



# DATA PRODUCT SPECIFICATION FOR L2 BULKFLX DATA PRODUCTS

Version 0-03  
Document Control Number 1341-00370  
2013-10-25

Consortium for Ocean Leadership  
1201 New York Ave NW, 4<sup>th</sup> Floor, Washington DC 20005  
[www.OceanLeadership.org](http://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



## Signature Page

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

OOI Senior Systems Engineer: \_\_\_\_\_

Date: \_\_\_\_\_

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: \_\_\_\_\_

## Table of Contents

1	Abstract .....	1
2	Introduction .....	1
2.1	Author Contact Information .....	1
2.2	Metadata Information .....	1
2.3	Instruments.....	5
2.4	Literature and Reference Documents .....	5
2.5	Terminology.....	5
3	Theory .....	6
3.1	Description .....	6
3.2	Mathematical Theory.....	6
3.3	Known Theoretical Limitations .....	8
3.4	Revision History .....	8
4	Implementation .....	8
4.1	Overview .....	8
4.2	Inputs.....	8
4.3	Processing Flow .....	9
4.4	Outputs.....	9
4.5	Computational and Numerical Considerations.....	9
4.6	Code Verification and Test Data Set.....	9
Appendix A	Example Matlab code from WHOI UOP group .....	1
Appendix B	Output Accuracy .....	16
Appendix C	Sensor Calibration Effects .....	16

## 1 Abstract

The Bulk Meteorological instrument addresses two purposes: 1) Observing the surface meteorology and 2) Providing the data required to compute the air-sea fluxes of heat, freshwater and momentum. The air-sea flux of heat is the sum of five separate heat fluxes: latent heat flux, sensible heat flux, net shortwave radiative heat flux, net longwave radiative heat flux and rain heat flux. The air-sea flux of freshwater is the difference between rain and evaporation. The air-sea flux of momentum is the vertical transfer of horizontal momentum from the air to the ocean and is called the wind stress.

This document describes the computation used to calculate the OOI Level 2 Bulk Flux products computed from data observations made with the Bulk Meteorological (METBK) family of instruments. This document is to be used by OOI programmers to construct appropriate processes to create the L2 Bulk Flux estimates of air-sea flux of heat and wind stress.

## 2 Introduction

### 2.1 Author Contact Information

Please contact Robert Weller ([rweller@whoi.edu](mailto:rweller@whoi.edu)), Janet Fredericks ([jfredericks@whoi.edu](mailto:jfredericks@whoi.edu)) or the Data Product Specification lead ([DPS@lists.oceanobservatories.org](mailto: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 OOI L2 Core Data Product Names and Descriptive Names for the products are:

<b>Name</b>	<b>Descriptive Name</b>
SPECHUM	Specific Humidity
SPHUM2M	Specific Humidity at 2 meters
WIND10M	Wind Velocity at 10 meters
TEMPA2M	Air Temperature at 2 meters
RAINRTE	Rain Rate
NETSIRR	Net Shortwave Irradiance
NETLIRR	Net Longwave Irradiance
LATNFLX	Latent Heat Flux (from Bulk Formulae method)
SENSFLX	Sensible Heat Flux (from Bulk Formulae method)
RAINFLX	Rain Heat Flux
HEATFLX	Net Heat Flux
FRSHFLX	Freshwater Flux
MOMMFLX	Momentum Flux (Wind Stress)

#### 2.2.2 Data Product Abstract (for Metadata)

This document describes how to compute the L2 BULKFLX data products from the Level 1 BULKMET (DCN 1341-00360) and VELPTMN (DCN 1341-00790) products.

The OOI surface buoys equipped with bulk meteorological sensor packages return the Level 1 BULKMET data, with each variable being observed at the height of each respective sensor: wind velocity, wind direction, incoming shortwave radiation, incoming longwave radiation, barometric pressure, rain rate, air temperature, and relative humidity. The BULKMET instrumentation also has sensors for ocean temperature and salinity, providing an observed near-surface temperature and salinity, each at its respective depth. The L1 VELPTMN provides a near-surface current as a

proxy for surface current. These basic observables are used with the metadata, providing the height/depth of the observations, and the empirical formulae called Bulk Formulae (see section 2.2.4) to produce surface meteorological variables at standardized heights (2 m above the sea surface for air temperature and humidity and 10 m above the sea surface for wind velocity) and also the air-sea flux products. The computation of the meteorological variables at the standardized heights provides products at common heights agreed upon by the community. The computation of the air temperature and humidity at 2m and the wind velocity at 10 m requires knowledge of the air-sea fluxes and has thus been combined with the production of the flux values to yield a combined L2 BULKFLUX data set.

SPECHUM: **Specific humidity** is defined as the ratio of the mass of water vapor ( $M_w$ ) to the mass of moist air ( $M_w + M_a$ ). [The Original Rotronic Humidity Manual, 2005]

SPHUM2M: The **specific humidity adjusted to a height of 2 meters** is derived from input and model-based values to be used in quality assessment procedures.

WIND10M: **Wind velocity adjusted to a height of 10m** is derived from the observed wind velocity.

TEMPA2M: **Air temperature adjusted to a height of 2 meters** is derived from the observed air temperature.

RAINRTE: The **rate of rain** is computed from the change of rain level relative to time, filtered to remove noise and the effects of evaporation on the rain level.

NETSIRR: **Shortwave** fluxes are in the wavelength band 0.3 to 3  $\mu\text{m}$ . Downwelling shortwave radiation at the surface ( $R_{s\downarrow}$ ) has a component due to the direct solar beam, and a diffuse component scattered from atmospheric constituents and reflected from clouds. Upwelling shortwave radiation ( $R_{s\uparrow}$ ) comes from reflection at the surface and the re-emergence of radiation backscattered from the upper ocean. In clear water, shortwave radiation penetrates to a depth of several tens of meters, influencing the thermal structure of the ocean surface layer. The ratio of downwelling to upwelling shortwave is the surface albedo ( $\alpha$ ), which depends on solar elevation, cloudiness and wavelength. For use in bulk algorithms, a single value of 0.055 for broadband albedo, based on the ratio of daily averaged upwelling and downwelling shortwave flux, has been found to be satisfactory. [Bradley & Fairall, 2006]

NETLIRR: **Longwave** fluxes range from 3 to around 50  $\mu\text{m}$  wavelength. Downwelling longwave radiation ( $R_{l\downarrow}$ ) originates from the emission by atmospheric gases (mainly water vapour, carbon dioxide and ozone), aerosols and cloud droplets. It is thus linked quite closely to the particular regional climate conditions. Upwelling longwave from the sea surface  $R_{l\uparrow}$  depends on the ocean skin temperature and surface emissivity ( $\epsilon$ ), with a small contribution due to reflection of the downwelling component. Emissivity is wavelength-dependent, and a spectrally integrated value of 0.97 is commonly used. Longwave absorption and emission take place in about the top 0.5 mm of water. [Bradley & Fairall, 2006]

LATNFLX: The **latent heat flux** is the transfer of energy as a result of evaporation or condensation (latent heat).

SENSFLX: The **sensible heat flux** is the transfer of energy as a result of direct contact between the ocean and air (sensible heat).

RAINFLX: Raindrops evaporate and cool as they fall. **Rain heat flux** is the cooling associated with rain falling at an air temperature less than the sea surface temperature.

HEATFLX: **Net heat flux** is the combined estimate of air-sea fluxes of heat, freshwater and

momentum using the Bulk Met observables and the Bulk Formulae method. The net heat flux is the sum of the short- and longwave fluxes, latent and sensible heat fluxes and the rain heat flux.

FRSHFLX: **Freshwater flux** is the difference between rate of rain and rate of evaporation.

