MESAVAGE AND GIRARD FORM CLASS TAPER FUNCTIONS DERIVED FROM PROFILE EQUATIONS

Thomas G. Matney and Emily B. Schultz[1]

Abstract—The Mesavage and Girard (1946) average upper-log taper tables remain a favorite way of estimating tree bole volume because they only require the measurement of merchantable (useable) height to an indefinite top diameter limit. For the direct application of profile equations, height must be measured to a definite top diameter limit, and this makes the collection of data more arduous, particularly in hardwoods, since in many cases both the height to the top diameter limit (stopper top) and the stopper top diameter must be measured. In this paper, Girard form class is used to convert a regular profile function into a Mesavage and Girard type taper model requiring only the measurement of merchantable height. The advantages of the new approach include: 1) the derived model's upper log tapers are species specific, 2) the model describes the taper in the butt log, and 3) pulpwood and sawtimber trees can use the same model and height measure. Like the Mesavage and Girard taper tables, the converted profile equations have the disadvantage of requiring an average form class, preferably by dbh class. The source code for the profile conversion algorithm in C++, or BASIC can be downloaded from www.timbercruise.com, www.cfr.msstate.edu, or by contacting the authors. The program is short and can be quickly embedded in any cruise program implementing profile functions.

INTRODUCTION

Tree volume tables based on Girard form class (Avery and Burkhart 1994), and the upper-log taper tables for Southern conifers and Eastern hardwoods (Mesavage and Girard 1946) are still in wide spread use throughout the entire Eastern US. This use continues in spite of the fact that for most commercial tree species, stand origins, and geographic regions, highly accurate tree profile functions are available. There are two primary reasons for the continued use of these out-of-date and flawed tables. These are: 1) many timber cruisers have relied on volumes derived from the taper tables for years and are very reluctant to change volume estimation systems even though they recognize that by changing to species specific profile functions the accuracy and consistency of tree volume estimates could be improved dramatically, and 2) height measurement in the field is quicker because heights are measured to an indefinite top (usable or merchantable). The heights of trees assigned to a profile function must be measured to a definite top, and when the tree is not merchantable to the targeted definite top, the stopper top must be estimated and recorded. The old taper tables are out-of-date because the character of timber has changed significantly due to genetics and management practices. They are also seriously flawed because they are not species, stand origin, or geographic region specific and make the assumption that upper-log tapers are independent of form class and provide no taper rates in the first log.

This paper describes a procedure developed to effectively convert a tree profile function into a form that requires only the field measurement of height to an indefinite top diameter. The procedure is a simple numerical algorithm that calculates the total height of a tree from dbh, form class height, and form class. Once this procedure is included in a timber cruise program, users of the program can take advantage of the accuracy of tree profile functions without increasing the complexity of field measurements.

METHODS

Profile equations are algebraically complex functions that predict diameter at any height (h) above ground from diameter breast height (dbh), total height (H), and the height h of the desired diameter. Outside bark profile equations (O) predict outside bark diameter (dob), and inside bark profile functions (I) predict inside bark diameter (dib). These functions can be written abstractly as dob = O(h, dbh, H) and dib = I(h, dbh, H).

---

[1] Thomas G. Matney, Professor, and Emily B. Schultz, Associate Professor, Mississippi State University, Forest and Wildlife Research Center, Mississippi State, MS 39762.

Girard form class (*F*) is defined as the percentage the scaling diameter (*sd*) of the first 16 foot log is of *dbh*.

$$F = 100 \frac{sd}{dbh} \tag{1}$$

Traditionally the assumption has been made that stump height is 1 foot, and log trim is 0.3 feet, and thus the height above ground (*g*) of the scaling diameter of the first 16 foot log is 17.3 feet. Stump heights are now around 0.5 feet, and the form class height could be set at 16.8 feet. Even though form class calculated at 16.8 feet would differ little from form class calculated at 17.3 feet, for current tree volume estimates to be comparable with past estimates form class height should be kept at 17.3 feet. However, if a departure from the height of 17.3 feet presents no problems, a form class height measurement point higher up on the bole will improve the accuracy of the profile function calculated from *dbh* and *F*. That is, when *dbh* and *F* are used to determine *H*, the closer the form class height *g* is to *H* the better the estimate of *H* will be.

