**Transplanting Wheat:**

**NWheat V 0.3 (24/06/96) from APSIM into DSSAT 4.5**

([Jump to recent](#Begin_Here); [Jump to NWheat-Subroutines.xlsx](file:///C:\Users\froyce\AppData\Roaming\Microsoft\Word\NWheat-Subroutines.xlsx))

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Questions arising**:

* To what extent do we need to change NWheat variable names to DSSAT names?
  + For inputs from shared files (weather, soil, management?) probably 100%
  + For outputs, probably 90% or more.
  + For internal program variables, case-by-case, depending whether they are shared with other modules
* What is the content of the dll\_export, include and “Sub-Program Arguments” declared in many of the subroutines? See [Appendix](#dll_export).
* Where does the control variable “Action” take its value? See [Action table](#Action_table), and [Appendix](#Action_code).
* What do [these mystery functions](#Mystery_functions) or sub-routines do?
* A principal change from Ceres to Apsim appears to have been an extreme form of “modularization”, wherein the large Ceres routines were divided into numerous (scores) of small sub-routines. My “nwheats.for subroutines” worksheet has about 60 sub-routines. These will mostly correspond to my new WH\_PHENOL, WH\_GROSUB, WH\_ROOTS, WH\_OPGROW, and WH\_OPHARV, each of which contains a single main sub-routine. So is my job to re-combine numerous small modules into a few large ones? Or should I maintain the multiple-subroutine structure within several fortran files, each of which is smaller than nwheats.for?

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**May 6, 2011**: Met with Senthold and Cheryl. We agreed that the first stage goal would be to achieve the same potential production as the APSIM version. Then we can work on water and N stresses. In the end, Senthold does not expect or require identical output. Rather, he said the “output clouds” should show the same trends. Although the origin of NWheat is CERES Wheat, they have both diverged so much from the point of origin, that it would probably be better to use CERES sorghum, millet or even maize as a template. Senthold gave me the [documentation](file:///C:\Users\froyce\Documents\Project_Areas\Wheat%20Model\Documentation%20from%20Senthold) and [source code](file:///C:\Users\froyce\Documents\Project_Areas\Wheat%20Model\Source%20Code%20from%20Senthold) files.

**May 12**: Began going through NWheat documentation and source code. Compared CERES alternatives: sorghum, millet, maize.

What is the CSCER-CSCRP module? CSCRP045.FOR and CSCER040.FOR each have long lists of variables with brief definitions. Is this Tony Hunt’s style?

Will I begin with Plant.FOR? [Compared variables](file:///C:\Users\froyce\AppData\Roaming\Microsoft\Word\CERES%20crop%20variables.txt) from each CERES crop in Plant.FOR.

**May 13**: Going through the NWheat code:

NWheat.FOR is a very large fortran file, compared to our models. Among the DSSAT fortran source files, only CSCRP045.FOR and CSCER040.FOR are larger (both wheat, oddly enough).

Possibly try to compile the four files Senthold gave me? Maybe not, if I am successful in the following plan: Compared input cultivar coefficients in NWheat to each relevant DSSAT model:

NWheat compared to MZ\_CERES, ML\_CERES, SG\_CERES, CSCERES\_Interface and CSCRP\_Interface

**May 16**: My plan is to:

1. Select the most similar DSSAT model:
   1. Keating et al (1997) explain that NWheat is most similar to CERES Wheat, V2
   2. According to Table 1, among other current CERES-based models, wheat and barley have the most similar cultivar coefficients.
   3. Looking at the barley code, it seems barley is a variation of the wheat model (as it should be).
   4. Cheryl did not favor using wheat as a template, since it is the least standardized. So probably barley falls into the same category.
   5. This leaves maize as the next-most-similar, in terms of cultivar coefficient correspondence. No stalk weight cultivar parameter, however.
   6. I should move on to looking at NWheat, and wait for Cheryl to return to decide on best template crop.
2. Attempt “transplant” by adapting NWheat components one-by-one into the relevant DSSAT module.
   1. Create a table showing NWheat components and the corresponding DSSAT (maize, or maybe several?) modules.
   2. Create a table for defining NWheat variables, and possibly relating to DSSAT variables.
3. Ideally, the hybrid model could be tested for functioning after each substitution, but unless NWheat variables are replaced with DSSAT variable names this could be difficult or even impossible.

Table 1: CERES cultivar coefficients that correspond to NWheat

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NWheat** | **Ceres Maize** | **Ceres Wheat** | **Ceres Barley** | **Ceres Millet** | **Ceres Grain Sorghum** |
| Sensitivity to vernalization | NA | Opt vernalize temp P1V | Opt vernalize temp P1V | NA | NA |
| Sensitivity to photoperiod | Photoperiod devel delay P2 | Photoperiod response P1D | Photoperiod response P1D | Critical photoperiod P2O, P2R | Critical photoperiod P2O, P2R, PSAT, PBASE |
| TT ear growth to maturity | TT silking to maturity P5 | Grain fill duration P5 | Grain fill duration P5 | TT grain fill to maturity P5 | TT grain fill to maturity P5 |
| Kernel per ear | Max kernels per plant G2 | Kernel # per canopy wt G1 | Kernel # per canopy wt G1 | NA | NA |
| Kernal max growth rate | Kernel filling rate G3 | Std kernel wt G2 | Std kernel wt G2 | NA | NA |
| Tiller max final weight | NA | mature tiller wt G3 | mature tiller wt G3 | NA | NA |

I should review the [notes](file:///C:\Users\froyce\AppData\Roaming\Microsoft\Sugarcane_model\CASUPRO_Development\Structure\CASUPRO-restructuring.docx) to my CASUPRO conversion to modular DSSAT.

Table

|  |  |  |  |
| --- | --- | --- | --- |
| **NWheat Section** | **Variables** | **Ceres Maize Module** | **Variables** |
| Version number (once) |  |  |  |
| Initialization (once) |  |  |  |
| Transpiration (daily) |  |  |  |
| Phenology (daily) |  |  |  |
| Biomass Accum (daily) |  |  |  |
| Leaf Area Dev (daily) |  |  |  |
| Senescence (daily) |  |  |  |
| Crop Nitrogen (daily) |  |  |  |
| End Crop (once) |  |  |  |
|  |  |  |  |

**June 6**: Met with Cheryl, and created a [basic strategy](file:///C:\Users\froyce\AppData\Roaming\Microsoft\Word\2011-06-06_mtg_with_Cheryl.docx) for migrating NWheat.

**June 7**:

“istage” stages (throughout entire nwheats.for module; use NE, EQ, LT, GT, LE, GE): (not sure where these ranges came from: different from those shown in [nwheat\_science.doc](../Files%20from%20Senthold/Documentation/nwheat_science.docx))

1. mature (0-0) eq lt le gt
2. emerg (1-2) eq lt gt ge
3. endjuv (2-3) eq le
4. endveg (3-4) eq ge
5. endear (4-5) eq lt
6. grnfil (5-6) eq le gt
7. fallow (8-8) eq lt ne
8. sowing (9-9) eq lt
9. germ (9-10)eq ne

Table 3: “Action” states (only in apsim\_nwheats subroutine) and the “if (Action.eq. . . .” statements that are used to call subroutines. Appear to be analogous to the “DYNAMIC” (sometimes “Control % DYNAMIC”) variable in WHAPS. Can I do a translation?

|  |  |  |
| --- | --- | --- |
| **“****Action” value - IF statement** | **Subroutine(s) called** | **“Dynamic” equivalent (tentative)** |
| if (Action.eq. MES\_Presence) | write(\*, \*) 'Module\_name = ', Module\_name | RUNINIT |
| if (Action.eq.MES\_Init) | call nwheats\_zero\_variables  call nwheats\_init | RUNINIT |
| if (Action .eq. MES\_Prepare) | call nwheats\_prepare | SEASINIT |
| if ((Action.eq.'sow').or.(Action.eq.'plant')) | call nwheats\_get\_other\_variables  call nwheats\_sow\_crop | RATE |
| if (Action.eq.MES\_Process) | call nwheats\_get\_other\_variables  call nwheats\_process  call nwheats\_crpmn ('harvest')  call nwheats\_set\_other\_variables | INTEGRATE |
| if (Action.eq.'harvest') | call nwheats\_harvest\_crop | OUTPUT |
| if (Action .eq. MES\_Post) | call nwheats\_post | OUTPUT |
| if (Action.eq.MES\_Get\_variable) | call nwheats\_send\_my\_variable |  |
| if (Action .eq. MES\_Set\_variable) | call nwheats\_set\_my\_variable |  |
| if (Action .eq. MES\_Event) | call nwheats\_capture\_event | OUTPUT |
| if (Action .eq. MES\_End\_run) | call nwheats\_end\_run | SEASEND |

Although I do not have the subroutine or code that controls “Action”, it looks like this is little more than a logical sequence, so maybe the criterion by which “Action” is set to each value will not be too difficult to deduce. If I substitute the appropriate DSSAT/WHAPS DYNAMIC variable for each ACTION variable, I will at least have a coherent sequence. Adjustments can be made later.

