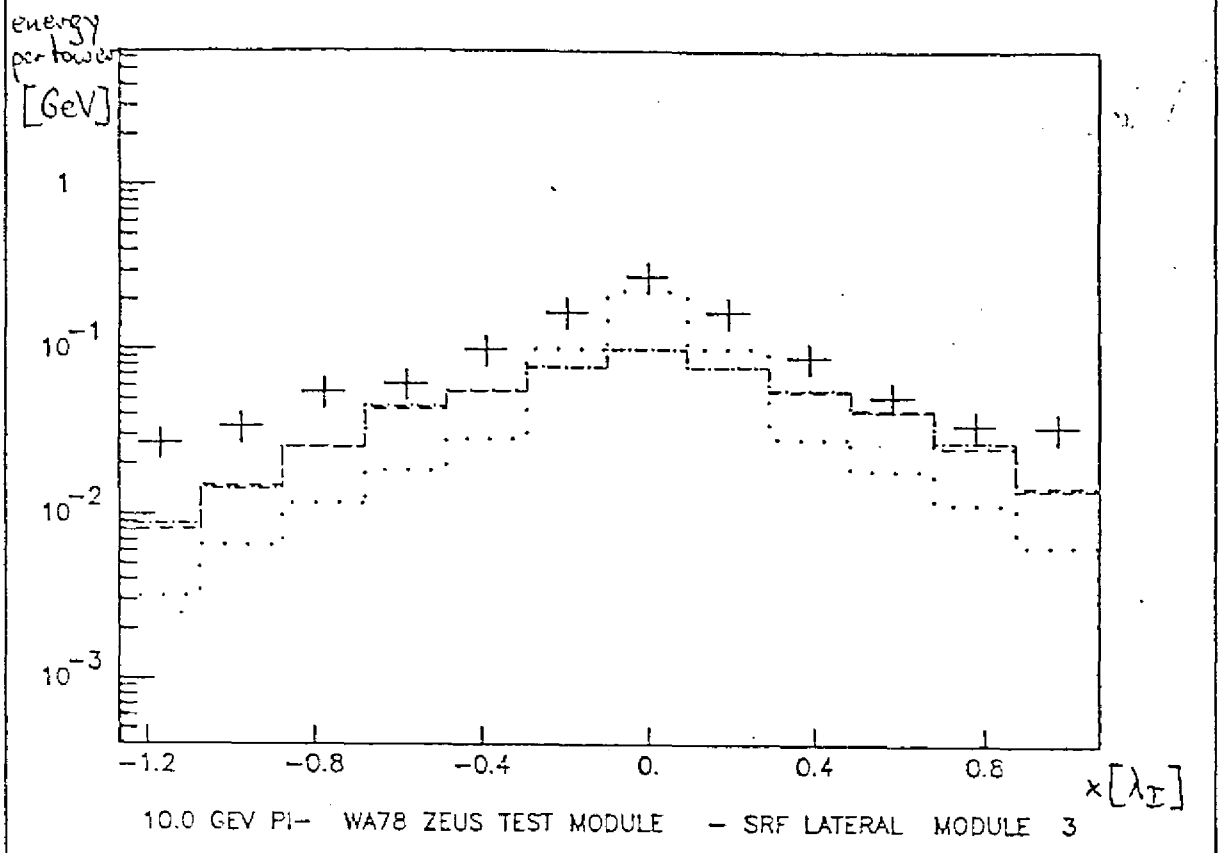

energy per tower [GeV]
1
$10^{-1}$
$10^{-2}$
$10^{-3}$
−1.2
−0.8
−0.4
0.
0.4
0.8
x[$\lambda_I$]
10.0 GEV PI− WA78 ZEUS TEST MODULE − SRF LATERAL MODULE 3

Fig. 3 (3 of 6)

