Manual

Version 0.5 of the manual covers four sections, i**nstallation of the program**, **how to run/ execute the program**, **how to use it**, **possible issues and some extra.** The manual here is the initial release version so it is brief and to the point, if you want to find a pre-documentation of the code, this one can be found in the documentation pdf at the same GitHub.

How to install:

Black tern is a pure C program, due to this can be installed into any machine that supports a gcc compiler. So far the code can be compiled and execute in BSD, OSX, Linux and covers any platform that supports a gcc compiler, so far it uses C99 standard and can be run on an X86-64 platform as any AMD/INTEL powered device or any ARM machine like the Apple-M1/Raspberry-Cortex.

Install instructions:

- Download the BKT0_6.zip file and data.txt file and unzip BKT0_6.zip.
- Move the data.txt file to the same folder created by unzipping BKT0_6.zip, if it tells you that there is other file with the same name it does not matter as maybe I just placed to copies of the same data.txt file.
- Go to the terminal in your system and navigate to the folder BKT0_6, usually you will do this by using the command cd. If lets us say your file is at downloads then you will usually do “cd /username/downloads/BKT0_6”, need to say that username its your computer username and if your system is case sensitive as BSD/Linux then you might need to put “cd /username/Downloads/BKT0_6”.
- Now that it is open, you have two options; you can just compile Blacktern as it is using the command: `gcc Blacktern0_6.c -L. -lprop -ldata -lcomp -lfields -o Blacktern0_6` or you can compile library by library and then blacktern, this is described in the next step. If you have some problems in BSD-Linux jump to the possible issues section.
- To compile library by library you will need to run the command: `gcc -c namethefile.c` for each file in the folder BKT0_6 except the Blacktern0_6.c, then you will need to link each file to their respective library. When you compiled using: `gcc -c namethefile.c` a set of .o files are created, each file goes inside a library so search for the .h files and take note of each file name inside then run: `ar -rc libraryname.a name ofthefile.o` and at the end just run: `gcc Blacktern0_6.c -L. -lprop -ldata -lcomp -lfields -o Blacktern0_6` I would never recommend this for someone that has short experience in .nix like systems.
- Run the command: `ls`
- After running the command you should see a new executable file named Blacktern0_6.
- It is done!, easy isn’t???!!!

How to run:

In order to run blacktern you should go to your terminal again and navigate to the folder where blacktern is located, this as below:

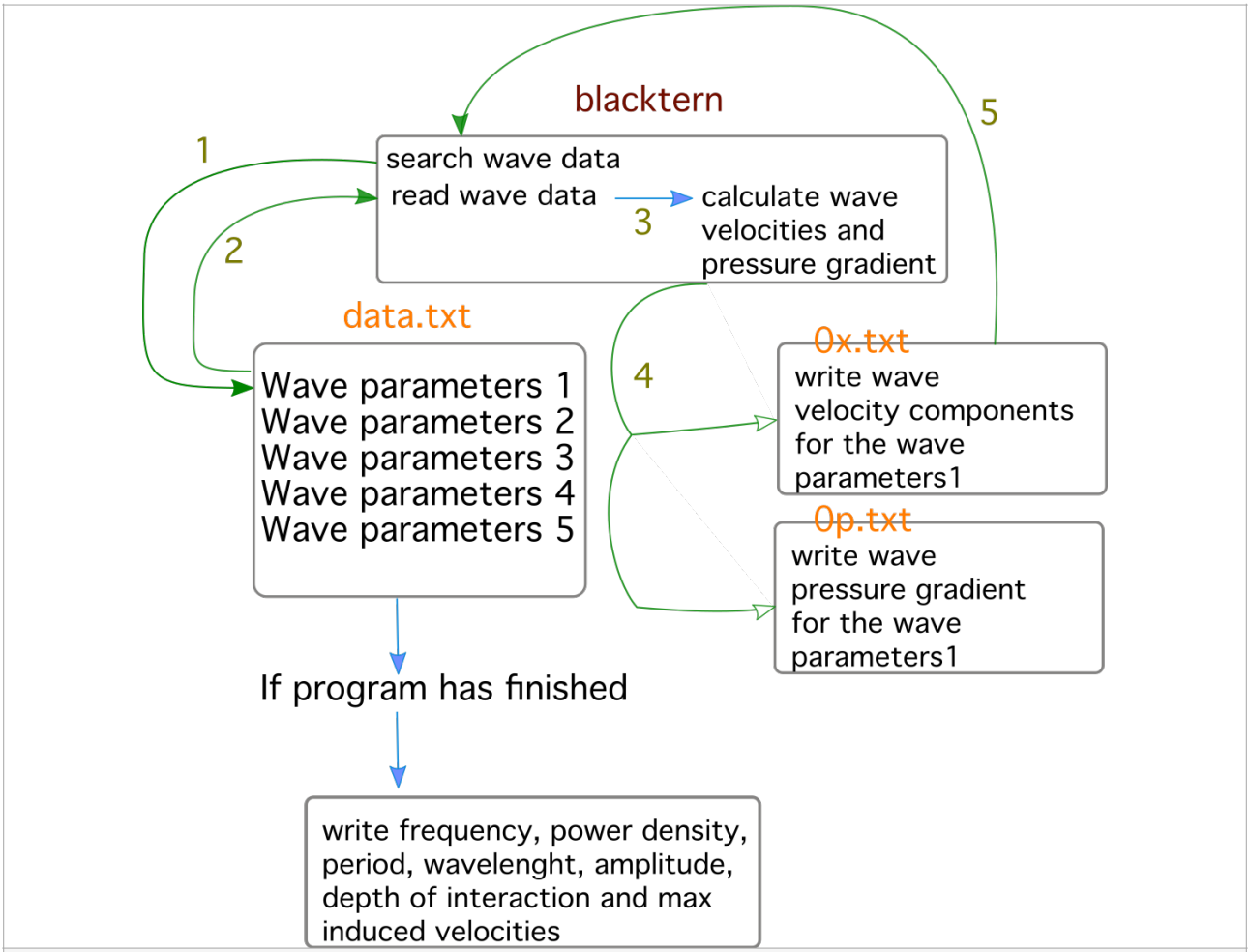
- Open the terminal and type: `cd </path to folder that contains the file/>`

Lets us say that your file is at your desktop in OSX, Linux or BSD, as when you open your terminal in any of those three you start from your user folder, then you will only need to type: `cd /Desktop`

Now that you are in the folder that contains the executable, you just need to type the name of the problem followed by a space and the path to the data file that contains the wave data: `./blacktern <name of file>.txt`

It is advised that your file must be in the same folder as blacktern, to simplify the process. If your file name is `data.txt`, then you will type: `./blacktern data.txt`

Blacktern will produce series of text files as an output, each output file contains the (x,y velocity components). The files will be named using numbers, if you have a file that contains the data for 5 wave types, then blacktern will produce 10 files. The structure of the main work behind can be seen below:



For every wave parameter 1 to 5, one different output file will be generated. Here in this case the process will repeat 1 to 10 during five times to create files 0x.txt, 0p.txt, 1x.txt, 1p.txt, 2x.txt, 2p.txt, 3x.txt, 3p.txt, 4x.txt, 4p.txt.

The output files will be named after its wave regime, for example if the conditions set for the wave parameters 1 are for a linear swell in deep waters and the conditions for wave parameters 2 have a swell moving in transitional water, then the names of the files will be:

`0x1.txt`

`0p1.txt`

`1xt1.txt`

`1pt1.txt`

The first number is the wave registry as wave period number 0, wave period number 1 and so on as blacktern calculates them; the x is just used to distinguish between the files containing the velocities (x) and the file containing the pressure gradient (p), as extra t for the second output will denote the regime this wave is moving, **no letter** will tell us that the wave moves in deep waters, **t** will tell us that the wave is on transitional waters. If the name structure of the files contains a 2 as:

`0x2.txt`

`0p2.txt`

`1xt2.txt`

`1pt2.txt`

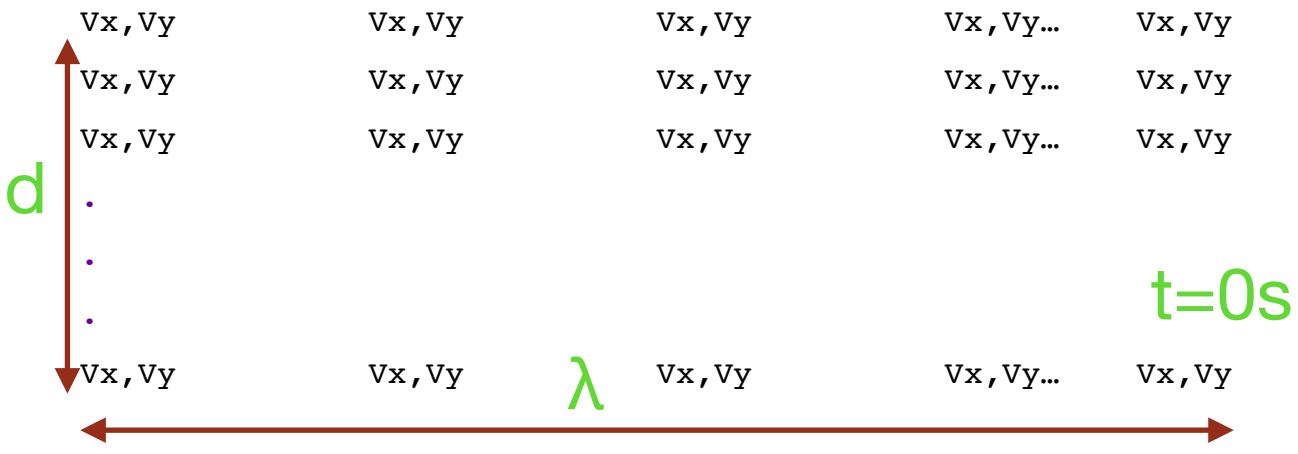
Then waves will be a second order wave, we can summarise this rules of naming files as:

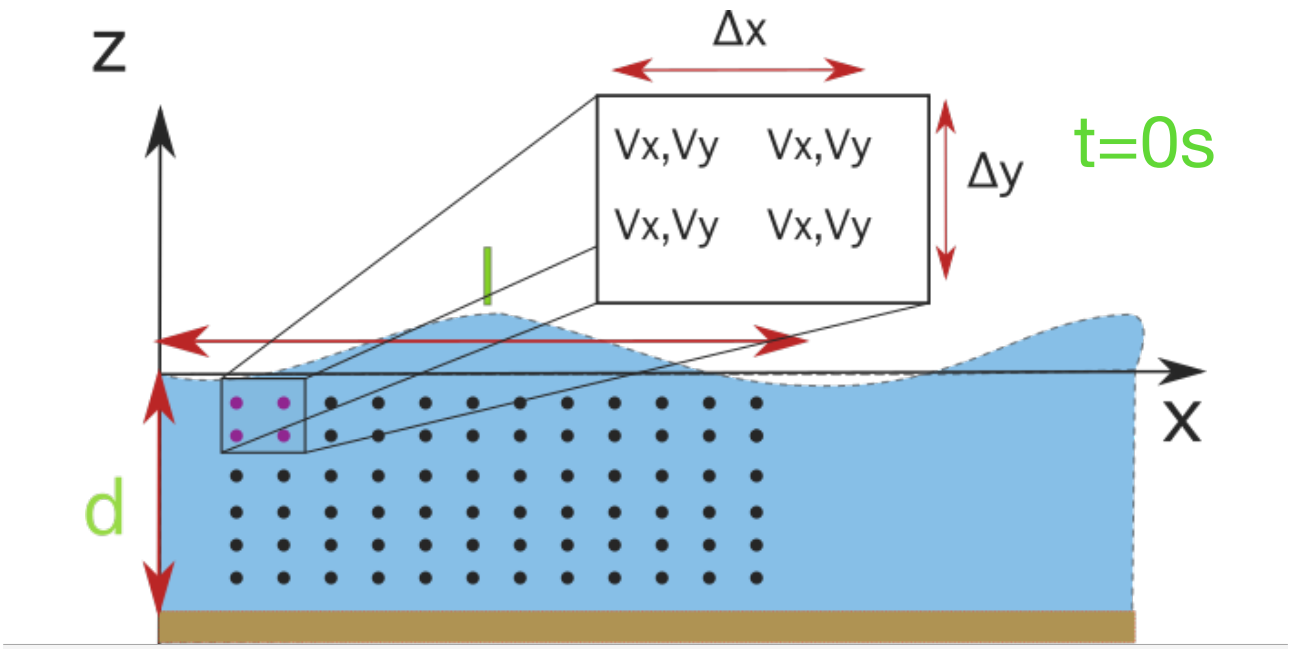
[wave record][x or p][deep regime][wave order].

