

CRIB Anapaw 入門

2015-2016年 山口研究室所属 坂口裕司

はじめに

Anapawの経験と知識がない方を読者に想定しています。この説明書によって、CRIBの環境におけるAnapawのためのコマンドライン上での動かし方、ヒストグラムの定義の仕方、変数の追い方の初歩が説明されています。

著者のPC環境で説明用の画像などが作られていますが、CRIBのPCと環境に大差はないので、この説明書と同じような操作でCRIBのAnapawは動くはずです。

ディレクトリ構造やファイル名やソースコードは2015年のCRIBのものを想定しています。

口伝のCRIB Anapawを文章化したもので、著者が推測しながら理解したものを多く含んでおり、間違いがある可能性が高いです。間違いを見つけたら、この説明書を訂正してもらい、代々後輩へと引き継いでもらえたら幸いです。

コンテンツ

● ヒストグラムを見るまで

- analogin, anapaw, a/l, book, start, hst
- コマンドの参考

● ヒストグラムの定義

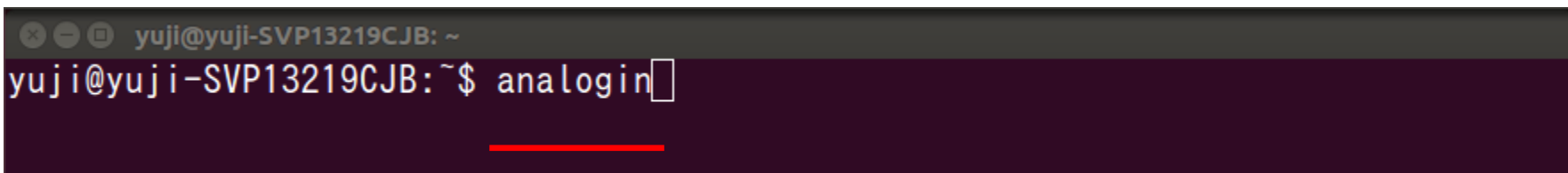
- Analyzer ID, 検出器 ID, 変数 ID
- anafileの構造
- gateの定義
- ヒストグラムの定義の具体的なやり方

● ソースコード内の変数

- ソースコード間にまたがる変数の追い方

ヒストグラムを見るまで

まずAnapawシステムに入る流れを説明します。



```
yuji@yuji-SVP13219CJB: ~$ analogin
```

端末から抜粋

端末で「analogin」と入力してエンターキーを押すと

ヒストグラムを見るまで

```
yuji@yuji-SVP13219CJB: ~
yuji@yuji-SVP13219CJB:~$ analogin

Your have already work directory. Start ANAPAW!

Local Environments are defined for ANAPAW.

ANAPAW_USER      = anapaw
CERNLIB          = /cern/lib
ANAPAW_HOME      = /home/yuji/usr/local/anapaw
ANAPAW_LIB       = /home/yuji/usr/local/anapaw/lib
ANAPAW_SOURCE    = /home/yuji/usr/local/anapaw/src
ANAPAW_MACROS    = /home/yuji/usr/local/anapaw/Macros
ANALOGON         = /home/yuji/10Be_plus_Alpha/crib/.analogon.kumac
USER_SOURCE      = /home/yuji/10Be_plus_Alpha/crib/src
BEAM_PRM        = /home/yuji/10Be_plus_Alpha/crib/src/beam.prm
PPAC_PRM        = /home/yuji/10Be_plus_Alpha/crib/src/ppac.prm
SSD_PRM         = /home/yuji/10Be_plus_Alpha/crib/src/ssd.prm
VME_PRM         = /home/yuji/10Be_plus_Alpha/crib/src/vme.prm
DALI_MAP        = /home/yuji/10Be_plus_Alpha/crib/src/dali.map
DALI_PRM        = /home/yuji/10Be_plus_Alpha/crib/src/dali.prm
PSD_MAP         = /home/yuji/10Be_plus_Alpha/crib/src/psd.map
PSD_VME_MAP     = /home/yuji/10Be_plus_Alpha/crib/src/psd_vme.map
PSD_PRM        = /home/yuji/10Be_plus_Alpha/crib/src/psd.prm
IHIT_MIN0      = 1

===== NOTICE ! =====

If you change ihit_min0 in enc_fera, use setana.

ex) > setana ihitmin 4

To execute ANAPAW, type anapaw <CR>.

If you change Local Environments, edit setlocal.

yuji-SVP13219CJB:~/10Be_plus_Alpha/crib> █
```