Chris Villalobos came over to “clone the repository” so that I could have a local copy of the changes that Cheryl made to CSM to begin to accommodate NWheat. I could not follow his rapid-fire, trial-&-error procedure, so next time I’ll just ask for help again.

**June 8**: Setting up a new MS Visual Fortran project:

1. New Project – Empty Project – Console Application
   1. Name: Nwheat
   2. Path: C:\NwheatCSM
   3. Solution: Nwheat
2. View - Solution Explorer
3. In Solution Explorer window, menu-click “Nwheat”, select “Add” and “Existing Item”. Navigate to “Source” subdirectory, and select all files.
4. Again menu-click “Nwheat”, then select “Properties”
   1. Configuration Properties – Fortran – Language – Compile lines with D in Column 1 - Yes (/d\_lines)
5. Again menu-click “Nwheat”, then select “Properties”
   1. Configuration Properties – Debugging – Working Directory: type: “C:\DSSAT\Wheat”
   2. Configuration Properties – Debugging – Command Arguments: type: “B DSSBatch.v45”
   3. When creating executables, in “Properties” we need to go to Build Events - Post-Build Event type the path to save the executable file (probably C:\DSSAT45\BIN), but that can wait a bit.
6. Check differences in genetic input files. The current versions are supplied by Git. Indeed there were some differences, and with the exception of the CASUPRO CUL and ECO files (see [CASUPRO log - June 8 entry](file:///C:\Users\froyce\AppData\Roaming\Microsoft\Sugarcane_model\CASUPRO_Development\Structure\CASUPRO-restructuring.docx#June_8_2011)) all were copied into the DSSSAT45\Genotype subdirectory. Also need to copy files located in SOURCE\DATA, and its supdirectories.

**The new model is WHAPS045**. Cheryl had originally called in NWAPS045, but using the standard wheat WH seemed to make more sense. WHAPS is selected in Plant.For (using CASE), and a main simulation file is called. To create that main file, I simply renamed MZ\_CERES.FOR to WH\_APSIM.FOR. Removed CASE selections of different maize versions from the new WH\_APSIM.FOR, but made few if any other changes at first. Since NWheat will be located within the DSSAT45\Wheat subdirectory, I also renamed the CUL, ECO and SPE files, as well as a maize FileX (FLSC8101.MZA 🡪FLSC8101.WHX), changing MZ to WH and maize to wheat. Compiled fairly easily, but getting it to run without errors too a few hours. The problem was needing to get the model “registered” in the proper places (DSSATPRO.V45, Simulation.cde, PATH.FOR, IPVAR.For and maybe afew other places, since Cheryl did some of this. In general, look where other models are listed or selected, although there are a number of files where it is not necessary to include the model to run, but rather to use sensitivity analysis on the coefficients.

Note that in DSSATPRO.v45 I had to change the line MWH C: \DSSAT45 DSCSM045.EXE **CSCER045** to MWH C: \DSSAT45 DSCSM045.EXE **WHAPS045**. I suppose this is what the DSSAT program does when the model is changed in File-Configuration (i.e., SCCAN to SCCSP).

Tested:

* Whether the model reads the File X and input files from the Wheat subdirectory as intended **YES.**
* Whether the outputs are identical to the MZCER from which it was borrowed. NO. Close, both the same, but why? This is only important because it could indicate some sort of hidden problem that may not necessarily go away “by itself.

Now, two alternatives:

1. Gut WH\_APSIM.FOR and begin rebuilding contents following nwheats\_crppr
2. Leave MZ\_CERES contents, and modify piece-by-piece following nwheats\_crppr

Alternative 1:

* A “cleaner” method

Alternative 2:

* Easier to obtain an immediately runnable version for iterative testing
* Has input (weather, soils, and genetic coefficients) and output code all in place.
* Seems to be a more usual method to use when creating a new crop

If alternative 2 does not show promise fairly quickly, I can always revert to alternative 1.

Eventually, all weather and soil inputs should be based on Senthold’s NWheat data.

**June 9**:

Plan the transition of nwheat into MZCER (now WH\_APSIM).

1. Which is the guide for calls: the entire nwheats.for file or the nwheats\_crppr subroutine?
   * Probably a combination
2. Finish excel sheet with all subroutines and functions
   * Add “calls to” and “called by”
3. Create a subset of calls from **nwheats\_crppr** and a similar list from WH\_APSIM.
4. Define the calls or transitions that correspond to DSSAT’s runinit, seasinit, emerge and integr.
5. Using the above, develop an ordered list of calls

**June 14**:

From where are the main Fortran files called? Apparently not from each other. Referring to the APSIM\_SoilN subroutine within the soiln.for module, a comment reads “This routine is the interface between the **main system** and the SoilN module”. We don’t seem to have the “main system”. I will probably have to follow the pseudo-code and papers.

[Ordered list of calls](file:///C:\Users\froyce\AppData\Roaming\Microsoft\Word\Ordered%20list%20of%20calls.docx) from Apsim\_nwheats (in nwheats.for) was completed. This, combined with the [nwheat\_science](file:///C:\Users\froyce\AppData\Roaming\Microsoft\Word\Documentation%20from%20Senthold\nwheat_science.docx) and [nwheat\_pseudo](file:///C:\Users\froyce\AppData\Roaming\Microsoft\Word\Documentation%20from%20Senthold\nwheat_pseudo.html) documents will be the guide for code transfer.

**June 16**:

How much of current content of WH\_APSIM.FOR could I comment out, possibly using some easily distinguishable code (!\*\*! or something . . .).

If I could get the model to “run”, outputting only zero-filled files, I could begin to plug NWheat components into the shell.

Or, I could begin to re-model the files called from WH\_APSIM.FOR one-by-one.

**June 20** Next steps:

From Cheryl’s list of “first round” sub-routines, I added a “Yes” column to the Excel list.

**June 21-24**:

In the WH\_APSIM.For module, I replaced MZSTGNAM with NWSTGNAM (NW = New Wheat)

I created a new CHARACTER DATA array, NWSTGNAM, and defined the members, using “[Comparison of Stage Names.txt](../Background/Comparison%20of%20Stage%20Names.txt)”.

**Focus on the control variables:** What I know about Nwheat control variables:

There are at least 7 different variables named with some variation of “stage”: istage, stagno, lstage, mxstag, xstage, zstage and fstage. Only istage and xstage are included in the outputs list in nwheat\_science.doc.

