

RubyDCL -- DCLの直接のラッパー

AdvancedDCL -- Rubyでの利用のための
さらなるラッパー

京都大学宙空電波科学研究センター

川那辺 直樹

DCLとは

1次元、2次元の描画と3次元の限られた描画機能
地図投影機能、座標変換機能、欠損値処理
が利用できる可視化ライブラリ

対応する言語

F77, F90, C, **Ruby**(Ruby-DCL)

•RubyDCL

DCLの1対1のインターフェースを持つライブラリ

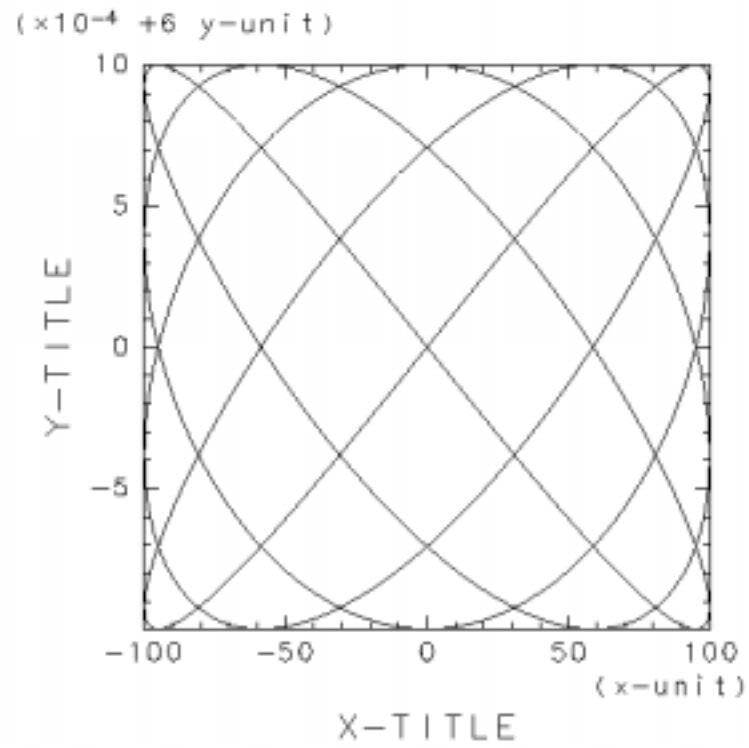
全サブルーチンはDCLというモジュールのメソッドとなっている。メソッド名はDCL(F77)のサブルーチン名をそのまま用いている。

