DATA PRODUCT SPECIFICATION FOR LOCAL RANGE TEST

Version 1-00
Document Control Number 1341-10005
2012-07-17

Consortium for Ocean Leadership
1201 New York Ave NW, 4th Floor, Washington DC 20005
www.OceanLeadership.org

in Cooperation with

University of California, San Diego
University of Washington
Woods Hole Oceanographic Institution
Oregon State University
Scripps Institution of Oceanography
Rutgers University

Document Control Sheet

Version	Date	Description	Author
0-01	2011-12-19	Initial Draft	M. Lankhorst
0-02	2012-04-23	Formatting update	M. Lankhorst
0-03	2012-06-13	Addressed comments from focused review.	M. Lankhorst
0-04	2012-07-06	Addressed comments from formal review. Added capability for n-dimensional Z and DATLIMZ coordinates.	M. Lankhorst
1-00	2012-07-17	Initial Release	E. Chapman

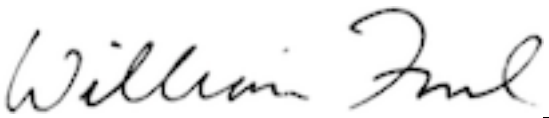
Signature Page

This document has been reviewed and approved for release to Configuration Management.

OOI Chief System Engineer: 

Date:2012-07-17

This document has been reviewed and meets the needs of the OOI Cyberinfrastructure for the purpose of coding and implementation.

OOI CI Signing Authority: 

Date: 2012-07-17

Table of Contents

1 Abstract

This document specifies the OOI Local Range Test, which is the computation to test whether a given data point falls within pre-defined ranges. These valid ranges are given in separate tables and are supposed to be site-specific, hence the name “local”. In addition, the valid ranges (i.e. permissible minimum and maximum values) can vary as functions of an independent parameter such as the depth in water where the measurement was taken.

2 Introduction

2.1 Author Contact Information

Please contact Matthias Lankhorst (mlankhorst@ucsd.edu) or the Data Product Specification lead (DPS@lists.oceanobservatories.org) for more information concerning the computation and other items in this document.

2.2 Metadata Information

2.2.1 Data Product Name

The parameter name for the quantity resulting from this computation is composed of the OOI Core Data Product Name that is being described, with “_LOCLRNG_QC” attached.

- [*DATAPRODUCT*]_LOCLRNG_QC

The OOI Core Data Product Descriptive Name for this parameter is

- Local Range Test

2.2.2 Data Product Abstract (for Metadata)

The local range test is a QC test that reports whether data points fall within given ranges.

2.2.3 Computation Name

The computation name is LOCLRNG.

2.2.4 Computation Abstract (for Metadata)

The local range test is a QC test that reports whether data points fall within given ranges. The ranges as well as the data points are given as functions along an independent coordinate axis. The algorithm will interpolate the ranges to the same coordinate values as the data points, and then determine whether the data points are inside the corresponding ranges. The coordinate may be multi-dimensional, in which case the data are presumed to be represented in a multidimensional space.

2.2.5 Instrument-Specific Metadata

None for this data product.

2.2.6 Data Product Synonyms

None.

2.2.7 Similar Data Products

The OOI also uses a Global Range Test (see OOI document no. 1341-10004), the ranges of which are typically not used in a site-specific context. In addition the Global Range Test does not allow the ranges to vary with an independent coordinate parameter (such as depth or time).

2.3 Instruments

This data product is not instrument-specific.

2.4 Literature and Reference Documents

None.

2.5 Terminology

2.5.1 Definitions

There are no definitions or acronyms specific to this document.

2.5.2 Acronyms, Abbreviations and Notations

There are no definitions or acronyms specific to this document.

2.5.3 Variables and Symbols

The following variables and symbols are defined here for use throughout this document:

DAT	Input data values
Z	Independent (i.e. coordinate axis) parameter for each DAT. Note that although the name suggests this be the vertical axis, any independent parameter can be used. Z can be multi-dimensional, in which case each column of Z specifies an independent coordinate axis.
DATLIM	Valid minimum and maximum values for DAT at the given DATLIMZ.
DATLIMZ	Independent coordinate parameters for DATLIM.
OUT	Output flags. 0 denotes values outside the valid range, 1 inside.