If the *dbh* and total height *H* of a tree are known, the scaling diameter of the first log *sd* = dib = *I*(*h, dbh, H*) = *I*(*g, dbh, H*). The scaling diameter of the first log is the value of the inside bark diameter at the form class height g (i.e., *h* = *g*). Thus, the form class equation (1) can be rewritten as

$$F = 100 \frac{I(g, dbh, H)}{dbh} \tag{2}$$

Equation (2) can be solved for the total height *H* using a number of procedures for solving for roots of an equation such as the bisection, Newton, secant, or regula falsi fixed point iterations (Burden and others 1981). Finding the value of *H* satisfying equation (2) by a fixed point iteration requires that the equation be written in the following form:

$$f(x) = f(x = H) = 100 \frac{I(g, dbh, x = H)}{dbh} - F = 0 \tag{3}$$

where

x = *H* has been used to designate that *x* and *H* are equivalent. In the case *F* is expressed as a ratio of outside bark diameters, *I*(*h, dbh, H*) equation (3) must be replaced by O(*h, dbh, H*).

The root (*x** = *H**) of the nonlinear equation (3) is the total height of the tree with the specified value of *dbh, g,* and *F*. Any of the above fixed point iterations can be used to find the root *H**, but the method of bisection, while being the slowest to converge to a solution, always converges. The other methods often diverge unless the initial guess of the solution falls inside an often narrow range of convergence for the method. The bisection method can be stated in algorithmic form:

Let *f*(*x*) be a piece-wise continuous function on a closed interval [A, B] with *f*(A)*f*(B) ≤ 0, then use the following steps to find the root of the equation *f*(*x*) = 0 having a specified maximum allowable error of MAXE.

1. Compute the mid-point M = (A + B)/2 of the interval [A, B]
2. Compute the maximum error of the iteration of the best estimate (M) of the equation's root as E = (B − A)/2
3. If E ≤ MAXE, the root of the equation is M. Otherwise
  a. If *f*(A)*f*(M) ≤ 0, set B = M. Otherwise, set A = M
  b. Repeat steps 1 to 3 until the error condition in step 3 is satisfied

EXAMPLE

To illustrate the solution procedure, consider the cubic foot volume of a cherry bark oak (*Quercus falcata* var. *pagodi folia* Ell.) tree with a *dbh* = 18 inches, *F* = 88, and a usable (merchantable) height of 3.5 16 foot logs. The cherry bark oak profile function used is presented in Matney and others 1985. For the example tree, the root of equation (3) is between A = 117.3 and B = 127.3. The solution for total height is between A = 117.3 and B = 127.3 because *f*(117.3) < 0 and f(127.3) > 0. The solution range is easy to find. Starting with a total height of 17.3 feet, the total height (*H*) is incremented by a fixed amount (INC) until f(*H*) ≥ 0. B is set to *H*, and A is set to B – INC. The INC for the example was 10 feet, but other increment values could be used. Table 1 summarizes the bisection of the interval [A, B] = [117.3, 127.3] with a maximum allowable error of 0.25 feet. The final total height estimate of 119.33 feet has an actual maximum error of 0.15625 which is less than 0.25 feet. Figure 1 presents the graph of equation (3) showing that the function is monotonically increasing and will cross 0 for some total height value exceeding the form class height.

Table 1—Determination of the total height of a tree with a d.b.h. of 18 inches and a form class of 88 using the method of bisection for solving for roots of nonlinear equations

