



**Datawave Marine Science**

**OFreq**

## **Output Writing Classes**

**Nicholas Barczak**

**May 19, 2013**

**Rev: 0.10**

**DMS1303-001-220-02**

## Revision History

Revision	Date	Changes	Approval
1.00	Dec 30, 1999	Initial Issue	

## Prepared By

Datawave Marine Science  
3500 27<sup>th</sup> PI West  
Apt 423  
Seattle, WA 98199

[www.dmsonline.us](http://www.dmsonline.us)

Confidential and proprietary property of Datawave Marine Science (DMS).  
May not be used for construction or provided to any third parties without prior written consent.  
Copyright 2012, Datawave Marine Science.

## Table Of Contents

1 Introduction.....	5
2 Class: fileList.....	5
2.1 Attribute: filePath.....	5
2.2 Attribute: outputFiles.....	5
2.3 Attribute: directions.....	5
2.4 Attribute: frequencies.....	5
2.5 Attribute: directionFile.....	5
2.6 Attribute: frequencyFile.....	5
2.7 Operation: writeFiles().....	5
3 Class: fileWriter.....	6
3.1 Attribute: fileName.....	6
3.2 Attribute: className.....	6
3.3 Attribute: version.....	6
3.4 Attribute: format.....	6
3.5 Attribute: header.....	6
3.6 Attribute: filePath.....	6
3.7 Attribute: outputsList.....	6
3.8 Operation: Constructor().....	7
3.9 Operation: Destructor().....	7
3.10 Virtual Operation: WriteFile().....	7
4 Class: fileWaveFrequencies.....	7
4.1 Attribute: frequencies.....	7
4.2 Operation: Constructor().....	7
5 Class: fileWaveDirections.....	8
5.1 Attribute: Directions.....	8
5.2 Operation: Constructor().....	8
6 Class: fileGlobalAcceleration.....	9
6.1 Attribute: dataList.....	9
6.2 Inherited Operation: Write().....	9
6.3 Operation: Constructor().....	9
7 Class: fileGlobalVelocity.....	10
7.1 Attribute: dataList.....	10
7.2 Inherited Operation: Write().....	10
7.3 Operation: Constructor().....	11
8 Class: fileGlobalPosition.....	11
8.1 Attribute: dataList.....	12
8.2 Inherited Operation: Write().....	12

8.3 Operation: Constructor()	12
9 Class: dataVector	13
9.1 Attribute: Frequency	13
9.2 Attribute: Value	13
10 Conclusion	13
11 References	14
Appendix A: Example of Output File	15

## 1 Introduction

---

This file covers the class definitions necessary for writing the output files from derived outputs.

## 2 Class: FileList

---

The fileList class contains and controls all the output files to write. This class gets used for each wave direction. It also specifies the folder path for each output file.

### 2.1 Attribute: FilePath

The filePath stores the path of the location to write files to. The fileList object updates with each iteration of wave directions. So the filePath object specifies the location of the files for each wave direction.

### 2.2 Attribute: OutputFiles

This is the major attribute for the object. It contains the vector of objects used to write output files.

### 2.3 Attribute: Directions

This is a vector that contains the list of wave directions. This may not be necessary since the system object contains the same information.

### 2.4 Attribute: Frequencies

This is a vector that contains the list of wave frequencies. This may not be necessary since the system object contains the same information.

### 2.5 Attribute: DirectionFile

This is an object for writing out the file of wave directions.

### 2.6 Attribute: FrequencyFile

This is an object for writing out the file of wave frequencies.

### 2.7 Operation: WriteFiles()

This is the operation to iterate through each object in the vector OutputFiles and trigger each object to write its output file. Before iterating through the files, the operation must also check that the folder path exists, or create the necessary folders.

## 3 Class: FileWriter

This is the main class used to define the file writing objects. Each fileWriter object is then customized for the specific type of output file to write.

### 3.1 Attribute: FileName

The filename to write. This is normally fixed. But don't create it as a constant. Future versions of the program may have some need to change the filename at runtime. The constructor will set a default filename for each object, based on its class.

### 3.2 Attribute: ClassName

This is the string that gets written in the output file. It gets manually set, but should match the name of the derivedOutput class that the file pulls from. It should not automatically use the class name of the derivedOutput class because some later implementations may involve more than one derivedOutput class.