**pset/get はサポートしていない。

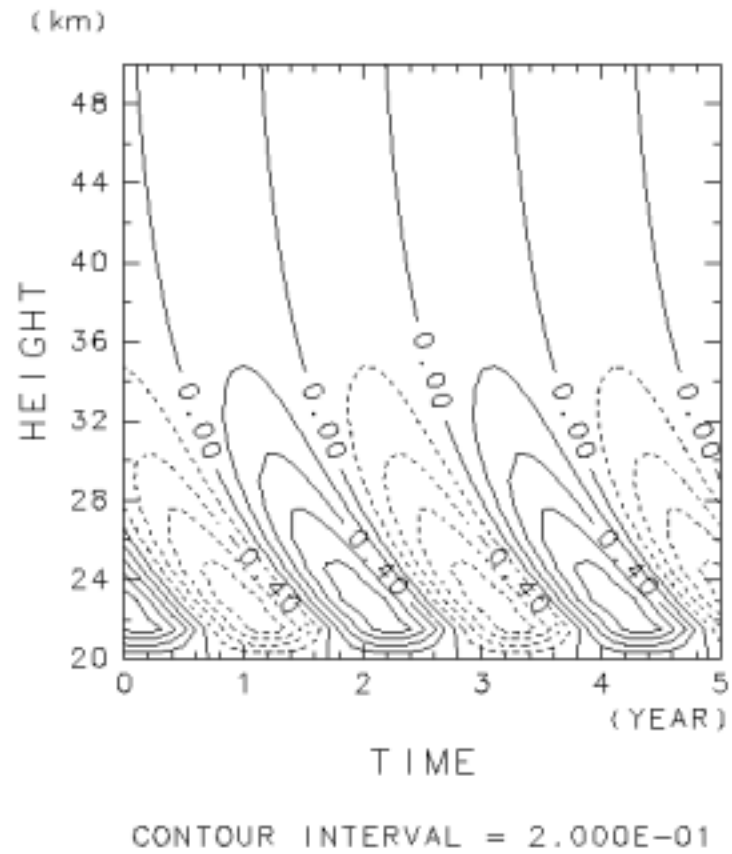
使用にあたって

多次元配列クラスライブラリNArrayが予めインストールされている必要がある。

hop.rb



u2d1.rb



•発展型描画ライブラリ(AdvancedDCL)

RubyDCLはRubyでの対話的な利用に必ずしも使いやすくはない。



発展型ライブラリ(AdvancedDCL)を開発

RubyDCLとの違い

- パッケージの再構成を行った。

 - それぞれのパッケージはモジュールとして実装されている。

- パラメタに関する変更

 - ◆パラメタ設定方法を2種類とした。

 - 一度設定されると再設定されるまで有効な設定

 - 一回の描画限り有効となる、テンポラリなパラメタ設定

Common	なし
Dev	SWPACK
Page	SLPACK
Frame	SLPACK
Fig	SGPACK, GRPACK, UMPACK, SCPACK
Axis	USPACK
Axis::Lin	U[XYZ]PACK
Axis::Log	ULPACK
DateAxis	UCPACK
Plot	UUPACK, SGPACK
Contour	UDPACK
Tone	USPACK, SGPACK
Vector	UGPACK
Map	UMPACK
Text	SGPACK
Arrow	SGPACK
Mark	UUPACK, SGPACK
ThreeD	SCPACK
Bar	UUPACK
Box	UUPACK
Gap	UUPACK
Errbar	UUPACK

パラメタに関する変更

- 一度設定されると再設定されるまで有効な設定

各モジュールに用意されているsetメソッドにより行う。

```
Plot.set("lineindex", 3)
```

コンターのラインインデクスを3に設定

```
Plot.lineindex=3
```

 として設定することも可能

パラメタ名がメソッド名となっている。

DCLではフレーム再設定などでリセットされる値(ウィンドウなど)を再設定せずに済む。

- テンポラリなパラメタ設定

各モジュールの描画メソッドにオプション引数として与えることで実現する。

```
Plot.draw(x, y, "lineindex"=>3)
```

プロットが終わるとlineindexは元に戻る。

これらはパラメタクラス(Parameter)を用いて実現している。

Parameterクラス

@parent	親パラメタ(Parameter)
@value	パラメタの値(Hash)
@typeinfo	パラメタの値のクラス(Hash)
@set_funcname	DCLにパラメタを設定するためのサブルーチン名(Hash)
@get_funcname	DCLからパラメタの値を得るためのサブルーチン名(Hash)
@set_call_pattern	サブルーチンを呼び出す際の引数の与え方(Hash)
@get_call_pattern	サブルーチンを呼び出す際の引数の与え方(Hash)
@sync	DCLとsyncしているか(true or false)

@value : {"nlev"=>15} **パラメタ名がHashのキーとなっている。**

pattern :	0	内部変数名とその値の組	udiset("nlev", 20)
	1	値のみ与える	uuslni(3)
	2	値が複数である。	grswnd(0.0, 10.0, 0.0, 10.0)

主なメソッド

`initialize(parent)`

初期化を行う。

`init(name, type, set_funcname, get_funcname, set_call_pattern, get_call_pattern)`

パラメタ名などをセットする。(値以外)

