Parallel Approach for the 90%-tile Wind and Wave Hindcast over Irish and Celtic Seas

Sebastian Kärsten
Otto-von-Guericke-Universität, Magdeburg, Germany.

Sabin Tabirca and Nandamudi Vijaykumar
National University of Ireland, Cork, Ireland.

Abstract: Hindcast has become a popular approach in attempting to understand past climate behavior especially in regions where observed data are of poor quality or completely unavailable. A hindcast of wind and wave atlas for Ireland (Irish and Celtic Seas as well as adjacent North Atlantic waters) has been recently developed. This produced a 40-year (1958 to 1997) database of wind fields (hourly) and wave conditions (every three hours). One important measure that is very useful is the percentile. This will allow an idea of the magnitude of the parameters. However, a sequential solution to obtain a 90%-tile value for both these parameters is a very time-consuming task due to the handling of a very large amount of information. Therefore, this paper describes the solution adopted to tackle this issue.

Key-words: Hindcast, Atmospheric Model, Wave Model, Percentiles, 90%-tiles

1. Introduction

The evaluation of climate change and linked environmental processes have drawn a lot of attention in the media since the 1990s, and have been a topic of fundamental research amongst both academic and commercial institutes around the world. Within the scope of coastal science and process work, many studies are being conducted as to how global climate changes will affect coastal regions, in terms of coastal flooding and linked river catchment functioning, erosion and many other coastal management concerns. These evaluations for the future greatly depend upon the use of numerical modelling techniques and upon their accuracy in predicting the operation of environmental systems.

1 The work was done in Boole Centre for Research in Informatics and Costal and Marine Research Centre of National University of Ireland, Cork.
Concern over global climate warming in the 21st Century has tended to focus on coastal and marine environments and outlines the associated risk to human and wildlife populations [Gat+92]. People living in coastal areas are now demanding that local authorities provide detailed information about the potential impacts of climate change. Recent reports for European coastal margins indicate increases in wind speeds, wave heights and heights of storm surges in recent years [Gun+98, Loz+04, WeiGay00]. For the Irish region, storm impact will be extremely significant for coastal erosion and will affect physical processes [Dev94, Loz+04]. In assessing the predicted risks of climate change, it is extremely valuable to be able to identify coastal areas that are at potentially high risk and environmentally sensitive to changes in wave regime.

Awareness of these issues underpins the need for an understanding of past wind and wave climate behavior for European waters. This was recognized at an international level and the research detailed here is part of a large scale European Union 5th Framework funded research project, HIPOCAS (Hindcast of Dynamic Processes of the Ocean and Coastal Areas of Europe) which was set up to generate a climatic hindcast for European waters. This project is linked to the evaluation of coastal processes and to assessing the wider uses of ocean environments (e.g., shipping and port management), particularly through the use of data incorporated into coastal GIS.

The overall project is based on data available from the reanalysis [Kal+96] developed by NCEP (National Center for Environmental Prediction), Washington, D.C. U.S.A. and NCAR (National Center for Atmospheric Research), Boulder, CO, U.S.A. However, this data still has limited use for coastal environments due to the coarse spatial and temporal resolution (200km x 6 hour). Therefore, a downsampling approach was adopted and by applying a regional model REMO (REgional MOdel) [JacPod97] for the North Atlantic to the NCEP data, the spatial resolution was increased from 2.5° to 0.1°. In order to guarantee a more accurate integration of the wave model in Irish waters the resolution was increased further to 0.1°. This downscaling was achieved using a limited area model HIRLAM (HIgh Resolution Limited Area Model) [Sass+00].

Wind fields are a required input to generate the wave conditions for shelf sea areas. The wind fields obtained from HIRLAM were then used by the wave model WAM Cycle 4 to generate the wave conditions such as wave heights, direction, swell, and wind seas. The area for the analysis was established according to scientific interest and covered the coastal areas of Ireland and surrounding seas from c. 48° to 58° N and 15° to 0° W (Figure 1). The output of this project is a data set consisting of wind fields and wave conditions for the area 48° to 58° N and 15° to 0° W. Wind fields were produced on hourly basis while wave conditions are available every three hours. Results were validated by using observation data from ship, buoys and satellite. Some analyses have already been performed for selected coordinates within the above mentioned grid. A detailed analysis over the grid as a whole is under progress. In order to give an idea of the magnitude of winds and
waves of the overall project for the Irish region, determining percentile values is very effective for this purpose. A x%-tile figure calculated for a certain coordinate within the defined grid separates the data in the following manner: the value obtained as the percentile shows that x% of the results are less than this obtained figure while the rest (1-x)% are greater. As an example, assume that it has been decided to determine 50%-tile figure. It means that for a coordinate, the percentile value shown indicates that 50% of the values are less than this obtained. In this case it is a median.