MOMMFLX: **Momentum flux** is the vertical transfer of horizontal momentum from the air to the ocean and is called the wind stress. It is the transfer of energy from the wind physically pushing against the water. The Momentum Flux is derived from the relative wind direction representing stress toward not stress from and includes the Northward and Eastward vector components of the average momentum flux relative to North True.

### 2.2.3 Computation Name

The Bulk Formulae Method

### 2.2.4 Computation Abstract (for Metadata)

Different humidity variables can be computed from each other. Relative humidity and specific humidity are the common humidity variables, with relative humidity typically being what is measured and specific humidity being what is required for subsequent computation of latent heat flux.

For clarity, there are two methods being used on OOI surface buoys to compute air-sea fluxes: 1) from the Bulk Meteorological observations and 2) from the Direct Covariance Flux System (DCFS) observations. Note that the Bulk Meteorological observations, taken together with surface currents and sea surface temperature, can be used to compute all the air-sea heat flux components (net longwave, net shortwave, latent, sensible, and rain), the air-sea freshwater flux, and the air-sea momentum flux. Note that the DCFS provides the ability to compute only two components of heat flux (latent and sensible), one part of the freshwater flux (evaporation), and the momentum flux. The computation of the latent and sensible heat flux and of the wind stress using the Bulk Meteorological observations uses formulae with parameterizations developed over the years by comparisons with DCFS observations; thus, this way of computing these flux components is called the Bulk Formulae method.

The Bulk Meteorological instrument is relatively low powered and will run continuously, typically sampling once per minute. Thus, the Bulk Meteorological data can be used with the Bulk Formulae method to provide continuous, one-minute records of the air-sea fluxes of heat, freshwater, and momentum. The Bulk Meteorological instruments provide the basis for computation of all the desired air-sea fluxes of heat, freshwater, and momentum. The DCFS instrument provides a second means to compute a subset of the fluxes; it makes rapid (e.g. 20 Hz) observations of turbulent three-dimensional wind velocity ( $u'$ ,  $v'$ ,  $w'$ ) and of fluctuations in air temperature ( $T'$ ) and humidity ( $q'$ ); computation of averages over time (~20 minutes) of products  $\langle u'w' \rangle$ ,  $\langle v'w' \rangle$ ,  $\langle w'q' \rangle$ , and  $\langle w'T' \rangle$  provide estimates, respectively of the two horizontal components of wind stress, of latent heat flux, and of sensible heat flux.

The net longwave irradiance, net shortwave irradiance, and rain heat flux do not have a significant sensitivity to the vertical structure of the lower atmosphere and their computation is straightforward once all the required derived data variables are computed. However, the vertical structures of the humidity, temperature, and wind velocity profiles are not only characterized by strong vertical gradients, roughly logarithmic, but also vary with the stability of the lower atmosphere. The more stable the lower atmosphere, the stronger the vertical gradients. The Bulk Formulae (Fairall et al., 2003) build in the stability dependent structure of the lower atmosphere and allow one to use meteorological data observed at known heights above the sea surface together with parameters called the Stanton and Dalton numbers at a fixed height of 10 m to iteratively compute sensible and latent heat flux together with wind stress.

The fluxes from the Bulk Formulae and DCFS methods will at times not agree, as the observations used to develop the parameterizations used in the Bulk Formulae method do not yet span the wide range of calm to energetic meteorological conditions to be seen in OOI deployments. There is science to be done in understanding the differences between fluxes from the two methods, and one goal of the DCFS deployment is to continue to improve the Bulk Formulae.

### 2.2.5 Instrument-Specific Metadata

The height of each sensor above or below the nominal sea surface must be recorded and kept as part of the metadata.

The recording period (length of time over which an observation is taken) is needed also as part of the metadata.

#### Quality Control Variables/User Auxiliary Data

During the computations associated with the Bulk Formulae Method data products are computed that must be saved to examine the quality of the Bulk Meteorological surface meteorological data products and the Bulk Formulae air-sea fluxes. Specifically these are:

#### TEMPSKN ( $T_{skin}$ )

the sea surface temperature; note should be extrapolated from observation depth to sea surface skin temperature using Fairall et al. (2006). During daytime heating and nighttime cooling near-surface vertical temperature gradients are present. It is the surface or skin temperature that is relevant to air-sea fluxes and this must be saved; it is determined in the Fairall et al. (2006) code using a simple ocean mixed layer model

#### STABTY ( $Z_U/L$ )

the stability parameter, the ratio of the height of the wind sensor to  $L$ , the Obukhov length scale; this should be archived as a quality control parameter to check performance of the iterative solution process in the Bulk Formulae and to document the stability of the lower atmosphere

#### CURRENT ( $U_s$ )

The momentum flux, sensible heat flux, and latent heat flux depend on the wind velocity relative to the ocean surface current. In some cases the surface current is not known and is assumed to be zero. In other cases, it is estimated from the current meter data nearest to the surface. To provide a record of how the processing was carried out, the surface current vector ( $U_s$ ) should be archived along with the observed wind velocity.

#### ZWINDSP, ZTMPAIR, ZHUMAIR, ZVELPTM

Sensor heights (depths) of METBK and VELPTMN instruments.

#### Processing flags JCOOLFL, JWARMFL, JWAVEFL

The COARE algorithm has 'switches' to turn on or turn off certain features. JCOOL turns on or off the implementation of cooling of the sea surface leading to a cooler skin temperature. JWARM turns on or off the implementation of a diurnal warm layer structure which is needed to extrapolate ocean temperature to the surface in the presence of string heating and low wind. In updates



to the COARE Bulk Formulae the effects of surface waves have been parameterized; JWAVE is a switch to turn on/off those effects.

### 2.2.6 Data Product Synonyms

Momentum flux is also known as wind stress.

### 2.2.7 Similar Data Products

The Direct Covariance Flux instrument (FDCHP) provides the ability to directly compute two components of heat flux (latent and sensible), one part of the freshwater flux (evaporation), and the momentum flux.

## 2.3 Instruments

For information on the instruments from which the L2 BULKFLUX core data product inputs are obtained, see the METBK Processing Flow document (DCN 1342-00360) and the Data Product Specification for the L1 Bulk Meteorological Data Products (DCN 1341-00360). These documents contain information on instrument classes and make/models; it also describes the flow of data from the METBK package through all of the relevant QC, calibration, and data product algorithms.

## 2.4 Literature and Reference Documents

Fairall, C. W., E. F. Bradley, J. E. Hare, A. A. Grachev, J. B. Edson, 2003: Bulk Parameterization of Air–Sea Fluxes: Updates and Verification for the COARE Algorithm. *J. Climate*, **16**, 571–591.  
**Doi:** [http://dx.doi.org/10.1175/1520-0442\(2003\)016<0571:BPOASF>2.0.CO;2](http://dx.doi.org/10.1175/1520-0442(2003)016<0571:BPOASF>2.0.CO;2)

Bradley, F. and C. Fairall, “A Guide to Making Climate Quality Meteorological and Flux Measurements at Sea”, NOAA Technical Memorandum OAR PSD-311, October 2006 (see DPS Artifacts >> BULKFLX >> NOAA-TM\_OAR\_PSD-311.pdf)

Rotronics, Inc, “The Original Rotronic Humidity Manual”, 09/01/2005 (See DPS Artifacts >> METBK >> RotronicHumidityHandbook.pdf)

Buck, A.L., 1981: *New equations for computing vapor pressure and enhancement factor*. *J. Appl. Meteor.* **20**, 1527–1532.

Payne, R.E., 1972: *Albedo of the sea surface*. *J. Atmos. Sci.*, **29**, 959-970.

Colbo & Weller, Accuracy of the IMET Sensor Package in the Subtropics, AMS 2009 (DOI: 10.1175/2009JTECHO667.1).

Bigorre and Weller, A Surface Mooring for Air–Sea Interaction Research in the Gulf Stream, Part II: Analysis of the Observations and Their Accuracies, AMS 2013 ( DOI: 10.1175/JTECH-D-12-00078.1).