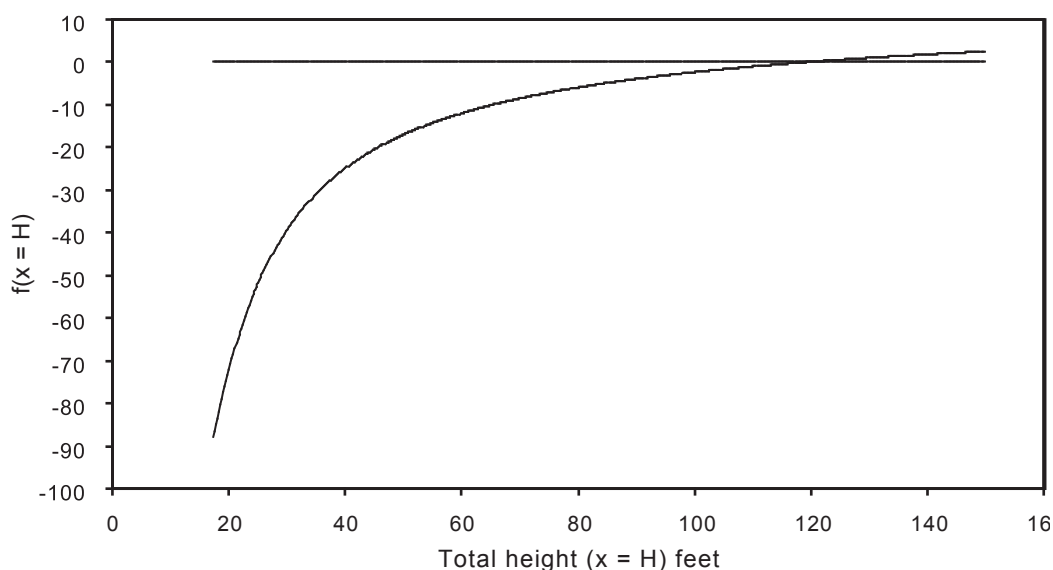| n	A	B	M = (A + B)/2	f(A)	f(B)	f(M)	Error = (B – A)/2
0	117.30	127.30	122.30	-0.21114	0.76453	0.29593	5.00000
1	117.30	122.30	119.80	-0.21114	0.29593	0.04751	2.50000
2	117.30	119.80	118.55	-0.21114	0.04751	-0.08050	1.25000
3	118.55	119.80	119.18	-0.08050	0.04751	-0.01617	0.62500
4	119.18	119.80	119.49	-0.01617	0.04751	0.01575	0.31250
5	119.18	119.49	119.33	-0.01617	0.01575	-0.00019	0.15625

Figure 1—Graph of the function f(x = H) for a cherry bark oak tree with dbh, form class height, and form class of 18 inches, 17.3 feet, and 88, respectively.

Given that total height is now known, the volume from stump height to any indefinite top limit $\leq H^* = 119.33$ feet is easily calculated. For example, the cubic foot ib and ob volumes of a 3.5 log tree are 116.33 and 129.82 cubic feet. The cubic volume computations were approximated by Smalian's cubic volume rule using a 4-foot bolt length. The initial values for A and B are readily obtained by setting $A = 17.3$, incrementing A by some value (INC) such as 10 until $F(A) \geq 0$, and then setting $B = A$ and $A = B - INC$.

C++ SOURCE CODE

The C++ source code for the bisection routine (HtFromDbhFC) for calculating total height from dbh and form class is presented in the appendix. The main program uses the dbh and form class from the Example to illustrate the computation of the total height and ib and ob cubic foot volume of the 3.5 log merchantable length. A Microsoft Visual C++ dialog project implementing the code can be directly accessed at the link www.timbercruise.com/Downloads/Software/HtFromDbhFormClass.zip or by visiting the Web site www.cfr.msstate.edu to obtain a. The links are case sensitive.

DISCUSSION

This procedure forces the profile function to have the desired form class. Like all procedures the algorithm will fail for illogical conditions. If the form class is too large, the scaling diameter of the first log will be greater than or equal to dbh inside bark and the procedure will fail to converge. Before calling the procedure the user should make sure that the ratio of scaling diameter to dbh inside bark is not too close to one. Otherwise, the procedure will converge provided the maximum height input is set sufficiently large. If the form class is too large, the routine will converge but to a ridiculously high value. If the form class is too low, the estimated total height may be less than the measured merchantable height. In either case, the specified form class for the tree will probably be erroneous.

The commercially available cruise program Timber Cruise (TCruise) implements the algorithm as an optional procedure for using profile functions. With the option selected, users can simply measure merchantable heights while improving their volume estimates by using a profile function that is specific to the tree species being measured. A trial version of TCruise is available at www.timbercruise.com.

In the case that the profile function is a function of Girard form class, a form class height other then 17.3 feet must be selected. For the profile functions inside TCruise published by Clark and others 1991 involving Girard form class, the program sets form class height to the top of the second log (i.e., 33.6 feet).

LITERATURE CITED

Avery, T.E.; Burkhart, H.E. 1994. Forest measurements. 4th Ed. McGraw-Hill, Inc., NY,NY. 408 pp.

Burden, R.L.; Douglas, J.D.; Reynolds, A.C.. 1981. Numerical analysis. 2nd Ed. Prindle, Weber, and Schmidt, Boston, MA. 598 pp.

Clark, A.; Souter, R.A.; Schlaegel, B.E. 1991. Stem profile equations for southern tree species. Department of Agriculture, Forest Service, Washington, DC., South Eastern Forest Experiment Station Research. Paper SE-282. 113 pp.