`3pt2.txt`=4th wave registry, pressure gradient, transitional waters, second order wave.

`7x1.txt`=7th wave registry, velocity gradient, deep waters, first order wave.

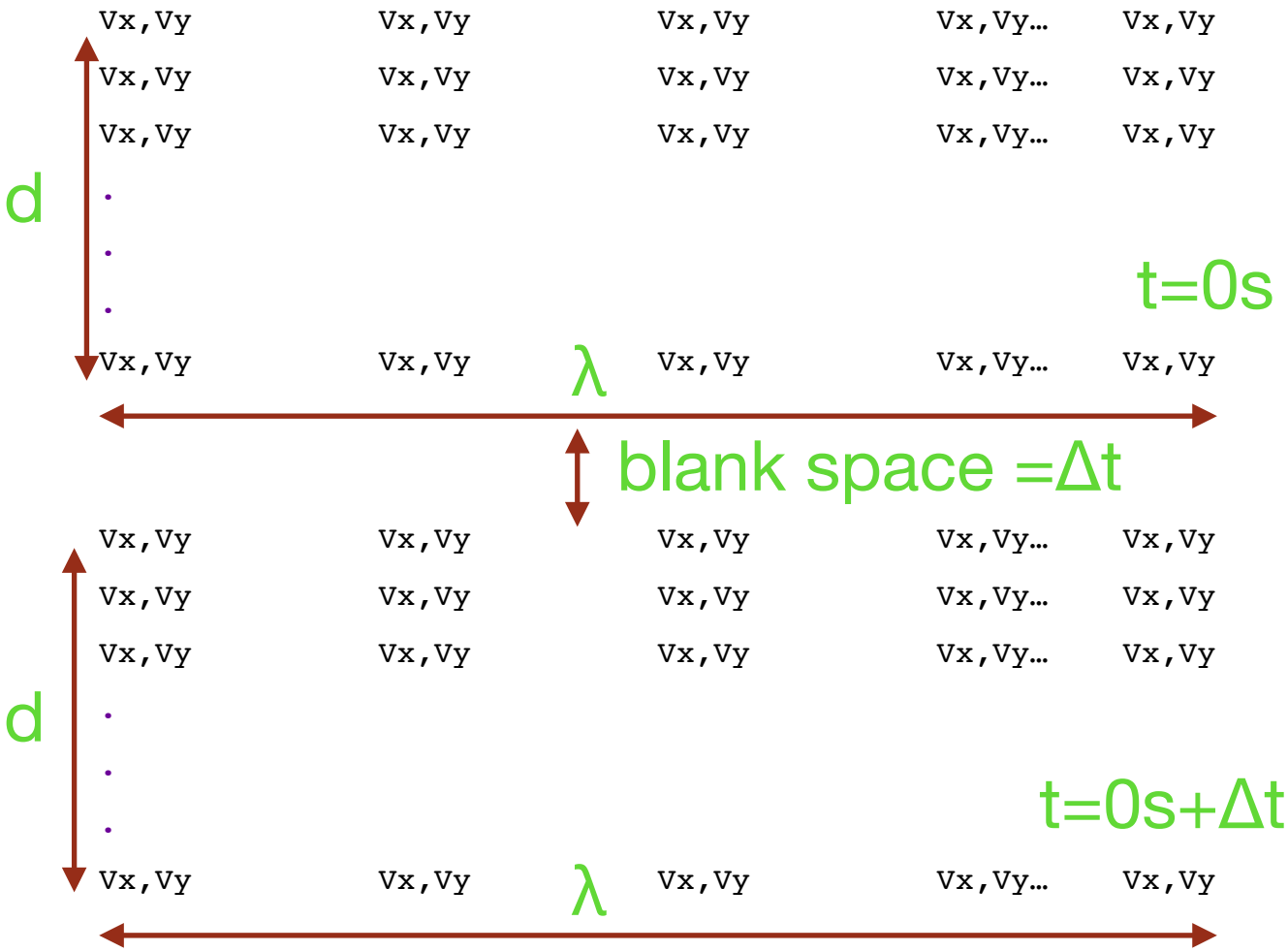
The output files will have and structure of columns and rows, where the pairs of data will represent the x,y velocity components. The components will be calculated in a length that goes from 0 to the maximum wavelength of the wave and from the mean water surface z=0 to the death where the wave is moving. The structure of the output and how it relates to the physical model can be seen in the figure below.