## 2.5 Terminology

### 2.5.1 Definitions

None.

### 2.5.2 Acronyms, Abbreviations and Notations

General OOI acronyms, abbreviations and notations are contained in the Level 2 Reference Module in the OOI requirements database (DOORS).

### 2.5.3 Variables and Symbols

Temperatures in degrees Kelvin are denoted by  $T$  and in Celsius by  $t$ , where  $T = t + 273.16$  K.

Molecular mass of dry air ( $m_a$ ) = 0.028965 (kg mol<sup>-1</sup>)

Universal gas constant  $R = 8.3145$  (J mol<sup>-1</sup> K<sup>-1</sup>)

Gas constant for dry air ( $R_a$ ) =  $R/m_a = 287.05$  (J kg<sup>-1</sup> K<sup>-1</sup>)

Molecular mass for water vapor ( $m_v$ ) = 0.018015 (kg mol<sup>-1</sup>)

Gas constant for water vapor ( $R_v$ ) =  $R/m_v = 461.53$  (J kg<sup>-1</sup> K<sup>-1</sup>)

Surface albedo = 0.055

## 3 Theory

### 3.1 Description

The specific humidity is a function of air temperature, atmospheric temperature and relative humidity, with a small correction to account for the effects of the sensors being over seawater.

### 3.2 Mathematical Theory

The theory described below is given in more detail in Bradley and Fairall, 2006.

From the ideal gas law, we obtain the

#### Density of dry air

$$\rho_a = (P - e) / R_a T \quad (\text{kg m}^{-3}),$$

with temperature  $T$  in kelvins and in  $(P - e)$  pressure is in Pascals (100 Pa = 1 mb). Note, to compute the dry air density, we must remove the pressure contribution of water vapor from the pressure gauge reading. At standard temperature (273.15 K = 0°C) and pressure (101325.0 Pa = 1013.25 mb),  $\rho = 1.29 \text{ kg m}^{-3}$ .

The behavior of water vapor in the atmospheric surface layer approximates that of an ideal gas so that the

#### Water vapor density

$$\rho_v = 100e / R_v T \quad (\text{kg m}^{-3}),$$

where, traditionally in meteorology,  $e$  is the partial pressure of water vapor in millibars (mb). The vapor density is also referred to as the absolute humidity.

By definition,

$$e = e_s * \text{RELHUMI}/100$$

#### Saturation vapor pressure

$$(e_s) = 6.1121 (1.0007 + 3.46 \times 10^{-6} * P) * \exp(17.502 * t / (t + 240.97)) \quad (\text{mb}),$$

where  $t$  (°C) is air temperature, and  $P$  is the atmospheric pressure (mb). This empirical expression is valid above pure water between -20° and 50°C (Buck, 1981).

SPECHUMI:

**Specific humidity** ( $q_a$ ) (Bradley and Fairall, 2006) is computed as follows:

$$q_a = \rho_v / \rho_a = 621.97 * e_s / (P - 0.378 * e_s) \quad (\text{g kg}^{-1} \text{ of moist air}),$$

where the factor 0.378 =  $(1 - m_v/m_a)$ , P (mb) is atmospheric pressure and 621.97 is  $100 * R_a/R_v$ .

**NETSIRR:**

The net shortwave flux,  $Q_{SW}$ , is computed by  $Q_{SW} = SW_{in} - \alpha (SW_{in})$ , where  $\alpha$  is the surface albedo based on the Payne (1972) formulation.

**NETLIRR:**

The net longwave flux,  $Q_{LW}$ , is computed as the difference between outgoing longwave and measured incoming longwave,  $LW_{in}$ , where the outgoing longwave is computed using a blackbody formula.

$$Q_{LW} = \epsilon \sigma T_s^4,$$

where  $\epsilon = 0.97$  and is the long-wave emissivity and sigma ( $\sigma$ ) is the Stefan Boltzmann constant,  $\sigma = 5.670373 \times 10^{-8} \text{ (W m}^{-2} \text{ K}^{-4}\text{)}$ .

If the observed temperature is deeper than a few cm, it should be extrapolated to the sea surface skin temperature,  $T_s$ , which should be used in this formula. In processing, there is the option to correct the observed surface temperature for the cool skin, that is, for the cooling on the very thin surface layer, taken to be 0.3 °C. If the cooling correction is to be made, JCOOLFL is set to 1.0 and the observed SST is reduced by 0.3 °C.

**RAINFLX:**

Rain heat flux is computed by determining a rain rate from the change in the rain level with time and assumes the rain falls at the dew point temperature,  $T_d$ .

$$\text{RAINFLX} = -c_w P_r (T_s - T_d),$$

where  $P_r$  is a rain rate as a mass flux computed from time change in the gauge level,  $c_w$  is the specific heat of water, and  $T_s$  is sea surface temperature.

**LATNFLX:**

Latent heat flux is computed by a bulk formula:

$$Q_{LH} = \rho L_e C_e |U_{vel}| (q_s - q_a),$$

where  $\rho$  is the density of the air (computed from known air temperature and humidity),  $L_e$  is the latent heat of evaporation,  $C_e$  is the parameterization coefficient for latent heat flux known as the Dalton number, the bars (| |) indicate the magnitude of the wind vector,  $q_s$  is the surface saturation specific humidity or the saturation humidity at the observed water skin temperature and atmospheric pressure, and  $q_a$  is the specific humidity of the air.

**FRSHFLX**

Knowing the latent heat flux, one can compute the rate of evaporation,  $E$ , and then with the precipitation rate,  $P_r$ , one can compute the freshwater flux,  $F$ .  $E = Q_{LH}/L_v$ , where  $L_v$  is the latent heat of vaporization and then  $F = E - P_r$ .

**SENSFLX:**

Sensible heat flux is computed by a bulk formula:

$$Q_{SH} = \rho c_p C_h |U_{vel} - U_s| (T_s - T_a),$$

where  $\rho$  is the density of the air (computed from known air temperature and humidity),  $c_p$  is the latent heat of evaporation,  $C_h$  is the parameterization coefficient for sensible heat flux known as the Stanton number, the bars (| |) indicate the magnitude of the wind vector,  $T_s$  is the sea surface temperature, and  $T_a$  is the temperature of the air.

**MOMMLFX:**

The wind stress or vertical flux of horizontal momentum is computed by a bulk formula:

$$\tau = \rho C_D |U_{vel} - U_s| (U_{vel} - U_s)$$

where  $\rho$  is the density of the air (computed from known air temperature and humidity),  $C_D$  is the parameterization coefficient for wind stress known as the drag coefficient, the bars (| |) indicate the magnitude of the wind minus surface current vector,  $U_s$  is the surface current vector, and  $U_{vel}$  is the observed wind vector.

### 3.3 Known Theoretical Limitations

The validity and accuracy of the Bulk Formulae approach has been verified by comparison with fluxes collected by Direct Covariance Flux methods. The empirical coefficients and their dependencies on stability, surface roughness, and other factors are determined by using fluxes from Direct Covariance Flux observations. Because there are relatively few such observations both at low wind speeds ( $\sim 5$  m/s) and high wind speeds ( $\sim 20$  m/s) the bulk formulae flux products are considered to be most accurate at wind speeds between 5 and 20 m/s.

### 3.4 Revision History

No revisions to date.

## 4 Implementation

### 4.1 Overview

The L1 BULKMET data products are used to compute the BULKFLX data products. Example code is provided. Specific humidity and rain rate are computed and provided as input to the functions that compute the other BULKFLX products. It is assumed that the BULKMET products have been averaged to provide hourly data for the computation of rain rate. It is also recommended that hourly averaged INPUT data be used in the computation of other BULKFLX data products.