### 3.3 Attribute: Version

Specifies the version of the program used, to allow the program awareness for backwards compatibility. Current version is 1.0

### 3.4 Attribute: Format

Specifies the format of the output data. Use keyword "ASCII" to represent text files.

### 3.5 Attribute: Header

The string of the nice header to print out at the top of every output file. Use the following text:

```
/*-----*- C++ -*-----*\n| O pen          | OpenSea:  The Open Source Seakeeping Suite\n| S eakeeping    | Version:   1.0\n| E valuation    | Web:       www.dmsonline.us/opensea\n| A nalysis      |\n/*-----*-*/
```

### 3.6 Attribute: FilePath

Acquired from the fileList object that called it. The filePath specifies the folder path that the output file needs to write into.

### 3.7 Attribute: OutputsList

A pointer to the derived outputsList object that generated all the derived outputs. This

needs to reference the entire list so that later implementations can access multiple derived outputs for file writing.

### **3.8 Operation: Constructor()**

Creates the object and sets key default attributes. Customizations are shown for each child class.

### **3.9 Operation: Destructor()**

Deletes the object.

### **3.10 Virtual Operation: WriteFile()**

Writes the output file. There is an example in the testing directory that shows how the output file should look. I have also included it in Appendix A at the end of this report. The general sequence follows this:

1. Header Attribute
2. Version Attribute
3. Format Attribute
4. Object Attribute
5. Some separator
6. Beginning of output data, organized by body.

## **4 Class: FileWaveFrequencies**

---

This object collects and writes out the list of wave frequencies.

### **4.1 Attribute: Frequencies**

The list of wave frequencies to write.

### **4.2 Operation: Constructor()**

The following items are used during the constructor to customize this class.

#### **4.2.1 Header Attribute**

Header =

```
/*-----*- C++ -*-----*\n| O pen          | OpenSea:  The Open Source Seakeeping Suite\n| S eakeeping    | Version:  1.0\n| E valuation    | Web:      www.dmsonline.us/opensea\n| A nalysis      |\n/*-----*- C++ -*-----*\\
```

This can really be coded as global constant variable. It won't change unless a new version of OpenSEA gets released.

#### 4.2.2 ClassName

className = "frequency"

#### 4.2.3 Version

version = 1.0

#### 4.2.4 Format

format = "ASCII"

#### 4.2.5 FileName

fileName = "frequencies.out"

## 5 Class: FileWaveDirections

---

This object writes out the wave directions output file.

### 5.1 Attribute: Directions

The list of wave directions to write.

### 5.2 Operation: Constructor()

The following items are used during the constructor to customize this class.

#### 5.2.1 Header Attribute

Header =

```
/*-----*- C++ -*-----*\n| O pen          | OpenSea:  The Open Source Seakeeping Suite\n| S eakeeping    | Version:  1.0\n| E valuation    | Web:      www.dmsonline.us/opensea\n| A nalysis      |\n/*-----*- C++ -*-----*\\
```



\\*-----\*/

This can really be coded as global constant variable. It won't change unless a new version of OpenSEA gets released.

### 5.2.2 ClassName

className = "direction"

### 5.2.3 Version

version = 1.0

### 5.2.4 Format

format = "ASCII"

### 5.2.5 FileName

fileName = "directions.out"

## 6 Class: FileGlobalAcceleration

---

This object writes out the global accelerations that were calculated by the globalAcceleration class in the OutputsList object.

### 6.1 Attribute: DataList

This is a vector of objects of the class dataVector<complex double>. Each object in the array contains results for one wave frequencies. Each objects is a vector itself meant to hold the data for a single wave frequency.

### 6.2 Inherited Operation: Write()

The write Operation gets the results from the derivedOutput object of class globalAcceleration. It stores those results in the object dataVector. Once it collected all the information for each frequency, it then writes that information out to the correct output file, formatted as shown in the test directory.

```
../500 Testing/Template Project - With Outputs/ofreq1/
```

It then repeats this process for each OutputsBody defined in the OutputsList object.

### 6.3 Operation: Constructor()

The following items are used during the constructor to customize this class.

### 6.3.1 Header Attribute

Header =

```
/*-----*- C++ -*-----*\n| O pen          | OpenSea:  The Open Source Seakeeping Suite\n| S eakeeping    | Version:   1.0\n| E valuation    | Web:      www.dmsonline.us/opensea\n| A nalysis      |\n/*-----*-*/
```

This can really be coded as global constant variable. It won't change unless a new version of OpenSEA gets released.

