



Technical Report
NREL/TP-XXXXX
September 2015

MExtemes Manual

Version 1.00

G. J. Hayman

Revised September 30, 2015
for MExtemes v1.00.00

NREL IS A NATIONAL LABORATORY OF THE U.S. DEPARTMENT OF ENERGY, OFFICE OF ENERGY
EFFICIENCY & RENEWABLE ENERGY, OPERATED BY THE ALLIANCE FOR SUSTAINABLE ENERGY, LLC.
CONTRACT No. DE-AC36-08GO28308

NOTICE

This report was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or any agency thereof.

Available electronically at <http://www.osti.gov/bridge>
Available for a processing fee to U.S. Department of Energy
and its contractors, in paper, from:
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
phone: 865.576.8401
fax: 865.576.5728
email: <mailto:reports@adonis.osti.gov>

Available for sale to the public, in paper, from:
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
phone: 800.553.6847
fax: 703.605.6900
email: orders@ntis.fedworld.gov
online ordering: <http://www.ntis.gov/help/ordermethods.aspx>

Cover Photos: (left to right) PIX 16416, PIX 17423, PIX 16560, PIX 17613, PIX 17436, PIX 17721



Printed on paper containing at least 50% wastepaper, including 10% post consumer waste.

Background

When striving for type certification of a wind turbine, engineers usually must perform a loads analysis for the certification agencies. This usually involves running hundreds or thousands of simulations modeling different conditions. Engineers usually have to generate extreme-event tables and plots from the simulation output.

Engineers at the National Wind Technology Center (NWTC) at the National Renewable Energy Laboratory (NREL) had created a program called MCrunch that included these features and many more, but MCrunch must store the entire dataset concurrently in memory. For such a large number of simulations, the memory requirements would exceed the amount available in an engineer's typical computer. Staff at the NWTC recently addressed this problem by creating MExtremes for the extreme-event tables and plots. This specialized tool processes files sequentially, so only one data file is stored in memory at a time.

Introduction

MExtremes is a MatLab-based tool created to post-process results from wind turbine tests, and aero-elastic, dynamic simulations. MExtremes computes extreme-value event tables and plots for one or more time-series files. The program reads a text-based settings file in conjunction with one or more time-series data files.

The program generates results in the form of MatLab plots, text output files, and/or Excel formatted files.

The extreme value calculations include minimum extreme value, maximum extreme value, the time of extreme event, the time-series file associated with the event, the design load case (DLC), and additional data channel values during any extreme event.

NREL distributes the source scripts, sample files, and a compiled version of MExtremes. If you do not own a MatLab license, you will have to use the compiled version. This requires you to download the free MatLab libraries (MCRInstaller.exe). If you own MatLab and use the source scripts, you will need the Statistics Toolbox.

Retrieving Files from the Archive

You can download the MExtremes archive from our web server at <https://nwtc.nrel.gov/MExtremes>. The file has a name similar to *MExtremes_v1.00.00.exe*, but may have a different version number. Create an MExtremes folder somewhere on your file system and put this file there. When you double click on it from Windows Explorer, it will create some files and folders. To use the scripts, you need to add MExtremes's *Source*, *Source\GUI*, *Source\GUI\colorspace*, and *Source\GUI\distinguishable_colors* folders to the MatLab search path. If you do not have Visual Studio 2010 SP1 run-time libraries installed on your computer then you will also need to install *Redist\vc_redist_x86.exe* (win32) or *Redist\vc_redist_x64.exe* (win64).

Distributed Files

The files included in the MExtremes archive are:

<i>ChangeLog.txt</i>	The list of changes to MExtremes.
<i>Documentation*.pdf</i>	The manual in PDF format.
<i>CertTest\Data*.out</i>	Time-series data used for the certification tests.
<i>CertTest*.m</i>	Scripts for the certification tests.
<i>CertTest*.mext</i>	The MExtremes input files used by the certification script.
<i>CertTest\NREL_Results*</i>	The results from the developer's certification test.
<i>Compiled\MExtremes_win32.exe</i>	The compiled, 32-bit Windows, MExtremes executable for those who do not own MatLab.
<i>Compiled\MExtremes_win64.exe</i>	The compiled, 64-bit Windows, MExtremes executable for those who do not own MatLab.
<i>Source*.m</i>	MExtremes source files.
<i>Source\GUI\colorspace*.*</i>	Color transformation code.
<i>Source\GUI\distinguishable_colors*.*</i>	Code to generate visually distinct colors.
<i>Redist\vc_redist_x86.exe</i>	Visual Studio 32-bit run-time libraries
<i>Redist\vc_redist_x64.exe</i>	Visual Studio 64-bit run-time libraries