`[](name)`

パラメタ名に対応する値を取り出す。

`[]=(name, val)`

パラメタの値の設定

`sync`

設定されている値をDCLに反映させる。

`set(param_hash)`

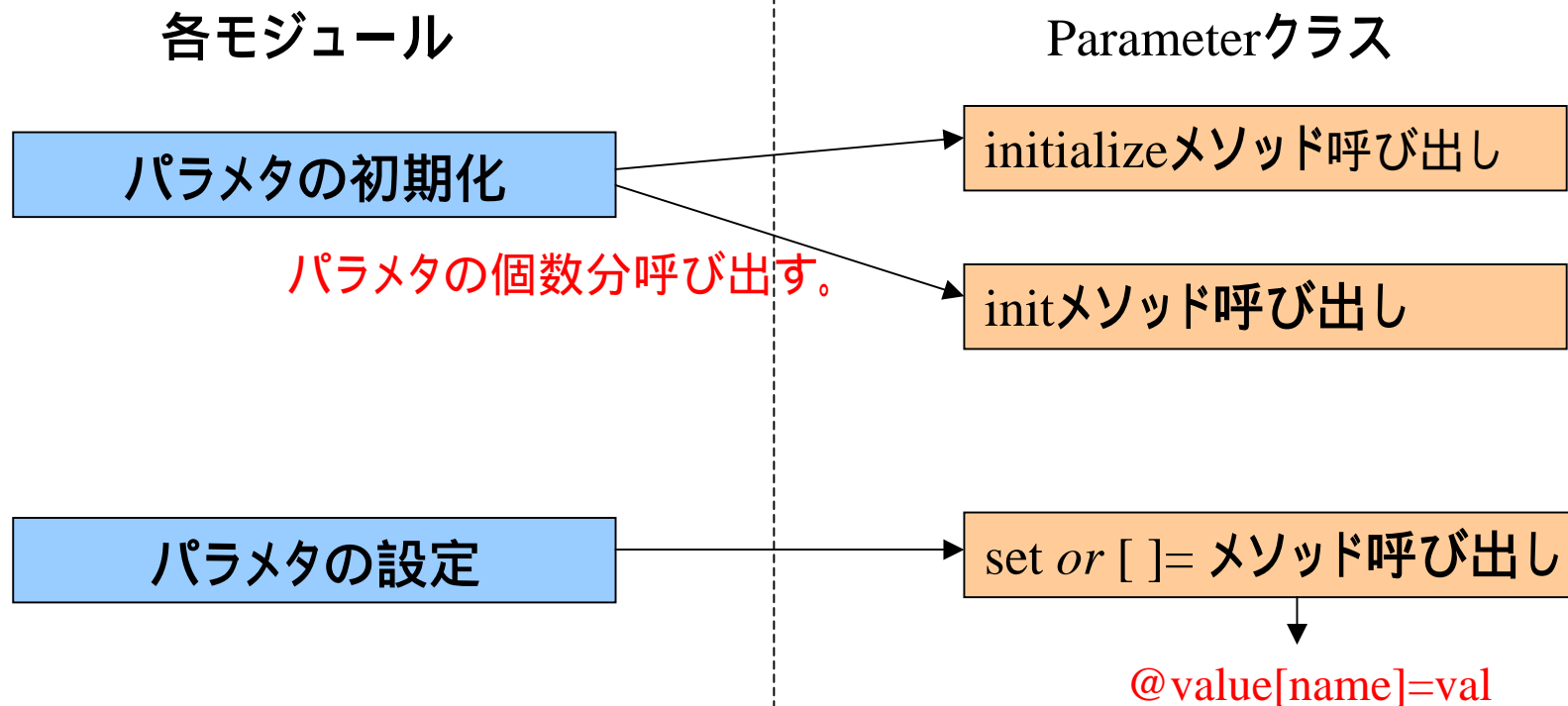
パラメタの値の設定(複数)、変更前のパラメタオブジェクトが返り値となる。

各モジュールは

@param	通常のパラメタ
@temp_param	テンポラリな設定用のパラメタ
@backup	テンポラリな設定後元に戻すためのバックアップ
@orig_param	リセット機能のための初期値を収めるパラメタ