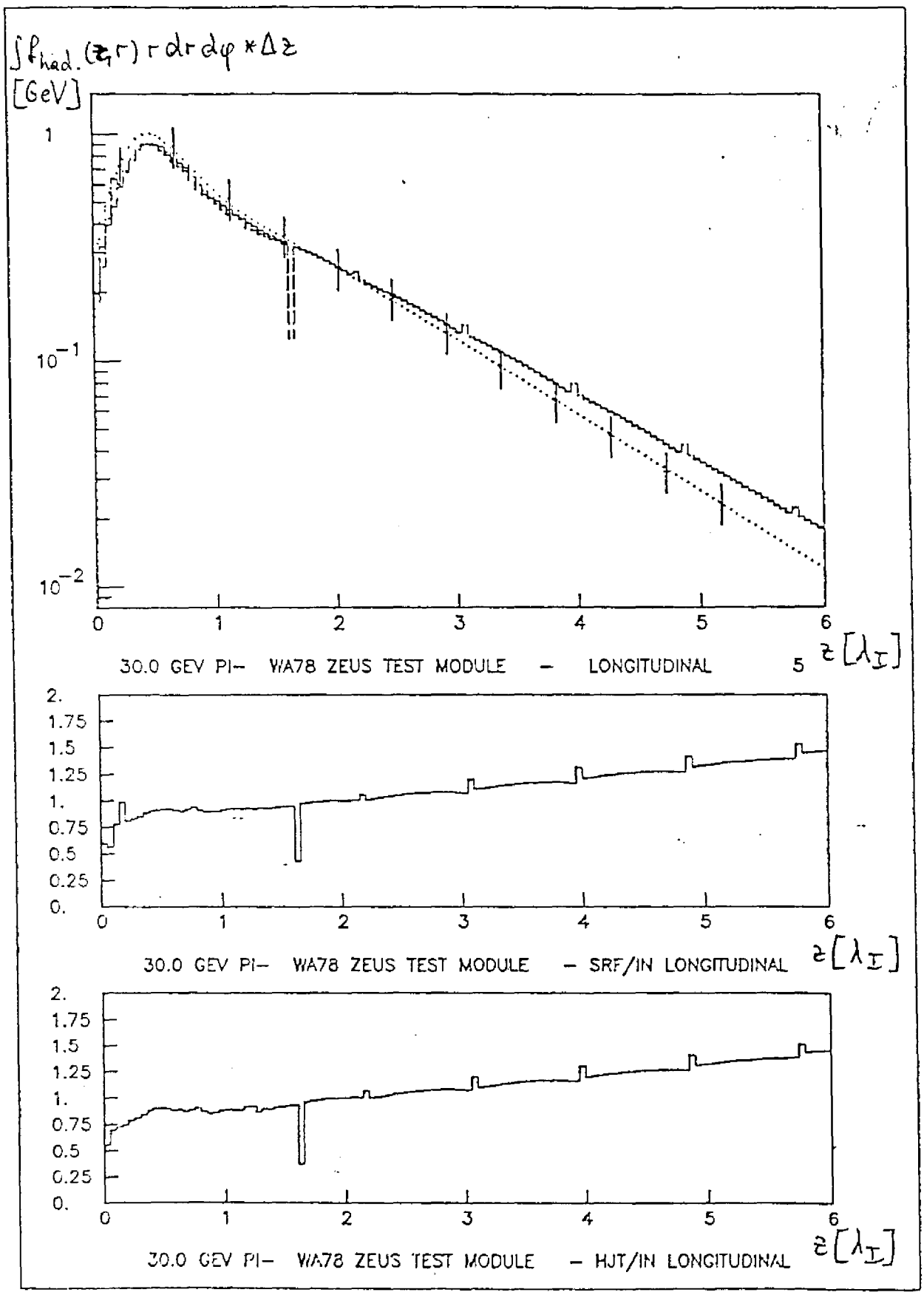

$\int f_{had.}(z,r)\, r\, dr\, d\varphi * \Delta z$
[GeV]
1
$10^{-1}$
$10^{-2}$
0 1 2 3 4 5 6
$z\,[\lambda_I]$
30.0 GEV PI- WA78 ZEUS TEST MODULE - LONGITUDINAL
2. 1.75 1.5 1.25 1. 0.75 0.5 0.25 0.
0 1 2 3 4 5 6
$z\,[\lambda_I]$
30.0 GEV PI- WA78 ZEUS TEST MODULE - SRF/IN LONGITUDINAL
2. 1.75 1.5 1.25 1. 0.75 0.5 0.25 0.
0 1 2 3 4 5 6
$z\,[\lambda_I]$
30.0 GEV PI- WA78 ZEUS TEST MODULE - HJT/IN LONGITUDINAL

Fig. 3 (4 of 6)