### 6.3.2 ClassName

className = "globalAcceleration"

### 6.3.3 Version

version = 1.0

### 6.3.4 Format

format = "ASCII"

### 6.3.5 FileName

fileName = "globalAcceleration.out"

## 7 Class: FileGlobalVelocity

---

This object writes out the global velocities that were calculated by the globalVelocity class in the OutputsList object.

### 7.1 Attribute: DataList

This is a vector of objects of the class `dataVector<complex double>`. Each object in the array contains results for one wave frequencies. Each objects is a vector itself meant to hold the data for a single wave frequency.

### 7.2 Inherited Operation: Write()

The write Operation gets the results from the derivedOutput object of class globalVelocity. It stores those results in the object dataVector. Once it collected all the information for each frequency, it then writes that information out to the

correct output file, formatted as shown in the test directory.

../500 Testing/Template Project - With Outputs/ofreq1/

It then repeats this process for each OutputsBody defined in the OutputsList object.

## 7.3 Operation: Constructor()

The following items are used during the constructor to customize this class.

### 7.3.1 Header Attribute

Header =

```
/*-----*- C++ -*-----*\n| O pen          | OpenSea:  The Open Source Seakeeping Suite | \n| S eakeeping    | Version:  1.0 | \n| E valuation    | Web:      www.dmsonline.us/opensea | \n| A nalysis      | | \n/*-----*- C++ -*-----*/
```

This can really be coded as global constant variable. It won't change unless a new version of OpenSEA gets released.

### 7.3.2 ClassName

className = "globalVelocity"

### 7.3.3 Version

version = 1.0

### 7.3.4 Format

format = "ASCII"

### 7.3.5 FileName

fileName = "globalVelocity.out"

## 8 Class: FileGlobalPosition

This object writes out the global positions that were calculated by the globalPosition class in the OutputsList object.

## 8.1 Attribute: DataList

This is a vector of objects of the class `dataVector<complex double>`. Each object in the array contains results for one wave frequencies. Each objects is a vector itself meant to hold the data for a single wave frequency.

## 8.2 Inherited Operation: Write()

The write operation gets the results from the `derivedOutput` object of class `globalAcceleration`. It stores those results in the object `dataVector`. Once it collected all the information for each frequency, it then writes that information out to the correct output file, formatted as shown in the test directory.

../500 Testing/Template Project - With Outputs/ofreq1/

It then repeats this process for each `OutputsBody` defined in the `OutputsList` object.

## 8.3 Operation: Constructor()

The following items are used during the constructor to customize this class.

### 8.3.1 Header Attribute

Header =

```
/*-----*- C++ -*-----*\n| O pen          | OpenSea:  The Open Source Seakeeping Suite\n| S eakeeping    | Version:  1.0\n| E valuation    | Web:      www.dmsonline.us/opensea\n| A nalysis      |\n/*-----*-*/
```

This can really be coded as global constant variable. It won't change unless a new version of OpenSEA gets released.

### 8.3.2 ClassName

className = "globalPosition"

### 8.3.3 Version

version = 1.0

### 8.3.4 Format

format = "ASCII"

### 8.3.5 FileName

fileName = "globalPosition.out"

## 9 Class: DataVector

---

The dataVector is a template class that can accommodate different variables. It is meant to represent as single data entry associated with the index of a single wave frequencies. The values for the data are actually a vector of template data type. The vector entries are the solutions associated with each motion.

### 9.1 Attribute: Frequency

An integer that represents the index of the wave frequency associated with this data object.

### 9.2 Attribute: Value

A vector containing the actual values associated with the data object. The size of the vector is not known before run time. It varies with the motion model associated with it.

## 10 Conclusion

---

This completes the definition of the outputs file writing objects.