の4つのパラメタオブジェクトを持つ。

Parameterクラスによるパラメタ設定



テンポラリなパラメタ設定

描画メソッド内

@param.sync

現在の設定をsyncする。

@backup = @temp_param.set(temp_param, @param)

テンポラリなパラメタ
を設定する

@temp_param.sync

テンポラリなパラメタをsyncする。

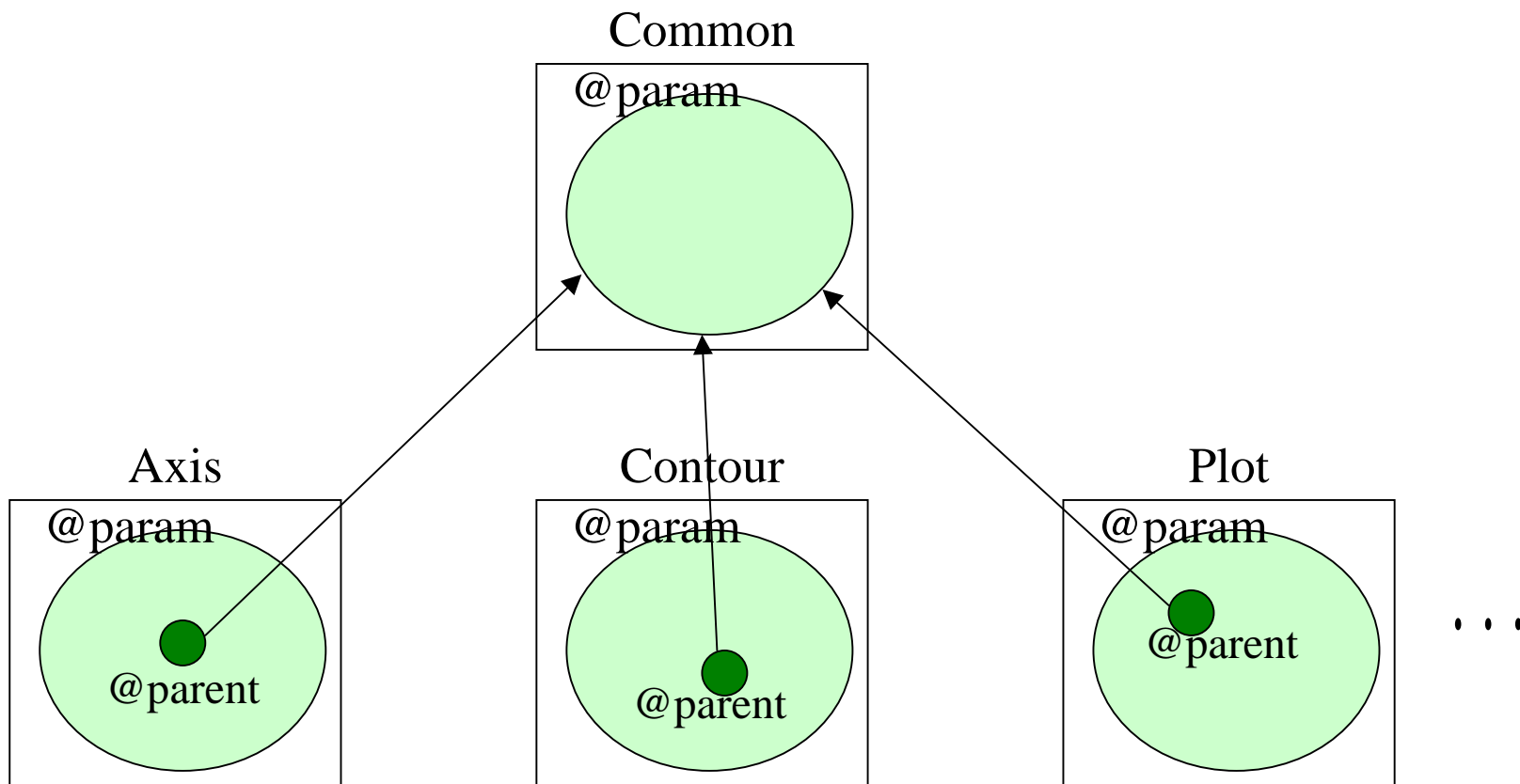
@backup.sync

テンポラリな設定を元に戻す。

テンポラリパラメタが無ければ呼ばない。

共通パラメタ

各モジュールの@paramの内部変数@parentをCommonモジュールの持つパラメタオブジェクト(@param)にしておく。



syncメソッドは値が設定されていないとき、@parentで値が設定されているか見に行く。あればその値をDCLにsyncさせる。

使い方

hop.rb

```
require "narray"
```

```
require "numru/advanceddcl"
```

```
include NumRu::AdvancedDCL
```

```
include NMath
```

```
Dev.open
```

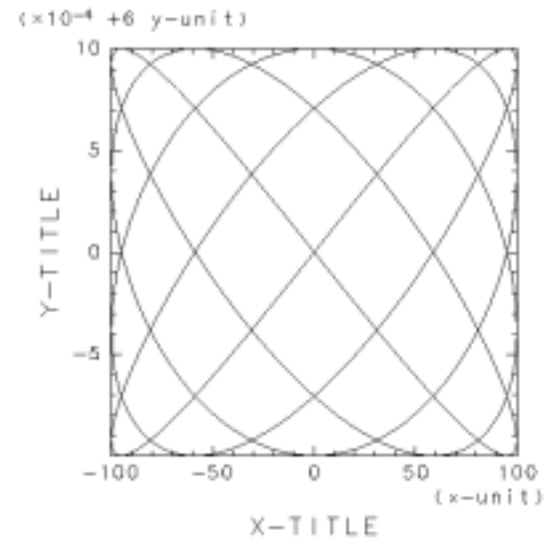
```
Frame.new
