

Algorithm xxx: The 2D Tree Sliding Window Discrete Fourier Transform

Lee F. Richardson

William F. Eddy

August 6, 2018

1 Introduction

This package implements of the 2D Tree Radix-2 Sliding Window Discrete Fourier Transform (SWDFT). This short manual describes installation, compiling the executable programs, callable routines, and shows how to incorporate the package into a larger program.

2 Installation

Download the zip file, unzip it, then navigate to the directory where the files are extracted. There are three executable programs to compile: one (**driver2d**) is located in the *src/* directory, and the other two (**timing** and **stability**) are located in the *tests/* directory. Compile and run the **driver2d** executable with:

```
cd src
make
./driver2d
```

The expected output is contained in the file *src/driver2d-expected-output.txt*. Similarly, compile and run the **timing** and **stability** executables with:

```
cd tests
make
./timing
./stability
```

The expected outputs are in *tests/timing-expected-output.txt* and *tests/stability-expected-output.txt*, respectively.

3 Routines and Executables

This package contains three executable programs, which call our C functions implementing 2D SWDFT algorithms. This section gives brief descriptions of both the C functions and the executable programs that call them.

3.1 C Functions

Our package includes one primary C function: **tswdft2d**. This function implements the 2D Radix-2 Tree Sliding Window Discrete Fourier Transform, described in the corresponding

manuscript. Detailed documentation for `tswdft2d` is provided in the `src/tswdft2d.c` file. All the macros and constants used in this function are defined in `src/tswdft2d.h`. The inputs for the `tswdft2d` function are:

- **x**. double complex *. Row-major 2D array with dimensions $N_0 \times N_1$
- **n0**. int. Window size in row direction (must be a power of two)
- **n1**. int. Window size in column direction (must be a power of two)
- **N0**. int. Number of rows in **x**
- **N1**. int. Number of columns in **x**.

The output is:

- **a**. double complex *. Row-major 4D array with dimensions $(N_0 - n_0 + 1) \times (N_1 - n_1 + 1) \times n_0 \times n_1$. The first two dimensions correspond to window position, and the final two dimensions correspond to frequency.

We also include two other functions, `swdft2d` and `swfft2d`, for testing purposes. These functions take the same input and output as `tswdft2d`.

3.2 Executables

We provide the following three executables:

- **driver2d**. Randomly generate a 2D complex-valued array, run the `tswdft2d` function on the 2D array, print the output.
- **timing**. Generate a 100×100 array, run three different algorithms on the array with varying window sizes, print out how long each algorithm takes.
- **stability**. Verifies that the `tswdft2d` and `swfft2d` functions give identical results.

3.3 Larger Programs

The `tswdft2d` function can be used in larger programs. First, include the following header files:

```
#include "tswdft2d.h"
#include "complex.h"
```

Then allocate a double complex pointer, run the program, and free the memory:

```
double complex *a;
a = tswdft2d(*x, m0, m1, N0, N1);
free(a);
```

where **x** is a 2D complex array, and the rest of the inputs are integers. See the driver program `src/driver2d.c` for an example. This example also shows how to access/print the elements of the array using some pre-defined macros.