

Doxygen

~ IGMModel を一例にした，
数値モデルのドキュメンテーションにおける Doxygen の利用

神戸大学 地球及び惑星大気研究室 M1

河合 佑太



はじめに

- Doxygen イン트로ダクション
 - Doxygen とは？
 - Doxygen の主な特徴
- 数値モデル開発における Doxygen の利用
 - dcmmodel におけるドキュメンテーション方法
 - Doxygen と RDoc の比較
 - シンタックス
 - 生成されるドキュメント
 - IGMModel における Doxygen の利用例
- まとめ

Doxygen インTRODダクシヨN

- * Doxygen とは ?
- * Doxygen の主な特徴

Doxygen(ドキシジェン)とは？

Dimitri van Heesch たちによって開発されたドキュメンテーションシジェネレーター

(Doxygen 公式サイト : <http://www.doxygen.org/>)

- 最新バージョン : 1.7.5.1 (2011-10-07 現在)
- 動作環境 (OS) :
GNU/Linux, Microsoft Windows, Mac OSX, Solaris など
- ライセンス : GNU GPL
- 対応プログラム言語 :
C++, C, Java, Python, IDL, C#, Fortran など

Doxygen の主な特徴 1

- 生成されるドキュメントの種類
 - オンラインドキュメントブラウザ (HTML 形式)
 - オフラインリファレンスマニュアル (LATEX 形式)
 - RTF, Post script, ハイパーリンク付き PDF, 圧縮 HTML, Unix man ページ形式もサポート
- コード構造の可視化
 - include 依存関係, 継承ダイアグラム, コラボレーション図, コールグラフの自動生成

Doxygen の主な特徴 2

- ソースコードのドキュメンテーションに限らず，普通のドキュメントも作成可能．
- 400 を越えるプロジェクトのドキュメンテーションに利用されている．
 - 例：GNU Standard C++ Library, MySQL, Open MPI, VTK など
(<http://www.stack.nl/~dimitri/doxygen/projects.html>)
- Fortran ソースに対応．
 - 気象分野では Fortran を用いる傾向が強いが，Fortran に対応している有名なドキュメンテーションツールは意外に少ない(?)．

数値モデル開発における **Doxygen** の利用

- * dcmmodel におけるドキュメンテーション方法
- * Doxygen と RDoc のシンタックス・生成物の比較
～ IGMModel における Doxygen の利用例

dcmodel におけるドキュメンテーション方法

- DCPAM, deepconv, spmodel 等では，従来から (Fortran 90/95 用に拡張した) RDoc を利用．
- IGModel では，実験的に Doxygen を利用．
 - もともと Doxygen を使った理由は IGModel のコードに Fortran 2003 の仕様を一部使っていたから．
 - Rdoc と Doxygen の使用感や自動生成されるドキュメントの品質の違いを評価するのがねらい．

dcmodel

DCPAM

spmodel

agcm 5

ispack

deepconv

IGModel

BPmodel

etc

<dcmodel プロジェクト>

- 地球流体電脳倶楽部の数値モデルプロジェクト

(<http://www.gfd-dennou.org/library/dcmodel/>)

シンタックス ~ RDoc との比較 1

[Doxygen]

```
!> ¥brief Initialize a variable of derived type SWSolver.  
!!  
!! - Todo  
!! - Implement unit test for SWSolverModInit  
!! @attention You must call this pricedure before calling any procedeures in this  
!! module.  
!!  
!! @param[in,out] swsol The variable of derived type SWSolver.  
subroutine SWSolverModInit(swsol, param)  
  Type(Aggrs), intent(inout) :: swsol  
  Type(ParamList), intent(in) :: param !< The variable containg the simulation ..
```

[RDoc]

```
subroutine SWSolverModInit(swsol, param)  
  !  
  ! Initialize a variable of derived type SWSolver.  
  !  
  !== Todo  
  ! * Implement unit test for SWSolverModInit  
  !== Attention  
  ! * You must call this pricedure before calling any procedeures in this module.  
  Type(Aggrs), intent(inout) :: swsol ! The variable of derived type SWSolver.  
  Type(ParamList), intent(in) :: param ! The variable containg the simulation ..
```

シンタックス ~ Rdoc との比較 2

[Doxygen]

```
!> ¥brief Calculate the sum of the specified array elemnts.  
!!  
!! This subroutine calculates the following equation:  
!! ¥f[  
!!     ¥sum_{i=1}^{n} a_i  
!! ¥f]  
subroutine Sum(array, n)
```

[RDoc]

```
subroutine Sum(array, n)  
! Calculate the sum of the specified array elemnts.  
!  
! * This subroutine calculates the following equation:  
! ¥[  
!     ¥sum_{i=1}^{n} a_i  
! ¥]
```

生成されるドキュメント ～ RDoc との比較

```
subroutine SWSolverModInit ( Type(Aggrs),intent(in)      swsol,  
                           Type(ParamList),intent(in)  param  
                           )
```

[Doxygen]
HTML format

Initialize a variable of derived type SWSolver.