## **11 References**

---

## Appendix A: Example of Output File

```

/*-----*- C++ -*-----*\
| O pen          | OpenSea:  The Open Source Seakeeping Suite
| S eakeeping    | Version:   1.0
| E valuation    | Web:      www.dmsonline.us/opensea
| A nalysis     |
\*-----*-*\
seafire
{
    version      1.0;
    format       ascii;
    object       accglobal;
}
// * * * * * //
body {
    name "body1";
    data {
        frequency 1
        value (
            3.49034295746184+2.95803989154981i
            12.4285966635775+2.73495886623547i
            25.7903399171931+2.59807621135332i
            70.1054123466879+2.39791576165636i
            347.234380478734+2.03715487874634i
            512.858510942829+1.93907194296653i
        );
    }
    data {
        frequency 2
        value (
            3.49034295746184+2.95803989154981i
            12.4285966635775+2.73495886623547i
            25.7903399171931+2.59807621135332i
            70.1054123466879+2.39791576165636i
            347.234380478734+2.03715487874634i
            512.858510942829+1.93907194296653i
        );
    }
    data {
        frequency 3
        value (
            3.49034295746184+2.95803989154981i
            12.4285966635775+2.73495886623547i
            25.7903399171931+2.59807621135332i
            70.1054123466879+2.39791576165636i
            347.234380478734+2.03715487874634i
            512.858510942829+1.93907194296653i
        );
    }
    data {
        frequency 4

