FDPSカーネルジェネレータ仕様書

Ver 0.0 — 2019/12/23

野村昴太郎

January 23, 2020

Contents

1 TODO

- カーネルジェネレータの仕様を決定する
- 関数をインライン展開する方法を考える
- FXXmat関連の演算や型推論が実装されていない
- "."演算子でFXXmatやFXXvecのメンバ変数を使った場合の実装がされていない
- FXXmatやFXXvecの演算を個別の演算に分解する機能が必要
- 現状勝手にアップキャストされてしまうが，低精度を維持したい場合，statementの先頭に型宣言ができる必要がある
-

2 履歴

2.1 2019/12/23

ファイルを作った.

2.2 2019/12/30

カーネルジェネレータ周りのTODOを追加.

3 はじめに

本書はFDPS向けのカーネル関数のソースコードジェネレータの仕様書である.

4 概要

FDPSのカーネル関数の最適化ソースコードを様々なアーキテクチャ向けに生成するためのジェネレータを作る．そのための言語仕様及びコンパイラの仕様を示す.

5 DSL仕様

本カーネルジェネレータは，これ向けのDSL(Domain Specific Language)で書かれたソースコードを元に，指定したアーキテクチャ向けに最適化されたC++のカーネルソースコードを生成する.

以下では本カーネルジェネレータで用いるDSLの仕様について定義する.

5.1 コメントアウト

二重のスラッシュ(//)以降のその行の文字列はコメントとして無視される.

5.2 予約語とその説明

- EPI (変数宣言時にEPIクラスの変数であることを示す)
- EPJ (変数宣言時にEPJクラスの変数であることを示す)
- FORCE (変数宣言時にFORCEクラスの変数であることを示す)
- static (変数宣言時にstaticメンバ変数であることを示す)
- function (関数宣言であることを示す)
- (S|U)(16|32|64) ((符号あり|なし)整数スカラー型)
- F(16|32|64)[vec|mat][2|3] (浮動小数点スカラー，ベクトル，マトリックス型)
- (sqrt|rsqrt|inv|min|max) (特殊関数．それぞれ平方根，逆数平方根，逆数，最小値，最大値)

5.3 利用可能な型

本DSLでは，FDPSで定義している(もしくはする予定の)整数型，スカラー型，ベクトル型，マトリックス型をサポートする．具体的には，以下の型が使用可能である．(現状，マトリックス型は対称行列のみをサポート)．

- (S|U)(16|32|64)
- F(16|32|64)
- F(16|32|64)vec[2|3]
- F(16|32|64)mat[2|3]

それぞれ，16 bit，32 bit，64 bitの(符号あり｜なし)整数型，浮動小数点スカラー型，浮動小数点ベクトル型，浮動小数点マトリックス型である．浮動小数点型は末尾に2もしくは3をつけることにより，次元を表すことができる．省略した場合は3次元である．

各型の値に演算子を作用させた場合の効果はFDPSのそれに準じる．ただし，+=や-=，*=，/=はFORCEクラスのメンバ変数にしか用いることができない．詳細は相互作用関数記述部5.6節で説明する．

5.4 変数宣言

関数宣言では，型の前にEPI，EPJもしくはFORCEをつけることでそれぞれのメンバ変数であることを示すことができる．また，先頭にstatic修飾子をつけることによりstaticメンバ変数にすることができる．また，変数宣言では変数名のあとにコロン(:)とFDPSのFull Particle Class(FP)のメンバ変数名を書かなくてはならない．これにより，EPI，EPJ及びFORCEクラスのどのメンバ変数とFPクラスのどのメンバ変数が対応関係にあるか示す事ができる．

```
EPI F32vec ri:r
EPJ F32vec rj:r
EPJ F32    mj:m
static EPI F32 eps:eps
```

EPI，EPJもしくはFORCEがつかない場合，カーネル関数オブジェクト初期化時に渡す引数として扱われる．この場合，コロンによって対応関係を示す必要はない．

```
F32 eps
```

5.5 関数記述部

本DSLでは，関数を記述し，相互作用関数記述部で利用することができる．関数記述部は，以下の記述になる．

```
function 関数名(引数0, $\cdots$){
  /* 計算部分 */
  return 返り値
}
```

関数記述内の計算部分では，予約語および変数宣言部にない任意のローカル変数を使用できる．型推論されるので，特に宣言なく使用できる．

5.6 相互作用関数記述部

相互作用関数記述部では，以下のような記述が可能である．

```
dr  = ri - rj
r2  = dr * dr
rinv = rsqrt(r2)
f += mj * ri * ri * ri * dr
u += mj * ri
```

相互作用関数記述部では，これまでに型を指定して宣言された変数をもとに型推論が行われるため，ローカル変数の宣言などを行う必要がなく，これまでに宣言されていない好きな変数名の変数に値を代入できる．

FORCEメンバ変数に対応する変数にのみ+=もしくは-=，*=，/=演算子を用いることができる(最大値とかはいらない？)．カーネルジェネレータは，EPIとEPJの2重ループを作り，その中でこれらの演算子を用いて書かれた記述をアキュムレートするカーネル関数を生成する．

また，カーネル記述部ではif文を用いることができる．

```
dr = ri - rj
r2 = dr*dr
if(r2 > 0.0 $&&$ r2 < 16.0){
  r2i = inv(r2)
  r6i = r2i * r2i * r2i
  f += 24 * r6i * r2i * (2*r6i - 1) * dr
  u += 4 * r6i * (r6i - 1)
}
```

6 カーネルジェネレータ仕様