* + **Action**: (I think the “APSIM\_\*\*.FOR” control files are basically [ordered call lists](#Action_states), which can probably be replicated by the DSSAT DYNAMIC variable)
    - Although action management is mentioned in Purpose of subroutine **nwheats\_crpmn**, the variable “Action” only appears in subroutine ***apsim\_nwheats***
    - In the residue.for file, “Action” is only found in subroutine *APSIM\_residue*
    - In the soiln.for file, “Action” is only found in subroutine *APSIM\_SoilN*
    - In the soilwats.for file, “Action” is only found in subroutine *APSIM\_soilwats*
  + **nwheats\_status** Takes on a numerical (integer?) value, apparently set in subroutine *nwheats\_send\_my\_variable*.
    - nwheats\_status = 0 (prior to planting; pre-plant?)
    - nwheats\_status = 1 (set to 1 in subroutine *nwheats\_sow\_crop*; so planted; crop in the ground?)
    - nwheats\_status = 10 (set to 10 at end of subroutine *nwheats\_harvest\_crop,* also in subroutine *nwheats\_crpmn;* crop “absent”; post-harvest?)
    - Only these values were encountered in nwheats.for.
    - nwheats\_status does not appear in the other 3 \*.for files Senthold gave me.
    - What does the nwheats\_status variable provide that istage does not?
  + **istage** most common flag variable; Main phenological stage variable? [Table 2](file:///C:\Users\froyce\AppData\Roaming\Microsoft\Word\Documentation%20from%20Senthold\nwheat_science.docx#Phenology) in nwheat\_science.doc?
    - istage integer stage number
    - nwheats\_set\_xstag subroutine
    - Stgdur(istage) (?)
    - cumph(istage) (?)
    - sumcbo(istage)
    - rtrate(istage)
    - csd1(istage), csd2(istage)
    - cnsd1(istage), cnsd2(istage)
    - rootfr (istage)
    - sumdtt(istage)
    - pgdd(istage)
  + **stagno** (OUTPUT) phenological stage number. Used as the argument for several arrays, as in *nwheats\_phase (****stagno****)*. May be where phenological stage (istage) number is determined
    - stagno integer ! (OUTPUT) phenological stage number
    - set in subroutine nwheats\_phase
    - nwheats\_germn (stagno)
    - nwheats\_cfail (stagno)
    - cumph(stagno)
    - stgdur (stagno) stgdur(stagno) = stgdur(stagno) + 1
    - sumdtt(stagno)
    - pgdd(stagno)
  + **lstage** In the pseudo code and actual code, appears only in “NWheat Phase.
    - lstage integer ! **stage number (last one) before it is changed**
    - subroutine nwheats\_phase (all occurances in this subroutine)
    - lstage = stagno
    - sumdtt(lstage)
    - pgdd(lstage)
    - cumph(lstage)
    - (least appearances of any of these control variables)
  + **Xstage** In pseudo code appears in SLA, Set Xstag, Set Zstag, Set Nconc, Event, Set Nfact, Ndmd, Height, and is mentioned in Nwheat Phase, but does not appear.
    - xstage real stage number
    - set in nwheats\_set\_xstag subroutine
    - Purpose: Set a growth stage index for use in plant nitrogen
  + **Zstage** A linearly interpolated (real?) value based upon xstage. In the pseudo code, only in Set Zstag and Set Nconc. In the Fortran code, nwheats\_set\_nconc, nwheats\_set\_zstag, and nwheats\_send\_my\_variable.
    - zadok's growth stage (?)
    - zstage = linear\_interp\_real (xstage,xs,zs,6)
  + **fstage** In the pseudo code, appears only in “NWheat Set Xstag”, which may be for N stress only. In the actual code, in nwheats\_set\_xstag and also in function nwheats\_dc\_code. F=”Fraction” of stage?
    - fstage real ! stage function (0-1)
  + **mxstag** In the pseudo code, appears only one time, in Nwheat Phase. In Fortran code appears almost exclusively as a function or array argument. Used to set phase in nwheat\_phase. Where set, unknown.
    - rtrate (mxstag) real ! root growth rate potential (mm /deg day)
    - rootfr (mxstag) real ! fraction of carbohydrate to roots (0-1)
    - character sname(mxstag)\*14 ! stage name (to write stage name?)
    - xstg**mn** (mxstag) real ! value at beginning of a stage
    - xstg**mx** (mxstag) real ! maximum value at end of a stage

“There are 9 crop stages in the NWheat module (Table 2), and commencement of each stage (except for sowing to germination which is driven by soil moisture) is determined by accumulation of thermal time (as modified by photoperiod and vernalisation effects)” [nwheat\_science.doc](file:///C:\Users\froyce\AppData\Roaming\Microsoft\Word\Documentation%20from%20Senthold\nwheat_science.docx#Phenology)

**Related variables**:

**stgdur(stagno)** sometimes stgdur(istage). This is an integer variable stand-in for istage, which is increased by +1 to essentially mark the end of each phase. It begins after istage reaches 1 (emergence) since from germination to emergence does not proceed by +1 (from 9 to 1).

One way to understand the “Action” control variable may be to compare its value in the nwheats.for code to the description in the nwheat\_pseudo.html pseudo code

**To summarize the control and subscript variables, in their possible order of importance**:

1. **istage**: Focus here: apparently the principal phenological stage variable (int) (same var name as WHAPS) Locus of calculation not found.
2. **stagno**: May be the variable from which istage is calculated
3. **lstage**: “L” may represent “last”, or previous, stage.
4. **mxstag**: May equal istage, but used as a stand-in argument in some functions or arrays.Locus of calculation not found.
5. **Xstage**: same value as istage, except real number as opposed to interger. Related to N stress.
6. **Zstage**: linearly interpolated value based on xstage; also related to N stress?
7. **fstage**: apparently related to Xstage, and therefore N stress.
8. **Action**: Seems little more than a ordering list; try to substitute DYNAMIC

As I begin replacing CERES code with APSIM code in each module, I will:

* Be confronted with the nuts-and-bolts of what I will need to do
* Need to devise ways of keeping up with substitution of variables
* Realize what information is missing, and how difficult it will be to infer what it should be

|  |  |  |  |
| --- | --- | --- | --- |
| **istage name** | **Number** | **WHAPS transition into** | **NWheat transition** |
| fallow | 7 |  |  |
| sowing | 8 | istage = sowing (when WH\_Phenol first runs; Do this just once in RUNINIT) |  |
| germ | 9 | IF (istage .EQ. sowing) then  istage = germ (on pdat) |  |
| emergence | 1 | Have not figured out yet. Seems should be P9, but does not seem to be. |  |
| endjuv | 2 |  |  |
| endveg | 3 |  |  |
| endear | 4 |  |  |
| grnfil | 5 |  |  |
| mature | 6 |  |  |
|  |  |  |  |

**What are the characteristics of each module with respect to substitution in this way?**

1. Control module (MZ\_CERES 🡪 WH\_APSIM)
   1. Continue within WH\_APSIM as necessary
   2. Begin in PLANT, with call to WH\_ APSIM (DONE)
   3. Introduce WH\_GROSUB and WH\_PHENOL (DONE)
   4. Introduce WH\_OPGROW, WH \_OPNIT, WH \_OPHARV, WH \_ROOTGR, WH \_NFACTO, WH \_NUPTAK and WH \_KUPTAK (DONE)
2. Initialization of crop parameters: (MZ\_GROSUB🡪 WH\_GROSUB; MZ\_PHENOL🡪WH\_PHENOL)
   1. Begin by simply substituting the Nwheat experiment weather, plant coefficients, and soil into respective DSSAT format. The genetic coefficients required some shifting/renaming
      1. Created new FileX, Genetic coefficient, and weather files to get running in the DSSAT shell. See [Apsim Wheat Input Files](file:///C:\Users\froyce\AppData\Roaming\Microsoft\Word\Apsim%20Wheat%20input%20files.docx) document.
      2. Before changing input files, check outputs now for evidence that inputs can be confirmed: ET.OUT, INFO.OUT, OVERVIEW.OUT, SoilTemp.OUT, SoilWat.OUT, Weather.OUT. Saved in [C:\DSSAT45\Wheat\Last runs](file:///C:\Users\DSSAT45\Wheat\Last%20runs).
3. Principal Nwheat subroutines:
   * 1. nwheats\_zero\_variables
     2. nwheats\_init subroutine
     3. others?
4. Transpiration (later)
   1. This may be handled by the standard (cross-crop) SPAM modules
   2. Is this used in no water stress simulations?
   3. Could be worth examine, since eventually will certainly be needed
5. Phenology (MZ\_PHENOL 🡪 WH\_PHENOL.for):
   1. I have started.
   2. Some initialization here
6. Biomass Accumulation (MZ\_GROSUB🡪WH\_GROSUB)
   1. Daily Photosynthesis Rate
   2. Plant component weights
7. Leaf Area Development (MZ\_GROSUB🡪 WH\_GROSUB?)
8. Senescence (MZ\_GROSUB🡪WH\_GROSUB)
9. Crop Nitrogen (later)

Next steps, from outline above:

* Do and confirm 1. (b), (c), and (d). (DONE, although part (a) will probably be ongoing.)
* Then take on 2. a.
  + APSIM files to use will be CROP.PAR, mexico.par, maybe RESIDUE.INI, Soil.par, SOILWATS.INI, AgMIP\_Mexico.con
  + New FileX: UFMX8901WHX
  + New weather file: (DONE)
  + New CUL and ECO files (used CROP.PAR only; see [Questions doc](../Background/Questions%20on%20Coefficients.docx) regarding some strange parameters)

**July 1-7**:

* + - * Change CUL file gradually, to keep debugger running:

IPVAR.FOR: WHAPS section and format 850 entered. May be finished.

