User Manual for the paper titled "Algorithm xxx: A Suite of Compact Finite Difference Schemes"

MANI MEHRA and KULDIP SINGH PATEL, Indian Institute of Technology Delhi, India.

Our MATLAB routine have one MATLAB function and three folders.

—`test.m`: This MATLAB function computes the error between analytic differentiation and corresponding compact finite difference approximation for a given function.

—`cfdm_periodic`: This folder has the MATLAB files for computing the compact finite difference approximations for periodic boundary condition in one and two dimensions.

—`cfdm_dirichlet`: This folder has the MATLAB files for computing the compact finite difference approximations for dirichlet boundary condition in one and two dimensions.

—`cfdm_Neumann`: This folder has the MATLAB files for computing the compact finite difference approximations for Neumann boundary in one dimension.

@@

The function `test.m` computes the error between analytic differentiation and corresponding compact finite difference approximation for a given function.

```
>>test
```

@@@

Compact finite difference schemes for periodic boundary in one dimension

In this case, `N` is the number of grid points, `p` is the order of accuracy desired, `x_l` is the left end of the interval and `x_r` is the right end of the interval. Differentiation matrix `D` is order of `N`. `f(x)` is a function defined as $f = (f_1, f_2, ..., f_N)^T$.

1.

`compact_first_periodic.m`

The function `compact_first_periodic.m` computes a differentiation matrix for first derivative approximation. The calling command for this function is

---

Author's address: M. Mehra (mmehra@maths.iitd.ac.in) and K. S. Patel (kuldip@maths.iitd.ac.in), Indian Institute of Technology Delhi, Hauz Khas, New Delhi-110 016

```
>>[D]=compact_first_periodic(N,p,x_l,x_r)
```

Permissible values for `p` are 4, 6, 8 and 10. Then first derivative approximation (f') for any function `f` can be written as

$$f' \approx D * f.$$

2.

`compact_second_periodic.m`
The function `compact_second_periodic.m` computes a differentiation matrix for second derivative approximation. The calling command for this function is

```
>>[D]=compact_second_periodic(N,p,x_l,x_r)
```

Permissible values for p are 4, 6, 8 and 10. Second derivative approximation (f'') for any function `f` can be written as

$$f'' \approx D * f.$$

3.

`compact_third_periodic.m`
The function `compact_third_periodic.m` computes a differentiation matrix for third derivative approximation. The calling command for this function is

```
>>[D]=compact_third_periodic(N,p,x_l,x_r)
```

Permissible values for `p` is 6. Third derivative approximation (f''') for any function `f` can be written as

$$f''' \approx D * f.$$

4.

`compact_fourth_periodic.m`
The function `compact_fourth_periodic.m` computes a differentiation matrix for fourth derivative approximation. The calling command for this function is

```
>>[D]=compact_fourth_periodic(N,p,x_l,x_r)
```

Permissible values for `p` are 4 and 6. Fourth derivative approximation (f'''') for any function `f` can be written as

$$f'''' \approx D * f.$$

@@

Compact finite difference schemes for periodic boundary in two dimensions

In this case, `N` is the number of grid points, `p` is the order of accuracy desired, `x_l` and `x_r` are the left end and the right end of the interval in x direction, `y_l` and `y_r` are the left end and right end of the interval in y direction. Differentiation matrix `D` is order of $\mathtt{N}^2$ and permissible values for `p` are 4, 6, 8 and 10. If $f(x, y)$ is a function defined as $f = (f_{1,1}, f_{2,1}, ..., f_{N,1}, f_{1,2}, f_{2,2}, ..., f_{N,2}, ..., f_{1,N}, f_{2,N}, ..., f_{N,N})^T$.

5.

`compact_first_periodic_2dx.m`
The function `compact_first_periodic_2dx.m` computes a differentiation matrix for first order partial derivative ($\frac{\partial}{\partial x}$) approximation. The calling command for this function is

```
>>[D]=compact_first_periodic_2dx(N,p,x_l,x_r)
```

First order partial derivative approximation $\left(\frac{\partial f}{\partial x}\right)$ for any function `f` can be written as

$$\frac{\partial f}{\partial x} \approx D * f.$$

6.

`compact_first_periodic_2dy.m`
The function `compact_first_periodic_2dy.m` computes a differentiation matrix for first order partial derivative approximation with respect to y variable $\left(\frac{\partial f}{\partial y}\right)$. The calling command for this function is

```
>>[D]=compact_first_periodic_2dy(N,p,y_l,y_r)
```

First order partial derivative approximation $\left(\frac{\partial f}{\partial y}\right)$ for any function `f` can be written as

$$\left(\frac{\partial f}{\partial y}\right) \approx D * f.$$

7.

`compact_second_periodic_2dxx.m`
The function `compact_second_periodic_2dxx.m` computes a differentiation matrix for second order partial derivative approximation. The calling command for this function is

```
>>[D]=compact_second_periodic_2dxx(N,p,x_l,x_r)
```

Second order partial derivative approximation $\left(\frac{\partial^2 f}{\partial x^2}\right)$ for any function `f` can be written as

$$\left(\frac{\partial^2 f}{\partial x^2}\right) \approx D * f.$$

8.

`compact_second_periodic_2dyy.m`
The function `compact_second_periodic_2dyy.m` computes a differentiation matrix for second order partial derivative approximation. The calling command for this function is

```
>>[D]=compact_second_periodic_2dyy(N,p,y_l,y_r)
```

Second order partial derivative approximation $\left(\frac{\partial^2 f}{\partial y^2}\right)$ for any function `f` can be written as

$$\left(\frac{\partial^2 f}{\partial y^2}\right) \approx D * f.$$

9.