![Map of the study area](image)

Figure 1: Geographical extent of the Hindcast project

The paper describes the experience of determining 90%-tile figures for the hindcast data. The calculation is a very simple one. It is necessary to sort the N values (wind fields and wave conditions) for each coordinate and determine the index that corresponds to 90% of this N. However, it is very time consuming. The reason is the distribution of the data. Data files are organized in time steps, i.e., for every time step YYYYMMDDHH (year, month, day and hour), the parameters are written for each coordinate. This means that in order to determine a 90%-tile value for a given coordinate, it is required to gather the values by reading several files referring to different time steps. This leads to opening and closing files all the time. Once the data is collected, it is sorted so that 90%-tile index and its corresponding value are determined.

Therefore this paper addresses the solution used in order to obtain the results. The solution implemented was to use parallel processing approach on a cluster of 100 nodes. Section 2 describes the structure of the data files, Section 3 discusses two different approaches and Section 4 reports the results. Finally Section 5 concludes the paper.

### 2. Structure of data files
As mentioned above there is a time period of 40 years from 1958 to 1997 consisting of 480 months. Since this is the only common feature of wind and wave data, they will be illustrated separately. For the wind data there is one single binary file per month, each around 60 megabytes. The exact file size depends upon the corresponding number of days for that specific month. Each of these 480 files is divided into its certain number of days; furthermore every day is split up by 24 due to the fact that wind fields were produced on hourly basis. All values are stored one after another starting with the 1st day of a month at 0:00 am to the 31st day at 23:00 pm. Additionally, the grid is regular ranging from c. 48° to 58° N and 15° to 0° W with an increment of 1/8° in both dimensions, hence it leads to a size of 121 longitudinal by 81 latitudinal values. Every wind vector consists of a zonal u and a meridional v components. The calculation of wind strength (meters/second) and wind direction then is as follows:

\[
\text{strength} = \sqrt{u^2 + v^2} \quad (1)
\]

\[
\text{direction} = \arctan\left(\frac{v}{u}\right) \quad (2)
\]

The data for one point in time and the entire grid is organized in three records (Figure 2). The header consists of an 8 byte unsigned integer of the form YYYYMMDDHH00 (year, month, day and hour) that contains all temporal information. These two appending zeros conveying no certain meaning, the format is just expandable to minutes but not in this context. The following second integer has the value 3 in all files, which indicates winds obtained at the height of 10 meters. The second record contains all zonal wind information. All these 121 x 81 = 9,801 values are stored as 4 byte single precision floating points. The third record contains all meridional information respectively. Each record begins and ends with a redundant integer, indicating the size of the record in bytes (white spaces in figure 2). One such triplet of records with the size of 78,444 bytes repeats itself 24 hours x {28, 29, 30, 31} days per file.

<table>
<thead>
<tr>
<th>header</th>
<th>u-record</th>
<th>v-record</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 Byte Unsigned Integer</td>
<td>4 Byte 121 x 81 Floats</td>
<td>121 x 81 Floats</td>
</tr>
</tbody>
</table>

Figure 2: Internal binary file syntax for entire grid and one hour

There are two minor remarks. The first one is that the hindcast actually begins on the 3rd of January 1958. That leads to a total number of (40 years x 365 days + 10 days (due to leap years) – 2 days (1st and 2nd of January 1958)) x 24 values/day = 350,952 values. Secondly every file except the last one (December 1997) ends with the entire first day of the following month. These 24 redundant values will be ignored and skipped in every file during the computation.
In order to compute the 90%-tile values for the entire grid sequentially it would be necessary to open all files and read in all wind speeds into a three dimensional array (longitude, latitude and time). For each grid coordinate then the temporal dimension is sorted out for that specific grid coordinate and the 90%-tile index of that array is calculated. Finally, the corresponding wind speed is returned. The big advantage is that every file would have been opened only once. Unfortunately, it is impossible neither in C nor in FORTRAN to allocate enough memory for one very big array to read in all data. One possible solution is to use only one linear time array and open all files for each grid coordinate:

```plaintext
Float array [350592];
For each x, y grid point {
    For each month, day and hour{
        Jump to u-value; Read u-value;
        Jump to v-value; Read v-value;
        Compute wind strength;
        Write result into array;
    }
    Sort array; Determine 90%-tile value;
}
```