### 4.2 Inputs

Hourly averages of the BULKMET L1b (real-time) and L1c (delayed mode) data products:

SHRTIRR ( $SW_{in}$ )	Downwelling Shortwave Irradiance [ $W/m^2$ ]
LONGIRR ( $LW_{in}$ )	Downwelling Longwave Irradiance [ $W/m^2$ ]
RELHUMI (RH)	Relative Humidity [%]
TEMPAIR ( $T_a$ )	Air Temperature [ $^{\circ}C$ ]
TEMPSRF (T)	Sea Surface Temperature [ $^{\circ}C$ ]
WINDAVG ( $U_{vel}$ )	Mean wind velocity, a vector
PRECIPM (PI)	Precipitation as level of rain in gauge [mm]
BARPRESS (P)	Barometric pressure [Pa]

Hourly averages of the VELPTMN L1b and L1c data product, when available:

VELPTMN-VLE	eastward sea water velocity [m/s]
VELPTMN-VLN	northward sea water velocity [m/s]

If not available, use zero as the surface current.

The year, month, day and time of the observation(s) are required.

The height of each sensor above the sea surface (below the sea surface) must be recorded and kept as part of the metadata.

The latitude and longitude of the METBK instruments are required.

### 4.3 Processing Flow

The specific steps necessary to create all calibrated and quality controlled data products for each OOI core instrument are described in the instrument-specific Processing Flow documents (DCN 1342-00360). This processing flow documents contain flow diagrams detailing all of the specific procedures (data product and QC) necessary to compute all levels of data products from the instrument and the order in which these procedures.

Step 1: Convert the air pressure (BARPRES) from pascals to millibars by dividing Pa by 100 (1 mbar = 100 Pa).

Step 2: Compute the specific humidity. See Appendix A-1 for example code.

Step 3: Compute the rain rate. See Appendix A-2 for example code. The code is valid for the computation of rain rate from hourly precipitation levels that will be telemetered. Minute data from recovered OOI BULKMET instruments will need to be treated by data experts to remove spikes and noise. Once the instrument and site-specific behaviors are well understood, this product may be modified to reference specific automated processing.

Step 4: Compute surface current speed  

$$\text{CURRENT} = \text{SQRT}(\text{VELPTMN\_VLE}^2 + \text{VELPTMN\_VLN}^2)$$

Step 5: Compute the remaining BULKFLX products. See Appendix A-3.

### 4.4 Outputs

The following L2 Products:

SPECHUM - Specific Humidity [g/kg] (with a precision of 0.1 g/kg)  
 SPHUM2M - Specific Humidity at 2 meters [g/kg] (with a precision of 0.1 g/kg)  
 WIND10M - Wind Velocity at 10 meters [m/s] (with a precision of 0.01 m/s)  
 TEMPA2M - Air Temperature at 2 meters [°C] (with a precision of 0.01 Deg C)  
 RAINRTE - Rain Rate [mm/hour] (with a precision of 0.01 mm/hr)  
 NETSIRR - Net Shortwave Irradiance [W/m<sup>2</sup>] (with a precision of 0.01 W/m<sup>2</sup>)  
 NETLIRR - Net Longwave Irradiance [W/m<sup>2</sup>] (with a precision of 0.01 W/m<sup>2</sup>)  
 LATNFLX - Latent Heat Flux [W/m<sup>2</sup>] (with a precision of 0.001 W/m<sup>2</sup>)  
 SENSFLX - Sensible Heat Flux [W/m<sup>2</sup>] (with a precision of 0.0001 W/m<sup>2</sup>)  
 RAINFLX - Rain Heat Flux in [mm/hour] (with a precision of 0.01 mm/hour)  
 HEATFLX - Net Heat Flux [W/m<sup>2</sup>] (with a precision of 0.01 W/m<sup>2</sup>)  
 FRSHFLX - Freshwater Flux [W/m<sup>2</sup>] (with a precision of 0.01 W/m<sup>2</sup>)  
 MOMMFLX - Momentum Flux (Wind Stress) [N/m<sup>2</sup>] (with a precision of 0.001 N/m<sup>2</sup>)

### 4.5 Computational and Numerical Considerations

#### 4.5.1 Numerical Programming Considerations

There are no numerical programming considerations for this computation. No special numerical methods are used.

#### 4.5.2 Computational Requirements

N/A

### 4.6 Code Verification and Test Data Set

The example input and output data are accessible through the following path:  
<https://alfresco.oceanobservatories.org/> and navigate to OOI >> REFERENCE >>  
 Data Product Specification Artifacts >> 1341-00370\_BULKFLX

The input file contains some input parameters already computed, such as surface current, rain rate and specific humidity.

A subdirectory contains sample Matlab code and example data for the testing of the conversion of precipitation level to rain rate. The sample data in the rain rate directory are not related to the data provided to demonstrate the computation of other flux parameters.

## Appendix A Example Matlab code from WHOI UOP group

### A.1 Compute bulk air specific humidity (rh2q.m)

```
function q=rh2q(t,p,hrh)
% q=rh2qs(atmp,bpr,hrh) ... % from S. Bigorre
% compute specific humidity in g/kg from relative humidity
% uses Tetens's formula from Buck (1981, JAM 20, 1527-1532)
% input: t air temperature in degC
%       p barometric pressure in mbars
%       hrh relative humidity in %
% output: q specific humidity in g/kg
% S. Bigorre 01/10/2008

%First, calculate saturated vapor pressure es in mb, using Buck:
es=6.1121.*exp(17.502.*t./(t+240.97)).*(1.0007+3.46e-6*p);

%Second, by definition of relative humidity, vapor pressure is:
e=es.*hrh/100; %where HRH is in % (typical value 70)

%Finally, compute specific humidity using:
q=.62197*e./(p-.378*e); % q in kg/kg
q=q*1000; % q in g/kg
```

### A.2 Generate rain rate from rain level (cal\_hrly\_prate.m)

```
% cal_hrly_prate.m

% calculate rain rate in mm/hr for sample hourly precip data
% this rain rate calculation is a simplified example for illustration
% purpose only
% it assumes all hourly positive precip recordings are due to the true
% rain events

% clear all; close all;

% load Argos hourly precip (mm) data from NTAS buoy
load sample_ntas_hrly_precip.mat;

nt=length(yday);

% identify rain events and accumulate the total precipitation level
prlev=zeros(nt,1);
prlev(1)=0;

diff_precip=[0;diff(precip)];

% set negative values to zero
ff=find(diff_precip<0);
diff_precip(ff)=0;

% accumulate precip level in mm
for k=2:nt;
    prlev(k)=prlev(k-1)+diff_precip(k);
end
```

```

% find rain rate in mm/hr
prate=[0;diff(prlev)];

% check precip, prlev, and prate values

% figure; hold on;
% plot(yday,precip,'r.-');
% plot(yday,prlev,'b.-');
% plot(yday,prate,'g.-');
% xlabel('yearday in 2012');
% legend('rain recorded (mm)','rain accumulated (mm)','rain rate
(mm/hr)','location','best');
% legend('boxoff');
% title('Sample Hourly Rain Level(mm) and Rain Rate (mm/hr)');
% saveas(gcf,'sample_ntas_hrly_prate','fig');
%
% note='sample hourly rain rate in mm/hr calculated based on hourly
precip data';
% save sample_ntas_hrly_prate.mat note lat lon mday year yday precip
prlev prate;

```

### A.3 Computation of Flux Parameters (except FRSHFLX – see A.4)

Example Matlab code is provided. The code is part of the WHOI Upper Ocean Processes Lab standard suite used in bulk flux processing. Code will have to be modified to deal with streaming hourly averages. The code can be described as follows:

The script **test\_do\_flux30a.m** (requires **qsat.m**) shows example input including hard-coded instrument heights, switches (**jcool**, **jwarm**) and parameters such as **albedo**. The code calls the processing function called **do\_flux30a.m** and includes also includes a list of the output parameters from the processing function. **do\_flux30a.m** takes the input array and prepares it for the requirements of the core algorithm (requires **grv.m**, **qsee.m**) The function **do\_cor30a.m** is where the bulk parameters are computed (requires **azeta.m**, **psit\_30.m** and **psiu\_30.m**),

