NP Package User's Guide

Version 1.0

August 26, 2014

Contents

1 Introduction

Program package NP allows one to calculate differential and total cross sections for the neutrinoproduction of pions off nuclei. So one can easily guess that NP stays for (Neutrinoproduction of Pions). It was written in FORTRAN and all code based on paper Kopeliovich *et al.*, "Neutrinoproduction of pions off nuceli", ... , `http://arxiv.org` .

Package contains the following files:

`npdemo.f`	Simple demonstration program
`npdemo-make`	Shell script to make executable file `npmake`
`npmain.f`	Main functions and subroutines
`npmain.inc`	Main include file
`npmath.f`	Mathematical routines (interpolation, integration etc.)
`npmath.fun`	Math include file (function types)
`npmath.inc`	Math include file (constants and common variables)
`npnucl.f`	Nuclei functions (densities etc.)
`nppion.f`	Pion functions (pion nucleon cross sections etc.)
`nppion.inc`	Pion include file
`npuser.pdf`	User's Guide (PDF format)
`npuser.ps`	User's Guide (Postscript format)

and can be downloaded from `http://www.fis.utfsm.cl/np`.

2 Quick Start

Those who don't want to use all features of the package can start with simple demonstration program (file `npdemo.f`). This program calculates for the scattering of neutrino with energy $E = 2\,\mathrm{GeV}$ off neon ($\nu\,\mathrm{Ne} \to \mu^-\pi^+\,\mathrm{Ne}$) few differential cross sections (at some values of kinematic variables x, y and t)

$$\frac{d\sigma}{dx}\,,\quad \frac{d\sigma}{dx\,dy}\,,\quad \frac{d\sigma}{dx\,dy\,dt}$$

and also total cross section σ_{tot}. Just download package, unpack it, create executable `npdemo` and run it:

`$ tar xzf np.tar.gz`	Unpack the package
`$ npdemo-make`	Create executable `npmake` using `g77` compiler
`$ npdemo`	Run demonstration program

And after some time you will have as an output something like:

```
--------------------------------------------------

 nu Ne20 --> mu- pi+ Ne20

--------------------------------------------------

--- Kinematic variables ---

 E =    2.000 GeV
 x =    0.184
 y =    0.432
 t =   -0.258 GeV^2

--- Coherent cross sections ---

 d sigma / dx dy dt =    0.037 10^{-40} cm^2/GeV^2
 d sigma / dx dy    =    5.082 10^{-40} cm^2
 d sigma / dx       =   19.159 10^{-40} cm^2
   sigma total      =   23.221 10^{-40} cm^2
```

3 Package usage

Package written in FORTRAN and uses different `COMMON` blocks which names started with `NP` (like `NPKIN`, `NPREAC` etc.). Most of them (with few exceptions) are defined in include files `npmain.inc`, `npmath.inc` and `nppion.inc`. This include files define also some mathematical (like π) and physical constants (like α_{em}, weak coupling G, lepton, meson and nucleon masses etc.). It is not implied direct usage of this files from the user's program. All necessary settings can be done via subroutines and functions of the user interface.

All calculations are written in `DOUBLE PRECISION` (`REAL*8`). So all functions and all function and subroutine arguments are `REAL*8`. Few subroutines have `INTEGER` arguments. In such cases argument type will be specified explicitly.

Through the whole package all dimensional variables are in GeV units (for example, $1\text{fm} = 5.067728853\,\text{GeV}^{-1}$). In rare cases (on some printouts) results are presented in other units (and it specified explicitly).

Package uses its own library of mathematical routines for interpolation, integration (for real and complex functions) and special functions. Mostly they are original versions except some special functions (like Bessel functions $J_0(x)$, $J_1(x)$, $Y_0(x)$ and $Y_1(x)$ borrowed from CERN library http://cern.ch/cernlib). Details of usage for all this routines are out of the scope of this text. But those who have some interest can easily find all necessary details and examples of usage in the source code.