- Todo
 - Implement unit test for SWSolverModInit

Attention:

You must call this procedure before calling any procedures in this module.

Parameters:

[in,out] **swsol** The variable of derived type SWSolver.
[in] **param** The variable containing the simulation parameter.

Definition at line 10 of file test1.f90

SWSolverModInit(*swsol*, *param*)

Subroutine :

swsol : Type(Aggrs), intent(inout)
: The variable of derived type SWSolver.
param : Type(ParamList), intent(in)
: The variable containing the simulation parameter.

Initialize a variable of derived type SWSolver.

Todo

- Implement unit test for [SWSolverModInit](#)

Attention

- You must call this procedure before calling any procedures in this module.

[RDoc]
HTML format

生成されるドキュメント ～ RDoc との比較

2.1.1 Function Documentation

2.1.1.1 subroutine SWSolverModInit (Type(Aggrs),intent(in) swsol, Type(ParamList),intent(in) param)

Initialize a variable of derived type SWSolver.

- Todo
 - Implement unit test for SWSolverModInit

Attention

You must call this procedure before calling any procedeures in this module.

Parameters

in, out	<i>swsol</i>	The variable of derived type SWSolver.
in	<i>param</i>	The variable containg the simulation parameter.

[Doxygen]
PDF format(latex)

[Doxygen]
man format

```
Functions
subroutine SWSolverModInit (swsol, param)
    Initialize a variable of derived type SWSolver.

Function Documentation
subroutine SWSolverModInit (Type(Aggrs), intent(in)swsol,
    Type(ParamList), intent(in)param)
    Initialize a variable of derived type SWSolver. .IP ". " 2 Todo

    · Implement unit test for SWSolverModInit

Attention:
    You must call this procedure before calling any procedeures in this
    module.

Parameters:
    swsol The variable of derived type SWSolver.
    param The variable containg the simulation parameter .

Author
Generated automatically by Doxygen from the source code.
```

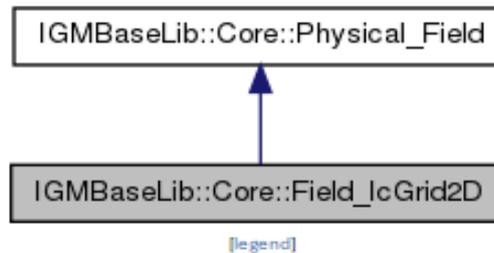
生成されるドキュメント ～コード構造の可視化 1

[Doxygen and GraphViz]
継承ダイアグラム, コラボレーション図

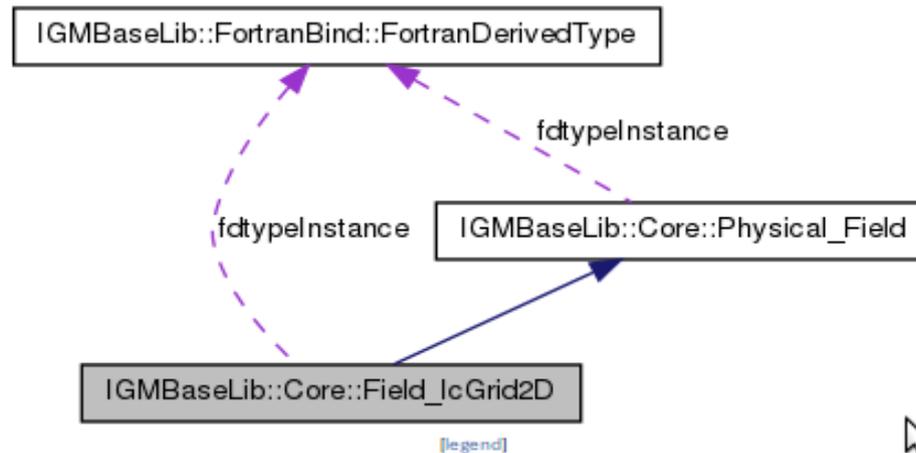
IGMBaseLib::Core::Field_IcGrid2D Class Reference

```
#include <Field_IcGrid2D_Manager.hpp>
```

Inheritance diagram for IGMBaseLib::Core::Field_IcGrid2D:



Collaboration diagram for IGMBaseLib::Core::Field_IcGrid2D:



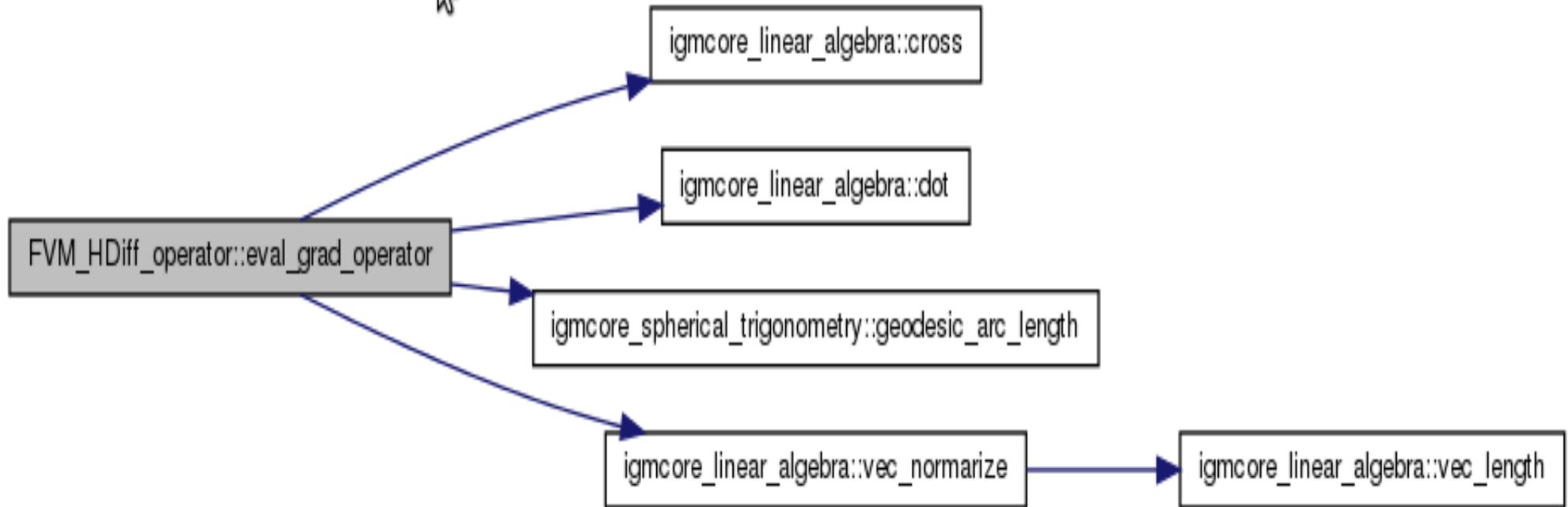
List of all members.

IGMBaseLib(IGModel project) の C++ ラッ
パー部分のソースコードのドキュメントの一部

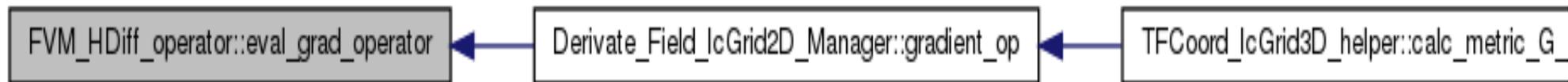
生成されるドキュメント ～コード構造の可視化 2

[Doxygen and GraphViz]
コールグラフ

Here is the call graph for this function:



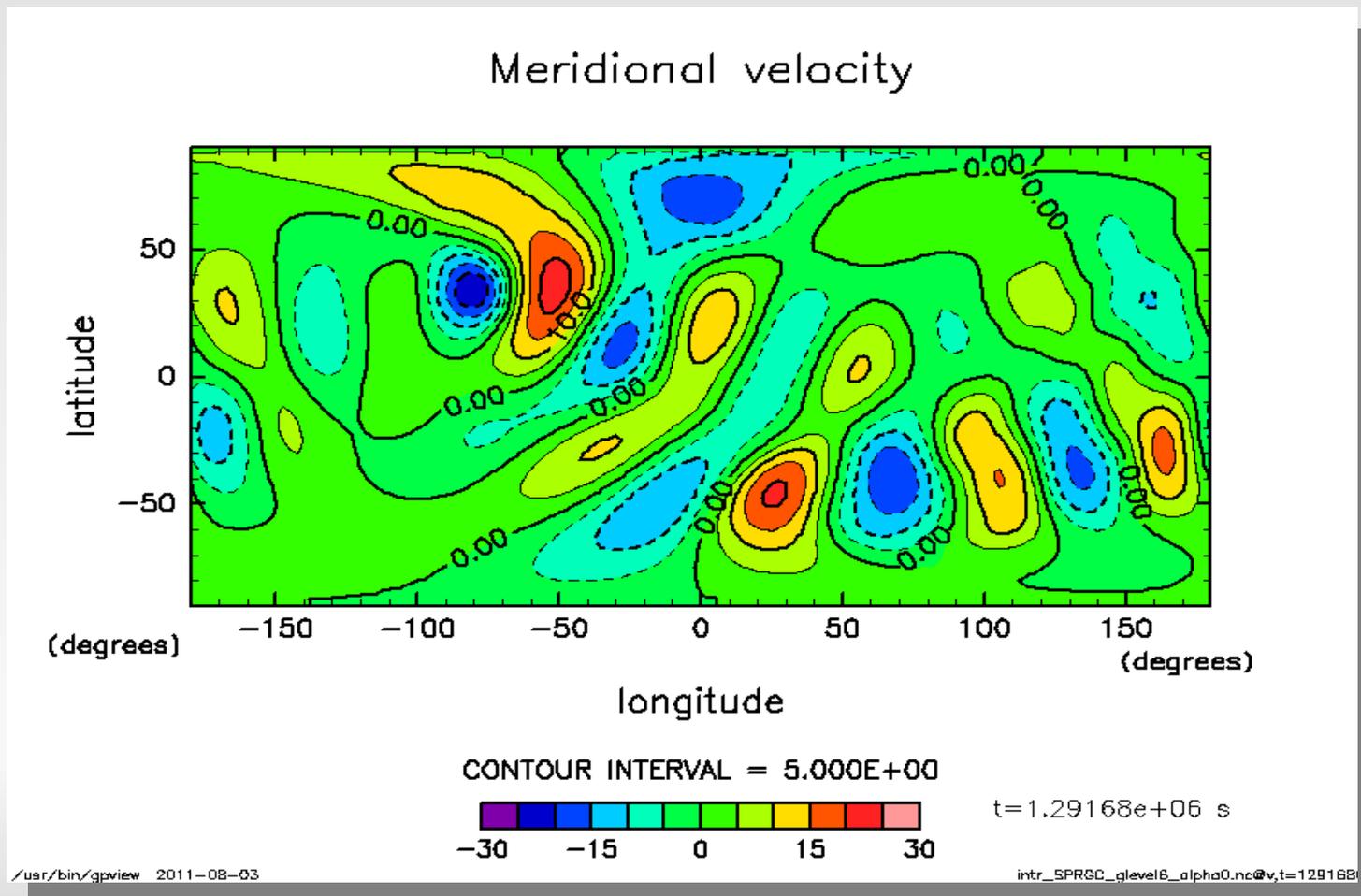
Here is the caller graph for this function:



IGMBaseLib(IGModel project) の微分演算を行う
Fortran ソースコードのドキュメントの一部

実際の利用例

- 正二十面格子大気モデル IGModel プロジェクト
<http://www.gfd-dennou.org/member/ykawai/work/IGModel.htm>



Doxygen と RDoc の比較

- シンタックス
 - Doxygen と RDoc 間のドキュメントの記述方法に，本質的な違いはあまりない．
 - RDoc の方がドキュメンテーション生成のために埋めこむ必要がある記述が少ない．
- 生成されるドキュメント
 - Doxygen が生成するドキュメントは，レイアウトや見た目が良い．また，出力できる形式が多様．
 - Doxygen は，継承・コラボレーション図，コールグラフなどを自動生成できる．
 - 完全な機能を使うには，Graphviz が必要．
 - このようなダイアグラムは，RDoc でも書ける (?)

まとめ

- ドキュメンテーション生成ツールに一つである Doxygen の紹介を行った。
 - 多数の出力形式, コード構造の可視化のサポート
 - Fortran ソースに対応
- (IGModel を例にして) 数値モデル開発における Doxygen の利用について, RDoc と比較しながら概要を説明した。
 - シンタックス
 - 生成されるドキュメントの様子

参考資料

- Doxygen 公式サイト

<http://www.doxygen.org/>

- RDoc 公式サイト

<http://rdoc.sourceforge.net/>

- RDoc Fortran 90/95 ソースコード解析機能強化版

<http://www.gfd-dennou.org/library/dcmmodel/rdoc-f95/>

- Graphviz

<http://graphviz.org/>

予備スライド

Doxygen の内部構造

<http://www.stack.nl/~dimitri/doxygen/arch.html>