Matney, T.G.; Hodges, J.D.; Sullivan, A.D.; Ledbetter, J.R. 1985. Tree profile and volume ratio equations for sweetgum and cherrybark oak trees. Southern Journal of Applied Forestry 9(10):222-227.

Mesavage, C.; Girard, J.W..1946. Tables for estimating board foot volume of timber. Department of Agriculture, Forest Service, Washington, DC. 94 pp.

APPENDIX

C++ source code in the example project using the bisection fixed point iteration method for determining total tree height from dbh and form class

```cpp
//
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

int main(int argc, char* argv[])

{
        // Outputs
        double csErrorMsg;
        double dCFVIB;
        double dCFVOB;
        double dTOT_HT;
        int nError;

        // Inputs
        double dDBH = 15.0;
        double dFC = 80.0;
        double dFC_HT = 17.3;
        double dGFC = 0.0;
        double dMAX_HT = 275.0;
        double dMHT = 3.5;

        double dSHt = 1.00;
        double dBoltLen = 4.0;

        // Calculate total tree height
        dTOT_HT = HtFromDbhFC(dDBH, dFC, dFC_HT, dMAX_HT,
                                        dGFC, &CBO_DIB, nError);

        // Convert merchantable height in logs to feet
        if(dMHT < 8.0)
        {
                dMHT = 16.0*dMHT;
        }

        // Convert to height above ground
        dMHT += dSHt; // Add stump height

        // Compute ob and ib cubic foot volume between the heights
        // of dSHt and dMHT
        dCFVIB = GetCubicVolume(dDBH, dTOT_HT, dGFC, &CBO_DIB,
                                        dSHt, dMHT, dBoltLen);

        dCFVOB = GetCubicVolume(dDBH, dTOT_HT, dGFC, &CBO_DOB,
                                        dSHt, dMHT, dBoltLen);

        }
```

```
                csErrorMsg = "";
                if(nError != 0)
                {
                        csErrorMsg = "Illogical input";
                }

                return 0;
}
*/

// Function for calculating total tree height from dbh, form class, and
// form class height
//
// Input
//
// dDbh        = diameter breast high
// dFC         = Girard type form class
// dFCHt       = form class height
// dMaxHt      = Maximum allowed total height
// dGFC        = Girard form class. This is for profile functions that
//                  use Girard FC. In this case the form class height dFCHt
//                  must be greater then 17.3 feet.
// pfIbProfile = pointer the the inside bark profile function
//
// Output
//
// Total height = return value
// nError       = error code (0= no error, 1 = illogical inputs)
//


double HtFromDbhFC(double dDbh, double dFC, double dFCHt, double dMaxHt,
                                     double dGFC, PROFILE pfIbProfile, int& nError)
{
   double dPHt, dTHt, dHtInc, dHtLower, dHtMid, dHtUpper, dFCTest, dMaxError;

   nError = 0;
        dPHt = dFCHt;
        dTHt = dFCHt;
   dHtUpper = dFCHt;
        dHtInc = (dMaxHt - dFCHt)/20.0;

   dFC = dFC/100.0F;
   while((dHtUpper = dHtUpper + dHtInc) < dMaxHt)
   {
      dTHt = dHtUpper;
      dFCTest = (*pfIbProfile)(dDbh, dTHt, dGFC, dPHt)/dDbh;
      if(dFCTest >= dFC)break;
   }

   if(dHtUpper > dMaxHt)
   {
      nError = 1; // Could not calculate a total ht for the FC. Check Dbh, Hm, and FC for logic
```

```
      return(dMaxHt);
    }

    dHtLower = dHtUpper - dHtInc;
    dMaxError = dHtUpper - dHtLower;
    dHtMid = (dHtLower + dHtUpper)/2.0F;

    while((dMaxError = dMaxError/2.0F) > 0.25F)
    {
      dTHt = dHtMid;
      dFCTest = (*pfIbProfile)(dDbh, dTHt, dGFC, dPHt)/dDbh;

      if(dFCTest <= dFC)
      {
        dHtLower = dHtMid;
      }
       else
      {
        dHtUpper = dHtMid;
      }

    dHtMid=(dHtLower + dHtUpper)/2.0F;
  }

  return(dHtMid);
}

// Cherry bark oak outside bark profile function
//
// Input
//
