# Module Interface Specification for STEM Moiré GPA

Alexandre Pofelski
macid: pofelska
github: slimpotatoes

November 16, 2017

# 1 Revision History

| Date | Version | Notes |
|---|---|---|
| 14/11/2017 | 1.0 | First draft |

# 2 Symbols, Abbreviations and Acronyms

The same Symbols, Abbreviations and Acronyms as in the SRS, the TestPlan and the MG (available in STEM Moiré GPA repository) are used in the Module Interface Specifications document.

addition to document

# Contents

# 3 Introduction

The following document details the Module Interface Specifications for STEM Moiré GPA. The full documentation and implementation can be found in STEM Moiré GPA repository.

# 4 Notation

The structure of the MIS for modules comes from [?], with the addition that template modules have been adapted from [?]. The mathematical notation comes from Chapter 3 of [?]. For instance, the symbol := is used for a multiple assignment statement and conditional rules follow the form $(c_1 \Rightarrow r_1|c_2 \Rightarrow r_2|...|c_n \Rightarrow r_n)$.

The following table summarizes the primitive data types used by STEM Moiré GPA.

| Data Type | Notation | Description |
|---|---|---|
| character | char | a single symbol or digit |
| integer | $\mathbb{Z}$ | an integer number |
| natural number | $\mathbb{N}$ | a natural number |
| real | $\mathbb{R}$ | a real number |

The specification of STEM Moiré GPA uses some derived data types: sequences, strings, and tuples. Sequences are lists filled with elements of the same data type. Strings are sequences of characters. Tuples contain a list of values, potentially of different types. In addition, STEM Moiré GPA uses functions, which are defined by the data types of their inputs and outputs. Local functions are described by giving their type signature followed by their specification.

# 5 Module Decomposition

The following table is taken directly from the Module Guide document for this project.

| **Level 1** | **Level 2** |
|---|---|
| Hardware-Hiding Module | |
| Behaviour-Hiding Module | Input<br>STEM Moiré GPA Control<br>STEM Moiré GPA GUI<br>User Input<br>SMH simulation<br>GPA<br>Mask<br>Unstrained region<br>Conversion<br>2D strain tensor |
| Software Decision Module | Fourier Transform<br>Least square fitting method<br>Phase calculation<br>Gradient<br>Generic GUI/Plot<br>Data structure<br>Object structure |

Table 1: Module Hierarchy

LIST ALL MIS to refer them in other document

# 6 MIS of Hardware Hiding Module (M 1)

## 6.1 Module

## 6.2 Uses

Data Structure

## 6.3 Syntax

### 6.3.1 Exported Access Programs

| Name | In | Out | Exceptions |
|---|---|---|---|
| | - | - | - |

## 6.4 Semantics

### 6.4.1 State Variables

### 6.4.2 Access Routine Semantics

():

- transition:
- output:
- exception:

# 7 MIS of STEM Moiré GPA Control Module (M 2)

## 7.1 Module

main

## 7.2 Uses

- STEM Moié GPA GUI
- Processing modules
  - Unstrained region
  - Conversion
  - SMH Simulation
  - GPA

– 2D Strain Tensors

- Data Structure

## 7.3 Syntax

### 7.3.1 Exported Access Programs

| Name | In | Out | Exceptions |
|---|---|---|---|
| | - | - | - |

## 7.4 Semantics

STEM Moiré GPA is designed to have the different steps of the process flow driven by user directly through GUI_SMG. The STEM Moiré GPA Control Module uses the events in STEM Moié GPA GUI to use the processing modules in the order defined by the user.

### 7.4.1 State Variables

### 7.4.2 Access Routine Semantics

():

- transition:
- output:
- exception:

# 8 MIS of STEM Moiré GPA GUI Module (M 3)

## 8.1 Module

GUI_SMG

## 8.2 Uses

- Generic GUI/Plot
- Data Structure

## 8.3 Syntax

### 8.3.1 Exported Access Programs

| Name | In | Out | Exceptions |
|---|---|---|---|
| | - | - | - |

## 8.4 Semantics

STEM Moiré GPA process flow is driven by user through GUI_SMG. User triggers the events that start the wished processing step.