Anapawのシステムのあるディレクトリへ入ることができます。

端末から抜粋

ヒストグラムを見るまで

最初に着目すべきAnapawのディレクトリ

「ls」というコマンドによってファイルとディレクトリが端末上に表示されます。

まず着目すべきディレクトリは以下です。

- rdf...ここに実験の生データが格納されています。
- ana...ここに「anafile」が格納されています。このanafileによってヒストグラムが定義されます。
- src...ここにソースファイルが格納されています。チャンネル、エネルギー、時間情報などの物理量の計算の指示はこのディレクトリ内にあるソースコードで行います。

ヒストグラムを見るまで

その後、「anapaw」と入力してエンターキーを押すとこの画面になります。

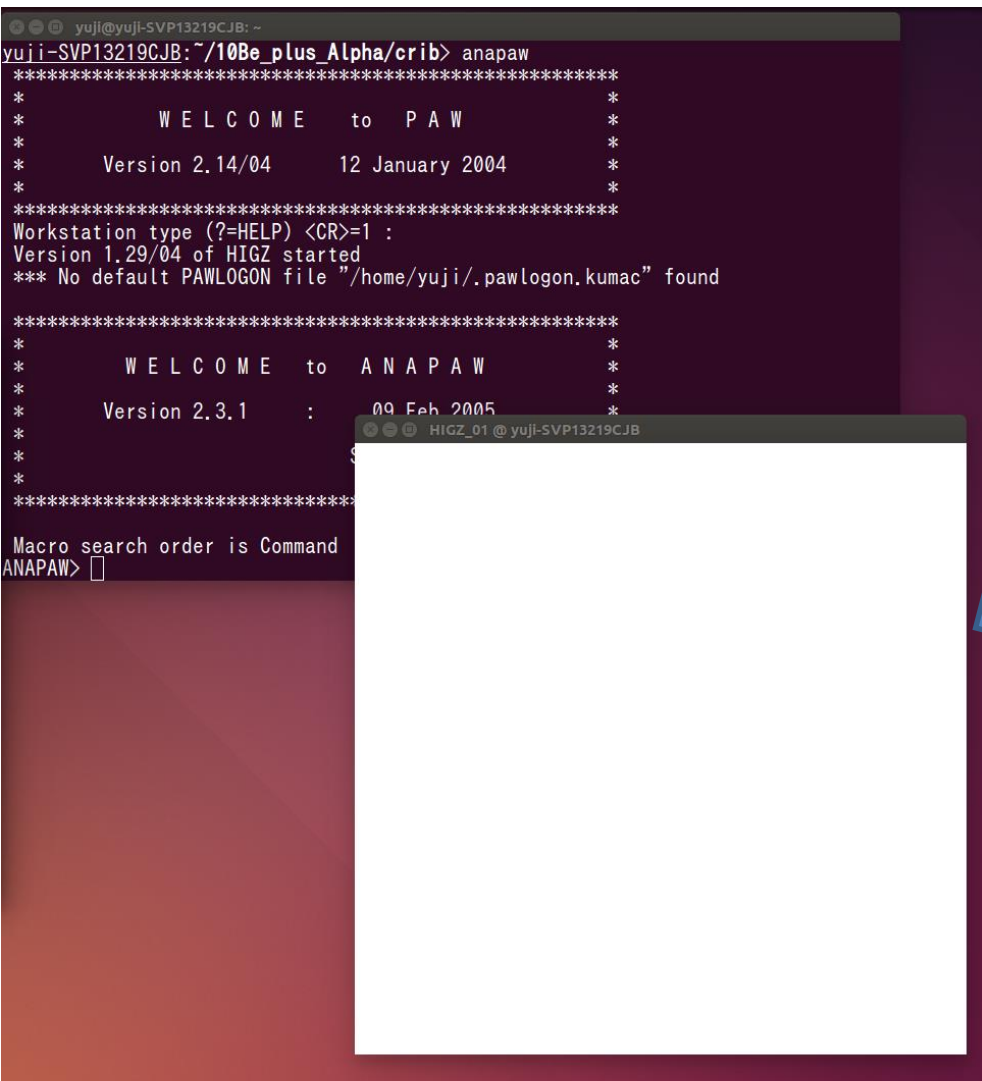
```
yuji-SVP13219CJB:~/10Be_plus_Alpha/crib> anapaw
*****
*                                     *
*      W E L C O M E   to   P A W      *
*                                     *
*      Version 2.14/04      12 January 2004   *
*                                     *
*****
Workstation type (?=HELP) <CR>=1 : █
```