```

```
value (  
  3.49034295746184+2.95803989154981i  
  12.4285966635775+2.73495886623547i  
  25.7903399171931+2.59807621135332i  
  70.1054123466879+2.39791576165636i  
  347.234380478734+2.03715487874634i  
  512.858510942829+1.93907194296653i  
);  
}  
data {  
  frequency 5  
  value (  
    3.49034295746184+2.95803989154981i  
    12.4285966635775+2.73495886623547i  
    25.7903399171931+2.59807621135332i  
    70.1054123466879+2.39791576165636i  
    347.234380478734+2.03715487874634i  
    512.858510942829+1.93907194296653i  
  );  
}  
data {  
  frequency 6  
  value (  
    3.49034295746184+2.95803989154981i  
    12.4285966635775+2.73495886623547i  
    25.7903399171931+2.59807621135332i  
    70.1054123466879+2.39791576165636i  
    347.234380478734+2.03715487874634i  
    512.858510942829+1.93907194296653i  
  );  
}  
data {  
  frequency 7  
  value (  
    3.49034295746184+2.95803989154981i  
    12.4285966635775+2.73495886623547i  
    25.7903399171931+2.59807621135332i  
    70.1054123466879+2.39791576165636i  
    347.234380478734+2.03715487874634i  
    512.858510942829+1.93907194296653i  
  );  
}  
data {  
  frequency 8  
  value (  
    3.49034295746184+2.95803989154981i  
    12.4285966635775+2.73495886623547i  
    25.7903399171931+2.59807621135332i  
    70.1054123466879+2.39791576165636i  
    347.234380478734+2.03715487874634i  
    512.858510942829+1.93907194296653i  
  );  
}  
data {  
  frequency 9
```



```
value (  
  3.49034295746184+2.95803989154981i  
  12.4285966635775+2.73495886623547i  
  25.7903399171931+2.59807621135332i  
  70.1054123466879+2.39791576165636i  
  347.234380478734+2.03715487874634i  
  512.858510942829+1.93907194296653i  
);  
}  
data {  
  frequency 10  
  value (  
    3.49034295746184+2.95803989154981i  
    12.4285966635775+2.73495886623547i  
    25.7903399171931+2.59807621135332i  
    70.1054123466879+2.39791576165636i  
    347.234380478734+2.03715487874634i  
    512.858510942829+1.93907194296653i  
  );  
}  
data {  
  frequency 11  
  value (  
    3.49034295746184+2.95803989154981i  
    12.4285966635775+2.73495886623547i  
    25.7903399171931+2.59807621135332i  
    70.1054123466879+2.39791576165636i  
    347.234380478734+2.03715487874634i  
    512.858510942829+1.93907194296653i  
  );  
}  
data {  
  frequency 12  
  value (  
    3.49034295746184+2.95803989154981i  
    12.4285966635775+2.73495886623547i  
    25.7903399171931+2.59807621135332i  
    70.1054123466879+2.39791576165636i  
    347.234380478734+2.03715487874634i  
    512.858510942829+1.93907194296653i  
  );  
}  
data {  
  frequency 13  
  value (  
    3.49034295746184+2.95803989154981i  
    12.4285966635775+2.73495886623547i  
    25.7903399171931+2.59807621135332i  
    70.1054123466879+2.39791576165636i  
    347.234380478734+2.03715487874634i  
    512.858510942829+1.93907194296653i  
  );  
}  
data {  
  frequency 14
```

```
value (  
3.49034295746184+2.95803989154981i  
12.4285966635775+2.73495886623547i  
25.7903399171931+2.59807621135332i  
70.1054123466879+2.39791576165636i  
347.234380478734+2.03715487874634i  
512.858510942829+1.93907194296653i  
);  
}  
data {  
frequency 15  
value (  
3.49034295746184+2.95803989154981i  
12.4285966635775+2.73495886623547i  
25.7903399171931+2.59807621135332i  
70.1054123466879+2.39791576165636i  
347.234380478734+2.03715487874634i  
512.858510942829+1.93907194296653i  
);  
}  
data {  
frequency 16  
value (  
3.49034295746184+2.95803989154981i  
12.4285966635775+2.73495886623547i  
25.7903399171931+2.59807621135332i  
70.1054123466879+2.39791576165636i  
347.234380478734+2.03715487874634i  
512.858510942829+1.93907194296653i  
);  
}  
data {  
frequency 17  
value (  
3.49034295746184+2.95803989154981i  
12.4285966635775+2.73495886623547i  
25.7903399171931+2.59807621135332i  
70.1054123466879+2.39791576165636i  
347.234380478734+2.03715487874634i  
512.858510942829+1.93907194296653i  
);  
}  
data {  
frequency 18  
value (  
3.49034295746184+2.95803989154981i  
12.4285966635775+2.73495886623547i  
25.7903399171931+2.59807621135332i  
70.1054123466879+2.39791576165636i  
347.234380478734+2.03715487874634i  
512.858510942829+1.93907194296653i  
);  
}  
data {  
frequency 19
```

```
value (
  3.49034295746184+2.95803989154981i
  12.4285966635775+2.73495886623547i
  25.7903399171931+2.59807621135332i
  70.1054123466879+2.39791576165636i
  347.234380478734+2.03715487874634i
  512.858510942829+1.93907194296653i
);
}
data {
  frequency 20
  value (
    3.49034295746184+2.95803989154981i
    12.4285966635775+2.73495886623547i
    25.7903399171931+2.59807621135332i
    70.1054123466879+2.39791576165636i
    347.234380478734+2.03715487874634i
    512.858510942829+1.93907194296653i
  );
}
}
body {
  name "body2";
  data {
    frequency 1
    value (
      3.49034295746184+2.95803989154981i
      12.4285966635775+2.73495886623547i
      25.7903399171931+2.59807621135332i
      70.1054123466879+2.39791576165636i
      347.234380478734+2.03715487874634i
      512.858510942829+1.93907194296653i
    );
  }
  data {
    frequency 2
    value (
      3.49034295746184+2.95803989154981i
      12.4285966635775+2.73495886623547i
      25.7903399171931+2.59807621135332i