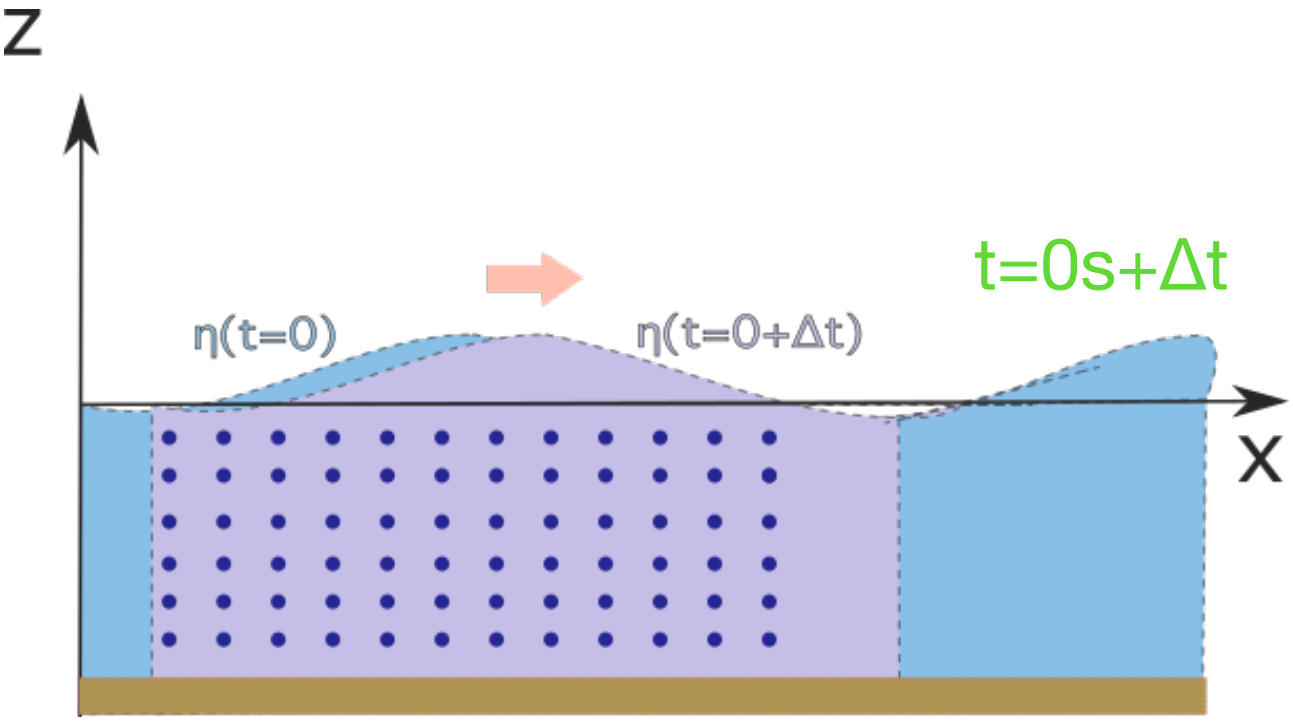




Δx and Δy are given by the user in the wave data file, this will be mentioned in the next section

As the velocities are calculated from 0 to l and 0 to d, the variation on the velocities on time will be calculated next from 0 to the maximum period of the wave. The output file will have a jump with a blank space after calculating all instant velocities at t=0s, then will jump to t=0s+Δt. This will be repeated until a full wave period is completed, thus meaning that a full wave cycle passed over the area were we are calculating the instant velocities.





The new values for each point in the grid will correspond to the new flow velocity values as the waves moves over the defined area.

The same process will be repeated for the pressure gradient at the wave files containing a p instead of an x in its name. In some of these files pressure gradient could be negative below the surface, this is indicative of a trough.

In the final step, blacktern will read the wave parameters as amplitude and period to calculate some spectral wave characteristics as:

- Wavelength of the given period at that depth.
- Frequency of the wave.
- Power density of the wave component at that wave height.
- Wave depth of interaction for deep water waves, thus telling us how far the Wakefield will reach in the water column.
- Maximum induced velocities by the waves in the surface, this as the maximum vectors on the z and z direction.

How to use it:

Step A

I order to use blacktern you will need wave data already processed, another tools will be added later. Blacktern uses two text files with extension .txt, the file must be composed solely by 5 rows that are separated by a space or tab space. The composition this file must have is:

- Wave period
- Wave height
- Δx
- Δy
- Δt

The structure will be:

```
T1    a1  Δx1  Δy1  Δt1
T2    a2  Δx2  Δy2  Δt2
T3    a3  Δx3  Δy3  Δt3
.
.
.
Tn    an  Δxn  Δyn  Δtn