`compact_mixed_periodic_2dxy.m`
The function `compact_mixed_periodic_2dxy.m` computes a differentiation matrix for mixed derivative approximation. The calling command for this function is

```
>>[D]=compact_mixed_periodic_2dxy(N,p,x_l,x_r,y_l,y_r)
```

Mixed derivative approximation $\left(\frac{\partial^2 f}{\partial x \partial y}\right)$ for any function `f` can be written as

$$\left(\frac{\partial^2 f}{\partial x \partial y}\right) \approx D * f.$$

@@

Compact finite difference schemes for Dirichlet boundary in one dimension

In this case, `N` is the number of grid points, `p` is the order of accuracy desired, `x_l` is the left end of the interval and `x_r` is the right end of the interval. Differentiation matrix `D` is of order `N` and permissible values for `p` are 4 and 6. $f(x)$ is a function defined as $f = (f_1, f_2, ..., f_N)^T$.

10.

`first_compact_dirichlet.m`
The function `first_compact_dirichlet.m` computes a differentiation matrix for first derivative approximation in case of Dirichlet boundary conditions. The calling command for this function is

```
>>[D]=first_compact_dirichlet(N,p,x_l,x_r)
```

First derivative approximation (f') for any function `f` can be written as

$$f' \approx D * f.$$

11.

`second_compact_dirichlet.m`
The function `second_compact_dirichlet.m` computes a differentiation matrix for second derivative approximation in case of Dirichlet boundary conditions. The calling command for this function is

```
>>[D]=second_compact_dirichlet(N,p,x_l,x_r)
```

Second derivative approximation (f'') for any function `f` can be written as

$$f'' \approx D * f.$$

@@@

Compact finite difference schemes for Dirichlet boundary in two dimensions

In this case, N is the number of grid points, p is the order of accuracy desired, x_l and x_r are the left end and the right end of the interval in x direction, y_l and y_r are the left end and right end of the interval in y direction. In this case, differentiation matrix D is order of N^2, permissible values for p are 4 and 6 and $f(x, y)$ is a function in two variable defined as $f = (f_{1,1}, f_{2,1}, ..., f_{N,1}, f_{1,2}, f_{2,2}, ..., f_{N,2}, ..., f_{1,N}, f_{2,N}, ..., f_{N,N})^T$.

12.

first_compact_dirichlet_2dx.m
The function first_compact_dirichlet_2dx.m computes a differentiation matrix for first order partial derivative $(\frac{\partial}{\partial x})$ approximation. The calling command for this function is

```
>>[D]=first_compact_dirichlet_2dx(N,p,x_l,x_r)
```

First order partial derivative approximation $\left(\frac{\partial f}{\partial x}\right)$ for any function f can be written as

$$\frac{\partial f}{\partial x} \approx D * f.$$

13.

first_compact_dirichlet_2dy.m
The function first_compact_dirichlet_2dy.m computes a differentiation matrix for first order partial derivative approximation with respect to y variable $\left(\frac{\partial f}{\partial y}\right)$. The calling command for this function is

```
>>[D]=first_compact_dirichlet_2dy(N,p,y_l,y_r)
```

First order partial derivative approximation $\left(\frac{\partial f}{\partial y}\right)$ for any function f can be written as

$$\left(\frac{\partial f}{\partial y}\right) \approx D * f.$$

14.

second_compact_dirichlet_2dxx.m
The function second_compact_dirichlet_2dxx.m computes a differentiation matrix for second order partial derivative approximation. The calling command for this function is

```
>>[D]=second_compact_dirichlet_2dxx(N,p,x_l,x_r)
```

Second order partial derivative approximation $\left(\frac{\partial^2 f}{\partial x^2}\right)$ for any function f can be written as

$$\left(\frac{\partial^2 f}{\partial x^2}\right) \approx D * f.$$

15.

second_compact_dirichlet_2dyy.m
The function compact_second_periodic_2dyy.m computes a differentiation matrix for second order partial derivative approximation. The calling command for this function is

```
>>[D]=second_compact_dirichlet_2dyy(N,p,y_l,y_r)
```

Second order partial derivative approximation $\left(\frac{\partial^2 f}{\partial y^2}\right)$ for any function f can be written as

$$\left(\frac{\partial^2 f}{\partial y^2}\right) \approx D * f.$$

16.

mixed_compact_dirichlet_2dxy.m
The function mixed_compact_dirichlet_2dxy.m computes a differentiation matrix for mixed derivative approximation in case of Dirichlet boundary conditions. The calling command for this function is

```
>>[D]=mixed_compact_dirichlet_2dxy(N,p,x_l,x_r,y_l,y_r)
```

Mixed derivative approximation for any function f can be written as

$$\frac{\partial^2 f}{\partial x \partial y} \approx D * f.$$

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

Compact finite difference schemes for Neumann boundary in one dimension

17.

compact_second_neumann.m
The function compact_second_neumann.m computes a differentiation matrix for second derivative approximation in case of Neumann boundary conditions. The calling command for this function is

```
>>[D,K]=compact_second_neumann(N,u_left,u_right,x_l,x_r)
```

where D is of order N, K is a N $\times$ 1 vector, u_left is the left Neumann boundary condition and u_right is the right Neumann boundary condition. Second derivative approximation (f'') for any function f can be written as

$$f'' \approx D * f + K.$$