//  dDbh = diameter breast high
//  dTHt = total tree height
//  dPHt = height of output diameter
//  dGFC = Girard form class. This is for profile functions that
//          use Girard FC.
//
// Output
//
//  Diameter outside bark at the height dPHt = return value
//

double CBO_DOB(double dDbh, double dTHt, double dGFC, double dPHt)
{
    static double dPar[3] = {0.779945, 0.126221, 1.0412989};
         double dDOB, dWorkA, dWorkB, dWorkC;

         if( (dPHt < dTHt) && (dPHt > 0.))
         {
                 dWorkA =dPar[0]*(pow(dTHt, dPar[1]));
                 dWorkB =-tan(dWorkA)/log(1.0 - pow((1.0-4.5/dTHt), dPar[2]));
                 dWorkA =dDbh/dWorkA;
                 dWorkC = -log(1.0 - pow((1.0 - dPHt/dTHt), dPar[2]))*dWorkB;
```

```
                        dDOB = dWorkA*atan(dWorkC);
              }
          else
          {
                  if(dPHt >= dTHt)
                  {
                          dDOB = 0.0;
                  }
                   else
                  {
                     dWorkA = dPar[0]*(pow(dTHt, dPar[1]));
                     dWorkA = dDbh/dWorkA;
                     dDOB = 1.570796327*dWorkA;
                  }
          }
      return (dDOB);
}

// Cherry bark oak inside bark profile function
//
// Input
//
// dDbh = diameter breast high
//  dTHt = total tree height
//  dGFC = Girard form class. This is for profile functions that
//          use Girard FC.
//
// Output
//
//  Diameter inside bark at the height dPHt = return value
//

double CBO_DIB(double dDbh, double dTHt, double dGFC, double dPHt)
{
        static double dPar[5] = {0.767290, 0.130027, 1.0560755F, 0.000,0.945};
   double dDIB, dWorkA, dWorkB, dWorkC, dWorkD;

        if( (dPHt < dTHt) && (dPHt > 0.))
        {
                dWorkA = dPar[3] + dPar[4]*dDbh;
                if(dWorkA < 0.0)dWorkA=0.0;
                dWorkB = dPar[0]*pow(dTHt, dPar[1]);
                dWorkC = -tan(dWorkB)/log(1.0 - pow((1.0 - 4.5/dTHt),dPar[2]));
                dWorkB = dWorkA/dWorkB;
                dWorkD = -log(1.0 - pow((1.0 - dPHt/dTHt), dPar[2]))*dWorkC;
                dDIB = dWorkB*atan(dWorkD);
        }
     else
        {
                if(dPHt >= dTHt)
                {
                        dDIB=0.;
```

```
                }
                else
                {
                 dWorkA = dPar[3] + dPar[4]*dDbh;
                 if(dWorkA < 0.0)dWorkA=0.0;
                 dWorkB = dPar[0]*(pow(dTHt, dPar[1]));
                 dWorkB = dWorkA/dWorkB;
                 dDIB = 1.570796327*dWorkB;
                }
        }
        return (dDIB);
}

// Cubic foot volume calculation function
//
// Input
//
// dDbh = diameter breast high
/ dTHt  = total tree height
//  dGFC = Girard form class. This is for profile functions that
//          use Girard FC.
// dBeginH = height of the large end of log
//  dEndH = height of the small end of log
//dBoltLen = bolt length for Smalian's cubic volume rule
// pfProfile = pointer to the profile function
//
// Output
//
// Cubic foot volume between the heights dBeginH and dEndH  = return value
//

double GetCubicVolume(double dDbh, double dTHt, double dGFC, PROFILE
pfProfile,
                                double dBeginH, double dEndH, double dBoltLen)
{

        double dPHt = dBeginH;
        double dDLE = (*pfProfile)(dDbh, dTHt, dGFC, dPHt);
        double dDSE;

        double dCFV = 0;

        while((dPHt = dPHt + dBoltLen) < dEndH)
        {
                dDSE = (*pfProfile)(dDbh, dTHt, dGFC, dPHt);
                dCFV += 0.002727*(dDSE*dDSE + dDLE*dDLE)*dBoltLen;
                dDLE = dDSE;
        }

        dBoltLen = dTHt - (dPHt - dBoltLen);
        dDSE = (*pfProfile)(dDbh, dTHt, dGFC, dEndH);
        dCFV += 0.002727*(dDSE*dDSE + dDLE*dDLE)*dBoltLen;

    return dCFV;
}
```