

NWTC Library – short overview of subroutines and functions

26-Jul-2012: A. Platt

24-Oct-2012: B. Jonkman

4-Dec-2012: A. Platt (v 1.05.02a)

7-Dec-2012: B. Jonkman (v1.06.00b)

12-Dec-2012: M. Buhl (v1.06.00c)

This documentation was developed for version 1.05.00 of the NWTC Library. Some changes may take place in later versions of the library.

Each file is listed separately with its MODULE and contained subroutines and functions. Unless noted otherwise, the listed routines are subroutines.

SingPrec.f90 (DoublePrec.f90)

Declares kind for single- or double-precision floating-point variables.

MODULE Precision: Stores constants to specify the KIND of variables. This module only contains constants.

NWTC Library.f90

Requires:

BufferRoutines.f90, ModMesh.f90, NWTC_Aero.f90, NWTC_IO.f90, NWTC_Library.f90, and NWTC_Num.f90.

Requires one, but not both, of the following files:

DoubPrec.f90 or SingPrec.f90.

Your project must include one, and only one, of the following files:

SysIVF.f90, SysGnuLinux.f90, SysGunWin.f90, SysIFL.f90, SysMatlab.f90, or SysIVF_Labview.f90.

Compilation order for command-line compilation:

SingPrec.f90 (DoubPrec.f90)
SysIVF.f90 (or other Sys*.f90 file)
NWTC_IO.f90
NWTC_Num.f90
NWTC_Aero.f90
ModMesh.f90
NWTC_Library.f90

Invoking programs should call NWTC_Init() to initialize data important to the use of the library.

MODULE NWTC_Library

| Name | Arguments | Description |
|-----------|--------------------------|--|
| NWTC_Init | ProgNameIn, ProgVerIn | Initialize <i>ProgName</i> and <i>ProgVer</i> if parameters have been passed. This routine then calls all required initialization routines. Write the version of the NWTC subroutine library that we are running |

SysIVF.f90 (SysGnuLinux.f90, SysGnuWin.f90, SysIFL.f90, SysMatlab.f90, SysIVF_Labview.f90)

Contains routines with system-specific logic and references. It also contains standard (but not system-specific) routines that it uses.

| Sys File | Intended Compiler/System |
|---------------------------|---|
| SysIVF.f90 | Intel Visual Fortran for Windows compiler |
| SysIFL.f90 | Intel Fortran for Linux compiler |
| SysGnuLinux.f90 | GNU Fortran for Linux compiler |
| SysGnuWin.f90 | GNU Fortran for Linux compiler |
| SysMatlab.f90 | Intel Visual Fortran for Windows compiler with Matlab's mex functions |
| SysIVF_Labview.f90 | Intel Visual Fortran for Windows compiler with references to IFPORT removed and no writing to the screen (output to a file named "Console.txt" instead) |

MODULE SysSubs:

| Name | Arguments | Description |
|-----------------------|-------------------------------------|--|
| FileSize | FileName, Size | Calls the routine FSTAT to obtain the size of the specify file or returns -1 on error. |
| FindLine | Str, MaxLen, StrEnd | Finds one line of text with a maximum length of <i>MaxLen</i> from the <i>Str</i> . It tries to break the line at a blank. |
| FlushOut | Unit | Flushes the buffer on the specified <i>Unit</i> . It is especially useful when printing "running..." type messages. |
| Get_Arg | Arg_Num, Arg, Error | Gets the Arg_Num'th argument from the command line. Note: The functionality in this routine was replaced by GET_COMMAND_ARGUMENT(), which will be available intrinsically in Fortran 2000. |
| Get_Arg_Num | Arg_Num | Gets the number of command line arguments. Note: The functionality in this routine was replaced by COMMAND_ARGUMENT_COUNT(), which will be available intrinsically in Fortran 2000. |
| Get_CWD | DirName, Status | Retrieves the path of the current working directory. |
| Get_Env (function) | EnvVar | Returns the string associated with the <i>EnvVar</i> environment variable in the OS. It returns the null string of the variable is not found. Note: The functionality in this routine was replaced by GET_ENVIRONMENT_VARIABLE(), which will be available intrinsically in Fortran 2000. |
| Is_NaN (function) | DbINum | Determines if a REAL(DbKi) variable holds a proper number. |
| OpenBinFile | Un, OutFile, RecLen, Error | Opens a binary output file. |
| OpenBinInpFile | Un, InFile, Error | Opens a binary input file. |
| OpenCon | | Opens the console for standard output. |