Colorspace Source Code

The colorspace transformation code used in MExtremes was written by Pascal Getreuer. The code's license may be found in Appendix A. The code was obtained from the MatLab Central file exchange website:

<https://www.mathworks.com/matlabcentral/fileexchange/28790-colorspace-transformations>

Distinguishable Colors Source Code

MExtremes uses code developed by Tim Holy to generate a set of maximally perceptually-distinct colors. The code's license may be found in Appendix B. The code was obtained from the MatLab Central file exchange website:

<https://www.mathworks.com/matlabcentral/fileexchange/29702-generate-maximally-perceptually-distinct-colors>

Certification Script

Before using MExtremes, you should run the certification script. It is a MatLab script called *MExtremes_certification_tests.m*; make sure that the MExtremes *Source*, *Source\GUI\colorspace*, and *Source\GUI\distinguishable_colors* folders are in the MatLab search path and the *CertTest* folder is the current directory, and then enter *MExtremes_certification_tests* at the MatLab command prompt.

MExtremes will run multiple times. The test procedure will generate outputs into the *CertTest* folder and compare the results to those stored in the *CertTest\NREL_Results* folder. The procedure will write the differences between the output results to a file called *CertTest.comp*. Scan through the file; the only differences should be the date and time stamps in the headers. If you run MExtremes with a version other than the one used in development, there may be slight differences in the last digits of some of the numbers.

Running MExtremes

Prior to running MExtremes, make sure that the MExtremes *Source*, *Source\GUI\colorspace*, and *Source\GUI\distinguishable_colors* folders are in the MatLab search path. From the MatLab command line, type:

```
MExtremes( settingsFile );
```

If the parameter *settingsFile* consists of a simple filename, the file must be located in the current MatLab working directory.

```
settingsFile = 'mysettingsfile.mlif'
```

Optionally, the *settingsFile* parameter may include a fully described path to the settings file.

```
settingsFile = 'C:\fatigue\mysettingsfile.mlif'
```

All output files generated by MExtremes will be placed in the same folder as the settings file.

Running MExtremes without MatLab

If you do not own a license to MatLab, you can run the compiled version of MExtremes, but it will work only with Microsoft Windows. We have done only the minimum required to make it work, but initial tests have shown that it is inferior to running within MatLab. For example, you will see none of the informational messages that would normally go to the MatLab *Command Window*. The program *appears* to do nothing at first and the command prompt returns immediately after you start it, which can give you the impression that it has completed. Eventually, output files will show up in the settings file folder.

To use the compiled version of MExtremes, you first have to install the MatLab Component Runtime (MCR). You need do this only once per version of MatLab. It must be compatible with the MatLab version used to compile MExtremes. You can find a compatible copy of the MCR Installer on the MExtremes Web page (see the Introduction). The file is very large, so be prepared to take some time downloading it. Once you have downloaded it, execute the file (*MCRInstaller_v*.*_win64.exe*, or *MCRInstaller_v*.*_win32.exe*, where *v*.** is the version number of the MCR) to install it, following the prompts.

Next, you have to modify the Windows search path to include the MCR *win32* folder (for example, *C:\Program Files\MATLAB\MATLAB Component Runtime\v77\bin\win32*). To do this in Windows XP, right click on *My Computer* and choose *Properties*. In the *System Properties* window, click on the *Environment Variables* button on the *Advanced* tab. In the *System Variables* list, select *Path* and click on *Edit*. Hit the *Home* key on your keyboard and enter the appropriate path followed by a semicolon. Click on *OK* several times to save your changes.

Using MCR requires that you copy *MExtremes_win32.exe* (or *MExtremes_win64.exe*) to any folder that contains MExtremes input files you want to process. To process an MExtremes input file, open a *Command Prompt* window (*Start, Programs, Accessories, Command Prompt*) and change the current directory to one containing your input file(s). In the window, enter *MExtremes_win32 <input file>* (or *MExtremes_win64.exe*). For example, in MExtremes's *CertTest* folder, enter *MExtremes_win32 Test_01.mext*.

