**Appendix C. Alaska sound cubic-foot volume algorithm**

The algorithm to estimate the percent of cubic-foot volume contained in the missing top (needed to calculate sound volume in Alaska) is too complex to easily portray; the program code is included here for reference. The algorithm is based partially on work from Flewelling and Raynes (1993) as well as unpublished regional reports that far pre-date this documentation.

DEF\_PCT\_MSTOP := NVL(C\_TREE\_VOL\_REC.CULLMSTOP,0);

DEF\_PCT\_CULL := NVL(C\_TREE\_VOL\_REC.CULL,0);

DEF\_PCT\_CULLFLD := NVL(C\_TREE\_VOL\_REC.CULL\_FLD,0);

SPN := c\_tree\_vol\_rec.spcd;

IF(C\_TREE\_VOL\_REC.STATECD=2 AND C\_TREE\_VOL\_REC.DIA>=5

AND C\_TREE\_VOL\_REC.HT>C\_TREE\_VOL\_REC.ACTUALHT) THEN

DEF\_PCT\_MSTOP := NIMS\_DEFECT\_MISSING\_TOP.AK\_MISSING\_PCT(C\_TREE\_VOL\_REC.SPCD,

C\_TREE\_VOL\_REC.DIA,

C\_TREE\_VOL\_REC.HT,

C\_TREE\_VOL\_REC.ACTUALHT);

IF(C\_TREE\_VOL\_REC.CYCLE<>0) THEN

DEF\_PCT\_CULL := (1-DEF\_PCT\_MSTOP/100)\*DEF\_PCT\_CULLFLD+DEF\_PCT\_MSTOP;

END IF;

END IF;

IF(DBH>=7) THEN

HIDDEN\_DEFECT := -1.7069+.2688\*DBH;

ELSE

HIDDEN\_DEFECT := 0;

END IF;

DEF\_PCT\_SND := HIDDEN\_DEFECT+DEF\_PCT\_CULL;

v\_volcfsnd := v\_volcfgrs\*(1-DEF\_PCT\_SND/100);

----------------------------------------------------------------------------------------------------------------------------

FUNCTION AK\_MISSING\_PCT(SPN IN NUMBER, DIA IN NUMBER, THT IN NUMBER, HT\_BREAK IN NUMBER) RETURN NUMBER IS

MISSING\_PCT NUMBER := 0;

TOP NUMBER := 4;

STUMP NUMBER := 1;

TAPEQ VARCHAR2(10) := '5';

TAPSPN NUMBER := 263;

TAPGEO VARCHAR2(10) := NULL;

BRKEQ VARCHAR2(10) := '000007';

BRKSPN NUMBER := 263;

CV4 NUMBER := 0;

CVB NUMBER := 0;

CVTOP NUMBER := 0;

BEGIN

IF(THT IS NULL) THEN

MISSING\_PCT := NULL;

ELSIF(HT\_BREAK IS NULL) THEN

MISSING\_PCT := NULL;

ELSE

IF SPN=242 THEN

TAPSPN := 242;

BRKSPN := 242;

ELSE

TAPSPN := 263;

BRKSPN := 263;

END IF;

CV4 := NIMS\_TAPER\_FLW\_TOP.SF\_CV(TAPSPN, DIA, THT, TOP, STUMP, TAPGEO, TAPEQ, BRKEQ, BRKSPN);

CVB := NIMS\_TAPER\_FLW\_TOP.SF\_CV\_BREAK(TAPSPN, DIA, THT, HT\_BREAK, TOP, STUMP, TAPGEO, TAPEQ, BRKEQ, BRKSPN);

CVTOP := CV4-CVB;

MISSING\_PCT := 100\*CVTOP/CV4;

END IF;

RETURN MISSING\_PCT;

END;

---------------------------------------------------------------------------

--Find cubic volume below any top diameter

--From file ALLCODE.FOR function SF\_CV

--Currently works only with west coast version of Flewelling Taper

--Need to consider DOB verses DIB methods for Rocky Mt equations

--and debug Inland equations when time permitts

FUNCTION SF\_CV(SPN IN NUMBER,

DBH IN NUMBER,

THT IN NUMBER,

TOP IN NUMBER,

STUMP IN NUMBER,

GEO IN VARCHAR2 := NULL,

EQN IN VARCHAR2 := NULL,

BARK\_EQ IN VARCHAR2 := NULL,

BARK\_SPN IN NUMBER := NULL) RETURN NUMBER IS

VOL NUMBER;

TOPHT NUMBER;

STUMPVOL NUMBER;

BEGIN

GEOSUB := GEO;

TAPEQ := EQN;

IF(TAPEQ='5') THEN -- Flewelling 2 point NW

SETUP\_W(SPN,DBH,THT,GEOSUB, BARK\_EQ, BARK\_SPN);

ELSIF(TAPEQ='6') THEN -- Flewelling 2 point INLAND

SETUP\_I(SPN,DBH,THT,GEOSUB, BARK\_EQ, BARK\_SPN);

ELSIF (TAPEQ='7') THEN -- Flewelling 2 point BLACK HILLS

SETUP\_B(SPN,DBH,THT,GEOSUB, BARK\_EQ, BARK\_SPN);

ELSIF (TAPEQ='8') THEN -- Flewelling 2 point ROCKY MTS

SETUP\_R(SPN,DBH,THT,GEOSUB, BARK\_EQ, BARK\_SPN);

ELSIF (TAPEQ='9') THEN -- Flewelling 2 point BLACK HILLS, DOB substitution

SETUP\_B(SPN,DBH,THT,GEOSUB, BARK\_EQ, BARK\_SPN);

ELSIF (TAPEQ='10') THEN -- Flewelling 2 point ROCKY MTS, DOB substitution

SETUP\_R(SPN,DBH,THT,GEOSUB, BARK\_EQ, BARK\_SPN);

END IF;

VOL := SF\_VOL2(THT);

IF(TOP>0) THEN

TOPHT := HT\_AT\_DIB(SPN,DBH,THT,TOP,GEOSUB, TAPEQ, BARK\_EQ, BARK\_SPN);

VOL := SF\_CUMV2(THT, TOPHT);

END IF;

IF(STUMP>0) THEN

STUMPVOL := SF\_CUMV2(THT,STUMP);

VOL := VOL-STUMPVOL;

END IF;

RETURN VOL;

END;

-------------------------------------------------------------------------------

--Fleweling NW Coast Setup

--Execute once per tree before getting dib

--Sets up stem form coeffients and

--Returns F Scaling Factor or ratio between Breast high DIB from regression and

--and DIB from taper equation.