COMGEN.BLK has variables added (initialized). May be finished.

So now the cultivar coefficients are written into DSSAT045.INP, where they await use by other modules.

Modified WH\_PHENOL and WH\_GROSUB to read all variables from DSSAT45.INP. Note that there are several !\*! markers for material to be removed.

* + - The tec and rue\_factor variables from CROP.PAR will not be retained as coefficients. They will be calculated within the model, according to the formulae on pg 299 of [Reyanga\_et-al\_1999.pdf](file:///C:\Users\froyce\AppData\Roaming\Microsoft\Word\Documentation%20from%20Senthold\Reyanga_et-al_1999.pdf).
  + SPE and ECO files will be left alone for the time being.

**July 8, 2011**: Next steps:

1. Create a [detailed summary](file:///C:\Users\froyce\AppData\Roaming\Microsoft\Word\Apsim%20Wheat%20input%20files.docx) (another document) of each new input file created: FileX, weather, soil, cultivar, ecotype and species. (OK)

**July 18-28, Aug 9-17, 22- , 2011**: go thru each module and describe; then distribute APSIM subroutines

1. Created a list of current WHAPS variables in [NWheat-Subroutines.xlsx](file:///C:\Users\froyce\AppData\Roaming\Microsoft\Word\NWheat-Subroutines.xlsx).
   1. Look at ModuleDefs to see whether there are more shared variables. (OK)
   2. Include weather Model Inputs (OK)
   3. Eventually will need soil Model Inputs
   4. Definitions and Units? This will certainly be necessary to substitute variables.
   5. Did a similar list of NWheat variables. Copied variables from [nwheat\_science.docx](file:///C:\Users\froyce\AppData\Roaming\Microsoft\Word\Documentation%20from%20Senthold\nwheat_science.docx#OUTPUTS), but need to see whether they are used under same names. In [NWheat-Subroutines.xlsx](file:///C:\Users\froyce\Documents\Project_Areas\Wheat%20Model\NWheat-Subroutines.xlsx), page Nwheat variables (2), removed duplicate Variable Names, after assuring each subroutine in which the variable appears is listed.
   6. Now that I have variables from both NWheat and WHAPS, what are possible next steps?
      1. Match old CROP.PAR input parameter names with new WHAPS045.CUL names and definitions. Recall some have been multiplied to create an easier-to-handle number. Pretty much done: See worksheet “Cultivar Characteristics” in the NWheat-Subroutines Excel file.
      2. Using [Main WHAPS subroutines](file:///C:\Users\froyce\AppData\Roaming\Microsoft\Word\WHAPS%20sub-routine%20descriptions.docx) and the [NWheat subroutines spreadsheet](file:///C:\Users\froyce\AppData\Roaming\Microsoft\Word\NWheat-Subroutines.xlsx), complete the preliminary plan for how to distribute APSIM subroutines among the WHAPS subroutines. Once this is done, I need to decide best place to start:
         1. cleaning out as much as possible, or
         2. substituting in and cleaning out as little as necessary.
2. **Aug 22:** Determine which WHAPS variables and code will probably have to be left largely as they are:
   1. Most broadly, variables ultimately passed to, or received from common modules, including CSM, InputModule, Management, SPAM, Utilities (Dates and Outputs are here), the multitude of Soil routines and Weather
3. Determine a logical order of substitution for the first several NWheat routines:
   1. Based on call order in NWheat, the first few routines:
      1. Zero variables. Performed locally in many DSSAT modules
      2. Initialize wheat module. Probably included in relevant DSSAT modules: WH\_PHENOL, WH\_GROSUB and maybe others.
      3. Get values of variables from other modules. In DSSAT, this is done in header declaration.
      4. Potential CH2O production based on photosynthesis
      5. PAR interception
      6. Actual CH2O production
   2. Based on call order in WHAPS:
      1. CSM –
         1. Year, day-of-year. Probably OK
         2. Sets value of CONTROL % DYNAMIC
            1. The Control variable relates to model processes (init, rate, integrate, output, etc), not phenological plant processes.
            2. If I substitute the appropriate DSSAT/WHAPS DYNAMIC variable for each ACTION variable into the NWheat code, I will at least have a coherent sequence and something that will run. Adjustments can be made later. See Table 3 above.
      2. Land – calls WH\_APSIM. OK as-is
      3. WH\_APSIM
         1. assigns numbers to phenology stage names, so these may need modifying
         2. calls most of the WH\_ files (Probably OK)
      4. WH\_PHENOL – [Phenlolgy Log](file:///C:\\Users\\froyce\\AppData\\Roaming\\Microsoft\\Word\\Phenology%20Log.docx). should be first to accept major pieces of NWheat.
         1. After much deliberation, I left the ISTAGE settings the same, only changing the written definitions to match NWheat.
         2. Begin placing NWheat phenology code. Use NWheat local variables (add them to declarations as necessary), and replace variables as necessary from Cultivar file.
      5. Create a new module for cold weather effects, particularly since Osvaldo thinks this is so important (from NWheat):
         1. ! calculate effects of cold weather
         2. call nwheats\_vernalization ()
         3. call nwheats\_cold\_hardening()
         4. call nwheats\_frost\_leaves ()
         5. call nwheats\_frost\_tillers ()
      6. WH\_GROSUB – next location for much of NWheat
4. Change in strategy: I gradually realized that NWheat does not have a “no stress” mode. Rather, stress is assumed away by supplying abundant N and H2O.
   1. After confirming this with Senthold, we decided that sooner rather than later, I would have to turn the DSSAT water and N stress back on, and make sure the DSSAT soil module is functioning.
   2. Since the NWheat code has been created without any “no stress” options, I will probably not save time by continuing to work in DSSAT “no stress” mode. So, the next step is to compare outputs between Water on and off. Wait for N until “water stress on” is at least functioning.
   3. Made sure planting dates and irrigation amounts and times were the same as the NWheat experiments. Looks like I had already adjusted Treatment 3 to these specifications.
   4. Runs OK. The few non-zero crop numbers are unchanged.
   5. Continued to add subroutines to WH\_GROSUB
5. WH\_OP\*.\* – there are 3 output files that may need to be modified to translate NWheat variable names into DSSAT output variable names.
6. without cleaning out others. But even if variable names are synchronized, this would require maintaining two systems of control variables.
   * + 1. This requires a running account of code, subroutines and variables introduced, changed, and removed.
     1. Gradually clean out (comment out?) existing WHAPS routines, while keeping all variable declarations intact to keep the model “running”, or at least prevent crashing.
     2. Try to substitute control variables:
        1. Where are control variables currently controlled in WHAPS?
           1. Dynamic: CMS.FOR
           2. ISTAGE (phenology): WH\_PHENOL.FOR
        2. Look up and compare those from NWheat to WHAPS (see [Table 3](#Table_3) above).
     3. As DSSAT code is removed, insert NWheat code, modifying the variable names as necessary?
        1. What NWheat variable names must be replaced with WHAPS variable names for the first, potential yield version?
           1. Those from standard input files, beginning with cultivar, weather and later soil
           2. Control variables?
           3. How about FileX experiment parameters: dates, planting geometry, etc?
           4. Output variable names.
           5. Variables exchanged with parts of the DSSAT program that are common to all models: in general, files within SOIL, SPAM, UTILITIES and WEATHER modules. Which of these are necessary for potential yield? Solar radiation and temperature related?
           6. Note that there are a substantial number of “Constant Values” throughout nwheats.for, which should eventually be moved to an input file.
7. Tentatively decide where to locate each Nwheat subroutine, among the APSIM NWheat files. See call mapping on worksheet “WHAPS Call order” in [NWheat-Subroutines.xlsx](../Background/NWheat-Subroutines.xlsx).
   1. WH\_ APSIM.for: Control module; much of what it does is call the following modules.
   2. WH\_GROSUB.for: growth subroutine; includes leaf senescence;
   3. WH\_PHENOL.for
   4. WH\_ROOTS.FOR
   5. WH\_OPGROW.FOR
   6. WH\_OPHARV.FOR
   7. Later: WH\_KUPTAK.FOR, WH\_NFACTO.for, WH\_NUPTAK.FOR and WH\_OPNIT.FOR
8. Go through the main WH\_\*\*\*.FOR files:
   1. Clear out old commented-out code
   2. Attempt to gradually replace existing CERES Maize functions with the corresponding Nwheat functions. It may be easiest to keep DSSAT variable names where possible, although maybe not.
   3. For the new CO2 code, should I create a new CO2 module? How do other DSSAT models handle this?
9. **January 13, 2012**:
   1. It has been 8 months since I started this model re-location project!
   2. In October 2011, I assured Senthold that I would have outputs by Christmas. That did not happen.
   3. Some months ago, I compared the number of lines in the original Nwheats.for to those in Nwheats.tmp, from which those transferred have been removed. Results were not encouraging enough to bother with.
10. **February 8, 2012:**
    1. It seems clear that the most efficient path to effective model comparison and use will be to incorporate everything from nwheat (except input and output code) into DSSAT.
       1. This mainly affects previous plans for code related to N, water, and the associated soil model.
       2. Assuming I can leave both the DSSAT and WHAPS soil models functioning, this will help compare and possibly integrate algorithms
       3. Unfortunately, this makes the initial job even larger.
       4. Major questions:
          1. How much code is in question and how often is it called?
          2. Where and how to incorporate the new soil-water code:
             1. Include as subroutines or spaghetti code?
             2. If subroutines, in existing modules, or create new files like WH\_Soil.FOR, WH\_WATBAL.FOR and/or others that parallel DSSAT?
             3. If spaghetti code, in what module? WH\_GroSub.for?
             4. Might Cheryl help with these questions?
          3. Answer the “how often” question
    2. Meeting w/ Senthold today
11. **Feb. 24**: Finished the soil-water module WH\_SW\_SUBS.FOR. I had proposed these 239 lines to be done by today, and they are. Of course, 7 working days of which 5 were fully or largely dedicated to this is sobering. The next block is **subroutine nwheats\_crppr.** The big question is whether this should be moved into its own subroutine, or mainly be included into WH\_GROSUB. Since the nwheats\_crppr “purpose” states: “biomass production, phenological stages, plant component development, water uptake and nitrogen uptake, and plant senescense”, I will have to break **nwheats\_crppr** up into various DSSAT modules, follow the function of each block or subroutine of code. That which is growth-related in to GROSUB. N-related, into a new N module, and so forth. Probably some would be added to the soil-water module I just “finished”. See [**road-map**](../Meetings/2012-02-24_Senthold.docx) I laid out for Senthold. He thinks we should be through by June
12. **Feb 27, 2012**: Early in the APSIM-to-DSSAT process, **nwheats\_crppr** provided the roadmap for what subroutine to call, and when. So, long ago I began transferring **nwheats\_crppr** into WHAPS. The questions now for **each subroutine** are
    1. Into which WHAPS subroutine to place the code?
       1. Phenology, growth, soil, roots or which N-related subroutine: uptake or deficit?
    2. Whether to introduce as a subroutine or integrate into the module as “spaghetti code?
       1. Is the code called more than once?
13. Classify each subroutine that is not already transferred. Take into account the context within CRPPR subroutine from which it is called
    1. nwheats\_set\_adf roots nwheats\_crppr (in WH\_ROOTS.FOR)
       1. linear\_interp\_real Figuring this out has help me understand some of the parameters currently in the cultivar file should be in the species file, to better adapt to tabex. The parameters to move are: fdsw (tabx), adf (taby), stage and stage (tabx) and afs (taby). These are a total of 3+3+2+2 or 10 parameters: FDSW(1,2,3), ADF(1,2,3), AFS(1,2) and STAG(1,2).
       2. nwheats\_ad\_rtloss (code transferred)
          1. nwheats\_root\_distrib (**left off here with nwheats\_crppr**) Note: dimmed subroutines will not be transferred now.
    2. nwheats\_rtdp roots nwheats\_crppr
    3. nwheats\_rtlv roots nwheats\_crppr
    4. add\_real\_array see “[equivalents](../Background/Syntax%20and%20Coefficient%20equivalents.txt)” doc
    5. subtract\_real\_array see “[equivalents](../Background/Syntax%20and%20Coefficient%20equivalents.txt)” doc
    6. nwheats\_germn phenology; nwheats\_phase
    7. nwheats\_set\_xstag (maybe) nitrogen?; nwheats\_phase
    8. nwheats\_set\_zstag (maybe) nitrogen or phenology; nwheats\_phase
    9. nwheats\_set\_nconc nitrogen; nwheats\_crppr
    10. nwheats\_event output?; nwheats\_crppr
    11. nwheats\_plnin nitrogen; nwheats\_crppr
    12. nwheats\_set\_nfact nitrogen; nwheats\_crppr
    13. nwheats\_partition growth; nwheats\_crppr
    14. nwheats\_translocate (maybe) growth; nwheats\_crppr
    15. nwheats\_tillering phenology or growth; nwheats\_crppr
    16. nwheats\_grnit nitrogen; nwheats\_crppr
    17. nwheats\_nuptk nitrogen; nwheats\_crppr
    18. subtract\_real\_array see “[equivalents](../Background/Syntax%20and%20Coefficient%20equivalents.txt)” doc
    19. subtract\_real\_array see “[equivalents](../Background/Syntax%20and%20Coefficient%20equivalents.txt)” doc
    20. nwheats\_respiration (removed)
    21. subtract\_real\_array see “[equivalents](../Background/Syntax%20and%20Coefficient%20equivalents.txt)” doc
    22. nwheats\_crtot output; nwheats\_crppr