Types of Output

MExtremes can generate the following types of output:

- Extreme event tables (in Excel and/or text format)
- Extreme event charts (in png or MatLab figure format)
- Extreme event scatter plots (in png or MatLab figure format)
- Matrices containing extreme event data in the MatLab save file format

The Settings File

Settings File Format

Use one of the sample *.mext* files found in the *CertTest* folder as a template. Except for a few blocks tables or the list of DLCs and their associated time-series files near the end, you should not add or remove any lines from the sample settings file. For time-series files, list the file names one per line and enclose them in double quotes. MExtremes ignores anything after the *==EOF==* line. The block tables you may need to edit in the settings file are associated with: data channels, calculated channels, event tables, and event charts. You must include the header line for each of these block tables even if the table contains zero entries. For example, even if you set *NumEETables* (see below in the Extreme Events section of this document) to zero, you must include the line:

```
TableName #ChansChanList #InfoChans InfoChanList ...
```

As long as the parameter(s) being read is (are) the first “word(s)” on the line and it is (they are) separated from the comment portion of the line with white space, MExtremes will not be affected. The amount of spacing is not important—use what-

ever looks good to you. You must separate any lists of numbers with white space—not commas. Block tables have their own special formats.

Title Line

The second line of the file is for a job title. The line can be of any length and contain anything you might find useful. Although MExtremes does not currently use this line for anything, it may someday include it in output files.

Job Options

There are currently three inputs in the Job Options section of the settings file. The first, *EchoInp*, allows you to request MExtremes to echo your settings file input parameters to another file so you can debug your settings file. The echo file has the same root name as your settings file and *echo* for an extension.

One common complaint about our old Crunch postprocessor is that if you add new channels to your data files, you have to change the channel numbers used in the settings file. We find that quite tedious and prone to error when we have more than 100 channels in our data files. We decided to allow users to specify channels by their names instead of numbers. If you want to take advantage of this feature, set *StrNames* to true and use the channel names surrounded by \$ anywhere channel numbers would normally go. Here's an example:

```
"Tower Clearance" 3 $TwrClrc1$ $TwrClrc1$ $TwrClrc1$ ...
```

If you enable *StrNames*, you don't have to use them exclusively; numbers will still work as MExtremes simply replaces all the *\$ChanName\$* strings found in the settings file with the channel numbers before processing the file. Use the *RealFmt* parameter to specify the formatting of fixed-point numbers in the output. The standard MatLab *sprintf* syntax applies. You must enclose this string in double quotes.

Input-Data Layout

MExtremes can automatically determine the number of columns and rows in your time-series data files, or you can specify these things explicitly. If you want MExtremes to determine the channel layout, set the number of input channels to zero and don't give it a list of channels.

If you want MExtremes to parse channel names and units from the settings file, specify the lines that contain such information. If you specify a zero for either, MExtremes assumes there is no such line. If you tell MExtremes which line contains names, it will use that line to determine the number of channels. It assumes that channel titles are a contiguous group of letters and/or symbols that do not contain any whitespace. For example, if the channel-names line is as follows:

```
Chan1 "Chan2"Chan_3, Chan-4 " Chan 5",
```

MExtremes would find six channels whose titles would be *<Chan1>*, *<"Chan2"Chan_3,>*, *<Chan-4>*, *<">*, *<Chan>*, and *<5">* (without the angle brackets). If your input file has a title line similar to this:

```
"Chan 1" "Chan 2" "Chan 3"
```

and it has three columns of data, you cannot use the auto-detection feature for columns because MExtremes will think it has six channels (three named *"Chan"*). If something such as this is the case, you should specify the channel information in the settings file.

Currently, you must either specify channel names and units within the settings file, or tell MExtremes to read them from the time-series data file(s).

If you choose auto-detection of channels and give a zero for the line containing the units, MExtremes will not include any unit strings in the output. If that is the case, you must not specify the units for calculated channels.