```

Every Δ is given by the user, lets say that we want to calculate the velocity field meter by meter, each second of a swell of 1m amplitude and period of T=10.5s. Then our file will be composed as:

```
10.5 1    1    1    1
```

Now if we want to calculate three wave swells with periods 10.5, 12.3, 14.5 and amplitudes of 1.2m, 1.5m, 0.8m then our text file will be composed as:

```
10.5 1    1    1    1
12.5 1    1    1    1
14.5 1    1    1    1
```

Of course, each Δ can be different, but if all wave components belong to the same analysis it will be advised to keep them at same Δ.

Step B:

The second file to be used needs to define the sea conditions on temperature, salinity and atmospheric pressure. This file has to be stored at the same path as Blacktern, this file need to be called pdata and have a .txt extension. The structure of the file is the same for the wave components, this structure can bee seen below:

```
Temp Sal  Pressure
```

Temperature need to be given in celsius, Salinity un PSU and Pressure in atmospheres, the calculations use the method given by the UNESCO for water density. An example of this could be the one detailed below:

```
15   35   0.98
```

The density was included as the gradient pressure can be sensitive at the temperature/ salinity of the place. The overall variation of the pressure gradient is small in percentage; however, its value is still significant. As the code is made in C but not in python the solutions that uses a power series of coefficients do not incurs in a large computational time larger than .01s.

After the file has been defined, now we only need to run blacktern and specify the file address or its name if the file is stored in same folder. Lets say our file is again named data.txt and blakctern is stored at same place as the data, then we will write:

```
./blacktern data.txt
```

The output will be stored in the same folder where blacktern us being used.