PROCEDURE SETUP\_W(SPN IN NUMBER,

DBH IN NUMBER,

THT IN NUMBER,

GEOSUB IN VARCHAR2 := NULL,

BARK\_EQ IN VARCHAR2 := NULL,

BARK\_SPN IN NUMBER := NULL) IS

BRKSPN NUMBER(5);

BEGIN

BRKEQ := BARK\_EQ;

BRKSPN := BARK\_SPN;

F := 1;

RH := BH/THT;

-- If bark equation is specified in default configuration, use it

IF BRKEQ IS NOT NULL THEN

DBT\_BH := NIMS\_BARK.PICK\_DBT(BRKEQ, BRKSPN, DBH, THT, GEOSUB);

ELSE

DBT\_BH := DBT\_BH\_W(SPN,DBH,THT,GEOSUB);

END IF;

DBH\_IB := DBH - DBT\_BH;

IF(SPN=202) THEN

SHP\_W\_DF(SPN,DBH,THT,GEOSUB);

ELSIF (SPN=263) THEN

SHP\_W\_WH(SPN,DBH,THT,GEOSUB);

ELSIF (SPN=242) THEN

SHP\_W\_RC(SPN,DBH,THT);

END IF;

TAPER;

YHAT\_BH := YHAT(THT,4.5);

F := DBH\_IB / YHAT\_BH;

END;

--------------------------------------------------------------------------------------------

--Function to select equation and compute double bark thickness

FUNCTION PICK\_DBT(BARK\_EQ IN VARCHAR2,

SPN IN NUMBER,

DIA IN NUMBER,

HT IN NUMBER := NULL,

GEOSUB IN VARCHAR2 := NULL) RETURN NUMBER IS

DBT NUMBER := 0;

BEGIN

IF BARK\_EQ = '000001' THEN --FVS EM

DBT := NIMS\_BARK\_NW.DBT\_BH\_FVS\_EM(SPN, DIA);

ELSIF BARK\_EQ = '000002' THEN --FVS NI

DBT := NIMS\_BARK\_NW.DBT\_BH\_FVS\_NI(SPN, DIA);

ELSIF BARK\_EQ = '000003' THEN --Flewelling Black hills

DBT := NIMS\_BARK\_NW.DBT\_BH\_FLW\_B(SPN, DIA, HT);

ELSIF BARK\_EQ = '000004' THEN --Flewelling Rocky Mountians

DBT := NIMS\_BARK\_NW.DBT\_BH\_FLW\_R(SPN, DIA, HT);

ELSIF BARK\_EQ = '000005' THEN --Flewelling N. Wyoming

DBT := NIMS\_BARK\_NW.DBT\_BH\_FLW\_WY(SPN, DIA, HT);

ELSIF BARK\_EQ = '000006' THEN --Flewelling Alaska

DBT := NIMS\_BARK\_NW.DBT\_BH\_FLW\_A(SPN, DIA, HT);

ELSIF BARK\_EQ = '000007' THEN --Flewelling West Coast

DBT := NIMS\_BARK\_NW.DBT\_BH\_FLW\_W(SPN, DIA, HT, GEOSUB);

ELSIF BARK\_EQ = '000008' THEN --Flewelling Inland

DBT := NIMS\_BARK\_NW.DBT\_BH\_FLW\_I(SPN, DIA, HT, GEOSUB);

ELSIF BARK\_EQ = '000009' THEN --Larson

DBT := NIMS\_BARK\_NW.DBT\_BH\_LARSON(SPN, DIA);

END IF;

RETURN DBT;

EXCEPTION

WHEN OTHERS THEN

RETURN 0;

END PICK\_DBT;

END NIMS\_BARK;

--------------------------------------------------------------------------------

--Fleweling Breast High DOB to Double bark Thickness for NW Coast

--Returns Double Bark thicknes at Breast hight

--From F\_WEST.FOR FUNCTION FDBT\_C1

FUNCTION DBT\_BH\_FLW\_W(SPN IN NUMBER, DBH IN NUMBER, THT IN NUMBER, GEOSUB IN VARCHAR2 := NULL)

RETURN NUMBER IS

DMEDIAN NUMBER := 0;

DFORM NUMBER := 0;

RATIO NUMBER := 0;

DBT\_BH NUMBER := 0;

BEGIN

IF(SPN=202) THEN

DMEDIAN := .566\*(THT-4.5)\*\*( .634 + .00074\*THT);

DFORM := DBH/DMEDIAN -1.0;

IF(GEOSUB IS NULL) THEN

RATIO := EXP(-2.4641+.04393\*LN(DBH)-.2922\*DFORM+.05964\*DFORM\*LN(DBH));

ELSIF(GEOSUB='01') THEN

RATIO := EXP(-2.5087+.03600\*LN(DBH)-.4086\*DFORM+.10120\*DFORM\*LN(DBH));

ELSIF(GEOSUB='02') THEN

RATIO := EXP(-2.5087+.03600\*LN(DBH)-.4086\*DFORM+.10120\*DFORM\*LN(DBH)+.117);

ELSIF(GEOSUB='03') THEN

RATIO := EXP(-2.5087+.03600\*LN(DBH)-.4086\*DFORM+.10120\*DFORM\*LN(DBH)+.121);

ELSIF(GEOSUB='04') THEN

RATIO := EXP(-2.5087+.03600\*LN(DBH)-.4086\*DFORM+.10120\*DFORM\*LN(DBH)+.133);

ELSIF(GEOSUB='05') THEN

RATIO := EXP(-2.5087+.03600\*LN(DBH)-.4086\*DFORM+.10120\*DFORM\*LN(DBH)+.025);

ELSIF(GEOSUB='06') THEN

RATIO := EXP(-2.5087+.03600\*LN(DBH)-.4086\*DFORM+.10120\*DFORM\*LN(DBH)+.028);

ELSIF(GEOSUB='07') THEN

RATIO := EXP(-2.5087+.03600\*LN(DBH)-.4086\*DFORM+.10120\*DFORM\*LN(DBH)+.088);

ELSIF(GEOSUB='08') THEN

RATIO := EXP(-2.5087+.03600\*LN(DBH)-.4086\*DFORM+.10120\*DFORM\*LN(DBH)+.028);

ELSE

RATIO := EXP(-2.4641+.04393\*LN(DBH)-.2922\*DFORM+.05964\*DFORM\*LN(DBH));

END IF;

ELSIF(SPN=263) THEN

IF(GEOSUB IS NULL) THEN

RATIO := .04504\*(1.0 +0.8307\*EXP(-.2048\*DBH));