If you want to specify channel layout, set *NumChans* to a number greater than zero. After the comment line describing the format for the channel table, enter *NumChans* lines with the channel names in double quotes, the units in double quotes, a fixed-point number for the scale, another for the offset, and another value for the partial safety factor (PSF) type. For the *PSF_Type* parameter valid options are 0 = do not apply a PSF, 1, 2, 3, or 4. The actual PSF for a specific *PSF_Type* is found in the Input Files section and is different for each design load case (DLC).

MExtremes cannot currently reorder or use a subset of channels, but it can rename channels from the original data file with this feature. It will apply scales and offsets as it reads the data. If you let MExtremes auto-detect channels, it cannot apply scales and offsets.

Unlike Crunch, MExtremes does not require that all files have the same number of lines. Like Crunch, it does require that the channel layout be the same for all files; that is, all data files must have the same number of channels and the same number of header lines.

MExtremes can also create new channels using typical MatLab expressions with references to other channels. (See the Calculated-Channels section.)

Calculated Channels

You can create new channels of data through the calculated-channels feature, which allows you to specify a single MatLab expression for each new channel. MExtremes numbers calculated channels in the order created; the number of the first one is one more than the number of input channels.

NumCChan specifies the number of new channels being created. If a calculated channel will need a seed to initialize a pseudo-random sequence, use *Seed* to store that value. Even if you don't use the random-number generator, you must include *seed* in the input file. A comment line describing the format for the lines describing the calculated channels is next. After that, enter one line for each calculated channel. These lines contain three fields; each should be enclosed in double quotes and separated by white space. The first field is for the channel name. The second is for the channel units (omit this if you are not using units). The last field is the MatLab-style expression. The time-series data is stored in a MatLab array named *FileInfo.Time* and has dimensions of (*nLines*, *nChannels*), where *nLines* are the number of lines in the time-series and *nChannels* is the number of channels found in the time-series.

Figure 1 is an example of the calculated channels section of the settings file when string names are used (*StrNames=true*):

```
----- Calculated Channels -----
2      NumCChan The number calculated channels to generate.
1234567890 Seed The integer seed for the random number generator
Col_Title Units Equation Put each field in quotes. Titles and..
"RootFMxy1" "(kN)" "sqrt( FileInfo.Time(:, $RootFxc1$).^2 +
      FileInfo.Time(:, $RootFyc1$).^2 )"
"RootMMxy1" "(kN.m)" "sqrt(FileInfo.Time(:, $RootMxc1$).^2 +
      FileInfo.Time(:, $RootMyc1$).^2 )"
```

Figure 1. Example input for calculated channels.

Extreme Events

NumEETables specifies the number of generated extreme event tables. *NumEECharts* specifies the number of generated extreme event charts. If you set *FindPeaks* to *True*, MExtremes will use a peak finding algorithm to find implied local peaks in the time-series data. *TimeChan* specifies the data channel column representing time. *WSChan* specifies the data channel column representing the wind speed channel. The next four parameters are only used if you set *BinModel* to a value other than 0 for a DLC (see Input Files section, below). These four parameters are ignored if *BinModel* is set to 0 for a given DLC. When *BinModel* is greater than 0, the data is placed into wind speed bins in the range, [*WSmin*, *WSmax*]. Any data outside this range will not be considered when generating bin-based extremes. *WSMaxBinSize* establishes the maximum bin size for these wind speed bins. Refer to the Input Files section for details on the binning models used by MExtremes.

If you wish to see detailed extreme event scatter plots for each channel listed in the Extreme Event Tables or Extreme Event Charts, set *ScatterPlots* to *True*. MExtremes will create two plots for each channel; one for the maximum extreme events and one for the minimum extreme events. The data points in these plots correspond to the extreme events for each time-series file analyzed. If a DLC is using wind speed binning (*BinModel* > 0) then the x-axis denotes the wind speed bin associated with the extreme event. Otherwise the x-axis is simply the wind speed value associated with the extreme event. If *PlotFormat* = 1, the scatter plots and charts are saved as PNG files. If *PlotFormat* = 2, then the plots and charts are saved in the MatLab figure format.