1. **Mar. 7**: New direction after meeting with Cheryl and Senthold. Cheryl suggested that we can essentially create a potential production switch for Nwheat, by setting all the stressors to one. In this way, we can compare output to abundant N and H2O output from the original Nwheat. This short-cut provides an alternative to my importation of all modules into a DSSAT-like framework, which seemed to me like the only way to be able to isolate the inevitable errors and problems in the transferred code. Additionally, all root, water, soil and Nitrogen code in a DSSAT format would more easily enable the development of alternative soil/water/N modules based on what the original Nwheat used. But Cheryl pointed out that we have a deliverable: Nwheat functioning in DSSAT, which assumes all DSSAT common inputs, soil etc. Although my method would represent a more careful way to achieve an essentially identical Nwheat, it is also a longer route. Cheryl’s suggestion represents a shortcut, assuming we are able to reasonably duplicate the original Nwheat model (that the Nwheat outputs will not be identical is assumed). Steps toward this new goal:
   1. Finish incorporating parameters into SPE file. **DONE**
   2. Find stressors, and set each to its “no stress” value. **DONE**
   3. Review nwheats\_crppr to see what code needs to be transferred, while skipping the remaining soil, water and N subroutines. May need to go thru code in nwheats.tmp and DSSAT version, to see what calls and code have been transferred.
      1. Transfer above
   4. Begin to track down processes within the transferred code, to see what needs linking through variables using the correct variable name.
   5. Update Cheryl.
2. Stressor locations:
   1. RTDP2 for rtdp\_stress\_switch ln 5130
   2. subroutine **nwheats\_set\_nfact** (not transferred; called by nwheats\_crppr)
      1. nfact(1) - affects grain N potential
      2. nfact(2) - affects photosynthesis
      3. nfact(3) - affects leaf senescence, grain N concentration & cell expansion
      4. nfact(4) - affects grain number **OK – set to 1**
   3. subroutine **nwheats\_set\_swdef\_new**
      1. swdef(3) is set to 1.0 in WH\_GROSUB **OK**
      2. this is **set\_swdef\_new** in WHAPS
      3. it is fully transferred, but not currently called anywhere: entirely skipped, so values of swdef should not change from 1.0.
         1. in original, called by nwheats\_prepare; nwheats\_crppr
      4. maybe skip entire routine if DSSAT water stress switch = ‘N’
      5. maybe add OR statement from DSSAT water stress switch at line 7093
   4. subroutine **nwheats\_biomp**
      1. may be mostly temperature stress, which is left in **Probably OK**
   5. subroutine **nwheats\_cfail Probably OK**
      1. may only apply, not calculate, stress factors
      2. code transferred to WH\_PHENOL
   6. function carbo and pcarb: We use carbo instead of pcarb to capture the decrease in growth with water content because carbo is pcarb after water stress (line 1211 – include an IF statement about water stress ON or OFF?) **Both Probably OK**
      1. **pcarbo** (potential CH2O) is from subroutine **nwheats\_ptcarb** (transferred)
      2. **subroutine nwheats\_biomp** (transferred)
         1. optfr (uses stress factors calculated elsewhere)
   7. **subroutine nwheats\_gndmd** (not transferred; not needed yet)  **Probably OK**
      1. delta\_grainc is the daily increment in grain weight (after stress)
   8. **real function nwheats**\_lfsen (not transferred; may be needed)
      1. sfactor ! stress factor for leaf senescence(0-1) **Check out senescense**
   9. **subroutine nwheats\_rtdp** (not transferred – root depth; not needed now)
      1. p\_rtdp\_stress\_switch (used line 3195)
      2. p\_rtdp\_swf\_switch (used line 3204) **Both Probably OK**
   10. **function nwheats\_swafc** (not transferred – soil-water - not needed now) **Probably OK**
   11. **subroutine nwheats\_watup\_new** (transferred; not called yet) **Probably OK**
   12. **function nwheats\_rtdp\_swf** (not transferred – soil-water - not needed now) **Probably OK**
       1. frpesw - soil-water below which stress occurs (0-1) (set to 0.25) **Probably OK**
   13. **subroutine nwheats\_plnin** (initialize plant nitrogen – not needed now) **Probably OK**
3. Stressor candidates:
   1. Nitrogen
      1. nfact(1-4) group **OK**
   2. Water
      1. Swdef(1-3) **OK**
   3. Other?
      1. ???
4. Ordered list of **relevant** (**non**-soil-water-nitrogen) subroutines and functions that should be installed before seriously linking and testing the model (as descrbed in 13.c. above). Notice that not only is the transfer important, but also the call timing and order. These comments refer to the list below.
   1. **Nwheats\_prepare (note:** these are **not** called during WHAPS RUNINIT, but it is not obvious why they need to be – except that nwheat calls them from MES-prepare**)**
      1. nwheats\_ptcarb (transferred to GROSUB, which is called from WH\_APSIM)
      2. nwheats\_biomp (transferred to GROSUB, which is called from WH\_APSIM)
   2. **nwheats\_process** (all called during INTEGRATE)
      1. nwheats\_crppr
         1. ~~nwheats\_set\_adf (mostly transferred, but roots, so probably stop)~~
         2. ~~nwheats\_rtdp (root depth)~~
         3. ~~nwheats\_rtlv (root length-volume)~~
         4. ~~add\_real\_array (rlvnew, rlv, nrlayr)~~ (Roots, so ignore)
         5. ~~nwheats\_set\_swdef\_new (transferred, but soil-water)~~
         6. ~~nwheats\_ptcarb (transferred)~~
         7. ~~nwheats\_biomp (transferred)~~
         8. ~~nwheats\_watup\_new (transferred, but soil-water)~~
         9. ~~subtract\_real\_array (rwu, swdep, nrlayr)~~ (Roots, so ignore)
         10. ~~nwheats\_watup\_new (transferred, but soil-water)~~
         11. ~~nwheats\_set\_swdef\_new (transferred, but soil-water)~~
         12. ~~nwheats\_vernalization (transferred)~~
         13. ~~nwheats\_cold\_hardening (transferred)~~
         14. ~~nwheats\_frost\_leaves (transferred)~~
         15. ~~nwheats\_frost\_tillers (transferred)~~
         16. nwheats\_phase (pieces transferred to PHENOL, but following need to be transferred and called, probably from GROSUB)
             1. ~~nwheats\_germn~~ (PHENOL?) (comparing to existing DSSAT code in WH\_PHENOL: both are soil-water based, so since we will ultimately go with DSSAT soil, leave this out)
             2. ~~nwheats\_cfail~~ (mostly commented out in nwheats.for; those lines and lines transferred to NW\_Phenol removed from nwheats.tmp.)
             3. ~~nwheats\_pgdin~~ (code changed somewhat, transferred to WH\_PHENOL)
             4. ~~nwheats\_set\_xstag~~ (although associated with N, transfer as xstage\_nw to StageFlags subroutine)
             5. ~~nwheats\_set\_zstag~~ (toi StageFlags in WH\_Phenol; is called during RUNINIT, so it can read Species parameters istageno, dc\_code, xs and zs, and export the finished products daily ) In WH\_PHENOL (line 1640) deciding between TABEX and ALIN functions. Check [FUNCTIONS.xls](../../Sugarcane_model/CASUPRO_Development/CASUPRO%20Versions/Daza_Files_ModeloProduccion/Documentos%20de%20Formulacion%20e%20Implementacion/DocsSoporteCASUPRO/FUNCTIONS.xls) Checked both, seemed to be the same, so I used ALIN since it’s variables have more intuitive names.
         17. nwheats\_set\_nconc (NO: critical N concentration)
         18. nwheats\_event (output; assume we will use DSSAT output files
         19. nwheats\_gpp (code transferred; no need to call)
         20. nwheats\_plwin (transferred to GROSUB)
         21. nwheats\_plnin (skip: initialize plant N)
         22. ~~nwheats\_lfemr~~ (Transferred to GROSUB: leaf emergence)
         23. nwheats\_set\_nfact (Skip: N availability)
         24. nwheats\_ptcarb (transferred)
         25. nwheats\_biomp (transferred)
         26. nwheats\_cdemand (transferred: biomass production)
         27. nwheats\_partition (transfer to GROSUB)
         28. nwheats\_translocate (GROSUB translocate carbohydrate)
         29. ~~nwheats\_tillering~~ (GROSUB)
         30. nwheats\_grnit (Skip: grain N uptake)
         31. nwheats\_nuptk (Skip: plant N uptake)
         32. subtract\_real\_array (snup\_no3, sno3, nrlayr) (Skip: N uptake related)
         33. subtract\_real\_array (snup\_nh4, snh4, nrlayr) (Skip: N uptake related)
         34. add\_real\_array (pnup, pl\_nit, mxpart) (Skip: N uptake related)
         35. ~~add\_real\_array~~ (growt, pl\_wt, mxpart) (try this one into GROSUB)
         36. ~~add\_real\_array~~ (transwt, pl\_wt, mxpart) (try this one into GROSUB)
         37. nwheats\_gpp\_stemwt (transferred; check for call)
         38. ~~nwheats\_respiration~~ (“remove the respiration terms as rue is on a net carbon basis”)
         39. **~~subtract\_real\_array~~**(Cresp, pl\_wt, mxpart) If Respiration is out, so is this subtraction of respiration
         40. sen\_la = **nwheats\_lfsen** () sfactor: stress factor for leaf senescence. What is “ti” and where is it calculated?
             1. real function **nwheats\_slan**