ELSIF(GEOSUB='01') THEN

RATIO := .04221\*(1.0 +0.8836\*exp(-.2145\*DBH));

ELSIF(GEOSUB='02') THEN

RATIO := .04221\*(1.0 +0.8836\*exp(-.2145\*DBH))\*(1.0+.118);

ELSIF(GEOSUB='03') THEN

RATIO := .04221\*(1.0 +0.8836\*exp(-.2145\*DBH))\*(1.0+.204);

ELSIF(GEOSUB='04') THEN

RATIO := .04221\*(1.0 +0.8836\*exp(-.2145\*DBH))\*(1.0+.105);

ELSIF(GEOSUB='05') THEN

RATIO := .04221\*(1.0 +0.8836\*exp(-.2145\*DBH))\*(1.0+.058);

ELSIF(GEOSUB='06') THEN

RATIO := .04221\*(1.0 +0.8836\*exp(-.2145\*DBH))\*(1.0+.061);

ELSIF(GEOSUB='07') THEN

RATIO := .04221\*(1.0 +0.8836\*exp(-.2145\*DBH))\*(1.0+.118);

ELSIF(GEOSUB='08') THEN

RATIO := .04221\*(1.0 +0.8836\*exp(-.2145\*DBH))\*(1.0+.071);

ELSE

RATIO := .04504\*(1.0 +0.8307\*EXP(-.2048\*DBH));

END IF;

ELSIF(SPN=242) THEN

IF(DBH>3.8) THEN

RATIO := 0.01949\*(1.0+15.599/DBH-29.212/DBH\*\*2);

ELSE

RATIO := 0.01949\*(1.0+15.599/3.8-29.212/3.8\*\*2);

END IF;

END IF;

DBT\_BH := RATIO\*DBH;

RETURN DBT\_BH;

END;

----------------------------------------------------------------------------------------

--Stem form shape paraemeters for west coast Western Hemlock

--From F\_WEST.FOR subroutine SHP\_W4

PROCEDURE SHP\_W\_WH(SPN IN NUMBER, DBH IN NUMBER, THT IN NUMBER, GEOSUB IN VARCHAR2 := NULL) IS

FF NUM\_TABLE;

DMEDIAN NUMBER;

DFORM NUMBER;

U1 NUMBER := 0;

U2 NUMBER := 0;

U3 NUMBER := 0;

U4 NUMBER := 0;

U5 NUMBER := 0;

U6 NUMBER := 0;

U7 NUMBER := 0;

U8 NUMBER := 0;

U9 NUMBER := 0;

U9A NUMBER := 0;

BEGIN

FF(13) := -3.1137977;

FF(14) := 1.1996084;

FF(15) := -.01195901;

FF(17) := -.066829984;

FF(18) := .026990398;

FF(19) := .24661021;

FF(21) := -8.3415555;

FF(22) := 2.4274384;

FF(23) := -5.6918023;

FF(24) := .56487213;

FF(25) := -7.6464554;

FF(26) := 5.1709049;

FF(27) := -2.7133381;

FF(28) := .38918349;

FF(29) := -7.0;

FF(34) := -1.2882571;

FF(35) := 35.688410;

FF(36) := .17995769;

FF(37) := 1.5565605;

FF(38) := 6.4397446;

FF(39) := -1.3439736;

FF(40) := 6.3442558;

FF(42) := 11.898092;

FF(43) := -3.6789851;

FF(44) := .15168209;

FF(45) := -1.3248733;

FF(46) := -.11788962;

FF(47) := -.015909154;

IF(GEOSUB='01') THEN -- Oregon coast

FF(25) := -7.511;

FF(34) := -1.215;

ELSIF(GEOSUB='02') THEN -- Oregon east valley

FF(25) := -7.687;

FF(34) := -1.355;

ELSIF(GEOSUB='03') THEN -- Washington north

FF(25) := -7.224;

FF(34) := -1.177;

ELSIF(GEOSUB='04') THEN -- Washington rain shasow

FF(25) := -7.355;

FF(34) := -1.373;

ELSIF(GEOSUB='05') THEN -- Washington South

FF(25) := -7.632;

FF(34) := -1.398;

ELSIF(GEOSUB='06') THEN -- Washington west

FF(25) := -7.646;

FF(34) := -1.188;

ELSIF(GEOSUB='07') THEN -- Oregon west valley

FF(25) := -7.687;

FF(34) := -1.355;

ELSIF(GEOSUB='08') THEN -- Washington coast

FF(25) := -7.911;

FF(34) := -1.449;

END IF;

DMEDIAN := .2855\*(THT-4.5)\*\*(.307 - .00505\*THT + .00001745\*THT\*THT+0.19\*LN(THT));

DFORM := DBH/DMEDIAN-1.0;

U7 := FF(13) + FF(14)\*( 1.0 - exp(FF(15)\*THT));

IF(U7<-7.0) THEN

U7 := -7.0;

END IF;

IF(U7>7.0) THEN

U7 := 7.0;

END IF;

RHI1 := exp(U7) / ( 1.0 + exp(U7) );

IF(RHI1>0.5) THEN

RHI1 := 0.5;

END IF;

U9A := FF(17) + FF(18)\*LN(THT) + FF(19)\*DFORM;

IF((RHI1 + U9A)> 0.75) THEN

U9A := 0.75-RHI1;

END IF;

IF(U9A>0) THEN

U9 := U9A;

ELSE

U9 := 0;

END IF;

U8 := FF(21)+FF(22)\*LN(THT) + FF(23)\*DFORM + FF(24)\*LN(THT)\*DFORM;

U1 := FF(25)+FF(26)/(THT/100) + FF(27)/(THT/100)\*\*2 + FF(28)/(THT/100)\*\*3;

U2 := FF(29);

U3 := FF(34)+ FF(35)/THT + FF(36)\*THT\*DBH/1000 + FF(37)\*DFORM;

U4 := FF(38) + FF(39)\*LN(THT) + FF(40)\*DFORM;

U5 := FF(42) + FF(43)\*LN(THT) + FF(44)\*DBH;

IF(U1<-7.0) THEN U1:=-7.0; END IF;

IF(U1> 7.0) THEN U1:= 7.0; END IF;

IF(U2<-7.0) THEN U2:=-7.0; END IF;

IF(U2> 7.0) THEN U2:= 7.0; END IF;

IF(U3<-7.0) THEN U3:=-7.0; END IF;

IF(U3> 7.0) THEN U3:= 7.0; END IF;

IF(U4<-7.0) THEN U4:=-7.0; END IF;

IF(U4> 7.0) THEN U4:= 7.0; END IF;