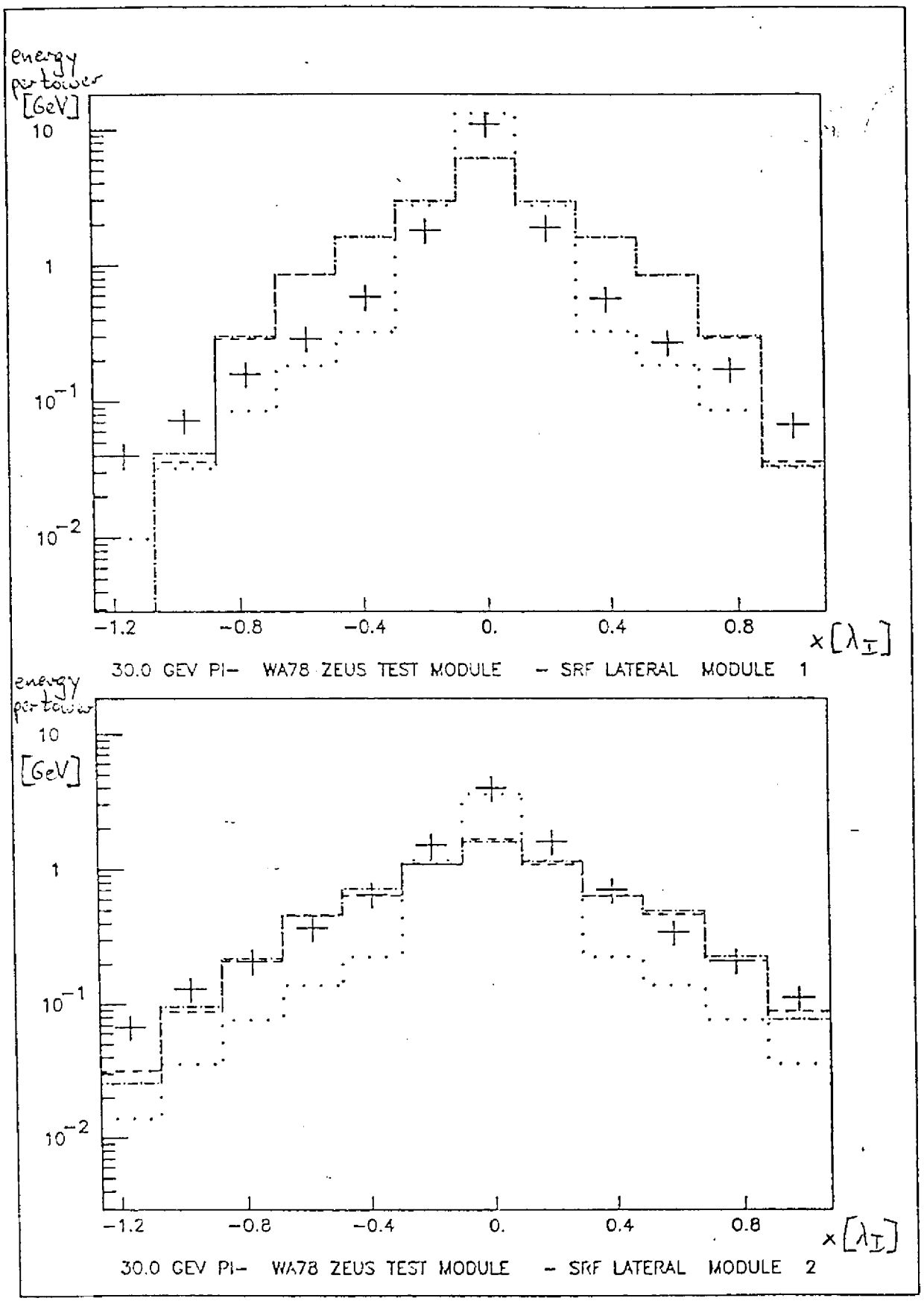

energy per tower [GeV]
10
1
$10^{-1}$
$10^{-2}$
−1.2
−0.8
−0.4
0.
0.4
0.8
x [$\lambda_I$]
30.0 GEV PI− WA78 ZEUS TEST MODULE − SRF LATERAL MODULE 1
energy per tower [GeV]
10
1
$10^{-1}$
$10^{-2}$
−1.2
−0.8
−0.4
0.
0.4
0.8
x [$\lambda_I$]
30.0 GEV PI− WA78 ZEUS TEST MODULE − SRF LATERAL MODULE 2

Fig. 3 (5 of 6)