TI = Fraction of a phyllochron interval which occurred as a fraction of today's daily thermal time

PHINT = Phyllochron interval. Number of GDD required for new leaf emergence, C.

frlf = dtt / PhInt (WHAPS-Grosub)

TI = DTT/(PHINT\*PC) (DSSAT-Grosub)

call nwheats\_lfemr (ti) (from nwheats\_crppr)

stage\_gpla needs to be passed from Phenol to Grosub.

* + - 1. **nwheats\_crtot** totals of crop variables (output; not necessary for initial testing)
    1. nwheats\_add\_sen\_roots (roots: probably skip**)**
  1. **Check nwheats.tmp for remaining code, and delete code already transferred but missed.**
     1. **nwheats\_crpmn** harvest crop1 and incorporate stover and roots (output; not necessary for initial testing)
     2. See following code **from nwheats\_crppr**:. The goto 900 leads to nwheats\_crtot, which is model output, and the end of the nwheats\_crppr subroutine, and (I think), the end of the model run.

if (istage.eq.mature .or. istage.eq.fallow) then

goto 900 !(this may end run)

else

endif

* + 1. Remove sections of nwheats\_crppr that have been transferred from nwheats.tmp. **DONE**

Begin connecting the modules and inspecting outputs. Note: Are modifications of cultivar parameters from subroutine nwheats\_read\_param incorporated? See [Outputs document](Analysis%20of%20Outputs.docx).