IF(U5<-7.0) THEN U5:=-7.0; END IF;

IF(U5> 7.0) THEN U5:= 7.0; END IF;

IF(U7<-7.0) THEN U7:=-7.0; END IF;

IF(U7> 7.0) THEN U7:= 7.0; END IF;

IF(U8<-7.0) THEN U8:=-7.0; END IF;

IF(U8> 7.0) THEN U8:= 7.0; END IF;

U6 := FF(45) + FF(46)\*LN(DBH) + FF(47)\*THT;

IF(U6<-6.0) THEN U6:=-6.0; END IF;

IF(U6> 6.0) THEN U6:= 6.0; END IF;

U6 := 1.0 + EXP(U6);

IF(U6<1.005) THEN U6:=1.005; END IF;

IF(U6>100.0) THEN U6:=100.0; END IF;

R1 := EXP(U1)/ (1.0 + EXP(U1));

R2 := EXP(U2)/ (1.0 + EXP(U2));

R3 := EXP(U3)/ (1.0 + EXP(U3));

R4 := EXP(U4)/ (1.0 + EXP(U4));

R5 := 0.5 + 0.5\*EXP(U5)/ (1.0 + EXP(U5));

A3 := U6;

RHI1 := EXP(U7) / (1.0 + EXP(U7));

IF(RHI1>0.5) THEN RHI1:=0.5; END IF;

RHLONGI := U9;

RHI2 := RHI1 + RHLONGI;

RHC := RHI2 + (1.0 - RHI2) \* EXP(U8)/( 1.0 + EXP(U8));

END;

----------------------------------------------------------------------------------------

--Stem form shape paraemeters for west coast Western Redcedar

--From F\_WEST.FOR subroutine SHP\_W5

PROCEDURE SHP\_W\_RC(SPN IN NUMBER, DBH IN NUMBER, THT IN NUMBER) IS

FF NUM\_TABLE;

DMEDIAN NUMBER;

DFORM NUMBER;

U1 NUMBER := 0;

U2 NUMBER := 0;

U3 NUMBER := 0;

U4 NUMBER := 0;

U5 NUMBER := 0;

U6 NUMBER := 0;

U7 NUMBER := 0;

U8 NUMBER := 0;

U9 NUMBER := 0;

U9A NUMBER := 0;

ID NUMBER;

BEGIN

FF(13) := -1.1917515;

FF(15) := .15632952;

FF(17) := .44805624;

FF(18) := -.10269221;

FF(21) := .828142;

FF(22) := 17.750205;

FF(25) := -17.95926;

FF(26) := 3.8308966;

FF(27) := 45.163788;

FF(29) := -1.754609;

FF(30) := -7.9230813;

FF(31) := -112.53761;

FF(34) := -.17943249;

FF(35) := .15534485;

FF(36) := -.032777026;

FF(38) := 8.5305448;

FF(39) := -.83350599;

FF(40) := 12.100013;

FF(42) := 8.3021283;

FF(43) := -150.0;

FF(45) := -7.1594841;

FF(46) := .11263465;

FF(47) := 22.396603;

DMEDIAN := .11\*(THT-4.5)\*\*( 1.08 + 0.0006\*THT);

DFORM := DBH/DMEDIAN -1.0;

U7 := FF(13) + FF(15)\*DFORM;

IF(U7<-7.0) THEN

U7 := -7.0;

END IF;

IF(U7>1.0) THEN

U7 := 1.0;

END IF;

RHI1 := exp(U7) / ( 1.0 + exp(U7));

U9A := FF(17) + FF(18)\*LN(THT);

IF((RHI1 + U9A)> 0.75) THEN

U9A := 0.75-RHI1;

END IF;

IF(U9A>0) THEN

U9 := U9A;

ELSE

U9 := 0;

END IF;

U8 := FF(21)+FF(22)/DBH;

U1 := FF(25) + FF(26)\*LN(DBH) + FF(27)/DBH;

U2 := FF(29) + FF(30)/DBH + FF(31)/(DBH\*DBH);

U3 := FF(34) + FF(35)\*DBH + FF(36)\*THT;

U4 := FF(38) + FF(39)\*DBH + FF(40)\*DFORM;

U5 := FF(42) + FF(43)/DBH;

IF(U1<-7.0) THEN U1:=-7.0; END IF;

IF(U1> 7.0) THEN U1:= 7.0; END IF;

IF(U2<-7.0) THEN U2:=-7.0; END IF;

IF(U2> 7.0) THEN U2:= 7.0; END IF;

IF(U3<-7.0) THEN U3:=-7.0; END IF;

IF(U3> 7.0) THEN U3:= 7.0; END IF;

IF(U4<-7.0) THEN U4:=-7.0; END IF;

IF(U4> 7.0) THEN U4:= 7.0; END IF;

IF(U5<-7.0) THEN U5:=-7.0; END IF;

IF(U5> 7.0) THEN U5:= 7.0; END IF;

IF(U7<-7.0) THEN U7:=-7.0; END IF;

IF(U7> 7.0) THEN U7:= 7.0; END IF;

IF(U8<-7.0) THEN U8:=-7.0; END IF;

IF(U8> 7.0) THEN U8:= 7.0; END IF;

U6 := FF(45) + FF(46)\*DBH + FF(47)/DBH;

IF(U6<-6.0) THEN U6:=-6.0; END IF;

IF(U6> 6.0) THEN U6:= 6.0; END IF;

U6 := 1.0 + EXP(U6);

IF(U6<1.005) THEN U6:=1.005; END IF;

IF(U6>100.0) THEN U6:=100.0; END IF;

R1 := EXP(U1)/ (1.0 + EXP(U1));

R2 := EXP(U2)/ (1.0 + EXP(U2));

R3 := EXP(U3)/ (1.0 + EXP(U3));

R4 := EXP(U4)/ (1.0 + EXP(U4));

R5 := 0.5 + 0.5\*EXP(U5)/ (1.0 + EXP(U5));

A3 := U6;

RHI1 := EXP(U7) / (1.0 + EXP(U7));

IF(RHI1>0.5) THEN RHI1:=0.5; END IF;

RHLONGI := U9;

RHI2 := RHI1 + RHLONGI;

RHC := RHI2 + (1.0 - RHI2) \* EXP(U8)/( 1.0 + EXP(U8));

END;

-----------------------------------------------------------------

--Calculater taper coefficients, given geometric properties

PROCEDURE TAPER IS

K NUMBER;

YC NUMBER;

SLOPE\_RHI NUMBER;

YI\_MIN NUMBER;

YI\_MAX NUMBER;

YI1 NUMBER;