端末から抜粋

再度エンターキーを押すと

ヒストグラムを見るまで

```
yuji@yuji-SVP13219CJB: ~
yuji-SVP13219CJB:~/10Be_plus_Alpha/crib> anapaw
*****
*                                     *
*      WELCOME to PAW                 *
*                                     *
*      Version 2.14/04   12 January 2004  *
*                                     *
*****
Workstation type (?=HELP) <CR>=1 :
Version 1.29/04 of HIGZ started
*** No default PAWLOGON file "/home/yuji/.pawlogon.kumac" found
*****
*                                     *
*      WELCOME to ANAPAW               *
*                                     *
*      Version 2.3.1   :   09 Feb 2005  *
*                                     *
*****
Macro search order is Command
ANAPAW> 
```



この様なウィンドウが開きます。

端末でコマンドを入力し、結果のヒストグラムがこのウィンドウに表示されます。

これでAnapawシステムに入った状態です。

パソコン画面から抜粋

ヒストグラムを見るまで

ここからはヒストグラムを見るまでの流れを説明します。

```
ANAPAW> a/l rdf/0100.rdf  
  
rdf/0100.rdf  Size : 336.9 MB  Block : 21564  
ANAPAW/EVTLOOP> book ana/run.ana  
  > book ana/run.ana  
ANAPAW/EVTLOOP> start█
```

端末から抜粋

「a/l 解析するファイル名」でLoopするファイルを選択します。
(オンライン解析ならファイル指定なし)

「book anafilename」でanafilenameを選択します。

「start」
でLoop開始です。

Loop...それぞれの検出器に対応する
物理量を表す変数を計算し、ヒストグ
ラムに詰める作業を繰り返すこと

ヒストグラムを見るまで

Loopが進むごとに数字が上がっていきます。最後までいけば勝手にLoopは止まります。エンタキーで途中で止めることもできます。

```
|=====/ | 6034/ 21564
ANAPAW-M : Interrupt Event Loop!

Blocks      : 6067
Total trigger : 414228
Valid Events : 414228
Valid/Total(%) : 100.00

ANAPAW/EVTLOOP> i
```

端末から抜粋

「i」
でanafileで定義したヒストグラムのID(HID)が表示されます。

ヒストグラムを見るまで

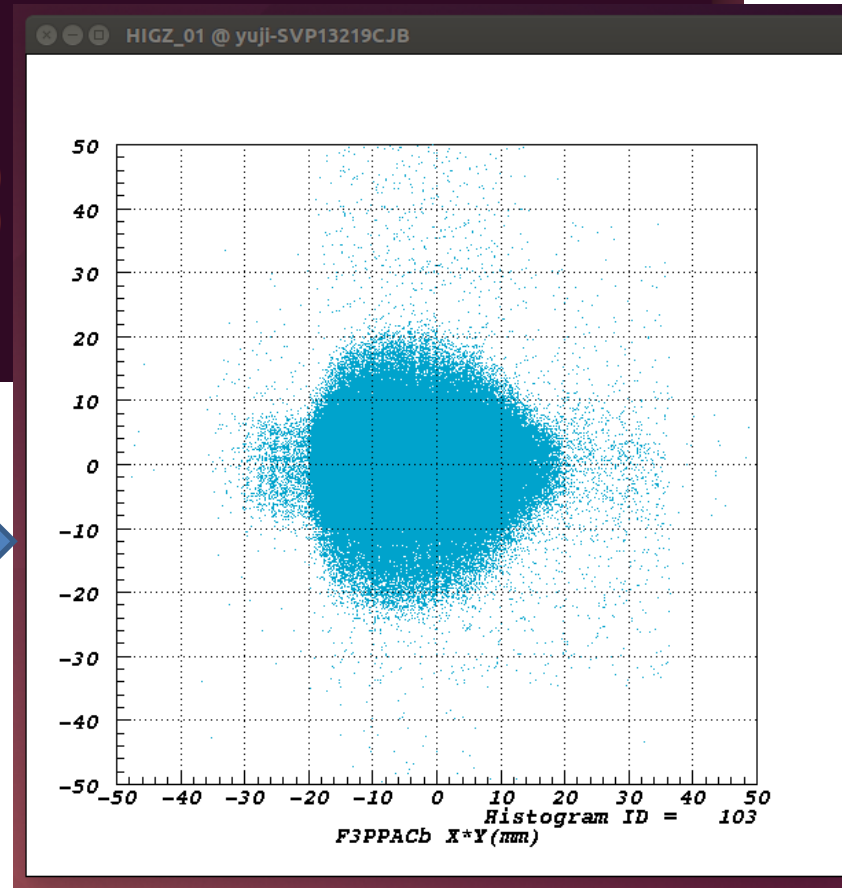
端末から抜粋

```
ANAPAW/EVTLOOP> i  
> i  
  
==> Histogram List
```