The Matlab m-files are located by following the following path:

<https://alfresco.oceanobservatories.org/> and navigate to OOI >> REFERENCE >> Data Product Specification Artifacts >> 1341-00370\_BULKFLX>>Code

#### A.3.1 test\_cal\_stratus9\_flux.m

```

%
% test_cal_stratus9_flux.m

% fluxes are calculated base on coare bulk air-sea flux algorithm version 3.0a
% obtained from Chris Fairall and Frank Bradley

clear all; close all;

fluxout='sample_stratus9_hrlyflux.mat';

% inpout parameters required:
% zwspd_in,zatmp_in,zhum_in,zsst_in,jcool,jwarm,albedo

% input variables in arrays(ntx1) required:
% hr,mn,sc,relwspd,sst,atmp,bpr,sh,swr,lwr,prate,grav,nlon;

% get required variables and parameters from infile='sample_stratus9_hrlymet.mat';

```



```

load sample_stratus9_hrlymet.mat
%meta experiment comment magvar mday year yday lat lon ...
           relwspd relwdir sst atmp bpr sh swr lwr prate ...
           zwspd_in zatmp_in zhum_in zsst_in currentdepth;

nt=length(yday);

% get yr mon day hr mn and sc for flux function
% year is the starting year of yday

[yr,mon,day,hr,mn,sc]=datevec(mday);

nlat=lat*ones(nt,1); % array of lat
nlon=lon*ones(nt,1); % array of lon
grav1=grv(lat); % single lat and grav
grav=grav1*ones(nt,1); % array of gravity constants

% adjusted heights for wind, atmp, and sh outputs

zwspd_out=10.0;
zatmp_out=2.0;
zhum_out=2.0;

% set jcool, jwarm, and albedo values

jcool=1;
jwarm=1;
albedo=0.055;

disp('calling do_flux30a function ...');

out=do_flux30a(hr,mn,sc,relwspd,sst,atmp,bpr,sh,swr,lwr,prate,grav,nlon,zwspd_in,zatmp_in,zhu
m_in,zsst_in,zwspd_out,zatmp_out,zhum_out,jcool,jwarm, albedo);

disp('finish do_flux30a ...');

%**** rename outputs for UOP Group ****
indx=1:nt;
yday=yday; % output old results
QB=out(:,1); % =hsb; % sensible heat
QH=out(:,2); % =h1b; % latent heat
Qs=out(:,3); % =Rns; % shortwave
Ql=out(:,4); % =Rnl; % longwave
QN=out(:,5); % =hnet_who; % net =qsen+qlat+qsw+qlw
taumag=out(:,6); % =taub; % stress
ws_h=out(:,7); % =u_zs; % adjusted wind speed
qq_h=out(:,8); % =qa_zs; % adjusted specific humidity
ta_h=out(:,9); % =t_zs; % adjusted air temp
tskin=out(:,10); % =ts-dter*jcool;
stability=out(:,11); % =zs/L %stability
ustar=out(:,12); % =usr; % ustar
drag=out(:,13); % =Cdn_10; % drag @ zu
wg=out(:,14); % =Wg; % gustiness factor
wbar=out(:,15); % =wbar; % webb mean vertical velocity
Qr=out(:,16); % =RF; % Qr rainfall heat flux

```

```

qstar=out(:,17); %=qsr; % qstar g/g instead of kg/kg??
tstar=out(:,18); %=tsr; % tstar, temp scaling parameter
CH=out(:,19); %=Chn_10;
CE=out(:,20); %=Cen_10;
Z0t=out(:,21); %=zot;
Z0q=out(:,22); %=zoq;
dt_wrm=out(:,23); %=dt_wrm;
dter=out(:,24); %=dter; % cool skin delta T
tk_pwp=out(:,25); %=tk_pwp;
tkt=out(:,26); %=tkt*1000; % thickness of cool skin
roughlen = out(:,27); % roughness length
tau_r = out(:,28); % momentum flux due to rainfall
rough = out(:,29); % roughness length

% add taudir, and stress components (taue,taun) to the outputs
taudir=relwdir;
[taue,taun]=md2vect(taumag,taudir);

disp(['save to output file: ' fluxout]);
eval(['save ' fluxout [' meta comment experiment mday year yday magvar ' ...
      ' lat lon currentdepth ' ...
      ' atmp sh bpr swr lwr ' ...
      ' sst prate relwspd relwdir ' ...
      ' zwspd_in zatmp_in zhum_in zsst_in ' ...
      ' QB QH Qs Ql QN ' ...
      ' taumag taudir taue taun ws_h qq_h ta_h tskin ' ...
      ' stability ustar drag wg wbar Qr ' ...
      ' tau_r rough roughlen ' ...
      ' qstar tstar CH CE Z0t Z0q dt_wrm dter ' ...
      ' tk_pwp tkt zwspd_out zatmp_out zhum_out jcool jwarm albedo;']);
%%%%%% end of the code *****

```

### A.3.2 function **do\_flux30a**

```

function out=do_flux30a(hr,mn,sc,wspd,ssst,atmp,bpr,sh,swr,lwr,prate,igrav,ilon,zin_wspd,
    zin_atmp,zin_hum,zin_sst,zout_wspd,zout_atmp,zout_hum,jcool,jwarm,albedo)

%***** modified version of cor3_0af.m *****
%***** reassign input data *****

qa=sh; % air specific humidity in g/kg

% wspd; % true wind speed, m/s; etl sonic anemometer
% sst; % sea snake temperature, C (0.05 m depth)
% ta=atmp; % air temperature, C (z=14.5 m)
% rs=swr; % downward solar flux, W/m^2 (ETL units)
% rl=lwr; % downward IR flux, W/m^2 (ETL units)
% org=prate; % rainrate, mm/hr
% (ETL STI optical rain gauge, uncorrected)
% lat=lat; % latitude, deg (SCS pcode)
% lon=lon; % longitude, deg (SCS pcode)

zu=zin_wspd; % anemometer ht (instrument height for wind)
zt=zin_atmp; % air T height (instrument height for air temp)
zq=zin_hum; % humidity height (instrument height for humidity)

```



```

imin=mn(ibg);
isec=sc(ibg);
%timex=[iyr imon iday ihr imin isec];
%jd=datenum(iyr,imon,iday,ihr,imin,isec)-datenum(iyr-1,12,31);
%      %year day number, Jan 1=1
%jdx(ibg)=jd;

```

```
%***** decode bulk met data ****
```

```

u=wspd(ibg); %wind speed or relative wspd
us=0; %surface current, us included in input wspd
tsea=sst(ibg);% bulk sea surface temp
t=atmp(ibg); % air temp
P=bpr(ibg); % air pressure in mb
q=qa(ibg); % air humidity
Rs=swr(ibg); % downward solar flux
RI=lwr(ibg); % doward IR flux
rain=prate(ibg);% rain rate in mm/hr
%lat =ilat(ibg);
%grav=grv(lat);% 9.78 gravity constant;
grav=igrav(1);% ==grv(lat),input directly to avoid the call;
lonx=ilon(1);% longitude for warm ladyer (daytime)

```

```
%***** variables for warm layer ***
```

```

%***** ntime=1e6*imon+1e4*iday+100*ihr+imin;
time=(ihr*3600+imin*60)/24/3600;
intime=time;
loc=(lonx+7.5)/15;

```

```

Rnl=.97*(5.67e-8*(ts-dter*jcool+273.16)^4-RI);
%oceanic broadband emissivity=0.97
Rns=(1-albedo)*Rs;%oceanic albedo=0.055 daily average