```

```
Axis.title('X-TITLE', 'x-unit', 'Y-TITLE', 'y-unit')
```

```
Axis.draw(x, y)
```

```
Plot.draw(x, y)
```

```
Dev.close
```



hop.rb (RubyDCL)

```
u2d1.rb
```

```
require "narray"
```

```
require "numru/advanceddcl"
```

```
include NumRu::AdvancedDCL
```

```
include NMath
```

```
Dev.open
```

```
Frame.new
```

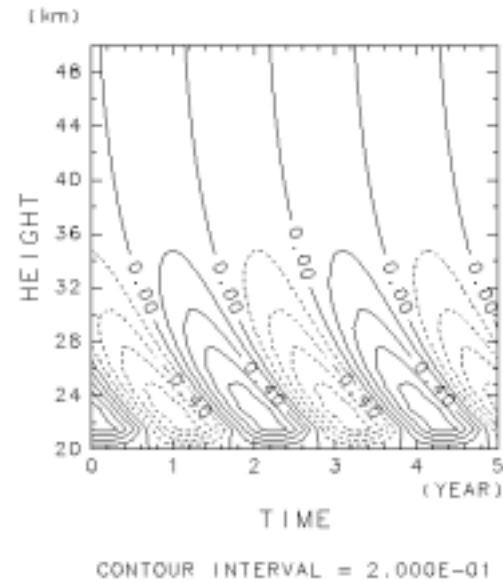
```
Fig.window=[tmin, tmax, zmin, zmax]
```

```
Axis.title('X-TITLE', 'x-unit', 'Y-TITLE', 'y-unit')
```

```
Axis.draw
```

```
Contour.draw(u)
```

```
Dev.close
```



```
u2d1.rb(RubyDCL)
```


kihon4.rb

SG+XV
INDEX3
INDEX5
SMALL
LARGE
LEFT
RIGHT
ROTATION+
ROTATION-