| | # | HID | Kind | Title |
|----|---|-----|------|-----------------|
| -> | 1 | 101 | (1) | PPACa cal |
| | 2 | 102 | (2) | F3PPACa X*Y(mm) |
| | 3 | 103 | (2) | F3PPACb X*Y(mm) |

```
ANAPAW/EVTLOOP> ht 103  
> ht 103
```

「i」で表示されたヒストグラムの中から目的のヒストグラムを選び、
「ht HID」
でウィンドウにヒストグラムが表示されます。



ヒストグラムを見るまで

コマンドの参考

コマンドを調べるにはRiken RIBF のチュートリアルが参考になります。

https://ribf.riken.jp/RIBFDAQ/index.php?plugin=attach&refer=Tools%2FAnalysis%2FANAPAW%2FTutorial&openfile=riken_guide.pdf

日本語のサイトも参考になります。

http://be.nucl.ap.titech.ac.jp/~sako/ANAPAW/ANAPAW_command.html

目的に応じて必要なコマンドを調べて端末から入力してみましょう。

xf, opt logz, opt linz, xsta, hcut, prx...

ヒストグラムを定義する

ヒストグラムを自分で定義するには

- anafileに必要なとなる Analyzer ID, 検出器 ID, 変数 ID を把握すること
- anafileの構造を把握すること
- gateの定義の仕方を把握すること
- ヒストグラムの定義の仕方を把握すること

が必要です。

ヒストグラムを定義する

Analyzer ID

検出器などで測定した物理量は用途別にディレクトリ「src」のenc_psd.f, enc_ppac.f等のソースコード内部の変数として扱われています。

ソースコードには1つずつAnalyzer IDが割り当てられています。

anafileでヒストグラムを定義するのに目的のソースコードのAnalyzer IDは何なのか調べたいとき、または別の人が書いたanafileのAnalyzer IDがどのソースコードを意味するのかを調べたいときは、ディレクトリ「src」内にある「usersrc.f」を参照してください。

ヒストグラムを定義する

Analyzer IDとソースコード

usersrc.fを見てみると、図のようになっています。

Analyzer IDは、2

ソースコード
「enc_ppac.f」
が対応する。

```
c
c Analyzer 2: PPAC
c
      If(AnalyzerFlag(2) Then
        call FND_EID(EvtData(1), 2, iadr_sub, nw_sub)
        If(nw_sub.le.0) Then
          EVTERR = .TRUE.
          Return
        EndIf
        Call Enc_PPAC( EvtData(iadr_sub), nw_sub,
          & val(1, 1, 2), nx, ny, naok(2) )
      EndIf
```

usersrc.f 一部抜粋

ヒストグラムを定義する

検出器 ID, 変数 ID

ソースコードには、検出器それぞれに割り当てられたIDと、検出器からの測定値由来の変数それぞれに割り当てられたIDが記載されています。

ヒストグラムを定義する

anafileの構造

anafileの構造は

- ① .Analyzer IDの指定部
- ② .xygateのファイルの指定部
- ③ .gateの定義部
- ④ .ヒストグラムの定義部

となっています。

・gate...ヒストグラムに入れるイベントの選別作用
gateをかけることで特定の領域内の変数を伴うイベントのみがヒストグラムに詰められます。

・xygate...2次元プロット上でのgate
「hcut」というコマンドを参照してください。

あるanafileを見ながら具体的な構造と文法を見ていきましょう。

ヒストグラムを定義する

anafileの構造

```
yuji@yuji-SVP13219CJB: ~  
ana[lys  
1  
2  
4  
5  
8  
6  
c  
c  
c coin, single, pileup  
xygate  
@, anapaw.cut  
gate  
11, 4, 1, 1, 3, 0.5, 1.5  
12, 4, 2, 2, 3, 1.5, 2.5  
13, 4, 3, 3, 3, 2.5, 3.5
```