YI2 NUMBER;

S0 NUMBER;

S1 NUMBER;

S3 NUMBER;

K2 NUMBER;

F\_A3 NUMBER;

G\_A3 NUMBER;

YB\_MIN NUMBER;

YB\_MAX NUMBER;

YB NUMBER;

BEGIN

-- Upper Segment

K := 1.0;

YC := K\*(1.0-RHC);

C2 := R5 \* YC;

C1 := 3.0\* (YC - C2);

SLOPE := -(3.0-R5)\*K/2.0;

-- Middle Segment

S1 := SLOPE\*(RHC-RHI2);

YI\_MIN := YC - S1\*(1.0+2.0\*R3)/3.0;

YI\_MAX := YC - S1\*(5.0+4.0\*R3)/9.0;

YI2 := YI\_MIN + R4 \* (YI\_MAX - YI\_MIN);

S0 := R3\*S1;

B1 := ( 6\*YC -6\*YI2 -2\*S0-4\*S1) / (-3\*YC +3\*YI2 +2\*S0 +S1);

B2 := S1 \* (1-R3)/ ( 0.5 -1.0/(B1+1.0));

B4 := S0;

B0 := YI2;

SLOPE\_RHI := R3\*S1/(RHC-RHI2);

-- Straight segment at inflections

YI1 := YI2 - slope\_RHI \* RHLONGI;

IF (RHLONGI > 0.0 ) THEN

E2 := (YI2-YI1)/RHLONGI;

E1 := YI1 - (E2\*RHI1);

ELSE

E1 := YI2;

E2 := 0.0;

END IF;

-- Lower Segment

S3 := -SLOPE\_RHI \* RHI1;

K2 := S3/R1;

F\_A3 := 1/(6\*A3\*A3) + LN( 1 -1/A3) + 1/( 3\*(A3-1)) + 2/(3\*A3);

G\_A3 := ( 1/(A3-1) - 1/A3 - 1/(A3\*A3) - 1/(A3-1)\*\*3 ) / F\_A3;

YB\_MIN := YI1 + (2\*S3+K2)/3 + (S3-K2)\*F\_A3 / (+1/(A3-1) -1/A3 -1/(A3\*\*2) -1/(A3\*A3\*A3));

YB\_MAX := YI1 + (2\*S3+K2)/3 + (S3-K2)/G\_A3;

YB := YB\_MIN + R2\*(YB\_MAX - YB\_MIN);

A0 := YI1;

A2 := ( YB - YI1 -(2\*S3+K2)/3)/F\_A3;

A1 := ( K2 - S3 + A2 \*(1/(A3-1) -1/A3 -1/A3\*\*2)) / 3;

A4 := S3;

END;

-------------------------------------------------------------------------------

--calculate the predicted dib, given relative height

--this is based on Eqns 1 thru 6 of the draft report.

--This routine may also takes the optional step of calculating 1st derivatives.

FUNCTION YHAT(THT IN NUMBER, HTUP IN NUMBER) RETURN NUMBER IS

SF\_YHAT NUMBER := 0;

X NUMBER := 0;

Y NUMBER := 0;

RH\_LENGTH NUMBER;

I\_SEG NUMBER;

DY\_DX NUMBER;

DD\_DH NUMBER;

BEGIN

RH := HTUP/THT;

-- Tests for invalid input

IF(RH>1.0) THEN

SF\_YHAT := 0.0;

IF(INEEDSL=1) THEN

SLOPE := 0.0;

END IF;

RETURN SF\_YHAT;

END IF;

IF(RH<0.0) THEN

SF\_YHAT := F;

IF(INEEDSL=1) THEN

SLOPE := 0.0;

END IF;

RETURN SF\_YHAT;

END IF;

-- upper segment

IF(RH>=RHC) THEN

X := (1.0 - RH)/(1.0-RHC);

IF(TAPEQ='7' OR TAPEQ='9') THEN

Y := C2\*X + (C1/2)\*X\*X - (C1/6)\*X\*X\*X;

ELSE

Y := X \* (C2 +X \* ( (C1/2.0) - (C1/6.0)\*X));

END IF;

RH\_LENGTH := 1-RHC;

I\_SEG := 1;

IF(ineedsl=1) THEN

DY\_DX := C2 + X\*(C1 - C1/2\*X);

END IF;

-- middle segment

ELSIF(RH>=RHI2) THEN

X := (RH - RHI2)/( RHC - RHI2);

IF(X>0.0) THEN

IF(TAPEQ='7' OR TAPEQ='9') THEN -- Black hills

Y := B0 +B4\*X - B2/((B1+1)\*(B1+2))\*X\*\*(B1+2) + B2/6.0\*X\*X\*X;

ELSE

Y := B0 +X\*(B4 + X\*(-B2/((B1+1)\*(B1+2))\*X\*\*(B1) + B2/6.0\*X ));

END IF;

ELSE

Y := B0;

END IF;

IF(INEEDSL=1) THEN

RH\_LENGTH := RHC-RHI2;

I\_SEG := 2;

IF(X>0) THEN

DY\_DX := B4 -B2/(B1+1.0)\*X\*\*(B1+1) + B2/2.0\*X\*X;

ELSE

DY\_DX := B4;

END IF;

END IF;

-- Straight segment

ELSIF(RHLONGI>0.0 AND RH>RHI1) THEN

Y := E1 + E2\*RH;

IF(INEEDSL=1) THEN

I\_SEG := 3;

DY\_DX := E2;

RH\_LENGTH := 1.0;

END IF;

-- lower segment

ELSE

X := (RHI1-RH)/RHI1;

IF(TAPEQ='7' OR TAPEQ='9') THEN

Y := A0 + (A4+A2/A3)\*X + A2/(2\*A3\*A3)\*X\*X + A1\*X\*X\*X + A2\*LN(1.0 - X/A3);

ELSE

Y := A0 + X\*((A4+A2/A3) + X\*(A2/(2\*A3\*A3) + A1\*X)) + A2\*LN(1.0 - X/A3);

END IF;

IF(INEEDSL=1) THEN

RH\_LENGTH := RHI1;

I\_SEG := 4;

DY\_DX := A4 + A2/A3 + A2/(A3\*A3)\*X + 3\*A1\*X\*X - A2/(A3-X);

END IF;

END IF;

SF\_YHAT := F\*Y;

IF(INEEDSL=1) THEN

-- the derivatives were calculated based on y and x

-- lets also calculate them based on dib and H.