      70.1054123466879+2.39791576165636i
      347.234380478734+2.03715487874634i
      512.858510942829+1.93907194296653i
    );
  }
  data {
    frequency 3
    value (
      3.49034295746184+2.95803989154981i
      12.4285966635775+2.73495886623547i
      25.7903399171931+2.59807621135332i
      70.1054123466879+2.39791576165636i
      347.234380478734+2.03715487874634i
      512.858510942829+1.93907194296653i
    );
  }
}
```

```
);
}
data {
  frequency 4
  value (
    3.49034295746184+2.95803989154981i
    12.4285966635775+2.73495886623547i
    25.7903399171931+2.59807621135332i
    70.1054123466879+2.39791576165636i
    347.234380478734+2.03715487874634i
    512.858510942829+1.93907194296653i
  );
}
data {
  frequency 5
  value (
    3.49034295746184+2.95803989154981i
    12.4285966635775+2.73495886623547i
    25.7903399171931+2.59807621135332i
    70.1054123466879+2.39791576165636i
    347.234380478734+2.03715487874634i
    512.858510942829+1.93907194296653i
  );
}
data {
  frequency 6
  value (
    3.49034295746184+2.95803989154981i
    12.4285966635775+2.73495886623547i
    25.7903399171931+2.59807621135332i
    70.1054123466879+2.39791576165636i
    347.234380478734+2.03715487874634i
    512.858510942829+1.93907194296653i
  );
}
data {
  frequency 7
  value (
    3.49034295746184+2.95803989154981i
    12.4285966635775+2.73495886623547i
    25.7903399171931+2.59807621135332i
    70.1054123466879+2.39791576165636i
    347.234380478734+2.03715487874634i
    512.858510942829+1.93907194296653i
  );
}
data {
  frequency 8
  value (
    3.49034295746184+2.95803989154981i
    12.4285966635775+2.73495886623547i
    25.7903399171931+2.59807621135332i
    70.1054123466879+2.39791576165636i
    347.234380478734+2.03715487874634i
    512.858510942829+1.93907194296653i
  );
}
```

```
);
}
data {
  frequency 9
  value (
    3.49034295746184+2.95803989154981i
    12.4285966635775+2.73495886623547i
    25.7903399171931+2.59807621135332i
    70.1054123466879+2.39791576165636i
    347.234380478734+2.03715487874634i
    512.858510942829+1.93907194296653i
  );
}
data {
  frequency 10
  value (
    3.49034295746184+2.95803989154981i
    12.4285966635775+2.73495886623547i
    25.7903399171931+2.59807621135332i
    70.1054123466879+2.39791576165636i
    347.234380478734+2.03715487874634i
    512.858510942829+1.93907194296653i
  );
}
data {
  frequency 11
  value (
    3.49034295746184+2.95803989154981i
    12.4285966635775+2.73495886623547i
    25.7903399171931+2.59807621135332i
    70.1054123466879+2.39791576165636i
    347.234380478734+2.03715487874634i
    512.858510942829+1.93907194296653i
  );
}
data {
  frequency 12
  value (
    3.49034295746184+2.95803989154981i
    12.4285966635775+2.73495886623547i
    25.7903399171931+2.59807621135332i
    70.1054123466879+2.39791576165636i
    347.234380478734+2.03715487874634i
    512.858510942829+1.93907194296653i
  );
}
data {
  frequency 13
  value (
    3.49034295746184+2.95803989154981i
    12.4285966635775+2.73495886623547i
    25.7903399171931+2.59807621135332i
    70.1054123466879+2.39791576165636i
    347.234380478734+2.03715487874634i
    512.858510942829+1.93907194296653i
```

```
);
}
data {
  frequency 14
  value (
    3.49034295746184+2.95803989154981i
    12.4285966635775+2.73495886623547i
    25.7903399171931+2.59807621135332i
    70.1054123466879+2.39791576165636i
    347.234380478734+2.03715487874634i
    512.858510942829+1.93907194296653i
  );
}
data {
  frequency 15
  value (
    3.49034295746184+2.95803989154981i
    12.4285966635775+2.73495886623547i
    25.7903399171931+2.59807621135332i
    70.1054123466879+2.39791576165636i
    347.234380478734+2.03715487874634i
    512.858510942829+1.93907194296653i
  );
}
data {
  frequency 16
  value (
    3.49034295746184+2.95803989154981i
    12.4285966635775+2.73495886623547i
    25.7903399171931+2.59807621135332i
    70.1054123466879+2.39791576165636i
    347.234380478734+2.03715487874634i
    512.858510942829+1.93907194296653i
  );
}
data {
  frequency 17
  value (
    3.49034295746184+2.95803989154981i
    12.4285966635775+2.73495886623547i
    25.7903399171931+2.59807621135332i
    70.1054123466879+2.39791576165636i
    347.234380478734+2.03715487874634i
    512.858510942829+1.93907194296653i
  );
}
data {
  frequency 18
  value (
    3.49034295746184+2.95803989154981i
    12.4285966635775+2.73495886623547i
    25.7903399171931+2.59807621135332i
    70.1054123466879+2.39791576165636i
    347.234380478734+2.03715487874634i
    512.858510942829+1.93907194296653i
```

```
    );  
  }  
  data {  
    frequency 19  
    value (  
      3.49034295746184+2.95803989154981i  
      12.4285966635775+2.73495886623547i  
      25.7903399171931+2.59807621135332i  
      70.1054123466879+2.39791576165636i  
      347.234380478734+2.03715487874634i  
      512.858510942829+1.93907194296653i  
    );  
  }  
  data {  
    frequency 20  
    value (  
      3.49034295746184+2.95803989154981i  
      12.4285966635775+2.73495886623547i  
      25.7903399171931+2.59807621135332i  
      70.1054123466879+2.39791576165636i  
      347.234380478734+2.03715487874634i  
      512.858510942829+1.93907194296653i  
    );  
  }  
}  
// ***** //
```