①. Analyzer ID の指定部

②. xygate のファイルの指定部

③. gateの定義部

④. ヒストグラムの定義部

③. gateの定義部

と

④. ヒストグラムの定義部

での決まりを理解すれば大体のことがので、この二つを以下で説明します。

```
hst1  
0, 2, 1, 1, 24, 1000, -50., 50., 'PPACa X(mm)'
```

```
hst2  
0, 2, 1, 1, 24, 2, 1, 1, 25, 1000, -50., 50., 1000, -50., 50., 'F3PPACa X*Y(mm)'  
0, 2, 2, 2, 24, 2, 2, 2, 25, 1000, -50., 50., 1000, -50., 50., 'F3PPACb X*Y(mm)'  
exit
```

あるanafileから抜粋

ヒストグラムを定義する

gateの定義

- ③. gateの定義部での数字にはそれぞれ意味があり、書く順番にも決まりがあります。
画像をもとに説明します。

あるanafileの一部抜粋

```
gate  
11, 4, 1, 1, 3, 0.5, 1.5  
12, 4, 2, 2, 3, 1.5, 2.5  
13, 4, 3, 3, 3, 2.5, 3.5
```

作成されるgateのIDをここで指定する。

該当する変数のgateをかける範囲。
ここでは1.5~2.5となる。

Analyzer ID, 検出器 ID, 検出器 ID, 変数 ID
という並び
gateの対象となる変数をこれによって指定している。

ヒストグラムを定義する

gateの定義

上に挙げたやり方で作られたgateから
or や and をとった新しいgateを作ることができます。

or で新しいgateを作る。

あるanafileの一部抜粋

```
or  
34, 32, 33, 36  
54, 52, 53, 56  
59, 57, 58  
and  
35, 34, 31  
55, 54, 31  
60, 59, 31
```

新しいgate ID

or をとるgate ID

and で新しいgateを作る。

and をとるgate ID

新しいgate ID

作られたgateは④. ヒストグラムの定義部に組み込まれ、ヒストグラムにイベントが詰め込まれていく際に作用します。

ヒストグラムを定義する

ヒストグラムの定義

- ④. ヒストグラムの定義部は図のように「hst1」から始まる1次元ヒストグラムの定義部と、「hst2」から始まる2次元ヒストグラムの定義部に分かれています。

```
hst1
0, 2, 1, 1, 24, 1000, -50., 50., 'PPACa X(mm)'
█

hst2
0, 2, 1, 1, 24, 2, 1, 1, 25, 1000, -50., 50., 1000, -50., 50., 'F3PPACa X*Y(mm)'
0, 2, 2, 2, 24, 2, 2, 2, 25, 1000, -50., 50., 1000, -50., 50., 'F3PPACb X*Y(mm)'
exit
```

あるanafileの一部抜粋

図のようにそれぞれ意味のある数字を順番通りに並べていくと、ヒストグラムの定義ができます。以下数字の意味と順番を解説します。

ヒストグラムを定義する

ヒストグラムの定義

1次元ヒストグラム

ヒストグラムの形の定義
bin数, 最小値, 最大値 の順

ヒストグラムの名前

gate ID
ここでgateを
指定できる

```
hst1  
0. 2, 1, 1, 24, 1000, -50., 50., 'PPACa X(mm)'
```

あるanafileの一部抜粋

ヒストグラムに詰める変数の指定。
Analyzer ID, 検出器 ID, 検出器 ID, 変数 ID の順

ヒストグラムを定義する

ヒストグラムの定義

2次元ヒストグラム

横軸の定義

数字の意味と順番は1次元のものと同じ。

```
hst2
0, 2, 1, 1, 24, 2, 1, 1, 25, 1000, -50., 50., 1000, -50., 50., 'F3PPACa X*Y(mm)'
0, 2, 2, 2, 24, 2, 2, 2, 25, 1000, -50., 50., 1000, -50., 50., 'F3PPACb X*Y(mm)'
exit
```