3.1 Output

All programs in the package use the same file index `F` for all output printouts (like error and warning messages, for example, in case of invalid values of kinematic variables provided by user etc.). By default this value is 6 (i.e. everything is send to the standard output). User can change this value and also set the level `L` of printout messages (0 - disable all printouts, 1 - only warning messages and some minimal information at the stage of setting reaction (default), 2 - some more printouts) with help of subroutine `SETOUT` with two `INTEGER` arguments:

`CALL` **SETOUT**`(F:INTEGER,L:INTEGER)`

If value of $\texttt{F} \leq 0$ then file index will not change. One can use this feature to change only printout level.

3.2 Setting reaction

First step in using this package is to set a reaction. One should call the subroutine `SETREAC` with four `INTEGER` arguments:

`CALL` **SETREAC**`(L:INTEGER,P:INTEGER,Z:INTEGER,A:INTEGER)`

where `L` and `P` denotes final lepton and pion

$$\texttt{L} = \begin{cases} \pm 1 & e^\pm \\ \pm 2 & \mu^\pm \\ \pm 3 & \tau^\pm \\ 0 & \nu(\bar{\nu}) \end{cases}, \qquad \texttt{P} = \begin{cases} \pm 1 & \pi^\pm \\ 0 & \pi^0 \end{cases},$$

while `Z` and `A` are atomic number and weight. For example, to setup a reaction from the demo program ($\nu\,\mathrm{Ne} \to \mu^-\pi^+\,\mathrm{Ne}$) one should set for a lepton `L=-2` (μ^-), for pion `P=1` (π^+) and for neon nucleus `Z=12` and `A=20`. This subroutine calculates various constants and interpolation data (for example, for nucleus density and thickness functions) of the specified reaction.

3.3 Kinematics

Packages provides set of subroutines to calculate regions for different kinematic variables: neutrino energy E, variables x, y and t. Instead of x, y one can use kinematic variables ν and Q^2. As far as reaction set with subroutine `SETREAC`, one can find neutrino energy threshold for this process:

`CALL` **KINEMIN**`(Emin)`

For neutrino energy `E` > `Emin`, one can find region, for example, for kinematic variable x

`CALL` **KINEX**`(E,xmin,xmax)`

Next step is for any fixed `x` in this interval (`xmin` $\leq$ `x` $\leq$ `xmax`) find corresponding y-region

```
    CALL KINEXY(E,x,ymin,ymax)
```

Instead of this chain of subroutine calls ($E \to x \to y$) one can use other sequences:
$E \to y \to x$:

```
    CALL KINEY(E,ymin,ymax)
    CALL KINEYX(E,y,xmin,xmax)
```

$E \to \nu \to Q^2$:

```
    CALL KINEN(E,numin,numax)
    CALL KINENQ(E,nu,Q2min,Q2max)
```

$E \to Q^2 \to \nu$:

```
    CALL KINEQ(E,Q2min,Q2max)
    CALL KINEQN(E,Q2,numin,numax)
```

As far as x and y (or ν and Q^2) are fixed one can find t-region

```
    CALL KINEXYT(E,x,y,tmin,tmax)
    CALL KINENQT(E,nu,Q2,tmin,tmax)
```

And finally (for completeness) package contains subroutines for changing kinematic variables ($x, y \Leftrightarrow \nu, Q^2$):

`CALL` **KINXEYNQ**`(E,x,y,nu,Q2)`	$E, x, y \ \to \nu, Q^2$
`CALL` **KINENQXY**`(E,nu,Q2,x,y)`	$E, \nu, Q^2 \to x, y$

3.4 Pions and nuclei

Package also provides different pion and nucleus functions. For example, for pion one can obtain total $\sigma_{tot}^{\pi N}(s)$ and elastic $\sigma_{el}^{\pi N}(s)$ cross sections off nucleon as a function of kinematic variable s

SigPNtot`(s)`
SigPNel`(s)`

Differential elastic cross section $d\sigma_{el}^{\pi N}(s, \cos\theta)/d\cos\theta$ is given by function of two variables s and $\cos\theta$ (here `c` stays for $\cos\theta$ in c.m.)