If you set *WrEEvtsTxt* to *true*, MExtremes will write tables of extreme events for each table group specified in the settings file and use a file name *RootName* followed by *.extr*, where *RootName* is the settings file name without the file extension. If *WrEEvtsXLS* is set to true, MExtremes will generate the same event tables in an Excel worksheet and use the value of *RootName* followed by *_EEvts.xlsx* (if *XLStype* is set to "xlsx") or *_EEvts.xls* (if *XLStype* is set to "xls"). For Excel output, *HeadTnS* sets the tint and shade of the header area (-1.0 to 1.0), with -1 being darkest and 1 lightest. Similarly, *DiagTnS* sets tint and shade of the diagonal (-1.0 to 1.0), with -1 being darkest and 1 lightest.

If *NumEETables* is greater than zero, you must include one line for each extreme event table below the tables block header line, which starts with the text: *TableName*. The table definition consists of: table name in double quotes, the number of extreme event channels included in the table, a white-space delimited list of channel names or indices, the number of informational channels, and the list of informational channel names or indices

If *NumEECharts* is greater than zero, you must include one line for each extreme event chart below the charts block header line, which starts with the text: *ChartName*. The chart definition consists of: chart name in double quotes, the y-axis label in double quotes, the chart type (*max_abs*, *max*, or *min*, corresponding to the maximum of the absolute value of extremes, the maximum value extremes, or the minimum value extremes, respectively), the number of extreme event channels included in the chart, and a white-space delimited list of channel names or indices.

```
----- Extreme Events -----
3      NumEETables      Number of tables of extreme events.
1      NumEECharts      Number of charts of extreme events.
true   FindPeaks       Should we use the peak finder?
1      TimeChan        The time step channel (used ...
2      WSChan          The primary wind-speed channel (used ...
0      WSmin           Minimum wind speed value to ...
31     WSmix           Maximum wind speed value to consider ...
2      WSMaxBinSize    Maximum width of a wind-speed bin.
false  ScatterPlots    Create scatter plots for each channel ...
1      PlotFormat      Format for saving any of the ...
true   WrEEvtsTxt      Write the extreme-event tables to a ...
true   WrEEvtsXLS     Write the extreme-event tables to an ...
"xlsx" XLStype        Type of Excel file (xls=2003,xlsx=2007).
5      ThemeColor      Excel color theme to use. In Excel ...
0.8    HeadTnS         Tint and shade of the header area ...
0.9    DiagTnS        Tint and shade of the diagonal ...
TableName #Chans      ChanList      #InfoChans
InfoChanList
"Tip Deflections" 6      23 24 26 27 29 30 3 1 2 5
"Tower Clearance" 3      32 33 34 3 1 2 5
"Blade-1 Loads" 4      134 52 135 55 3 1 2 5
ChartName YLabel      Type      #Chans      ChanList
"Tip Deflectons" "Maximum Magnitude of Tip Deflections" max_abs 6
23 24 26 27 29 30
```

Figure 2. Example of the Extreme Events section of the settings file.

Input Files

Set the *FileFormat* parameter to 1 when using ascii-formatted FAST output files. Set *FileFormat* to 2 when using binary FAST output files. *NumDLCs* specifies the number of DLC groups specified in the settings file. You may only set this to zero if you are using the *@Restore* command. The remainder of the input files section is split up into *NumDLCs* design load cases (DLCs). A DLC group consists of a DLC header

line, followed by a list of input time-series files for that DLC, one line for each file in double quotes. The header line includes: the number of files in the DLC, followed by a list of four partial safety factors (PSFs), the BinModel value (0, 1, or 2), and the name of the DLC in double quotes. The PSFs are used when a PSF_Type was specified in the channels table. The BinModel options follows the specifications given in IEC 61400-1 edition 3 and are as follows:

0 : Do not bin extremes, instead take the extreme value across all DLCs for a given channel (after applying any partial safety factors). And a single extreme for a given DLC.

1 : On a DLC-basis, assign each input file (within the DLC) to a wind speed bin, according to the file's mean wind speed value. Find the extremes for each input file within a DLC and then sort these into their respective wind speed bins. Find the mean of the extremes for a given bin. This becomes the extreme value of the bin. Then find the overall extreme value for the DLC based on these mean extreme values. If the overall extreme value across all DLCs corresponds to a mean extreme value for a binned input file, then we determine the closest extreme event to this calculated mean extreme. The maximum extreme event, corresponding to this calculated mean extreme is taken as the closest extreme event which is larger than the mean. For minimum extremes take the closest extreme event which is smaller than the calculated mean extreme. Also, for plotting and simplified reporting we will include the calculated mean value, in addition to the closest actual extreme event data value to the mean. We need the closest actual event in order to report when the event occurred in the dataset as well as the values of other channels at that time.