DD\_DH := DY\_DX \* F /( RH\_LENGTH \* THT);

IF(I\_SEG<>2 AND I\_SEG<>3) THEN

DD\_DH := -DD\_DH;

END IF;

SLOPE := DD\_DH;

END IF;

RETURN SF\_YHAT;

END;

--------------------------------------------------------------------------------

--Find height at which a specified DIB occurs.

--From SF\_HS

--This program calls BRK\_UP several times. I could not see where anything from

--those calls were being used so I did not include for now. CHECK OUT LATER.

--I Think I found it, when using rocky MT eq need to convert dob to dib but since

--I am getting the same ht as Natcruise I will leave for now.

FUNCTION HT\_AT\_DIB(SPN IN NUMBER,

DBH IN NUMBER,

THT IN NUMBER,

DIBUP IN NUMBER,

GEO IN VARCHAR2 := NULL,

EQN IN VARCHAR2 := NULL,

BARK\_EQ IN VARCHAR2 := NULL,

BARK\_SPN IN NUMBER := NULL) RETURN NUMBER IS

HTUP NUMBER;

HI2 NUMBER;

DI2 NUMBER;

HI1 NUMBER;

DI1 NUMBER;

D NUMBER;

D1 NUMBER;

D2 NUMBER;

RZ NUMBER;

DBASE NUMBER;

IBREAK NUMBER;

ITER NUMBER;

TOOHIGH NUMBER := THT;

TOOLOW NUMBER := 0;

ADJUST NUMBER :=0;

EPSILON NUMBER := .0001;

TOL NUMBER := .0005;

ERR NUMBER := 0;

EPS NUMBER := 0;

HTRY NUMBER;

DTRY NUMBER;

HHIGH NUMBER;

HLOW NUMBER;

EHIGH NUMBER;

ELOW NUMBER;

BEGIN

GEOSUB := GEO;

TAPEQ := EQN;

-- Step 1. Calculate diameter at inflection point (DI2)

-- and inflection height (HI2)

HI2 := RHI2\*THT;

DI2 := DIB(HI2,SPN,DBH,THT,GEOSUB, TAPEQ, BARK\_EQ, BARK\_SPN);

-- Make a first guess for H; choice of formula

-- depends on whether we are above or below inflection.

IF(DIBUP>DI2) THEN --check the straight region

TOOHIGH := HI2;

IF(RHLONGI>0.) THEN

HI1 := RHI1\*THT;

DI1 := PICK\_DIB(TAPEQ, SPN, DBH, THT, HI1, GEOSUB, BARK\_EQ, BARK\_SPN);

IF(DIBUP<DI1) THEN

TOOLOW := DI1;

RH := RHI2 - (RHI2-RHI1)\*(DIBUP-DI2)/(DI1-DI2);

GOTO L100;

END IF;

TOOHIGH := DI1;

ELSE

DI1 := DI2;

END IF;

-- SOLUTION IS BELOW THE INFLECTION

DBASE := PICK\_DIB(TAPEQ, SPN, DBH,THT,0.0,GEOSUB, BARK\_EQ, BARK\_SPN);

IF(DBASE<DIBUP) THEN

HTUP := 0.0;

INEEDSL := 0;

RETURN HTUP;

END IF;

-- Note: if there were an analytic solution, it would go here

RZ := ((DIBUP-DI1)/(DBASE-DI1))\*\*0.25;

RH := (1.0-RZ)\*RHI1;

ELSE -- Solution is above the inflection

TOOLOW := DI2;

RZ := 1.0-(DIBUP/DI2)\*\*2.0;

RH := RHI2 + (1-RHI2)\*RZ;

END IF;

<<L100>>

HTUP := RH\*THT;

INEEDSL := 1;

IBREAK := 0;

-- NEWTON SOLUTION FOR ROOTS

<<L110>>

ITER := 0;

<<L20>>

ITER := ITER+1;

IF(ITER>30) THEN

GOTO L200;

END IF;

D := PICK\_DIB(TAPEQ, SPN, DBH,THT,HTUP,GEOSUB, BARK\_EQ, BARK\_SPN);

ERR := D - DIBUP;

IF (ERR<0.0) THEN

TOOHIGH := HTUP;

END IF;

ADJUST := - ERR/SLOPE;

HTUP := HTUP+ADJUST;

IF(HTUP>THT) THEN

HTUP := (HTUP-ADJUST+THT)/2;

END IF;

IF(HTUP<0.0) THEN

HTUP := (HTUP-ADJUST)/2;

END IF;

RH := HTUP/THT;

IF(ABS(ADJUST)>TOL\*THT OR ABS(ERR)>EPSILON) THEN

GOTO L20;

END IF;

IF(SLOPE>0) THEN

IF(IBREAK<2) THEN

IBREAK := IBREAK+1;

IF(IBREAK = 1) THEN

HTUP := 0.8\*HTUP;

END IF;

IF(IBREAK=2) THEN

HTUP := HTUP +(THT-HTUP)\*0.25;

END IF;

GOTO L110;

END IF;

END IF;

-- NOTE: 1 STEP HAS BEEN TAKEN AFTER CONVERGENCE CRITERIA MET.

INEEDSL := 0;

RETURN HTUP;

-- MODIFIED BISECTION APPROACH (slightly easier convergence criteria)

<<L200>>

INEEDSL := 0;

ITER := 0;

HHIGH := TOOHIGH;

HLOW := TOOLOW;

D1 := DIB(HHIGH,SPN,DBH,THT,GEOSUB, TAPEQ, BARK\_EQ, BARK\_SPN);

D2 := DIB(HLOW,SPN,DBH,THT,GEOSUB, TAPEQ, BARK\_EQ, BARK\_SPN);

EHIGH := D1-DIBUP;

ELOW := D2-DIBUP;

IF(ELOW\*EHIGH>0) THEN

HTUP := 0;

RETURN HTUP;

END IF;

EPS := EPSILON\*2.0;

<<L210>>

HTRY := (HHIGH+HLOW)/2;

DTRY := DIB(HTRY,SPN,DBH,THT,GEOSUB, TAPEQ, BARK\_EQ, BARK\_SPN);

ERR := DTRY-DIBUP;

IF(ABS(ERR)<EPS) THEN

HTUP := HTRY;

RETURN HTUP;

END IF;

IF(ITER>40) THEN

HTUP := HTRY;

RETURN HTUP;

END IF;

ITER := ITER+1;

IF(ERR>0.0) THEN

HLOW := HTRY;

ELSE

HHIGH := HTRY;

END IF;

GOTO L210;

RETURN HTUP;

END;

----------------------------------------------------------------------------------

--Estimate Dib at any height (any species, any method)

--Optionally estimate slope d(DIB)/d(H)