3 Theory

3.1 Description

DATLIM defines a valid range as a function of DATLIMZ, both defined in a look-up table. The algorithm determines if data points DAT, given as a function of Z, fall within this valid range. For each data point DAT, the limits DATLIM (given at coordinates DATLIMZ) are interpolated to the coordinate values Z such that they correspond to DAT. Then, a mathematical comparison (greater than or equal, lesser than or equal) determines whether the data point falls in the valid range or not.

3.2 Mathematical Theory

Contained in Description above.

3.3 Known Theoretical Limitations

The validity of the output is limited by the quality of the ranges given in the lookup tables.

3.4 Revision History

None.

4 Implementation

4.1 Overview

The software first interpolates the given DATLIM to the actual Z parameters of the data DAT, yielding a valid minimum and maximum value for each Z. Then, a simple application of the **>** and **<** operators determines whether the actual DAT falls within this range or not.
Appendix A contains exemplar source code in MatLab.

4.2 Inputs

DAT	Column vector of class double
Z	Array of class double, same number of rows as DAT. Number of columns defines number of dimensions of coordinate space. Z and DATLIMZ must have same number of columns.
DATLIM	Two-column array of class double
DATLIMZ	Vector of class double, same number of rows as DATLIM

4.3 Processing Flow

Prior to calling this algorithm, the proper ranges must be extracted from the lookup tables. Once these input parameters are known, no special considerations are needed in terms of processing flow other than executing code like the exemplar code.

4.4 Outputs

OUT	Array of class logical, same size as DAT

4.5 Computational and Numerical Considerations

4.5.1 Numerical Programming Considerations

None.