2 : For DLC 2.1, 2.2 and 5.1, the extreme value will be based on Model 1, except that only the largest half the bin extremes will be used when computing the mean of the max and only the smallest half of the bin extremes will be used when computing the mean of the min.

```
----- Input Files -----
1  FileFormat  Flag determining input file format. 1 = ascii, 2 = binary
2  NumDLCs     The number of Design Load Cases.
3  1.1  1.3  1.5  1.7  0  "DLC 2.3"  (NumDLCfiles, PSF1, PSF2 ...
".\Data\DLC2.3_1.out"
".\Data\DLC2.3_2.out"
".\Data\DLC2.3_3.out"
2  1.0  1.15  1.4  1.6  1  "DLC 2.1"  (NumDLCfiles, PSF1, PSF2 ...
".\Data\DLC2.1_1.out"
".\Data\DLC2.1_2.out"
```

Figure 3. Example of the input files section of the settings file.

Saving or Restoring Data

You can save all the processed time-series data into a single MatLab save file using the `@Save` command. Simply create a separate line starting with `@Save` followed by the filename you wish to store the data into. If you previously saved time-series data, you can restore that data using the `@Restore` command. Create a separate line starting with `@Restore` followed by the filename which contains the data you are restoring. MExtremes is capable of restoring data via the `@Restore` command, parsing new time-series data via the DLC groups, and then saving the combined information into a new save file using the `@Save` command.

```
----- Input Files -----
@Save "MyData.dat"
@Restore "OldData.dat"
1  FileFormat  Flag determining input file format. 2
NumDLCs     The number of Design Load Cases.
3  1.1  1.3  1.5  1.7  0  "DLC 2.3"  (NumDLCfiles, PSF1, PSF2
".\Data\DLC2.3_1.out"
```

Figure 4. Example of the @Save and @Restore commands.

Limitations

MExtremes has the following limitations:

- All files must have the same channel layout.
- Computers must have sufficient virtual memory to contain the longest file's data. All time-series data are stored as four-byte, single-precision fixed-point numbers to save space.
- All files must have the same time step.
- Channel names and units may not contain any white space.

Future Work

- write out the input data after MExtremes applies scales and offsets, filters the data, and adds calculated channels.

Acknowledgments

Greg Hayman of the NWTC wrote MExtremes. This program is based on MCrunch which was written by Marshall Buhl. Jason Jonkman, Marshall Buhl, Rick Damiani, and Andrew Platt created the outline for the technical functionality of MExtremes.

Feedback

If you have problems with MExtremes, please contact Greg Hayman. If he has time to respond to your needs, he will do so, but please do not expect an immediate response. Please send your comments or bug reports to:

Greg Hayman
NWTC/3811
National Renewable Energy Laboratory
1617 Cole Blvd.
Golden, CO 80401-3393
United States of America

Web: <http://wind.nrel.gov>
Email: gregory.hayman@nrel.gov

References

Buhl, M.L., Jr. (2008). *MCrunch User's Guide*. NREL/EL-500-43139. Golden, CO: National Renewable Energy Laboratory, 15 pp.

International Electrotechnical Commission, "Wind turbines – Part 1: Design Requirements" (2005). IEC 61400-1:2005(E), 179 pp. Geneva, Switzerland.

The MathWorks. "The MathWorks – MatLab and Simulink for Technical Computing." <http://www.mathworks.com>. Accessed March 28, 2008.

Veers, P.S.; Winterstein, S.R.; Nelson, D.V.; Cornell, C.A. (1989). "Variable Amplitude Load Models for Fatigue Damage and Crack Growth." Potter, J.M. and Watanabe, R.T., eds. *Development of Fatigue Loading Spectra*, ASTM STP 1006, p. 172.

Appendix A – Colorspace License

Copyright (c) 2005-2010, Pascal Getreuer
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Appendix B – Distinguishable Colors License

Copyright (c) 2010-2011, Tim Holy
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.