| Name | Arguments | Description |
|------------------|------------------------------------|--|
| OpenUnflnpBEFile | Un, InFile, RecLen, Error | Opens a binary input file with data stored in Big Endian format (created on a UNIX machine). Data are stored in <i>RecLen</i> -byte records. |
| ProgExit | StatCode | Stops the program. If the compiler supports the EXIT routine, pass the program status to it. Otherwise, do a STOP |
| UsrAlarm | | Generates an alarm to warn the user that something went wrong. |
| WrNR | Str | Writes out a string to the screen without following it with a new line. |
| WrOver | Str | Writes out a string that overwrites the previous line. |
| WrScr | Str | Writes out a string to the screen. Break long messages into multiple lines. |

NWTC_Num.f90

Contains numeric-type routines with non-system-specific logic and references.

It also contains global numeric-related variables.

MODULE NWTC_Num:

| Name | Arguments | Description |
|--|---|---|
| InterpBin <i>(function interface)</i> | XVal, XAry, YAry, ILO, AryLen | Returns a y-value that corresponds to an input x-value by interpolating into the arrays. It returns the first or last <i>YAry()</i> value if <i>XVal</i> is outside the limits of <i>XAry()</i> . Note: This is an interface for InterpBinComp and InterpBinReal and will call the appropriate one (depending if <i>YAry</i> is complex or real). |
| InterpStp <i>(function interface)</i> | XVal, XAry, YAry, Ind, AryLen | Returns a y-value that corresponds to an input x-value by interpolating into the arrays. It uses the passed index as the starting point and does a stepwise interpolation from there. This is especially useful when the calling routines save the value from the last time this routine was called for a given case where <i>XVal</i> does not change much from call to call. When there is no correlation from one interpolation to another, InterpBin() may be a better choice. It returns the first or last <i>YAry()</i> value if <i>XVal</i> is outside the limits of <i>XAry()</i> . Note: This is an interface for InterpStpComp and InterpStpReal and will call the appropriate one (depending if <i>YAry</i> is complex or real). |
| AddOrSub2Pi | OldAngle, NewAngle | This routine is used to convert <i>NewAngle</i> to an angle within 2π of <i>OldAngle</i> by adding or subtracting 2π accordingly; it then sets <i>OldAngle</i> equal to <i>NewAngle</i> . This routine is useful for converting angles returned from a call to the ATAN2() FUNCTION into angles that may exceed the $-\pi$ to π limit of ATAN2(). This routine assumes that the angle change between calls is not more than 2π in absolute value. <i>OldAngle</i> should be SAVED in the calling routine. |
| BSortReal | RealAry, NumPts | This routine sorts a list of real numbers. It uses the bubble sort algorithm, which is only suitable for short lists. |
| Cross_Product <i>(function)</i> | Vector1, Vector2 | This function computes the cross product of two 3-element arrays: $Cross_Product = Vector1 \times Vector2$ (resulting in a vector). |
| EqualRealNos <i>(function)</i> | ReNum1, ReNum2 | This function compares 2 real numbers and determines if they are "almost" equal, <i>i.e.</i> within some relative tolerance. |
| GetSmlRotAngs <i>(function)</i> | DCMat, ErrStat | This subroutine computes the angles that make up the input direction cosine matrix, <i>DCMat</i> . |
| GL_Pts | IPt, NPts, Loc, Wt, ErrStat | Returns the non-dimensional (-1:+1) location of the given Gauss-Legendre Quadrature point and its weight. The values came from Carnahan, Brice; Luther, H.A.; Wilkes, James O. (1969) "Applied Numerical Methods." |
| IndexCharAry <i>(function)</i> | CVal, CAry | Returns an integer index such that $CAry(IndexCharAry) = CVal$. If no element in the array matches <i>CVal</i> , the value -1 is returned. The routine performs a binary search on the input array to determine if <i>CVal</i> is an element of the array; thus, <i>CAry</i> must be sorted and stored in increasing alphabetical (ASCII) order. The routine does not check that the array is sorted. The routine assumes that <i>CVal</i> is type CHARACTER and <i>CAry</i> is an array of CHARACTERS. |

