Algorithms in FAST v8

Bonnie Jonkman

March 18, 2014

1 Definitions and Nomenclature

Module Name	**Abbreviation in Module**	**Abbreviation in this Document**
ElastoDyn	ED	ED
AeroDyn	AD	AD
ServoDyn	SrvD	SrvD
SubDyn	SD	SD
HydroDyn	HydroDyn	HD
MAP	MAP	MAP
FEAMooring	FEAM	FEAM
InflowWind	InfW	InfW

Table 1: Abbreviations for modules in FAST v8

2 Initializations

3 Input-Output Relationships

3.1 Input-Output Solves (Option 2 Before 1)

This algorithm documents the procedure for the Input-Output solves in FAST, assuming all modules are in use. If an individual module is not in use during a particular simulation, the calls to that module's subroutines are omitted and the module's inputs and outputs are neither set nor used.

1: **procedure** CalcOutputs_And_SolveForInputs()
2: $y\_ED \leftarrow$ ED_CalcOutput($p\_ED, u\_ED, x\_ED, xd\_ED, z\_ED$)
3:
4: $u\_AD \leftarrow$ TransferOutputsToInputs(y_ED)
5: $y\_AD \leftarrow$ AD_CalcOutput($p\_AD, u\_AD, x\_AD, xd\_AD, z\_AD$)
6:
7: $u\_SrvD \leftarrow$ TransferOutputsToInputs(y_ED, y_AD)
8: $y\_SrvD \leftarrow$ SrvD_CalcOutput($p\_SrvD, u\_SrvD, x\_SrvD, xd\_SrvD, z\_SrvD$)
9:
10: u_ED(not platform reference point) $\leftarrow$ TransferOutputsToInputs(y_SrvD, y_AD)
11: $u\_HD \leftarrow$ TransferMeshMotions(y_ED)
12: $u\_SD \leftarrow$ TransferMeshMotions(y_ED)
13: $u\_MAP \leftarrow$ TransferMeshMotions(y_ED)
14: $u\_FEAM \leftarrow$ TransferMeshMotions(y_ED)
15:
16: ED_HD_SD_Mooring_InputOutputSolve()
17:
18: *If AeroDyn or ServoDyn had states to update, we should do this:*
19: $u\_AD \leftarrow$ TransferOutputsToInputs(y_ED)
20: $u\_SrvD \leftarrow$ TransferOutputsToInputs(y_ED, y_AD)
21: *However, they don't so we'll omit these steps for efficiency.*
22: **end procedure**

Note that inputs to *ElastoDyn* before calling CalcOutput() in the first step are not set in CalcOutputs_And_SolveForInputs(). Instead, the *ElastoDyn* inputs are set depending on where CalcOutputs_And_SolveForInputs() is called:

- At time 0, the inputs are the initial guess from *ElastoDyn*;
- On the prediction step, the inputs are extrapolated values from the time history of ElastoDyn inputs;
- On the first correction step, the inputs are the values calculated in the prediction step;
- On subsequent correction steps, the inputs are the values calculated in the previous correction step.

3.2 Input-Output Solve for $HydroDyn$, $SubDyn$, MAP, $FEAMooring$, and the Platform Reference Point Mesh in $ElastoDyn$

This procedure implements Solve Option 1 for the accelerations and loads in $HydroDyn$, $SubDyn$, MAP, $FEAMooring$, and $ElastoDyn$ (at its platform reference point mesh). The other input-output relationships for these modules are solved using Solve Option 2.

1: **procedure** ED_HD_SD_Mooring_InputOutputSolve()
2:
3: $y\_MAP \leftarrow$ CalcOutput($p\_MAP, u\_MAP, x\_MAP, xd\_MAP, z\_MAP$)
4: $y\_FEAM \leftarrow$ CalcOutput($p\_FEAM, u\_FEAM, x\_FEAM, xd\_FEAM, z\_FEAM$)
5:
6: ▷ Form u vector using loads and accelerations from u_HD, u_SD, and platform reference input from u_ED
7:
8: $u \leftarrow$ u_vec(u_HD, u_SD, u_ED)
9: $k \leftarrow 0$
10: **loop** ▷ Solve for loads and accelerations (direct feed-through terms)
11: $y\_ED \leftarrow$ ED_CalcOutput($p\_ED, u\_ED, x\_ED, xd\_ED, z\_ED$)
12: $y\_SD \leftarrow$ SD_CalcOutput($p\_SD, u\_SD, x\_SD, xd\_SD, z\_SD$)
13: $y\_HD \leftarrow$ HD_CalcOutput($p\_HD, u\_HD, x\_HD, xd\_HD, z\_HD$)
14: **if** $k \geq k\_max$ **then**
15: exit loop
16: **end if**
17: $u\_HD\_tmp \leftarrow$ TransferMeshMotions(y_ED, y_SD)
18: $u\_SD\_tmp \leftarrow$ TransferMeshMotions(y_ED)
$\cup$ TransferMeshLoads(y_HD, u_HD_tmp)
19: $u\_ED\_tmp \leftarrow$ TransferMeshLoads($y\_HD, y\_SD, y\_MAP, y\_FEAM, u\_HD\_tmp, u\_SD\_tmp, u\_MA$
20:
21: $U\_Residual \leftarrow u -$ u_vec($u\_HD\_tmp, u\_SD\_tmp, u\_ED\_tmp$)
22:
23: **if** last Jacobian was calculated at least DT_UJac seconds ago **then**
24: Calculate $\frac{\partial U}{\partial u}$
25: **end if**
26: Solve $\frac{\partial U}{\partial u}\delta u = -U\_Residual$ for δu
27:
28: **if** $\|\delta u\|_2 <$ tolerance **then** ▷ To be implemented later
29: exit loop
30: **end if**
31:
32: $u \leftarrow u + \delta u$
33: Transfer u to u_HD, u_SD, and u_ED▷ loads and accelerations only
34: $k = k + 1$
35: **end loop**
36: ▷ Transfer non-acceleration fields to motion input meshes

37:

38: u_HD(not accelerations) $\leftarrow$ TransferMeshMotions(y_ED, y_SD)

39: u_SD(not accelerations) $\leftarrow$ TransferMeshMotions(y_ED)

40:

41: $u\_MAP \leftarrow$ TransferMeshMotions(y_ED)

42: $u\_FEAM \leftarrow$ TransferMeshMotions(y_ED)

43: **end procedure**