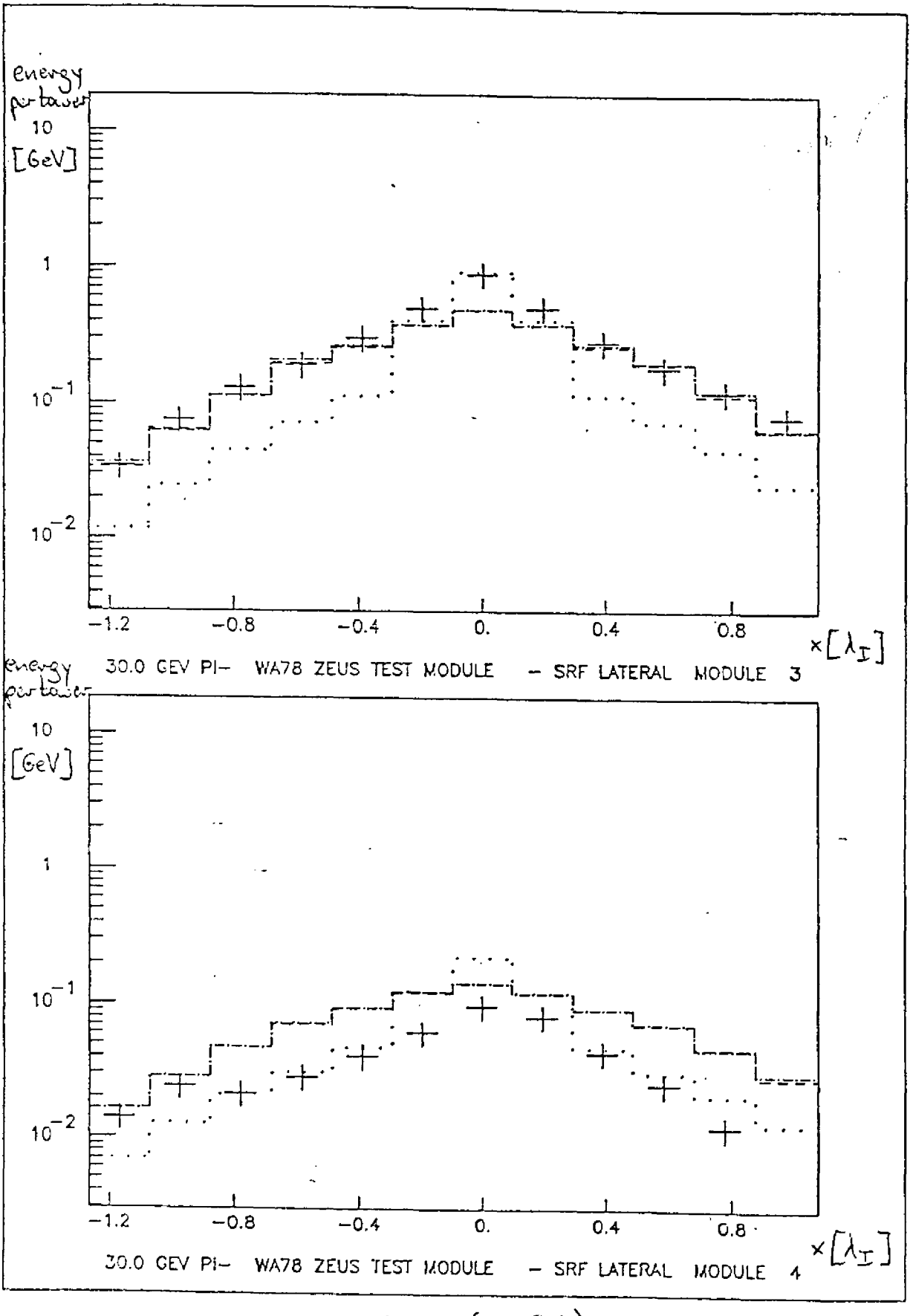
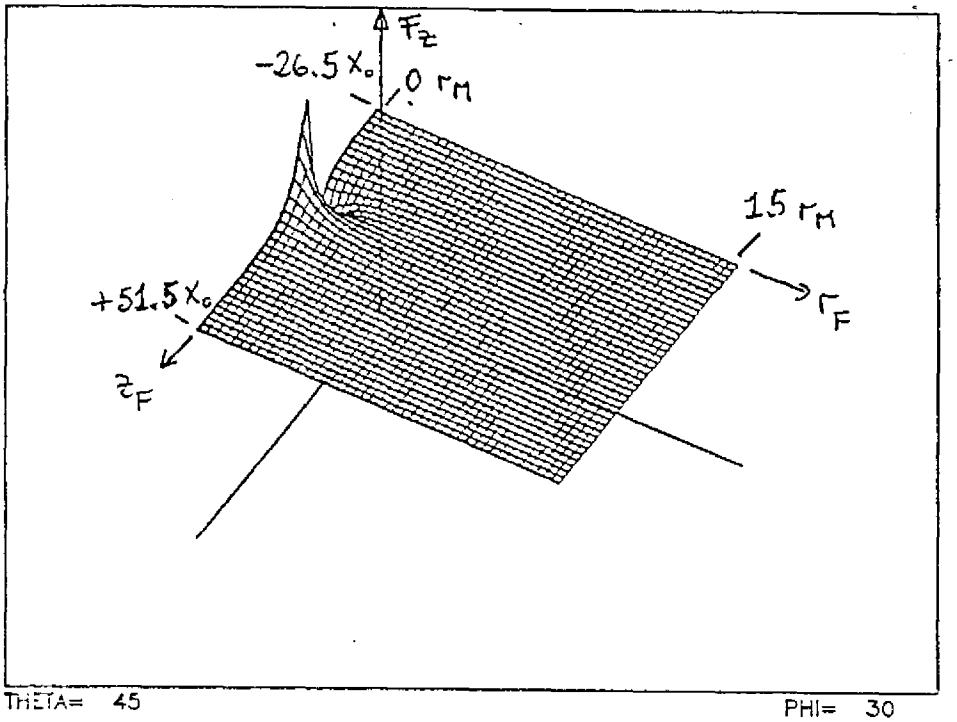