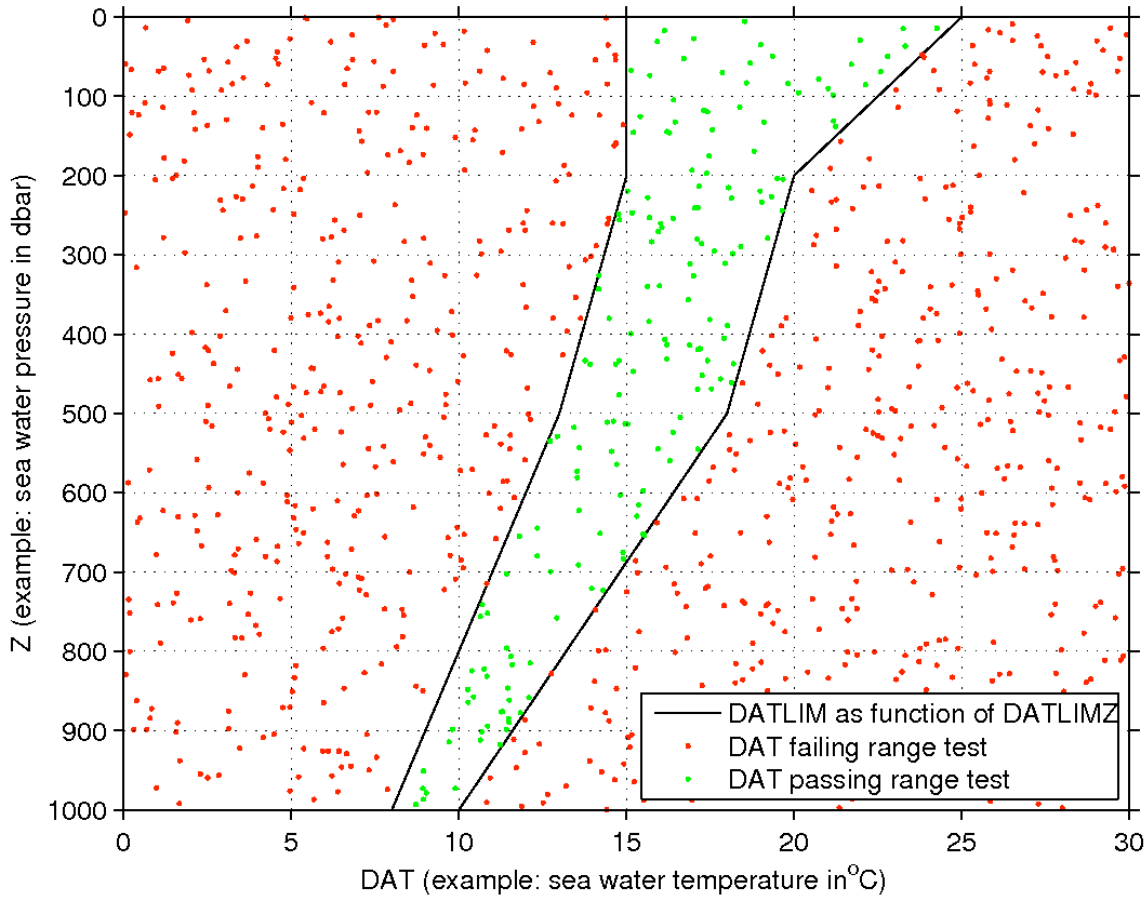
4.5.2 Computational Requirements

None.

4.6 Code Verification and Test Data Set

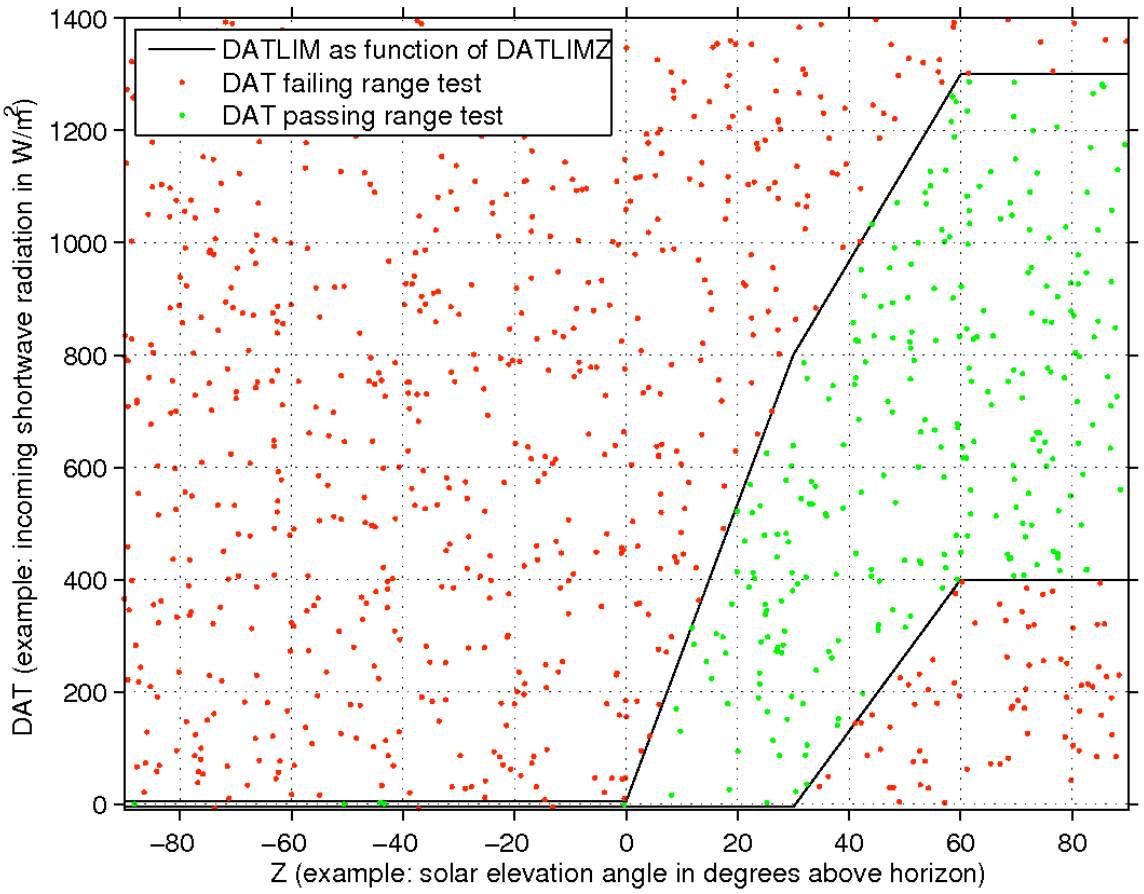
The following table and figure show an example that can be thought of as sea water temperature as a function of pressure (which effectively means depth):