### 8.4.1 State Variables

### 8.4.2 Access Routine Semantics

():

- transition:
- output:
- exception:

# 9 MIS of Imput Module (M 4)

## 9.1 Module

Input

## 9.2 Uses

- STEM Moié GPA GUI
- Data Structure

## 9.3 Syntax

### 9.3.1 Exported Access Programs

| Name | In | Out | Exceptions |
|---|---|---|---|
| | - | - | - |

## 9.4 Semantics

### 9.4.1 State Variables

### 9.4.2 Access Routine Semantics

():

- transition:
- output:
- exception:

# 10 MIS of SMH Simulation (M 5)

## 10.1 Module

SMHSim

## 10.2 Uses

- Fourier Transform
- Input
- Data Structure

## 10.3 Syntax

### 10.3.1 Exported Access Programs

| Name | In | Out | Exceptions |
|---|---|---|---|
| | - | - | - |

## 10.4 Semantics

### 10.4.1 State Variables

### 10.4.2 Access Routine Semantics

():

- transition:
- output:
- exception:

# 11 MIS of GPA Module (M 6)

## 11.1 Module

GPA

## 11.2 Uses

- Mask
- Fourier Transform
- Phase
- Gradient
- Data Structure

## 11.3 Syntax

### 11.3.1 Exported Access Programs

| Name | In | Out | Exceptions |
|---|---|---|---|
| | - | - | - |

## 11.4 Semantics

### 11.4.1 State Variables

### 11.4.2 Access Routine Semantics

():

- transition:
- output:
- exception:

# 12 MIS of Mask Module (M 7)

## 12.1 Module

GPA

## 12.2 Uses

- Mask
- Fourier Transform
- Phase
- Gradient
- Data Structure

## 12.3 Syntax

### 12.3.1 Exported Access Programs

| Name | In | Out | Exceptions |
|---|---|---|---|
| | - | - | - |

## 12.4 Semantics

### 12.4.1 State Variables

### 12.4.2 Access Routine Semantics

():

- transition:
- output:
- exception:

# 13 MIS of Unstrained region (M 8)

## 13.1 Module

URef

## 13.2 Uses

- Least Square Fit
- Input
- Data Structure

## 13.3 Syntax

### 13.3.1 Exported Access Programs

| Name | In | Out | Exceptions |
|---|---|---|---|
| | - | - | - |

### 13.4 Semantics

#### 13.4.1 State Variables

#### 13.4.2 Access Routine Semantics

():

- transition:
- output:
- exception:

## 14 MIS of Conversion Module (M 9)

### 14.1 Module

MtoCConv

### 14.2 Uses

- Input
- Data Structure

### 14.3 Syntax

#### 14.3.1 Exported Access Programs

| Name | In | Out | Exceptions |
|---|---|---|---|
| | - | - | - |

### 14.4 Semantics

#### 14.4.1 State Variables

#### 14.4.2 Access Routine Semantics

():

- transition:
- output:
- exception:

# 15 MIS of 2D Strain Tensor Module (M 10)

## 15.1 Module

2D_Strain

## 15.2 Uses

Data Structure

## 15.3 Syntax

### 15.3.1 Exported Access Programs

| Name | In | Out | Exceptions |
|---|---|---|---|
| | - | - | - |

## 15.4 Semantics

### 15.4.1 State Variables

### 15.4.2 Access Routine Semantics

():

- transition:
- output:
- exception:

# 16 MIS of Fourier Transform Module (M 11)

*# 2D Fourier transform*

## 16.1 Module

FTCalc

## 16.2 Uses

Data Structure

## 16.3 Syntax

### 16.3.1 Exported Access Programs

| Name | In | Out | Exceptions |
|---|---|---|---|
| $\mathcal{FT}$ | $f : \mathbb{R}^2 \to \mathbb{R}$ | $f : \mathbb{R}^2 \to \mathbb{C}$ | - |
| i$\mathcal{FT}$ | $f : \mathbb{R}^2 \to \mathbb{C}$ | $f : \mathbb{R}^2 \to \mathbb{R}$ | - |

## 16.4 Semantics

### 16.4.1 State Variables

None

### 16.4.2 Access Routine Semantics

*# Calculate the 2D Fourier transform of a function f*

$\mathcal{FT}(f(x, y))$:

- output: $\widetilde{f}(\nu, \mu)$ such that

$$\forall(\nu, \mu) \in \mathbb{R}^2 \wedge \forall(x, y) \in \mathbb{R}^2,\ \widetilde{f}(\nu, \mu) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-2i\pi(\nu x + \mu y)} dxdy$$

- exception:

*# Calculate the 2D inverse Fourier transform of a function $\widetilde{f}$*

i$\mathcal{FT}(\widetilde{f}(\nu, \mu))$:

- output: $f(x, y)$ such that

$$\forall(x, y) \in \mathbb{R}^2 \wedge \forall(\nu, \mu) \in \mathbb{R}^2,\ f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \widetilde{f}(\nu, \mu) e^{2i\pi(\nu x + \mu y)} dxdy$$

- exception:

# 17 MIS of Gradient Module (M 12)

*# 2D Gradient*

## 17.1 Module

GradCalc

## 17.2 Uses

Data Structure

## 17.3 Syntax

### 17.3.1 Exported Access Programs

| Name | In | Out | Exceptions |
|---|---|---|---|
| grad | $f : \mathbb{R}^2 \to \mathbb{R}$ | $f : \mathbb{R}^2 \to \mathbb{R}^2$ | - |