energy per tower
10
[GeV]
1
$10^{-1}$
$10^{-2}$
−1.2
−0.8
−0.4
0.
0.4
0.8
$\times[\lambda_I]$
30.0 GEV PI− WA78 ZEUS TEST MODULE − SRF LATERAL MODULE 3
energy per tower
10
[GeV]
1
$10^{-1}$
$10^{-2}$
−1.2
−0.8
−0.4
0.
0.4
0.8
$\times[\lambda_I]$
30.0 GEV Pi− WA78 ZEUS TEST MODULE − SRF LATERAL MODULE 4

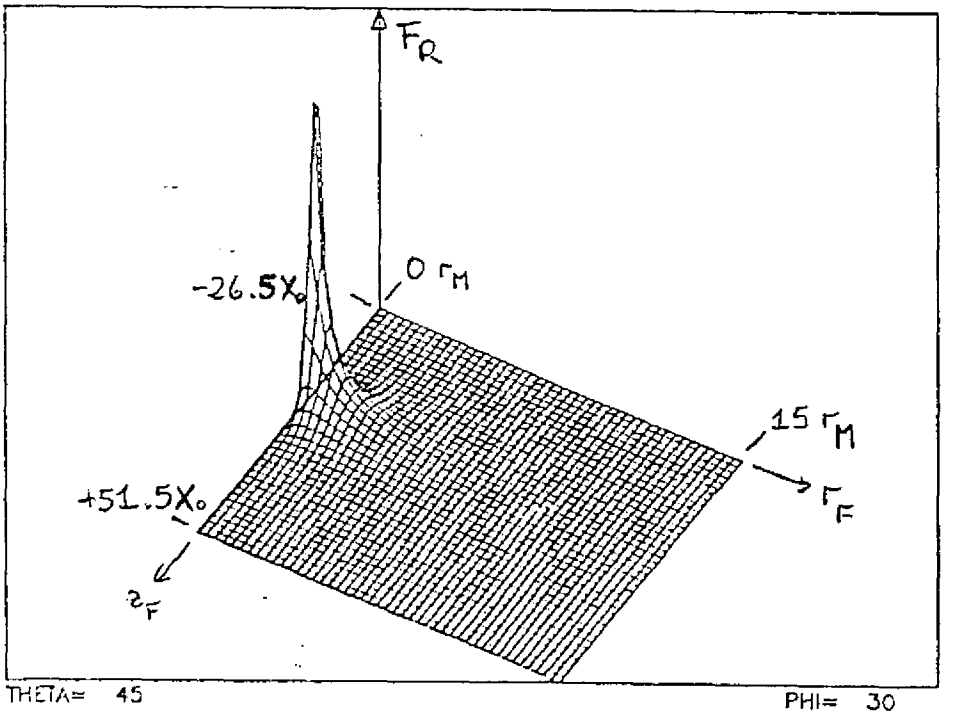
Fig. 3 (6 of 6)

VARIABLE STEP INTEGRATED VECTOR FIELDS ***




THETA= 45 PHI= 30

LONGITUDINAL COMPONENT OF FIELD FOR E 1.00 GEV




THETA= 45 PHI= 30

RADIAL COMPONENT OF FIELD FOR E 1.00 GEV

Fig. 4