DATLIM	DATLIMZ	DAT	Z	OUT
15 25 15 20 13 18 8 10	0 200 500 1000	(1000 random points between 0 and 30)	(1000 random points between 0 and 1000)	(see figure below)



The following table and figure show an example that can be thought of as incoming shortwave radiation as a function of solar elevation angle:

DATLIM	DATLIMZ	DAT	Z	OUT
-5 5 -5 5 -5 800 400 1300 400 1300	-90 0 30 60 90	(1000 random points between -10 and 1400)	(1000 random points between -90 and 90)	(see figure below)



Appendix A Example Code

```
% DATAQC_LOCALRANGETEST   Data quality control algorithm testing
%    if measurements fall into a user-defined valid range.
%    This range is not constant but varies with measurement location.
%    Returns 1 for presumably good data and 0 for data presumed bad.
%
% Time-stamp: <2012-07-06 15:33:41 mlankhorst>
%
% USE CASE SCENARIO: A time series of measurements DAT is given at a
%   time-varying altitude Z, e.g. air temperature from a rising
%   weather balloon. The purpose of this routine is to identify
%   data points that fall within an expected range, which varies
%   with altitude (e.g. at the ground, accept temperatures -20..35
%   degrees C, but at 10000m altitude accept -80..-40 degrees C).
%
%   Z can have more than one dimension, in which case the locations
%   are interpreted as being in a multi-dimensional space. Use this
%   feature e.g. if your temperature ranges are further categorized
%   by season/month, in which case one dimension might be the
%   altitude, and the other the time of year.
%
% USAGE:   OUT=dataqc_localrangetest(DAT,Z,DATLIM,DATLIMZ);
%
%          OUT: Boolean, 0 if value is outside range, else 1.
%
%          DAT: Input dataset, a numeric real scalar or column
%               vector.
%          Z:   Location of measurement DAT. Must have same number
%               of rows as DAT and same number of columns as DATLIMZ.
%          DATLIM: Two-column matrix with the minimum (column 1)
%               and maximum (column 2) values considered valid.
%          DATLIMZ: Matrix with the locations where DATLIM is
%               given. Must have same number of rows as DATLIM and
%               same number of columns as Z.
%

function out=dataqc_localrangetest(dat,z,datlim,datlimz);

  checkinput(dat,'DAT');
  checkinput(z,'Z');
  checkinput(datlim,'DATLIM');
  checkinput(datlimz,'DATLIMZ');

  [numlim,ndim]=size(datlimz);

  [tmp1,tmp2]=size(datlim);
  if tmp1~=numlim
    error('DATLIM and DATLIMZ must have same number of rows.')
  end
  if tmp2~=2
    error('DATLIM must have exactly 2 columns.')
  end

  [num,tmp2]=size(z);
  if tmp2~=ndim
    error('Z must have same number of columns as DATLIMZ.')
  end

  if ~isvector(dat)
```

```
    error('DAT must be vector.');
  end
  dat=dat(:);
  if num~=length(dat)
    error('Length of DAT must match number of rows in Z.')
  end

  if ~all(datlim(:,2)>datlim(:,1))
    warning(['Second column values of DATLIM should be greater than' ...
             ' first column values.'])
  end

  if ndim==1
    lim1=interp1(datlimz,datlim(:,1),z);
    lim2=interp1(datlimz,datlim(:,2),z);
  else
    F=TriScatteredInterp(datlimz,datlim(:,1));
    lim1=F(z);
    F=TriScatteredInterp(datlimz,datlim(:,2));
    lim2=F(z);
  end

  ff=find((isnan(lim1))|(isnan(lim2)));
  lim1(ff)=max(datlim(:,2));
  lim2(ff)=min(datlim(:,1));

  out=(dat>=lim1)&(dat<=lim2);




function checkinput(x,txt)

  if ~isnumeric(x)
    error('%s must be numeric.',txt)
  end
  if ~ismatrix(x)
    error('%s must be a matrix.',txt)
  end
  if ~all(isreal(x(:)))
    error('%s must be real.',txt)
  end
```