```

```
%***** set condition dependent stuff *****
```

```

Le=(2.501-.00237*tsea)*1e6;
cpv=cpa*(1+0.84*q/1000);
rhoa=P*100/(Rgas*(t+tdk)*(1+0.61*q/1000));
visa=1.326e-5*(1+6.542e-3*t+8.301e-6*t*t-4.84e-9*t*t*t);
Al=2.1e-5*(tsea+3.2)^0.79;

```

```
%***** apply warm layer *****;
```

```

if jwarm==1; %do warm layer
  chktime=loc+intime*24;
  newtime=(chktime-24*fix(chktime/24))*3600;
  if icount>1 %not first time thru
    if newtime>21600 & jump==1
      e=[num2str(icount) ' ' num2str(newtime) ...
        ' ' num2str(jtime) ' ' num2str(jump) ...
        ' ' num2str(q_pwp) ];
      disp(e)
    else
      jump=0;
      if newtime <jtime %re-zero at midnight
        jamset=0;
        fxp=.5;
        tk_pwp=19;

```

```

tau_ac=0;
qcol_ac=0;
dt_wrm=0;
jump=0;
else
%**** set warm layer constants ***
rich=.65; %crit rich
ctd1=sqrt(2*rich*cpw/(Al*grav*rhow));
ctd2=sqrt(2*Al*grav/(rich*rhow))/(cpw^1.5);
%*****

dtime=newtime-jtime;
%delta time for integrals
qr_out=Rnl+hs_old+hl_old+RF_old;
%total cooling at surface
q_pwp=fxp*Rns-qr_out;
%tot heat abs in warm layer
if q_pwp<50 & jamset==0
% do nothing??
%check for threshold
else
jamset=1;
%indicates threshold crossed
tau_ac=tau_ac+max(.002,tau_old)*dtime;
%momentum integral
if qcol_ac+q_pwp*dtime>0
%check threshold for warm layer existence
for i=1:5 %loop 5 times for fxp
fxp=1-(0.28*0.014*(1-exp(-tk_pwp/0.014)) ...
+0.27*0.357*(1-exp(-tk_pwp/0.357)) ...
+0.45*12.82*(1-exp(-tk_pwp/12.82)))/tk_pwp;
%fg=fpaul(tk_pwp);fxp=fg(1);
qjoule=(fxp*Rns-qr_out)*dtime;
if qcol_ac+qjoule>0
tk_pwp=min(19,ctd1*tau_ac/sqrt(qcol_ac+qjoule));
end;
end;% end i loop
else %warm layer wiped out
fxp=0.75;
tk_pwp=19;
qjoule=(fxp*Rns-qr_out)*dtime;
end;% end sign check on qcol_ac
qcol_ac=qcol_ac+qjoule; %heat integral

%***** compute dt_warm *****
if qcol_ac>0
dt_wrm=ctd2*(qcol_ac)^1.5/tau_ac;
else
dt_wrm=0;
end;
end;% end threshold check
end;% end midnight reset
if tk_pwp<ts_depth
dsea=dt_wrm;
else
dsea=dt_wrm*ts_depth/tk_pwp;

```

```

    end;
    end; % end 6am start first time thru
    end;% end first time thru check
    jtime=newtime;
    end;% end jwarm, end warm layer model appl check

    ts=tsea+dsea;
    qs=qsee([ts P]);
    qsx(ibg)=qs;
    tsx(ibg)=ts;
    a=.018;
    b=.729;
    twave=b*u;
    hwave=a*u.^2.*(1+.015*u);

    %***** uop add adjusted output heights to the original code
    %***** allow zsu zst and zsq to be different instead of fixed 10m
    %**** set data for basic flux alogithm
    x=[u us ts t q s q Rs RI rain zi P zu zt zq grav jcool jwave twave hwave zsu zst zsq albedo] ;
    % 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23

    %***** call modified LKB routine *****

    y=do_cor30a(x);

    %***** output from routine *****
    hsb=y(1); % sensible heat flux W/m2
    hlb=y(2); % latent heat flux
    taub=y(3); % wind stress
    zo=y(4); % velocity roughness length
    zot=y(5); % temp roughness length
    zoq=y(6); % humidity roughness length
    L=y(7); % Obukhov length
    usr=y(8); % ustar
    tsr=y(9); % tstar
    qsr=y(10); % qstar [g/g]
    dter=y(11); % cool skin delta T
    dqr=y(12); % " " " q
    tkt=y(13); % thickness of cool skin
    RF=y(14); % heat flux due to rainfall
    wbar=y(15); % webb mean vertical velocity m/s
    Cd=y(16); % ==drag @ zu transfer coeffs relative to ut
    Ch=y(17); %
    Ce=y(18); % Dalton
    Cdn_10=y(19); % 10m neutral drag coeffs=drag @ 10 [includes gustiness]
    Chn_10=y(20); % neutral transfer coeff for heat
    Cen_10=y(21); % neutral transfer coeff for moisture
    Wg=y(22); % gustiness factor m/s
    u_zs=y(23); % wind speed at zs height
    t_zs=y(24); % temp at zs height
    qa_zs=y(25); % specific hum at zs height
    rh_zs=y(26); % relative hum at zs heigh
    stability=y(27); % zu/L stability parameter height/L
    tau_r=y(28); % momentum flux due to rainfall
    rough=y(29); % roughness Reynolds number rr;

```

```

%***** save various parts of data *****
%***** create whoi type out file
out(ibg,1)=-hsb; %sensible heat (negative), opposite of chris
out(ibg,2)=-hlb; %latent heat (negative), opposite of Chris
out(ibg,3)=Rns; %shortwave
out(ibg,4)=-Rnl; %longwave (negative), opposite of Chris
out(ibg,5)=hnet; % net =qsen+qlat+qsw+qlw
out(ibg,6)=taub; % taumag
out(ibg,7)=u_zs; % adjusted wind speed
out(ibg,8)=qa_zs; % adjusted specific humidity
out(ibg,9)=t_zs; % adjusted air temp
out(ibg,10)=ts-dter*jcool; % sst=tskin
out(ibg,11)=stability; %whoI zu/L instead of L for stability
out(ibg,12)=usr; % ustar
out(ibg,13)=Cdn_10; % drag coefficient @ zu
out(ibg,14)=Wg; % gustiness factor
out(ibg,15)=wbar; % webb mean vertical velocity
out(ibg,16)=RF; % === Qr rainfall heat flux
out(ibg,17)=qsr; % qstar g/g instead of kg/kg??
out(ibg,18)=tsr; % tstar, temp scaling parameter
out(ibg,19)=Chn_10; % transfer coeff
out(ibg,20)=Cen_10; %
out(ibg,21)=zot; % roughness length due to temp
out(ibg,22)=zoq; % due to hum
out(ibg,23)=dt_wrm;
out(ibg,24)=dter; % cool skin delta T
out(ibg,25)=tk_pwp;
out(ibg,26)=tk*1e3; % thickness of cool skin
out(ibg,27)=zo; % roughlen
out(ibg,28)=tau_r; % momentum flux due to rain
out(ibg,29)=rough; % roughness Reynolds number;

%e=[num2str(imon) ' ' num2str(iday) ' ' num2str(ihr) ...
% ' ' num2str(dter) ' ' num2str(dt_wrm) ' ' num2str(tk_pwp) ];
if ( rem(ibg,1440)==0 )
    disp(ibg);
end

hs_old=hsb;
hl_old=hlb;
RF_old=RF;
tau_old=taub;
icount=icount+1;
end; % data line loop

% plots for uop out
%figure;plot(jdx,out(:,3),'-o');xlabel('Julian Day');ylabel('Sensible Heat Flux (W/m^2)');
%figure;plot(jdx,out(:,4),'-o');xlabel('Julian Day');ylabel('Latent Heat Flux (W/m^2)');
%figure;plot(jdx,out(:,8),'-o');xlabel('Julian Day');ylabel('Stress (N/m^2)');
%figure;plot(jdx,out(:,26),jdx,out(:,25));xlabel('Julian Day');ylabel('DT cool and warm (C)');