**June 28, 2012**: Most of the work during the past 4 months has been in [thermal time](Thermal%20Time%20Correction.docx) and no-stress model [outputs](Analysis%20of%20Outputs.docx). However, these comments are of a general nature. After reading the description of the thermal time calculation on page 7 of the [NWheat Science](../Documentation/nwheat_science.docx) paper (one of the original references that Senthold gave me when we began), I broached the idea to Senthold that the code I am working with may not be the code that produced the outputs I am trying to match; it may either be different or incomplete. Of course, the implications should this be the case, are serious. Senthold later noticed the same passage himself, and decided that I should introduce the 8-period interpolation of average temperature. Actually, this is the worst, but not the only source of significant uncertainty in this process:

* References to functions that are not described in the text. Most are probably simple, and intuitively self-explanitory (l\_bound , bound, divide), but others are a little trickier (linear\_interp\_real, subtract\_real\_array) . Recently, I found a website which seems to contain the code for these functions.
* Not all cultivar and species parameters explicit. Even parameters as fundamental as optimal and maximum temperature for calculating thermal time were not in the input files, and had to be pulled from literature on the model.
* Daily output data from the original APSIM NWheat version are very limited compared to DSSAT. Most importantly, there is no output of daily thermal time or its accumulation. The closest variable, DC code, is 4 calculations removed from daily thermal time, and so is not an efficient way to assure that the thermal time calculations match.

Since an adjustment to the vernalization parameter creates very close outputs, I should make one more check of the vernalization coefficient calculations before introducing the major change in the thermal time calculation.

**EURIKA**! Found the error: **reqvd** was set at 70 instead of 50. After changing this, the DC Code matched exactly, while using VSEN (p1v) equal to 1.00. Source of error: no idea. I can’t find any reference to this parameter, except that it was on my list to ask Senthold about last week (can’t remember whether I actually confirmed it with him). That alone would have let me solve this problem weeks ago, if I had kept a list of relative confidence in each parameter.

I renamed the reqvd parameter as VREQ, and moved it to the ECOTYPE file.

Next, I will adjust so that other outputs match. Return to [Analysis of Outputs](Analysis%20of%20Outputs.docx).

Senthold suggested that there may be a little stress causing the differences, since the potential in Nwheat is only abundant water and N.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Jan 22, 2013**: Back after working mostly in the [soil-water](Soil%20Log.docx) and [N documents](Nitrogen%20Log.docx).

The three equivalent ways of running the model with no stress **have very similar outputs**:

1. No H2O or N stress (potential production)
2. H2O stress only, with abundant water applied.
3. H2O and N stress, with abundant water and N applied (development condition)

Testing the model under H2O and N stress, biomass and yield generally behave as expected, but not to the degree expected (crop cannot be killed). The stress indices may not be having sufficient impact. So, the following work is indicated:

* N and H2O stress may not be functioning properly. This will be best judged by comparing output to an identically-stressed simulation, using the same climate, soil and genotype. Lacking comparative runs, I could examine outputs and try to find inconsistencies between available water and N, and plant growth. But this is a much less direct and more time-consuming path.
* After (or before) checking and fixing the above, WHAPS must be tested against APSIM using other soil and climate inputs.
* Also, [Output files are incomplete](file:///C:\Users\froyce\Documents\Project_Areas\Wheat%20Model\Progress%20Narrative%20Files\Output%20Files.docx). Numerous variables whose names have been changed need to be translated back into DSSAT. This could be done by anyone with patience. If I do it, I will need some tips from Cheryl.
* Finally, I should provide some sort of documentation:
  + Method used for transferring code
  + Describe any aspects not described in either APSIM or DSSAT documentation
  + List known problems (high LAI?)

**Jan 22, 2013**: Senthold said he would ask Davide to do some N and H2O stress runs using the same soil, climate and genotype. In the meantime, update the stress reporting in PlantGro.Out, as described in the [Output files](C:\\Users\\froyce\\Documents\\Project_Areas\\Wheat Model\\Progress Narrative Files\\Output Files.docx) .

Along with this, I will ask him to include H2O and N stress indices in the output:

swdef(1)

swdef(2)

swdef(3)

Nfact(1)

Nfact(2)

Nfact(3)

Nfact(4)

**Jan 24, 2013**: Davide sent files, but he could not do the detailed stress breakdown daily.

His first water stress run #1 reduces the irrigation from 28 to 9 applications of 30 mm each. Water stress run #2 further reduces the irrigation to 5 applications (no other changes). There are some other minor differences in his Mexico.PAR files, compared to the original from Senthold which we copied for the WHAPS version.

The N stress run lowered the initial Soil water content from 0.211 to 0.111 per layer, and changed the N fertilization rate from 150+50+50 to 50+20.

**Jan 25, 2013**: The stress runs in APSIM differ substantially from those in DSSAT. Unfortunately, Davide could not output the various dimensions of Nwheat H2O and N stress (I determined that from the code, see below).

I will use a copy of Nwheat consolidated.xlsx called [Nwheat stress comparison.xlsx](../Daily%20Outputs/Nwheat%20stress%20comparison.xlsx) to compare APSIM to DSSAT, beginning with water stress. The first step will be to substitute in the stressed outputs on the mt\_0 (APSIM output) and the PlantGro-new (DSSAT output) pages.

Unfortunately, the latest Nwheat outputs from Davide lack some basic variables for comparison: stem weight (stem\_in\_t), leaf weight (leaf\_wt), Continue to complete comparisons in [Nwheat stress comparison.xlsx](file:///C:\Users\froyce\Documents\Project_Areas\Wheat%20Model\Daily%20Outputs\Nwheat%20stress%20comparison.xlsx) to see whether other variables are missing.

Examining the graphs in [Nwheat stress comparison.xlsx](file:///C:\Users\froyce\Documents\Project_Areas\Wheat%20Model\Daily%20Outputs\Nwheat%20stress%20comparison.xlsx), page “WHAPS-vs-NWheat\_stress”, DSSAT LAI exceeds APSIM LAI long before DSSAT Above Ground Weight exceeds APSIM. However, DSSAT root weight is most consistently greater that APSIM, from DAP 48 to end of run. Grain weight is initially the same, but rapidly diverges after DAP 96. How is this similar to, or different from the no-stress comparison? Only LAI is divergent in the no-stress runs.

Regarding the Stress indices in APSIM output files:

* n\_stress = n\_stress + (1 - nfact2)
* w\_stress = w\_stress + (1 - swdf2)

So, since in nwheats.for nfact2 = nfact(2) and swdf2 = swdef(2), the stress outputs are cumulative (should have been obvious). Anyway, I can make the PlantGro stress outputs match this, at least for the time being. PlantGro has two water stress (WSPD & WSGD) and one N stress indices.

Water stress in PlantGro.Out:

WSPD = SWF\_AV = (1.0 - SWFAC) = 1- swdef(photo\_nw)

WSGD = TUR\_AV = (1.0 - TURFAC) = 1 - swdef(cellxp)

Nitrogen stress in PlantGro.Out:

NST\_AV = (1 - nfact2)

When I get the stem and leaf weight water-stress outputs,

* Could part of the problem lie in the ability of DSSAT to simulate only water stress, whereas with APSIM, water stress could lead to simultaneous N stress? Will compare N stress indices in the no-stress vs stress APSIM runs. Looking at no-stress file [mt\_0\_dec-4.out](../Daily%20Outputs/mt_0_dec-4.out) vs water stress file [WS2\_0.out](../Daily%20Outputs/Stress%20Runs/02%20Water%20Stress%202%20(WS2)/Outputs/WS2_0.out), the N stress also varies even though N application has not.
* The divergence in the models is so large, that they must only be compared with DSSAT in the ‘stress-on’ mode. The DSSAT no-stress options will have to be fixed later.
* (**Begin Here for material for report**) Compare H2O and N stress at four levels: 1) luxury inputs; 2) 60% reduction in water + luxury N; 3) further 40% reduction in water + luxury N and 4) same heavy water reduction + IC water reduction + N reduction. DSSAT shows much less N stress, even with no N added. It is possible that the DSSAT soil is working as intended, with this amount of mulch. Or maybe this extreme amount of residue has uncovered a weakness in DSSAT.