DSigPNel`(s,c)`

Pion-nucleon ratio of the real to imaginary parts $\alpha_{\pi N}(s) = \mathrm{Re}\, T/\mathrm{Im}\, T$ of the forward hadronic amplitudes and slope $B_{\pi N}(s)$ are given by functions of one kinematic variable s

AlpPN(s)

BslPN(s)

All pion functions give results for selected type of pion averaged on protons and neutrons of the selected nucleus, i.e. if one, for example, has defined reaction (via `SETREAC`) for π^+ and nucleus with $Z = 12$ and $A = 20$ then function `SigPNtot` gives $(12\sigma_{tot}^{\pi p} + 8\sigma_{tot}^{\pi n})/20$. So if one needs, for example, total cross section for π^- on proton just set in `SETREAC` argument `P=-1` and for nucleus `Z=1` and `A=1`.

Package provides also some other nucleus functions, but most of them are specific for the used approach and for general use, probably, one can mention only the simplest one - nucleus density $\rho_A(r)$

Rho(r)

All this functions were mentioned only for completeness. Normally if one needs only cross sections for the neutrinoproduction of pions off nuclei there are no needs in their direct usage.

3.5 Neutrinoproduction cross sections

Differential and total neutrinoproduction cross sections can be calculated in coherent and incoherent channel. To select type of the process one should call subroutine `SETSIGT` with one `INTEGER` argument `T` (default value is 0 - coherent, and 1 - incoherent)

CALL **SETSIGT**(T:INTEGER)

As it was already mentioned above, all dimensional variables are in GeV units. But for all neutrinoproduction differential and total cross sections one can change default value `U` $= 1.0$ (which corresponds to cross sections in GeV^{-2}) to some other reasonable value. It can be done with help of subroutine `SETSIGU`:

CALL **SETSIGU**(U)

where `U` is some new unit in GeV^{-2}. For example, to have results for cross sections in fm^2 one should call this subroutine with `U` $= 5.067728853^2$. One more example can be found in the demonstration program `npdemo.f`.

Main differential cross section $d\sigma_{\nu A}(E, x, y, t)/dx\,dy\,dt$ for neutrinoproduction of pions off nuclei A is given by function `SigNAxyt`

SigNAxyt(E,x,y,t)

where E - neutrino energy, and x, y, t are standard kinematic variables. One can use also variables ν and Q^2. Corresponding cross section $d\sigma_{\nu A}(E, \nu, Q^2, t)/d\nu\,dQ^2\,dt$ is given by function `SigNAnQt`

SigNAnQt(E,nu,Q2,t)

All other cross sections calculated on its basis by integrating on some kinematic variables.

The accuracy of such integrations can be chosen with help of subroutine `SETSIGA` with one parameter

`CALL` **SETSIGA**`(Eps)`

Default value for `Eps` is 0.01. That is enough for the accuracy of the whole approach by itself. But to make some estimations of cross sections one can reduce this value to 0.1 and still have reasonable accuracy ($\leq 10\%$) but for noticable shorter time. Changing this parameter will affect the following differential and total cross sections:

SigNAxy`(E,x,y)`	$d\sigma_{\nu A}/dx\,dy$
SigNAx`(E,x)`	$d\sigma_{\nu A}/dx$
SigNAy`(E,y)`	$d\sigma_{\nu A}dy$
SigNAnQ`(E,nu,Q2)`	$d\sigma_{\nu A}d\nu\,dQ^2$
SigNAn`(E,nu)`	$d\sigma_{\nu A}d\nu$
SigNAQ`(E,Q2)`	$d\sigma_{\nu A}dQ^2$
SigNA`(E)`	$\sigma_{\nu A}$

And the final remark. The code for the function `SigNA` is optimized. In fact it even integrates over total kinematic volume using some special kinematic variables (not x, y nor ν, Q^2). So if one needs total cross section it's better to use just this function. For all other cross sections (like `SigNAx`,`SigNAxy` etc.) one should feels free to write his own routines (for example, to have distributions on some other kinematic variable) on a basis of main routine `SigNAxyt`.