```

### A.3.3 function **do\_cor30a.m**

```
function y=do_cor30a(x)
```

```

% modify cor30a.m for more output variables
% for whoi flux Jan19, 2006

%version with shortened iteration; modified Rt and Rq
%uses wave information wave period in s and wave ht in m
%no wave, standard coare 2.6 charnock: jwave=0
%Oost et al. zo=50/2/pi L (u*/c)^4.5 if jwave=1
%taylor and yelland zo=1200 h*(L/h)^4.5 jwave=2

%% input data: x=[u us ts t qs q Rs RI rain zi P zu zt zq grav jcool jwave twave hwave zsu zst
zsq albedo] ; %%

u=x(1); %wind speed (m/s) at height zu (m)
us=x(2); %surface current speed in the wind direction (m/s)
ts=x(3); %bulk water temperature (C) if jcool=1, interface water T if jcool=0
t=x(4); %bulk air temperature (C), height zt
Qs=x(5)/1000; %bulk water spec hum (g/kg) if jcool=1, ...
Q=x(6)/1000; %bulk air spec hum (g/kg), height zq
Rs=x(7); %downward solar flux (W/m^2)
RI=x(8); %downward IR flux (W/m^2)
rain=x(9); %rain rate (mm/hr)
zi=x(10); %PBL depth (m)
P=x(11); %Atmos surface pressure (mb)
zu=x(12); %wind speed measurement height (m)
zt=x(13); %air T measurement height (m)
zq=x(14); %air q measurement height (m) (specific humidity in g/kg)
grav=x(15); % ==grv(lat)
jcool=x(16); %implement cool calculation skin switch, 0=no, 1=yes
jwave=x(17); %implement wave dependent roughness model
twave=x(18); %wave period (s)
hwave=x(19); %wave height (m)
% add the adjusted height
zsu=x(20);
zst=x(21);
zsq=x(22);
albedo=x(23);

%***** set constants *****
Beta=1.2;
von=0.4;
fdg=1.00;
tdk=273.16;
%grav=grv(lat); % modified gravity, input directly

%***** air constants *****
Rgas=287.1;
Le=(2.501-.00237*ts)*1e6;
cpa=1004.67;
cpv=cpa*(1+0.84*Q);
rhoa=P*100/(Rgas*(t+tdk)*(1+0.61*Q));
visa=1.326e-5*(1+6.542e-3*t+8.301e-6*t*t-4.84e-9*t*t*t);

%***** cool skin constants *****
Al=2.1e-5*(ts+3.2)^0.79;
be=0.026;

```



```

cpw=4000;
rhow=1022;
visw=1e-6;
tcw=0.6;
bigc=16*grav*cpw*(rhow*visw)^3/(tcw*tcw*rhoa*rhoa);
wetc=0.622*Le*Qs/(Rgas*(ts+tdk)^2);

%***** wave parameters *****
lwave=grav/2/pi*twave^2;
cwave=grav/2/pi*twave;

%***** compute aux stuff *****
Rns=Rs*(1-albedo);
Rnl=0.97*(5.67e-8*(ts-0.3*jcool+tdk)^4-RI);
%***** Begin bulk loop *****

%***** first guess *****
du=u-us;
dt=ts-t-.0098*zt;
dq=Qs-Q;
ta=t+tdk;
ug=.5;
dter=0.3;
dqr=wetc*dter;
ut=sqrt(du*du+ug*ug);
% whoi using zsu (could be 10 or 2m)instead of 10m
%u10=ut*log(10/1e-4)/log(zu/1e-4);
u10=ut*log(zsu/1e-4)/log(zu/1e-4);
usr=.035*u10;
zo10=0.011*usr*usr/grav+0.11*visa/usr;
Cd10=(von/log(zsu/zo10))^2;
Ch10=0.00115;
Ct10=Ch10/sqrt(Cd10);
zot10=zsus/exp(von/Ct10);
Cd=(von/log(zu/zo10))^2;
Ct=von/log(zt/zot10);
CC=von*Ct/Cd;
Ribcu=-zu/zi/.004/Beta^3;
Ribu=-grav*zu/ta*((dt-dter*jcool)+.61*ta*dq)/ut^2;
nits=3;
if Ribu<0;
    zetu=CC*Ribu/(1+Ribu/Ribcu);
else;
    zetu=CC*Ribu*(1+27/9*Ribu/CC);
end;
L10=zu/zetu;
if zetu>50;
    nits=1;
end;
usr=ut*von/(log(zu/zo10)-psiu_30(zu/L10));
tsr=-(dt-dter*jcool)*von*fdg/(log(zt/zot10)-psit_30(zt/L10));
qsr=-(dq-wetc*dter*jcool)*von*fdg/(log(zq/zot10)-psit_30(zq/L10));

tk=.001;

charn=0.011;

```

```

if ut>10
    charn=0.011+(ut-10)/(18-10)*(0.018-0.011);
end;
if ut>18
    charn=0.018;
end;

%disp(usr)

%***** bulk loop *****
for i=1:nits;

    zet=von*grav*zu/ta*(tsr*(1+0.61*Q)+.61*ta*qsr)/(usr*usr)/(1+0.61*Q);
    %disp(usr)
    %disp(zet);
    if jwave==0;zo=charn*usr*usr/grav+0.11*visa/usr;end;
    if jwave==1;zo=50/2/pi*lwave*(usr/cwave)^4.5+0.11*visa/usr;end;%Oost et al
    if jwave==2;zo=1200*hwave*(hwave/lwave)^4.5+0.11*visa/usr;end;%Taylor and Yelland
    rr=zo*usr/visa; % roughness Reynolds number
    L=zu/zet;
    zoq=min(1.15e-4,5.5e-5/rr^.6);
    zot=zoq;
    usr=ut*von/(log(zu/zo)-psiu_30(zu/L));
    tsr=-(dt-dter*jcool)*von*fdg/(log(zt/zot)-psit_30(zt/L));
    qsr=-(dq-wetc*dter*jcool)*von*fdg/(log(zq/zoq)-psit_30(zq/L));
    Bf=-grav/ta*usr*(tsr+.61*ta*qsr);
    if Bf>0
        ug=Beta*(Bf*zi)^.333;
    else
        ug=.2;
    end;
    ut=sqrt(du*du+ug*ug);
    Rnl=0.97*(5.67e-8*(ts-dter*jcool+tdk)^4-Rl);
    hsb=-rhoa*cpa*usr*tsr;
    hlb=-rhoa*Le*usr*qsr;
    qout=Rnl+hsb+hlb;
    delsq=Rns*(.065+11*tk-6.6e-5/tk*(1-exp(-tk/8.0e-4))); % Eq.16 Shortwave
    qcol=qout-delsq;
    alq=Al*qcol+be*hlb*cpw/Le; % Eq. 7 Buoy flux water

    if alq>0;
        xlamx=6/(1+(bigc*alq/usr^4)^.75)^.333; % Eq 13 Saunders
        tkt=xlamx*visw/(sqrt(rhoa/rhow)*usr); %Eq.11 Sub. thk
    else
        xlamx=6.0;
        tkt=min(.01,xlamx*visw/(sqrt(rhoa/rhow)*usr)); %Eq.11 Sub. thk
    end;

    dter=qcol*tkt/tcw;% Eq.12 Cool skin
    dqr=wetc*dter;

end;%bulk iter loop

tau=rhoa*usr*usr*du/ut; %stress
hsb=-rhoa*cpa*usr*tsr;
hlb=-rhoa*Le*usr*qsr;