--From SF\_DS and SF\_2PT

--H input R\*4 Section height

--DIB output R\*4 Diameter inside bark

FUNCTION DIB(HTUP IN NUMBER,

SPN IN NUMBER,

DBH IN NUMBER,

THT IN NUMBER,

GEO IN VARCHAR2 := NULL,

EQN IN VARCHAR2 := NULL,

BARK\_EQ IN VARCHAR2 := NULL,

BARK\_SPN IN NUMBER := NULL) RETURN NUMBER IS

DIB NUMBER;

SL1 NUMBER;

DOBUP NUMBER;

DIBUP NUMBER := 0;

DBT\_UP NUMBER;

BEGIN

GEOSUB := GEO;

TAPEQ := EQN;

--Reminder: GEOSUB is tied to volume taper so the volume package must be run first is geosub

--is not explicitly provided. When a stand alone TAPER package is created there will be an

--issue on whether to have a single allowabe taper equation ro leave it linked to whatever type

--of volume was run last.

IF NOT (L\_SPN=SPN AND L\_DBH=DBH AND L\_HT=THT

AND NVL(L\_TAPEQ,'YY')=NVL(TAPEQ,'YY')

AND NVL(L\_BRKEQ,'YY')=NVL(BRKEQ,'YY')

AND NVL(L\_GEOSUB,'YY')=NVL(GEOSUB,'YY')) THEN

IF (TAPEQ='5') THEN

SETUP\_W(SPN, DBH, THT, GEOSUB, BARK\_EQ, BARK\_SPN);

ELSIF (TAPEQ='6') THEN

SETUP\_I(SPN,DBH,THT,GEOSUB, BARK\_EQ, BARK\_SPN);

ELSIF (TAPEQ='9') THEN

SETUP\_B(SPN,DBH,THT,GEOSUB, BARK\_EQ, BARK\_SPN);

ELSIF (TAPEQ='10') THEN

SETUP\_R(SPN,DBH,THT,GEOSUB, BARK\_EQ, BARK\_SPN);

ELSIF (TAPEQ='37') THEN

SETUP\_A(SPN,DBH,THT,GEOSUB, BARK\_EQ, BARK\_SPN);

END IF;

L\_SPN := SPN;

L\_DBH := DBH;

L\_HT := THT;

L\_TAPEQ := TAPEQ;

L\_GEOSUB := GEOSUB;

L\_BRKEQ := BRKEQ;

END IF;

IF(HTUP>THT) THEN

IF(INEEDSL=1) THEN

SLOPE := -1;

END IF;

ELSE

RH := HTUP/THT;

IF(TAPEQ='9') THEN -- Black Hills using outside bark projections

DOBUP := YHAT(THT,HTUP);

DBT\_UP := BRK\_B(HTUP,SPN,DBH,THT,DOBUP);

DIBUP := DOBUP-DBT\_UP;

ELSIF(TAPEQ='10' ) THEN -- Rocky Mt. using outside bark projections

DOBUP := YHAT(THT,HTUP);

DBT\_UP := BRK\_R(HTUP,SPN,DBH,THT,DOBUP);

DIBUP := DOBUP-DBT\_UP;

ELSE -- All Others YHAT returns DIB

DIBUP := YHAT(THT,HTUP);

END IF;

SL1 := SLOPE;

DIB := DIBUP;

END IF;

RETURN DIB;

END;

---------------------------------------------------------------------------------------------

--Temporaly put flewellig eqs here untill able to put bark eqn as input in main taper program

FUNCTION PICK\_DIB(TAPEQ IN VARCHAR2,

SPN IN NUMBER,

DBH IN NUMBER,

THT IN NUMBER,

HTUP IN NUMBER,

GEOSUB IN VARCHAR2 := NULL,

BARK\_EQ IN VARCHAR2 := NULL,

BARK\_SPN IN NUMBER := NULL,

D2 IN NUMBER := 0,

HT2 IN NUMBER := 0,

DBTBH IN NUMBER := 0,

DBT2 IN NUMBER := 0) RETURN NUMBER IS

DIA NUMBER := 0;

DUMMY NUMBER := 0;

SPC VARCHAR2(8);

BEGIN

IF (TAPEQ='5') THEN -- Flewelling 2 point NW

DIA := DIB\_W(HTUP,SPN,DBH,THT,GEOSUB, BARK\_EQ, BARK\_SPN);

ELSIF (TAPEQ='6') THEN -- Flewelling 2 point INLAND

DIA := DIB\_I(HTUP,SPN,DBH,THT,GEOSUB, BARK\_EQ, BARK\_SPN);

ELSIF (TAPEQ='9') THEN -- Flewelling 2 point BLACK HILLS, DOB substitution

DIA := DIB\_B(HTUP,SPN,DBH,THT,GEOSUB, BARK\_EQ, BARK\_SPN);

ELSIF (TAPEQ='10') THEN -- Flewelling 2 point ROCKY MTS, DOB substitution

DIA := DIB\_R(HTUP,SPN,DBH,THT,GEOSUB, BARK\_EQ, BARK\_SPN);

END IF;

RETURN DIA;

END;

----------------------------------------------------------------------------------

--West Coast

--TAPEQ='5'

FUNCTION DIB\_W(HTUP IN NUMBER,

SPN IN NUMBER,

DBH IN NUMBER,

THT IN NUMBER,

GEO IN VARCHAR2 := NULL,

BARK\_EQ IN VARCHAR2 := NULL,

BARK\_SPN IN NUMBER := NULL) RETURN NUMBER IS

DIB\_UP NUMBER;

BEGIN

DIB\_UP := DIB(HTUP, SPN, DBH, THT, GEO, '5', BARK\_EQ, BARK\_SPN);

RETURN DIB\_UP;

END;

---------------------------------------------------------------------------

--for the two-point system, calculates cubic volume to any height

--From file ALLCODE.FOR function SF\_CUMV2

FUNCTION SF\_CUMV2(THT IN NUMBER, TOPHT IN NUMBER) RETURN NUMBER IS

PI4 NUMBER := 0.785398163;

VOL NUMBER;

AI NUMBER;

X NUMBER;

XS NUMBER;

X2 NUMBER;

X3 NUMBER;

X4 NUMBER;

X5 NUMBER;

X6 NUMBER;

X7 NUMBER;

BI NUMBER;

AREA1 NUMBER;

AREA2 NUMBER;

VTIP NUMBER;

BEGIN

RH := TOPHT/THT;

IF(RH<=RHI1) THEN

X := RH/RHI1;

XS := 1.0-X;

AI := FYSQA(XS,FALSE);