| Name | Arguments | Description |
|------------------------|---|--|
| LocateBin | XVal, XAry, Ind, AryLen | Finds the lower-bound index of an input x-value located in an array. On return, <i>Ind</i> has a value such that $XAry(Ind) \leq XVal < XAry(Ind+1)$, with the exceptions that $Ind = 0$ when $XVal < XAry(1)$, and $Ind = AryLen$ when $XAry(AryLen) \leq XVal$. Note: If the index doesn't change much between calls, <i>LocateStp()</i> may be a better option. |
| LocateStp | XVal, XAry, Ind, AryLen | Finds the lower-bound index of an input x-value located in an array. On return, <i>Ind</i> has a value such that $XAry(Ind) \leq XVal < XAry(Ind+1)$, with the exceptions that $Ind = 0$ when $XVal < XAry(1)$, and $Ind = AryLen$ when $XAry(AryLen) \leq XVal$. It uses the passed index as the starting point and does a stepwise search from there. This is especially useful when the calling routines save the value from the last time this routine was called for a given case where <i>XVal</i> does not change much from call to call. When there is no correlation from one interpolation to another, a binary search may be a better choice. |
| Mean (function) | Ary, AryLen | Function to calculate the mean value of a vector array. |
| MPI2Pi | Angle | Ensures that <i>Angle</i> lies between $-Pi$ and Pi . |
| SetConstants | | Computes some useful constants based upon Pi and IEEE arithmetic. |
| RombergInt | f, a, b, R, err, eps, ErrStat | Used to integrate a function <i>f</i> over the interval $[a, b]$ (<i>f</i> is an external function). This routine is useful for sufficiently smooth (e.g., analytic) integrands, integrated over intervals which contain no singularities, and where the endpoints are also nonsingular. |
| SmlRotTrans | RotationType, Theta1, Theta2, Theta3, TransMat, ErrTxt | This routine computes the 3x3 transformation matrix, <i>TransMat</i> , to a coordinate system <i>x</i> (with orthogonal axes x_1, x_2, x_3) resulting from three rotations (<i>Theta1</i> , <i>Theta2</i> , <i>Theta3</i>) about the orthogonal axes (X_1, X_2, X_3) of coordinate system <i>X</i> . All angles are assumed to be small, as such, the order of rotations does not matter and Euler angles do not need to be used. This routine is used to compute the transformation matrix (<i>TransMat</i>) between undeflected (<i>X</i>) and deflected (<i>x</i>) coordinate systems. <i>See the subroutine in the file NWTC_Num.f90 for more details.</i> |
| SortUnion | Ary1, N1, Ary2, N2, Ary, N | Takes two sorted arrays and finds the sorted union of the two. Note: If the same value is found in both arrays, only one is kept. However, if either array has multiple occurrences of the same value, the largest multiple will be kept. Duplicates should be eliminated externally if this is not desirable. |
| StdDevFn (function) | Ary, AryLen, Mean | Calculates the standard deviation of a population contained in <i>Ary</i> . |

NWTC_IO.f90

Contains I/O-related variables and routines with non-system-specific logic.

MODULE NWTC_IO:

| Name | Arguments | Description |
|----------------------------------|---|--|
| AdjRealStr <i>(interface)</i> | NumStr | Removes leading spaces and trailing zeros from strings created by real numbers. |
| AllocAry <i>(interface)</i> | Ary, AryDim1, [AryDim2], [AryDim3], Descr, ErrStat | Allocates logical, character, integer, and real arrays. Values are passed for <i>AryDim2</i> , and <i>AryDim3</i> when 2 or 3 dimensional arrays are requested. Note: This interface will call the appropriate allocation subroutine depending on the type and dimensionality of the array requested. This interfaces to: <ul style="list-style-type: none">– character array allocation subroutines (AllCAry1, AllCAry2, AllCAry3)– logical array creation subroutines (AllLAry1, AllLAry2, AllLAry3)– integer array allocation subroutines (AllIAry1, AllIAry2, AllIAry3)– real array allocation subroutines (AllRAry1, AllRAry2, AllRAry3) |
| CheckArgs | InputFile, ErrStat | Checks for command-line arguments. |
| CheckIOS | IOS, Fil, Variable, VarType, TrapErrors | Checks the I/O status and prints either an end-of-file or an invalid-input message, and then aborts the program. |
| CloseEcho | | Closes the echo file and sets <i>Echo</i> to false. |
| Conv2UC | Str | Converts all the text in <i>Str</i> to upper case. |
| CountWords <i>(function)</i> | Line | Function that counts the number of "words" in a line of text. It uses spaces, tabs, commas, semicolons, single quotes, and double quotes ("whitespace") as word separators. |
| CurDate <i>(function)</i> | | Function that a character string encoded with the date in the form dd-mmm-ccyy. |
| CurTime <i>(function)</i> | | Function that returns a character string encoded with the time in the form "hh:mm:ss". |
| DispNVD <i>(interface)</i> | -- ProgDesc, Name/Ver | Displays the name of the program, its version, and its release date. Note: This interface will call the appropriate allocation subroutine depending on the type and number of arguments passed. This interfaces to: <ul style="list-style-type: none">– DispNVD0 – no inputs. The global variables ProgName and ProgVer are used– DispNVD1 – Single input of type ProgDesc.– DispNVD2 – Two arguments of character type containing the name and version info |
| Flt2LStr <i>(function)</i> | FltNum | Converts a REAL to a left-justified string. |
| GetNewUnit | UnIn | Returns a unit number not currently in use. |
| GetNVD <i>(function)</i> | ProgDesc | Returns a string with the program name, version, and date (converts data structure to single string) |
| GetPath | GivenFil, PathName | Parses the path name from the name of the given file. It counts everything before (and including) the last "\" or "/". |
| GetRoot | GivenFil, RootName | Parses the root file name from the name of the given file. It counts everything after the last period as the extension. |
| GetTokens | Line, NumTok, Tokens, Error | Parses <i>Line</i> for <i>NumTok</i> "tokens" and return them in the <i>Tokens</i> array. This routine differs from GetWords() in that it uses only spaces as token separators. |
| GetWords | Line, Words, NumWords | Retrieves <i>NumWords</i> "words" from a <i>Line</i> of text. |

| Name | Arguments | Description |
|---|--|---|
| NameOFile | InArg, OutExten, OutFile, ErrStat | Get the name of the input file from the <i>InArg</i> th command-line argument. Remove the extension if there is one, and append <i>OutExten</i> to the end. |
| NormStop | | Performs a normal termination of the program. |
| Num2LStr <i>(function interface)</i> | Num | Converts a floating point number to a left-aligned string. It eliminates trailing zeroes and the decimal point on floating point numbers. Note: This is an interface to several the functions Int2LStr, R2LStr4, R2LStr8, and R2LStr16. It will call the appropriate one depending on the type of <i>Num</i> . |
| OpenBin | Un, OutFile, RecLen, ErrStat | Opens a binary output file. |
| OpenBInpFile | Un, InFile, ErrStat | Opens a binary input file. |
| OpenEcho | Un, OutFile, ErrStat | Opens a formatted output file for the echo file. |
| OpenFInpFile | Un, InFile, ErrStat | Opens a formatted input file. |
| OpenFOutFile | Un, OutFile, ErrStat | Opens a formatted output file. |
| OpenFUnkFile | Un, OutFile, FailAbt, Failed, Exists, ErrStat | Opens a formatted output file and returns a flag (<i>Exists</i>)telling if it already existed. |
| OpenUInBEFile | Un, InFile, RecLen, ErrStat | Opens an unformatted input file of <i>RecLen</i> -byte data records stored in Big Endian format. |
| OpenUInfile | Un, InFile, ErrStat | Opens an unformatted input file. |
| OpenUOutfile | Un, OutFile, ErrStat | Opens an unformatted output file. |
| PathIsRelative <i>(function)</i> | GivenFil | Determine if the given file name is absolute or relative. A path is considered an absolute path one that satisfies one of the following criteria: 1) It contains "/" or "\" 2) It starts with "/" or "\" All others are considered relative. |