```

```

%***** rain heat flux *****

dwat=2.11e-5*((t+tdk)/tdk)^1.94;%! water vapour diffusivity
dtmp=(1.+3.309e-3*t-1.44e-6*t*t)*0.02411/(rhoa*cpa);    %!heat diffusivity
alfac= 1/(1+(wetc*Le*dwat)/(cpa*dtmp));    %! wet bulb factor
RF= rain*alfac*cpw*((ts-t-dter*jcool)+(Qs-Q-dqer*jcool)*Le/cpa)/3600;
%***** add the momentum flux due to rainfall by kelan *****

tau_r =0.85 *rain/3600*du; %%du==wind spd relative to sea surface

%***** Webb et al. correction *****
wbar=1.61*hlb/Le/(1+1.61*Q)/rhoa+hsb/rhoa/cpa/ta;%formulation in hlb already includes webb
%wbar=1.61*hlb/Le/rhoa+(1+1.61*Q)*hsb/rhoa/cpa/ta;
hl_webb=rhoa*wbar*Q*Le;
%***** compute transfer coeffs relative to ut @meas. ht *****
Cd=tau/rhoa/ut/max(.1,du);
Ch=-usr*tsr/ut/(dt-dter*jcool);
Ce=-usr*qsr/(dq-dqer*jcool)/ut;
%***** 10-m neutral coeff realtive to ut *****
Cdn_10=von*von/log(zsu/zo)/log(zsu/zo); %von=0.4; == Frank's CD
Chn_10=von*von*fdg/log(zsu/zo)/log(zsu/zot);
Cen_10=von*von*fdg/log(zsu/zo)/log(zsu/zoq);

%***Kelan adds azeta() to adjust met variables to new heights
%***copy from cor3_0af.for
% zsu % adjusted wind speed height
% zst % adjusted air temp height
% zsq % adjusted specific humidity height
% ts - dter*jcool === sst of Bradley's;

% kelan add this function call azeta.m in matlab version
% Frank missed this call in his cor30a.for version
zetU=azeta(von,grav,ta,Q,usr,tsr,qsr,zsu);
zetT=azeta(von,grav,ta,Q,usr,tsr,qsr,zst);
% same as above if zst=zsus
zetQ=azeta(von,grav,ta,Q,usr,tsr,qsr,zsq);
% same as above if zsq=zst=zsus

u_zs=usr/von*(log(zsu/zo)-psiu_30(zetU));
t_zs=(ts-dter*jcool)+tsr/von*(log(zst/zot)-psit_30(zetT));
q_zs=(Qs-dqer)+qsr/von*(log(zsq/zoq)-psit_30(zetQ)); % !kg/kg
qa_zs=1000.*q_zs; % !g/kg

e_zs=q_zs*P/(0.62197+0.378*q_zs); % !mb

%call humidity(t_zs,p,esat_zs); % !mb
esat_zs = (1.0007+3.46e-6*P)*6.1121*exp(17.502*t_zs/(240.97+t_zs));
rh_zs=e_zs/esat_zs;
% don't change this output orders
y=[hsb hlb tau zo zot zoq L usr tsr qsr dter dqer tkt RF wbar Cd Ch Ce Cdn_10 Chn_10
.q!
Cen_10 ug u_zs t_zs qa_zs rh_zs zu/L tau_r rr ];
% 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29

```

% did not find RT RQ used in any where, they both were initialized  
% to zeros. Not going to carry these two variables.

#### A.3.4 Compute bulk water spec humidity (**qsee.m**)

```
function s=qsee(y)
x=y(:,1);
p=y(:,2);
es=6.112.*exp(17.502.*x./(x+240.97)).*98.*(1.0007+3.46e-6*p);
s=es*621.97./(p-.378*es);
```

#### A.3.5 function **grv.m** – acceleration due to gravity is calculated as a function of the latitude

```
function g=grv(lat)
gamma=9.7803267715;
c1=0.0052790414;
c2=0.0000232718;
c3=0.0000001262;
c4=0.0000000007;

phi=lat*pi/180;
x=sin(phi);
g=gamma*(1+c1*x.^2+c2*x.^4+c3*x.^6+c4*x.^8);
```

#### A.3.6 function **azeta.m**

```
function zetL=azeta(von,glat,ta,Q,usr,tsr,qsr,z)
%input glat directly
%function zetL=azeta(von,lat,ta,Q,usr,tsr,qsr,z)

% modify of Frank's function zeta

% TO EVALUATE OBUKHOVS STABILITY PARAMETER Z/L FROM AVERAGE
% TEMP T IN DEG C, AVERAGE HUMIDITY Q IN GM/GM, HEIGHT IN M,
% AND FRICTIONAL VEL,TEMP.,HUM. IN MKS UNITS
% SEE LIU ET AL. (1979)

%ta=t+toK;
tv=ta*(1.+0.61*Q);
tvsr=tsr*(1.+0.61*Q)+0.61*ta*qsr;
sgn=sign(tvsr); % !added this to avoid program
if(abs(tvsr) < 1.e-3) %then !failure when TVSR is very small
    tvsr=sgn*tvsr;
end;
%glat=grv(lat); % modify of 9.78 %% already applied
ob=tv*usr*usr/(glat*von*tvsr);
zetL=z/ob;
```

#### A.3.7 function **psit\_30.m**

```
function psi=psit(zet)
x=(1-15*zet).^5;
```

```

psik=2*log((1+x)/2);
x=(1-34.15*zet).^3333;
psic=1.5*log((1+x+x.*x)/3)-sqrt(3)*atan((1+2*x)/sqrt(3))+4*atan(1)/sqrt(3);
f=zet.*zet./(1+zet.*zet);
psi=(1-f).*psik+f.*psic;

ii=find(zet>0);
if ~isempty(ii);
    %psi=-4.7*zet;
    c=min(50,.35*zet);
    psi(ii)=-((1+2/3*zet(ii)).^1.5+.6667*(zet(ii)-14.28)./exp(c(ii))+8.525);
end;

```

### A.3.8 function **psiu\_30.m**

```

function psi=psiuo(zet)

x=(1-15*zet).^25;
psik=2*log((1+x)/2)+log((1+x.*x)/2)-2*atan(x)+2*atan(1);
x=(1-10.15*zet).^3333;
psic=1.5*log((1+x+x.*x)/3)-sqrt(3)*atan((1+2*x)/sqrt(3))+4*atan(1)/sqrt(3);
f=zet.*zet./(1+zet.*zet);
psi=(1-f).*psik+f.*psic;
ii=find(zet>0);
if ~isempty(ii);

    %psi(ii)=-4.7*zet(ii);
    %c(ii)=min(50,.35*zet(ii));
    c=min(50,.35*zet);
    psi(ii)=-((1+1.0*zet(ii)).^1.0+.667*(zet(ii)-14.28)./exp(c(ii))+8.525);
end;

```

### A.4 Compute FRSHFLX

The code below is not in the example code nor is output data available.

```

Lv = (2.5e6) ; % Lv is the latent heat of vaporization
rho =1025; %rho is seawater density (1025 kg/m3 is fine).
%QH is the latent head flux (LATNFLX)
erate = QH / Lv / rho * 1000*60*60; % mm/h
FRSHFLX = erate - RAINRTE

```

## **Appendix B          Output Accuracy**

The accuracy of the specific humidity calculated by this algorithm is a function of the accuracy of the algorithm inputs: atmospheric pressure, air temperature, and relative humidity. Using the accuracies specified in the OOI requirements database (DOORS) for the algorithm inputs:

0.2 mb for atmospheric pressure

0.05 °C for air temperature

1% for Relative Humidity

The resulting specific humidity accuracy should meet the requirement of 0.3 g/kg and a have a precision of 0.1 g/kg

For an accuracy analysis of the flux measurements, see Bigorre, et al. (2013) , Colbo & Weller (2009) and F. Bradley and C. Fairall (2006), section “B3. Estimate of Turbulent Flux Errors”.

## **Appendix C          Sensor Calibration Effects**

<N/A>