縦軸の定義

数字の意味と順番は1次元のものと同じ。

あるanafileの一部抜粋

exit がanafileの終わりを意味しています。以降が無視されるので、デバッグなどにも活用することができます

「book」というコマンド入力の際に書き換えたanafileを指定しLoopすれば、目的のヒストグラムを得ることができます。

ソースコード内の変数

解析の為には、エネルギーなどの測定値、に由来する変数を計算する必要があります。

その計算の為には変数の追い方を把握する必要があります、その初歩を説明します。

ソースコード内の変数

測定値由来の変数はAnapawでは「val1」や「val2」などと名付けられた配列でソースコード間で共有されています。(恐らく)

enc_recoil.fを見てみると図のような記載があります。

```
c Input:
c   Analyzer #5 ENC_PSD    --> val1
c   Analyzer #8 ENC_PSD_VME--> val2
c -----
```

enc_recoil.fから抜粋

「val1にはenc_psd.fで計算されている変数が、val2にはenc_psd_vme.fで計算されている変数が格納されている。」という意味です。

ソースコード内の変数

val1,2は配列であり、どの要素がどの測定値由来の変数に該当するのかを把握する必要があります。

enc_recoil.fを見てみると

```
c First layer... PSD 3,2,1

AdE1 = val1(5, 151) ! dE1X MeV
AdE2 = val1(5, 153)
AdE3 = val2(8, 24)

ID = val1(2, 151) + 30
T1a= val2(7, ID)

ID = val1(2, 153) + 30
T2a= val2(7, ID)
```

enc_recoil.fから抜粋

図のようにval1(5,151)などと記載されています。
このval1(5,151)が何の物理量を表しているかを調べるやり方を説明します。

ソースコード内の変数

val1(5,151)の「5,151」という数字は変数 IDと関係しており、変数 IDは通常ソースコードの上部に記載されていると上で説明しました。enc_recoil.fの中ではval1はenc_psd.fから来た配列でした。enc_psd.fの上部を見てみると図のようになっています。

```
ID: 151 Strip with highest pulse height (PSD1-x)
     152                               (PSD1-y)
     153 Strip with highest pulse height (PSD2-x)
     154 Strip with highest pulse height (PSD2-y)
     155 Strip with highest pulse height (PSD3-x)
     156 Strip with highest pulse height (PSD3-y)
Word#: 1      2      3      4      5
       ID      ID_max Acal_max Tmax  Ecal_max
       6      7
       T-RF1 T-RF2

Tmax: valid for PSD-1x & PSD-2x
Acal_max: Maximum Acal of the plane
Ecal_max[MeV] = Acal_max * CommonGain
```

enc_psd.fから抜粋

「val1(5,151)」の「5」に対応するのが恐らく「Word#」で「5」は「Ecal_max」です。要素を表す左側の数字はこの「Word#」のようです。

「151」に対応するのが変数 IDである「151」で「Strip with highest pulse height (PSD1-x)」であり、val1(5,151)は「PSD1-xのストリップの中で最も高いエネルギーを表している。」と推定できます。

ソースコード内の変数

ソースコード上部に載っているコメントアウトされた文をもとにして変数の追い方を説明しましたが、CRIBのソースコードは色々な人が書き換えるので、コメントアウトの文と実際の変数が対応していない可能性があります。

コメントアウトの文を参考にしながらコードを読んでいき、本当に対応しているのか、確認をしておきましょう。

ソースコードを書き換えた場合、コンパイルしなければ変更が反映されません。Anapawのmakeは「makeana」です。

ソースコードで目的の物理量を計算できるようになり、anafileでヒストグラムの定義ができるようになり、コマンドライン上でAnapawを操作できるようになれば、Anapawでの解析の初歩はできるようになったといえます。

実際のデータを自分の頭を使い解析していき、わからないことは調べ、覚えていくことでAnapawを使いこなせるようになると思います。

Anapawインストールの参考

CRIBに在籍していたDaid KahlのWebサイトにAnapawのインストール方法が載っています。

<http://www.cns.s.u-tokyo.ac.jp/~daid/physix/anapaw.html>

CRIBで実験したDaheeがAnapawのインストールを文章にまとめてくれたものを、この説明書と同じディレクトリに置いておきます。

著者はDaheeのやり方が上手いきました。それでもなお、フォントのバグ、hcutのバグ、のデバッグが必要みたいです。デバッグができた方は是非文章化してこの説明書に加えてください。