4 Short Reference

BOLD denotes type `REAL*8` and *ITALIC* type `INTEGER`

— Output —

`CALL SETOUT(`*F*`,`*L*`)`	set output file F and printout level L

— Reaction —

`CALL SETREAC(`*L*`,`*P*`,`*Z*`,`*A*`)`	set L - lepton, P -pion, nucleus Z and A

— Kinematics —

`CALL KINEMIN(`**Emin**`)`	get neutrino energy threshold E_{min}		
`CALL KINEX(`**E,xmin,xmax**`)`	E	$\to$	x_{min}, x_{max}
`CALL KINEXY(`**E,x,ymin,ymax**`)`	E, x	$\to$	y_{min}, y_{max}
`CALL KINEY(`**E,ymin,ymax**`)`	E	$\to$	y_{min}, y_{max}
`CALL KINEYX(`**E,y,xmin,xmax**`)`	E, y	$\to$	x_{min}, x_{max}
`CALL KINEN(`**E,numin,numax**`)`	E	$\to$	ν_{min}, ν_{max}
`CALL KINENQ(`**E,nu,Q2min,Q2max**`)`	E, ν	$\to$	Q^2_{min}, Q^2_{max}
`CALL KINEQ(`**E,Q2min,Q2max**`)`	E	$\to$	Q^2_{min}, Q^2_{max}
`CALL KINEQN(`**E,Q2,numin,numax**`)`	E, Q^2	$\to$	ν_{min}, ν_{max}
`CALL KINEXYT(`**E,x,y,tmin,tmax**`)`	E, x, y	$\to$	t_{min}, t_{max}
`CALL KINENQT(`**E,nu,Q2,tmin,tmax**`)`	E, ν, Q^2	$\to$	t_{min}, t_{max}
`CALL KINEXYNQ(`**E,x,y,nu,Q2**`)`	E, x, y	$\to$	ν, Q^2
`CALL KINENQXY(`**E,nu,Q2,x,y**`)`	E, ν, Q^2	$\to$	x, y

— Pions and nuclei —

SigPNtot(**s**)	total $\sigma^{\pi N}_{tot}(s)$
SigPNel(**s**)	elastic $\sigma^{\pi N}_{el}(s)$
DSigPNel(**s,c**)	elastic differential $d\sigma_{\pi N}(s, \cos\theta)/d\cos\theta$
AlpPN(**s**), **BslPN**(**s**), **Rho**(**r**)	ratio $\alpha_{\pi N}(s)$, slope $B_{\pi N}(s)$, density $\rho_A(r)$

— Neutrinoproduction cross sections —

`CALL SETSIGT(`*T*`)`	set type (0 - coherent, 1 - incoherent)
`CALL SETSIGU(`**U**`)`	set unit (**U** in GeV^{-2})
`CALL SETSIGA(`**Eps**`)`	set accuracy
SigNAxyt(**E**,**x**,**y**,**t**)	$d\sigma_{\nu A}/dx\,dy\,dt$
SigNAnQt(**E**,**nu**,**Q2**,**t**)	$d\sigma_{\nu A}/d\nu\,dQ^2\,dt$
SigNAxy(**E**,**x**,**y**)	$d\sigma_{\nu A}/dx\,dy$
SigNAx(**E**,**x**)	$d\sigma_{\nu A}/dx$
SigNAy(**E**,**y**)	$d\sigma_{\nu A}/dy$
SigNAnQ(**E**,**nu**,**Q2**)	$d\sigma_{\nu A}/d\nu\,dQ^2$
SigNAn(**E**,**nu**)	$d\sigma_{\nu A}/d\nu$
SigNAQ(**E**,**Q2**)	$d\sigma_{\nu A}/dQ^2$
SigNA(**E**)	total $\sigma_{\nu A}$