Figure 3. Sequential Algorithm for the Wave 90% Problem.

Compared to wind data there are major differences for waves. All wave files are text files where each wave height (unit in meters) is stored in one line together with its longitude and latitude coordinates as an ASCII string. Here is an extract from the file ‘DaHs_1958010303_0’:

```
-15.00  58.00  3.09232545  310.64188
-14.50  58.00  3.05402374  310.83081
-14.00  58.00  3.00010562  310.82391
```

The first column is the longitude, the second represents the latitude and the third one is the wave height in meters. The fourth represents direction and it is not considered and it will be skipped during computation. The wave files don’t have headers; all temporal information are stored in the filename itself (‘DaHs_YYYMDDHH_0’, year, month, day and hour). Unlike wind fields each wave file contains all information of the whole grid but for only one point in time. Another difference is that the wave grid is not regular; in fact it consists of three different granularities, the closer the coast the finer the resolution. The increments are 1/2°, 1/6° and 1/12° (Figure 4).

Values are generated every 3 hours, that leads to a total number of (40 years x 365 days + 10 days – 2 days (see above)) x 8 values/day = 116,864 files. The size of every wave file is around 245 kilobytes. Unfortunately all the wave files don’t have exactly the same size. Most files have 4,730 lines or grid points but 480 files
of them only in May 1974 and February 1984 do have slightly fewer lines, which means that the wave data is irregular in all three dimensions (longitude, latitude and time).

![Figure 4: Three granularities for wave heights](image)

3. Parallel approach
The parallel program was done in C using MPI-2 [MPI00] to compute the 90%-tile values for winds and waves. The first parallel attempt was to adopt the sequential solution from Figure 3 with only some little changes. A longitudinal partitioning was used to divide the grid as a whole into smaller chunks. Therefore, every processor gets only a column of the grid to compute the 90%-tile values. The optimal number of processors is 11, because 121 longitudinal values are divisible by this. In that case every processor has to handle the same amount of values which takes nearly the same time, so the cluster is optimally balanced. The longitudinal start and end coordinates depend on the rank of the processor and range from rank*11 to rank*11+10 (see Figure 5). After the start of the computation all processors always open and close the same file with a shared file pointer that is provided within MPI-2 I/O. Using that shared file pointer means that there is a synchronization and a cluster update of the file pointer after every read or seek operation of any processor in this file. Due to the fact that there are two components (u and v) to be read for winds it is not possible to tap the full potential of the shared file pointer. There always has to be a back seek, because the u and v components are not back-to-back. Once all files have been read and the
resulting arrays for 11 grid points are created per processor, the 90%-tile values can be determined. The result is written to an output file using a second shared file pointer to guarantee that the results are written in the correct order. The output file is a text file with the same structure as the DaHs files for waves (see above). Then the computation repeats with the next line.

![Figure 5: Spatial partitioning (processors 1 – 11)](image)

This computation with 11 processors has not been adopted, because it would have taken approximately 54 days, even though it has been working correctly. As mentioned above it is a big disadvantage that each file is opened multiple times. Each CPU tries to cache the current file, thus the cache memory is flipped permanently, which means a loss of the very fast cache benefit. Once open, there are a lot of seek operations, because the information of only a single coordinate is needed. These reasons slow down the process significantly. But there are some more reasons for the runtime problem. Due to the shared file pointer there is cluster synchronization after every read or seek operation, which means that the resulting reading time is always the sum of the slowest file operations on any processor. Furthermore the entire cluster has been very busy in general; some processors are busier than others which does not balance the computation.