| Name | Arguments | Description |
|------------------------------------|--|--|
| PremEOF | Fil, Variable, TrapErrors | Write out an EOF message and aborts the program. |
| ProgAbort | Message, TrapErrors | Outputs fatal error messages and stops the program. |
| ProgPause | | Pauses the program and requires the user enter an <Enter> to resume execution. |
| ProgWarn | Message | Outputs non-fatal warning <i>Message</i> and returns to the calling routine. |
| ReadAry <i>(interface)</i> | UnIn, Fil, Ary, AryLen, AryName, AryDescr, ErrStat | Reads in <i>AryLen</i> values into the array <i>Ary</i> from the next <i>AryLen</i> lines of the input file. Note: This is an interface to the subroutines ReadCAry, ReadIAry, ReadLAry, and ReadRAry. It will call the appropriate one depending on the type of <i>Ary</i> . ReadRAry can read values separated by white space from the same line of the input file as well. |
| ReadAryLines <i>(interface)</i> | UnIn, Fil, Ary, AryLen, AryName, AryDescr, ErrStat | Reads in <i>AryLen</i> values into the array <i>Ary</i> from the next <i>AryLen</i> lines of the input file. Note: This is an interface to the subroutines ReadCAryLines, ReadDAryLines, and ReadRAry. It will call the appropriate one depending on the type of <i>Ary</i> . |
| ReadCom | UnIn, Fil, ComName, ErrStat | Reads a comment from the next line of the input file. |
| ReadFASTBin | UnIn, FASTdata, ErrLev, ErrMsg | Reads the contents of a FAST binary output file and stores it in FASTdata. The name of the data file is input through the FASTdata structure by the calling procedure. |
| ReadNum | UnIn, Fil, Word, VarName, ErrStat | Reads a single word from a file and tests to see if it's a pure number (no true or false). |
| ReadOutputList | UnIn, Fil, CharAry, AryLenRead, AryName, AryDescr, ErrStat | Reads a <i>AryLen</i> values into a real array from the next <i>AryLen</i> lines of the input file. |
| ReadStr | UnIn, Fil, CharVar, VarName, VarDescr, ErrStat | Reads a string from the next line of the input file. |

| Name | Arguments | Description |
|-------------------------------|--|--|
| ReadVar <i>(interface)</i> | UnIn, Fil, Var, VarName, VarDescr, ErrStat | Reads in variable <i>Var</i> from the next line of the input file. <i>Var</i> can be of type CHARACTER, DOUBLE, INTEGER, LOGICAL, or REAL. Note: This is an interface to the subroutines <i>ReadCVar</i> , <i>ReadDVar</i> , <i>ReadIVar</i> , <i>ReadLVar</i> , and <i>ReadRVar</i> . It will call the appropriate one depending on the type of <i>Var</i> . |
| WaitTime | WaitSecs | Waits for <i>WaitSecs</i> before proceeding. |
| WrFileNR | Unit, Str | Writes out the string, <i>Str</i> , to the file connected to <i>Unit</i> without following it with a new line. |
| WrML | Str | Writes out the string, <i>Str</i> , in the middle of a line. |
| WrPr | Str | Writes out a prompt with text <i>Str</i> to the screen without following it with a new line, though a new line precedes it. |
| WrScr1 | Str | Writes out the string, <i>Str</i> , to the screen after a blank line. |

NWTC Aero.f90

This module contains aerodynamics routines with non-system-specific logic and references. It also contains global aerodynamics-related variables.

MODULE NWTC_Aero:

| Name | Arguments | Description |
|------------------------------|--|--|
| AeroInt | ISeg, Alpha, Re, AF_Table, IntData, DoCl, DoCd, DoCm, DoCpmin, ErrStat | Finds the Re-bounding tables and then calls GetCoef() to get the desired coefficients for the two tables and then interpolates between them. |
| CompDR | NumSeg, RLoc, HubRad, RotorRad, DimenInp, DelRLoc, ErrStat | Computes the segment lengths from the local radii and the rotor radius. It prints and error if the list of radii is not realizable. |
| GetAF | AF_File, AF_Table, ISeg | Get airfoil data from either a new NWTC-style or an old AeroDyn-style airfoil file. |
| GetCoef <i>(function)</i> | ISeg, Alpha, AlfaTab, CoefTab, NumRows, Ind, ErrStat | Interpolation routine for airfoil section coefficients. |
| GetCoefs | ISeg, Alpha, Re, AF_Table, ClInt, CdInt, CmInt, CpminInt, DoCl, DoCd, DoCm, DoCpmin, ErrStat | Finds the Re-bounding tables and then calls GetCoef() to get the desired coefficients for the two tables and then interpolates between them. |