1. Luxury water and N experiment, to compare these stress levels under abundant conditions: Neither model showed water stress. Both showed slight N stress: DSSAT during the last 4 days of the experiment, after max grain yield was reached and LAI was at zero. Nwheat had four days of slight N stress during the emergence phase. This may be the cause of slightly lower leaf and stem weights in Nwheat compared to DSSAT, but even before stress disappears, weights become equal in the two models.
2. 60% reduction in water + luxury N (Davide’s WS1 option) LAI suffers more in Nwheat than WHAPS, so the existing discrepancy here is aggravated. Yield is substantially lower in Nwheat, although above ground weight seems to be different only by the amount of grain weight (no leaf or stem weights provided for these runs of Nwheat). N stress shows no change for either model, so discrepancies must be from water stress or N stress that is not reported in Nwheat. Water stress is different in the two models, beginning earlier in DSSAT but Nwheat surpasses around the same time the grain weights diverge.
3. Low water application with luxury N (Davide’s WS2 option), running DSSAT with water and N stress enabled (see [Nwheat stress comparison.xlsx](file:///C:\Users\froyce\Documents\Project_Areas\Wheat%20Model\Daily%20Outputs\Nwheat%20stress%20comparison.xlsx)).
   * 1. DSSAT: no stress until DAP 48, when water stress appears, gradually at first then strengthening. A water application ends stress for 4 days, then starts again and builds to full water stress, which diminishes late in experiment. Mild N stress also appears for 13 days, beginning with the endveg stage.
     2. Nwheat: slight water stress for 3 days beginning DAP 54, then returning at DAP 63. Full water stress at DAP 77, continuing until the end of the experiment. The same early N stress appears DAP 11 and lasts four days. N stress re-appears at DAP 49, before there is any water stress reported (but only one three types of water stress is reported). This N stress rises to 0.49, then diminishes and disappears on DAP 78.
4. Lower initial soil water, low water application and low N application (Davide’s WS3 option).
   * 1. Water stress in DSSAT and Nwheat are fairly similar, although DSSAT is higher stress early (thru DAP 70) and Nwheat higher for the rest of the season.
     2. N stress could not be more different: Nwheat quite high, and DSSAT, none.

The most outstanding features of the lower-stress runs is the huge difference in yield (DSSAT much higher) and N stress (DSSAT much lower). The higher DSSAT yield could be partially related to the higher DSSAT LAI (a chronic difference between the models). Additionally, it is strange that for Nwheat, N stress appears before water stress, even though N application remains at the highest level. In the higher stress run, DSSAT LAI, above ground weight and root weight are all much lower than Nwheat, yet the yields are very close! Water stress is also a fairly good match, but the N stress is not at all similar.

Possible sources of the discrepancies:

* There is, and has been a problem with LAI. This likely contributes, but is probably not the primary cause.
* In the “water stress only” run (WS2), the drastic drop in APSIM yield, from about 8 t/ha to just over 2 t/ha may be linked to high, unexpected N stress. This N stress may indicate an error in the APSIM setup, or APSIM soil model.
* In the higher water stress plus higher N stress run (WS3), early water stress in the DSSAT system may be the cause of low above ground and root weight, compared to APSIM. Yet, the grain yield is very close. This probably reflects the relatively high N stress in Nwheat, which strongly affects grain yield. The high water stress in DSSAT is not due to greater biomass or LAI (both are lower in DSSAT). This may be a difference in handling residue/mulch.
* Comparing plant N between the model versions, it appears that DSSAT version is taking much more N from the stem and leaves than the APSIM version, thus creating more grain. The overall N in Tops is very close, while DSSAT has much more in Grain at the end of the season, and a similar amount less in stems and leaves. Check algorithm(s) for transferring N to grain. Candidates: subroutine nwheats\_gnptl; subroutine nwheats\_grnit; subroutine nwheats\_gndmd;
* The DSSAT version works well when water is abundant, but when water is limiting, then too much grain is produced. Although I do not have weights for other above-ground parts, the N per area numbers suggest that the grain is produced at the expense of stems and leaves. Here seem two possibilities:
  + Grain demand is higher in DSSAT. But wouldn’t this show up in production with luxury N, also?
  + Grain demand is the same, but grain is allowed to extract more N from other plant parts. What code determines the extent to which grain demand can pull N out of stems and leaves? nwheats\_gnptl (called by nwheats\_cdemand and nwheats\_grnit.
    - “N available in stover is the difference of its N content and the minimum it's allowed to fall to.”

NUPC on page “N uptake-WS2” in [Nwheat stress comparison.xlsx](file:///C:\Users\froyce\Documents\Project_Areas\Wheat%20Model\Daily%20Outputs\Nwheat%20stress%20comparison.xlsx) (dimmed headers are yet to be corrected)

**Feb 4th, 2013**: Senthold told me that the money ran out with January, so to send my report. The 6-month project could not be finished in 20 months.

**Final report**.

The objectives:

* Describe what was done
* Include all relevant code
* Provide direction for next person to pick this up when funding is available
* Send a final report to Senthold with a copy to Davide and Cheryl

asking about the logic of the WS2 outputs.

* examine soil water and soil N indices for clues

When I get leaf and stem weight:

* Compare leaf weight vs LAI differences: could a divergence in SLA be causing the discrepancy here?
* Where does DSSAT first show signs of diverging from APSIM under water stress?

**Apr 23, 2013**: Gave [report](Nwheat%20to%20DSSAT%20Report.docx) plus NWheat DSSAT and FORTRAN files to Cheryl:

* relevant files from C:\X-DSSAT45-Nwheat\Wheat (not all data subfolders), plus soil.sol and all UFMX weather files
* C:\NwheatCSM\Source files

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Appendix**:

What does the following code imply (from **residue.for**, but similar code is common to all 4 Nwheat files):

\* ================================================================

subroutine APSIM\_residue (Action, data\_string)

\* ================================================================

implicit none

**dll\_export** apsim\_residue

include 'const.inc' ! Global constant definitions

include 'residue.inc' ! residue common block

include 'string.pub'

include 'engine.pub'

include 'error.pub'

\*+ Sub-Program Arguments

character **Action\*(\*)** ! Message action to perform

character data\_string\*(\*) ! Message data

Cheryl thinks the control variable “Action” is equivalent to “Dynamic”. Where does “Action” take its value? The Apsim\_nwheats subroutine is based on statements such as:

else if (Action.eq.MES\_Init) then

call nwheats\_zero\_variables ()

call nwheats\_init ()

But “Action” is not updated in any of the code I have.

A related variable, “nwheats\_status”, is updated in “subroutine nwheats\_send\_my\_variable”, via a call to “respond2get\_integer\_var” subroutine or function (cannot locate any declaration).

What is the nature of frequently-used function, “fill\_real\_array”? It is not described in our 4 APSIM Fortran files. \

Other mystery functions (sub-routines):

* subtract\_real\_array
* add\_real\_array
* pnup
* bound\_check\_real\_var
* Report\_date\_and\_event
* rep\_evnt
* bndchk
* cw\_pgdin
* new\_postbox
* post\_real\_array
* message\_send\_immediate
* respond2get\_integer\_var
* **bound\_check\_real\_var** (fstage, 0.0, 1.0, 'fstage')

From nwheats.for:

Sub-Program Arguments

! Note - these names are the same as those in the plant common block.

! If this block is included in this procedure, these will

! need to be renamed. They are named the same for consistency.

\*

real carb ! (INPUT) biomass supply for plant growth (g/plant)

real gro\_wt (\*) ! (OUTPUT) actual biomass uptake by plant parts (g/part)

real pl\_dmd (\*) ! (INPUT) biomass demand for plant parts (g/part)

Do fdsw and adf constitute a set of three ordered pairs (like a TABEX)? If so, this would mean changes to Cultivar file, and possibly an addition to Ecotype

Fdsw 0.0 0.5 1.0 ! fraction of drainable soil water (0-1)

Adf 1.0 1.0 0.0 ! aeration deficit (1 = no stress)