The second attempt aims to avoid most of these disadvantages with dividing the data into temporal chunks (Figure 6). While using 24 processors each of them only has to open for example 480 / 24 = 20 files for the wind data, which leaves the cache happier. Additionally, the internal file jumping is reduced for winds, because each of the u and v records are read as a whole with the contents of the entire grid. Furthermore the pretty slow MPI-2 I/O file operations are replaced by much faster C commands like fscanf or fprintf for text files and fread or fseek for
binary wind files. From not using shared file pointers it follows that the cluster now synchronizes after every grid point. Thus the imbalance of the cluster in general is compensated a bit, because temporarily very busy processors have a greater chance to catch up. Every processor then broadcasts its partial result array to any other participating processor. After that the array is sorted with quicksort on any processor and the 90%-tile value is detected. Finally the results are sent to the root (processor #0) and only the root writes them into the resulting output text file to maintain the correct order.

Figure 6: Time partitioning

The following pseudo code is for waves and can be adopted for winds with only some few changes:

```c
float array[#points][#files / #procs];
float result[#files];
// #points: total number of grid points
// #files: total number of files
// #procs: total number of processors
// rank: ID of each processor (0..#procs-1)
for files (rank * #files / #procs) to
   (((rank+1) * #files / #procs) - 1)
{
   open file;
   for each grid point x,y {
      read value; write value into array[x, y, file];
   }
   close file;
}
```
for each grid point \( x, y \) {
    broadcast one dimension of \( \text{array}[x, y] \)
    append \( \text{array}[x, y] \) to \( \text{result[]} \)
    quicksort \( \text{result[]} \);
    determine 90%-tile value;
}
if rank is not root then send 90%-tile value to root;
else{
    receive results from other processors;
    write results to output file;
}

Figure 7. Parallel Algorithm for the Wave 90\% Problem.

One possible optimization approach is to approximate the 90%-tile values rather than computing exact results. Therefore each processor computes all 90%-tile values for the entire grid but only for its own partial time period. After that the average is computed for each grid point over all processors. In fact the accuracy of the results is acceptable, but the improvement of the running time is too little. The time need to read all files overbalances the time to sort the arrays and determine the 90%-tile value. In both cases (approximation and exact results) all files have to be read first. Thus the speed advantage affects only the quicksort algorithm.

4. Execution times

It is very important to give some information about the execution times for the sequential solution described in Figure 2. This is actually a computation of more than 26 Gb of external data stored in 480 files. The program was run a machine with the following specifications PIII@1.2GHz each with 512 K cache, 1Gb of RAM. The computation of the 90%-tile value for a point \( x, y \) took in average around 4 hours since all the 480 files had to be opened and read. Therefore, the total computation time could have been around \( 121 \times 81 \times 4 = 39204 \) hours = 1633 days.

Fortunately, the temporal partitioning of the problem reduced dramatically the execution times. The execution of the parallel program was done on the 100 node DELL Beowulf cluster of Boole Centre for Research in Informatics at National University of Ireland, Cork. The specifications of each node are PIII@1.26GHz each with 512 K cache, 1Gb of RAM and 18 Gb rpm SCSI hard drive. The parallel program from Figure 7 was implemented in C using MPI 2 (MPICH library). The execution times of this parallel program are presented in Figure 8 across various number of processors. It is clear that the major component of the time factor comes while reading data from the files assign to each processor which also includes overloads for cache dependencies. It is very important to point out that the program did not give any good results when it was run with less than 4 processors mainly due to cache dependency. This was also reflected in the fact the execution times are laid out on two levels of times. The program was run with a
number of processors between 15 and 30 the execution times were in the interval 62-65 minutes. Another level which is 59-61 minutes was obtained for a number of processors between 35 and 75. However, the smallest execution time was obtained for 98 processors and it was around 54.6 minutes.

Figure 8. Execution Time of Waves

5. Conclusion
Weather data in general is often connected with handling a huge amount of data. Fortunately in that special case the focus is only on a small cutout of the grid. Nevertheless there are a lot of small mistakes that can be made, which can increase the running time of the computation dramatically. Now that there are cluster solutions for winds and waves handling approximately 26 Gigabytes each, a focus to bigger amounts of data in Terabytes is possible.

References


Parallel Approach for the 90% Wind and Wave Hindeast