Example:

- We provide a file called data.txt with very simple wave parameters, this file has some long linear waves in deep waters and some long linear waves in transitional waters. Download blacktern.c and the file at the same folder in your machine.
- Now that you already downloaded the code and the file, just open your terminal at the place and type: `gcc blacktern.c -o blacktern`
- There must be now an executable named blacktern in your folder too, now just check again (just to be sure), that the data file data.txt is in same folder than blacktern.
- Now type the name of the program followed by a space and the name of the data file as: `./blacktern data.txt`
- Now the program will ask you two things first the depth where the waves are moving, just enter a value and click enter.
- The program will ask you now the latitude of the buoy systems where you got this data, enter the values un decimal system as 35.456 on the example. Now click enter.

- If everything went well, now yo have several new text files from `0x1.txt`, `0p1.txt` `1x1.txt`, `1p1.txt` and one extra file called `spectra.txt` that contains certain spectral information on the wave parameters calculated using data.txt.

Possible issues:

1) Blacktern can be compiled in BSD, OS-X and Linux or any system supporting gcc and pure C, however; some compilation instructions may change. It is known that for some linux-unix systems compile any C program using the library math.h, will need a flag. To do this then just compile using the next instruction:

- `gcc Blacktern0_6.c -L. -lprop -ldata -lcomp -lfields -o Blacktern0_6 -lm`

2) Black tern needs the name of file if this one is stored at the folder than the executable, if not we will need to path to the data file. AS an example lets us say that blacktern is at Desktop and the `data.txt` is at downloads, in OS-X or Linux we will use the next:

- Open your terminal at desktop.
- Type: `./BKT0_6 /Downloads/data.txt`

Remember that blacktern will store the files at the place where blacktern is located, so the files won't be located where the `data.txt` file is.

3) Can't find the output files?, as mentioned in the last paragraph the output files will be stored at the same location as the executable.

4) Compilation gives an error?, usually the program even if small is tested to run each time so could be two things: I made a mistake and did not notice it or must be your machine or gcc installation. If any error arises please sent a message to try to reproduce it, this with the data and way that this occurred.

5) Values are wrong compared to other tools!, usually the values are checked against other tools using the analytical expressions (Mathematica/MATLAB); however, could be that something is wrong. Values were accurate compared to these two software tools using the same formulations, the values sometimes differ due to the decimals being cut. It could happen also that the method that calculates the wavelength differs from the ones used in other tools, another suggestion is that sometimes the user confuses the amplitude with the wave height when writing the data.txt file and then values are incorrectly calculated.

Extra:

Why blacktern asks for the buoy latitude and the depth of propagation?.

Well blacktern will possibly be a standalone code, however it is not meant to be one as this is just a piece of a larger program to analyse data. Black tern aims to have something like:

Data stream -> code path ways->code[calculation]->code pathways->Data produced.

The code pathways is a very rough manner to talk about a non-defined source of data, as this will be modified meanwhile the code grows. The aim is that the program will just plug into several types of data stream, this to maintain flexibility; by now the data streams that are implemented are just temporary but a good way to start!.

The data that is handed uses a double or float point value; however, it only prints X.XXX length?.

The calculus in fact in black tern use double so the numbers can be quite long to keep all the possible decimals, however when we save data we only keep the dimensions till what would be 1cm. Cause you know its a bit too much!.

Blacktern uses linear and weakly linear theory, is that still a thing?.

Well blacktern aims to do fast calculations and a good agreement with the values. It is indeed true that even a weakly linear theory as Stokes dealing with non-linear functions in a Fourier expansion way might be “debatable for some mathematicians/physicists” and definitely for most of the engineers/oceanographers out there. But meanwhile a solution is found to do it as fast as possible with strong non-linearities (there is one, it exists a rough approach) this will be postponed for later (Poncaire might forgive us for this!).

Blacktern is accurate?.

Well first, the license clearly says that there no responsibility over the results, however in the theories that blacktern uses it is quite a good approach. The results are from time to time, compared to the solutions produced with tools as Mathematica with the exact analytical expressions.

And so far, so good!. But if there is any mistake please sent a message!.

Please:

As a request, if you use it at least tell about how it went and how bad or good this is done!.