## 17.4 Semantics

### 17.4.1 State Variables

### 17.4.2 Access Routine Semantics

*# Calculate the 2D gradient of a 2D function f*

grad($(f(x, y))$):

- output:$\nabla f(x, y)$ such that

$$\forall (x, y) \in \mathbb{R}^2,\ \nabla f(x, y) = \begin{bmatrix} \frac{\partial f}{\partial x}(x, y) \\ \frac{\partial f}{\partial y}(x, y) \end{bmatrix}$$

- exception:

# 18 MIS of Least Square Fit Method Module (M 13)

*# 2D linear least square method to fit a function f*

## 18.1 Module

LSFMCalc

## 18.2 Uses

Data Structure

## 18.3 Syntax

### 18.3.1 Exported Access Programs

| Name | In | Out | Exceptions |
|---|---|---|---|
| lsfm | $f : \mathbb{R}^2 \to \mathbb{R}$ | $f : \mathbb{R}^2 \to \mathbb{R}$ | - |

## 18.4 Semantics

### 18.4.1 State Variables

### 18.4.2 Access Routine Semantics

*# Calculate the 2D fit of a function f using the linear least square method on a domain* $U = ([x_0, x_1]; [y_0, y_1]) \in \mathbb{R}^2$

lsfm(f,U):

- output: $fit(x, y) = ax + by$ such that

$$\forall (x, y) \in U,\ E(a, b) = \int_{x_0}^{x_1} \int_{y_0}^{y_1} [f(x, y) - fit(x, y)]^2 dxdy \ \text{ is minimized}$$

$$\Rightarrow \frac{\partial E}{\partial a} = 0 \wedge \frac{\partial E}{\partial b} = 0 \Rightarrow a = \frac{\int_{x_0}^{x_1} \int_{y_0}^{y_1} xf(x, y)dxdy}{\int_{x_0}^{x_1} \int_{y_0}^{y_1} x^2 dxdy} \wedge b = \frac{\int_{x_0}^{x_1} \int_{y_0}^{y_1} yf(x, y)dxdy}{\int_{x_0}^{x_1} \int_{y_0}^{y_1} y^2 dxdy}$$

- exception:

# 19 MIS of Phase Operation Module (M 14)

## 19.1 Module

PhaseCalc

## 19.2 Uses

Data Structure

## 19.3 Syntax

### 19.3.1 Exported Access Programs

| Name | In | Out | Exceptions |
|---|---|---|---|
| unwrap | $f : \mathbb{R}^2 \to [-\pi, \pi[$ | $f : \mathbb{R}^2 \to \mathbb{R}$ | - |
| wrap | $f : \mathbb{R}^2 \to \mathbb{R}$ | $f : \mathbb{R}^2 \to [-\pi, \pi[$ | - |

## 19.4 Semantics

### 19.4.1 State Variables

### 19.4.2 Access Routine Semantics

wrap($f$):

- output: $g$ such that

$$\forall(x,y) \in \mathbb{R}^2, \exists k \in \mathbb{Z} | g(x,y) = f(x,y) + 2k\pi \wedge g(x,y) \in [-\pi, \pi[$$

- exception:

unwrap($f$):

- output: $g$ such that

$$\begin{aligned} &\forall(x,y) \in \mathbb{R}^2, \exists k \in \mathbb{Z} | g(x,y) = f(x,y) + 2k\pi \wedge g \text{ is continous} \\ \Rightarrow &\forall(x,y) \in \mathbb{R}^2, \exists k \in \mathbb{Z} | \lim_{(x,y)\to(x_0,y_0)} g(x,y) = g(x_0,y_0) = f(x_0,y_0) + 2k\pi \end{aligned}$$

- exception:

# 20 MIS of Data Structure Module (M 15)

## 20.1 Module

DataStruct

## 20.2 Uses

## 20.3 Syntax

### 20.3.1 Exported Access Programs

| Name | In | Out | Exceptions |
|---|---|---|---|
| set | Metadata | - | - |

## 20.4 Semantics

### 20.4.1 State Variables

### 20.4.2 Access Routine Semantics

():

- transition:
- output:
- exception:

# 21 MIS of Generic GUI/Plot Module (M 16)

## 21.1 Module

GUIGene

## 21.2 Uses

Hardware-Hiding Data Structure

## 21.3 Syntax

### 21.3.1 Exported Access Programs

| Name | In | Out | Exceptions |
|---|---|---|---|
| | - | - | - |

## 21.4 Semantics

### 21.4.1 State Variables

### 21.4.2 Access Routine Semantics

():

- transition:
- output:
- exception:

# 22 Appendix