VOL := (AI) \* PI4 \* RHI1;

ELSIF (RH<RHI2) THEN

AREA1 := (E1+E2\*RHI1)\*\*2;

AREA2 := (E1+E2\*RH )\*\*2;

VOL := (RH-RHI1) \* (AREA1+AREA2+ SQRT(AREA1\*AREA2))/3.0\*PI4;

VOL := VOL + VA1;

ELSIF (RH<RHC) THEN

X := (RH-RHI2)/(RHC-RHI2);

XS := X;

BI := FYSQB(XS);

VOL := (BI-BI0) \* PI4 \* (RHC-RHI2);

VOL := VOL + VA1 + VA2;

ELSIF (RH<1) THEN

X := (1-RH)/(1-RHC);

X3 := X\*X\*X;

X4 := X3\*X;

X5 := X4\*X;

X6 := X5\*X;

X7 := X6\*X;

VTIP := PI4\*( C2\*C2/3\*X3 + C1\*C2/4\*X4 +C1\*(C1/20-C2/15)\*X5

-C1\*C1/36\*X6 +C1\*C1/252\*X7)\*(1-RHC);

VOL := VTOT - VTIP;

ELSE

VOL:=VTOT;

END IF;

VOL := VOL\*VFACT;

RETURN VOL;

END;

---------------------------------------------------------------------------

--Evaluate the integral of y\*\*2 at point t t=0 at ground, 1 at inflection

--The FIRST call to this function must have argument t = 0

FUNCTION FYSQA(T IN NUMBER, FIRST BOOLEAN) RETURN NUMBER IS

X NUMBER;

Z NUMBER;

YZINT NUMBER;

Y2INT NUMBER;

FYSQA NUMBER;

BEGIN

-- Note: C's in this routine are UNRELATED to upper segment taper coe's

IF(T<0.0) THEN

FYSQA:=0.0;

RETURN FYSQA;

END IF;

IF(FIRST) THEN

CE0 := A0 + A3\*A4 +1.5\*A2 + A1\*A3\*\*3.0;

CE1 := -A3\*A4 -2.0\*A2 -3.0\*A1\*A3\*\*3.0;

CE2 := A2/2.0 + 3.0\*A1\*A3\*\*3.0;

CE3 := -A1\*A3\*\*3.0;

END IF;

Z := 1.0 - T/A3;

Y2INT := (CE0\*CE0) \*Z

+(2.0\*CE0\*CE1) \*Z\*\*2.0 /2.0

+(2.0\*CE0\*CE2 + CE1\*CE1) \* Z\*\*3.0 /3.0

+(2.0\*(CE0\*CE3+ CE1\*CE2)) \* Z\*\*4.0 /4.0

+(2.0\*CE1\*CE3 + CE2\*CE2) \* Z\*\*5.0 /5.0

+(2.0\*CE2\*CE3) \* Z\*\*6.0 /6.0

+(CE3\*CE3) \* Z\*\*7.0 /7.0

+ 2.0 \* A2 \* CE0 \*(Z\* LN(Z) - Z)

+ 2.0 \* A2 \* CE1 \*(Z\*\*2 /2.0 \*LN(Z) - Z\*\*2.0/4.0)

+ 2.0 \* A2 \* CE2 \*(Z\*\*3 /3.0 \*LN(Z) - Z\*\*3.0/9.0)

+ 2.0 \* A2 \* CE3 \*(Z\*\*4 /4.0 \*LN(Z) - Z\*\*4.0/16.0)

+ A2\*A2 \* ( Z\*LN(Z)\*LN(Z) - 2.0\*Z\*LN(Z) + 2.0\*Z);

YZINT := Y2INT\*(A3);

IF(FIRST) THEN

BAAS := YZINT;

END IF;

FYSQA := YZINT - BAAS;

RETURN FYSQA;

END;

---------------------------------------------------------------------------

--Find cubic volume below any top diameter or broken top

--From file ALLCODE.FOR function SF\_CV

--Currently works only with west coast version of Flewelling Taper

--Need to consider DOB verses DIB methods for Rocky Mt equations

--and debug Inland equations when time permitts

FUNCTION SF\_CV\_BREAK(SPN IN NUMBER,

DBH IN NUMBER,

THT IN NUMBER,

HT\_BREAK IN NUMBER,

TOP IN NUMBER,

STUMP IN NUMBER,

GEO IN VARCHAR2 := NULL,

EQN IN VARCHAR2 := NULL,

BARK\_EQ IN VARCHAR2 := NULL,

BARK\_SPN IN NUMBER := NULL) RETURN NUMBER IS

VOL NUMBER;

TOPHT NUMBER;

STUMPVOL NUMBER;

BEGIN

GEOSUB := GEO;

TAPEQ := EQN;

IF(TAPEQ='5') THEN -- Flewelling 2 point NW

SETUP\_W(SPN,DBH,THT,GEOSUB, BARK\_EQ, BARK\_SPN);

ELSIF(TAPEQ='6') THEN -- Flewelling 2 point INLAND

SETUP\_I(SPN,DBH,THT,GEOSUB, BARK\_EQ, BARK\_SPN);

ELSIF (TAPEQ='7') THEN -- Flewelling 2 point BLACK HILLS

SETUP\_B(SPN,DBH,THT,GEOSUB, BARK\_EQ, BARK\_SPN);

ELSIF (TAPEQ='8') THEN -- Flewelling 2 point ROCKY MTS

SETUP\_R(SPN,DBH,THT,GEOSUB, BARK\_EQ, BARK\_SPN);

ELSIF (TAPEQ='9') THEN -- Flewelling 2 point BLACK HILLS, DOB substitution

SETUP\_B(SPN,DBH,THT,GEOSUB, BARK\_EQ, BARK\_SPN);

ELSIF (TAPEQ='10') THEN -- Flewelling 2 point ROCKY MTS, DOB substitution

SETUP\_R(SPN,DBH,THT,GEOSUB, BARK\_EQ, BARK\_SPN);

END IF;

VOL := SF\_VOL2(THT);

IF(TOP>0) THEN

TOPHT := HT\_AT\_DIB(SPN,DBH,THT,TOP,GEOSUB, TAPEQ, BARK\_EQ, BARK\_SPN);

IF(HT\_BREAK<TOPHT) THEN

TOPHT := HT\_BREAK;

END IF;

VOL := SF\_CUMV2(THT, TOPHT);

END IF;

IF(STUMP>0) THEN

STUMPVOL := SF\_CUMV2(THT,STUMP);

VOL := VOL-STUMPVOL;

END IF;

RETURN VOL;

END;