resp Documentation

Release 0.1

Me

Nov 23, 2017

CONTENTS:

1 Sparse Matrix Operations API **3**

1.1 Basic types 3

1.2 Constructors 3

1.3 Operations 3

2 Example of usage of the library **5**

3 Indices and tables **7**

Index **9**

This preliminary project has the aim of defining a API for dealing with Sparse Matrix operations in the most flexible way. We try to use SphinX equipped with the sphinx-fortran package with the objective of defining the API by using autodoc tools.

CHAPTER

ONE

SPARSE MATRIX OPERATIONS API

In this file we will put out proposition for the API of the sparsematrix operations to be performed. We will profit of this file also to learn the good prectice to create developer's documentation for the library

1.1 Basic types

type `resp/`**`resp_matrix`**

Basic entity of the library, represents a Matrix with a defined sparsity level Here we list the important attributes that we would like to expose

Type fields

- **`% ndummy`** *[integer]* :: Placehoder for some componenets of the structure we would like to expose

1.2 Constructors

We here indicate the basic routines for the construction of the derived types

function **`resp_matrix_create`**(*dict_options*)

initialize the resp_matrix object, that might be filled and allocated afterwards

Parameters **dict_options** *[dictionary,pointer]* :: dictionary containing the initialization options

Return **sm** *[*resp_matrix*]*

subroutine **`resp_matrix_set`**(*sm*)

allocate the memory

Parameters **sm** *[*resp_matrix*]* :: matrix to be allocated, should be initialized first

1.3 Operations

subroutine **`resp_gemm`**(*a*, *b*, *c*)

perform a matrix-matrix multiplication this routine is note meant to be a constructor for the matrix C

Parameters

- **a** *[*resp_matrix*,in]* :: use the documentation of *`resp_matrix`*
- **b** *[*resp_matrix*,in]*

- **c** *[*resp_matrix*,inout]* :: output matrix, should have bes set first

CHAPTER

TWO

EXAMPLE OF USAGE OF THE LIBRARY

In this section we might write some code snippets related on how the proposed API should work Ideally, the code snippets should be extracted from actual programs that will be used in the non-regression test of the library (as soon as I figure out how to do that...)

I now add another line in the index page

CHAPTER THREE

INDICES AND TABLES

- genindex
- modindex
- search

INDEX

R

resp_gemm() (fortran subroutine), **3**
resp_matrix (fortran type in module resp), **3**
resp_matrix_create() (fortran function), **3**
resp_matrix_set() (fortran subroutine), **3**