Category A

It's Getting Hot in Here

Loretta Brown, Junior
David Schiowitz, Junior
Advanced Computer Studies
Silver High School, Silver City, NM

Adventures in Supercomputing
Team Number: 068
Teachers: Mrs Larisch
Ms Trujillo
Mentor: Bob Schiowitz

Acknowledgments

The authors would like to acknowledge the following persons for their help in the selection of the topic, development and execution of the program, and board design:

- Mrs. Peggy Larisch, Teacher, Silver High School, Advanced Computers
- Ms. Nadie Trujillo, Teacher, Silver High School, Biology
- Mr. Berry Estes, Nuclear Engineer (Retired), Sandia Natural Laboratories
- Mr. Bob Schiowitz, South Zone Archeologist, U.S. Forest Service, Silver City, NM
- Ms. Kelly Trinkle, Student, Silver High School

Table of Contents

1.0 Introduction

1.1 Purpose

The purpose of this project is to develop a computer program that is

- capable of investigating specific input parameters associated with and characteristic of a “prescribed burn” operation, and
- present recommended responses, that is, appropriate managed actions.

The program shall be capable of identifying characteristics of the “prescribed burn” that, even though these input parameters are variable in the overall equation can be controlled to a certain extent thereby providing on-scene operations personnel with an enhanced level of knowledge of how the “prescribed burn” may possibly behave taking into consideration the identifiable input conditions. A “prescribed burn” operation is defined as follows:

> “a planned burning of flammable materials (such as dense undergrowth, dead trees, old trees, and excessive ground cover) in a specified area and under controlled conditions.”

The controlled conditions are represented in the program as the identifiable input parameters and are actually variable quantities, as noted above, that have to be considered as part of the “burn” problem. The program evaluates the aggregate set of input parameters and, based on these specified conditions, calculated recommended burn actions commensurate with the existing environmental conditions foe the removal of the undesired materials. The on-scene operations staff can utilize the program to determine the execution process for taking action on the prescribed burn.

1.2 Scope

The program is created to establish a well-developed model that predicts the behavior of a prescribed ground burn. The program uses the predefined fuel models, the averaged percentage values of the wind direction and speed and the slope of the terrain as inputs to establish a prediction on the scorch height, flame length,

1.3 Computer Program

The computer program was created using the programming language of C. C is one of the best choices for this particular project because it is able to store and calculate a large amount of data which is used in the complex equations. Many forest service officers operating prescribed burns require an easy to use and object oriented program for fast predictions on the possible behavior of the fire. The program fireLib(the modified program) is difficult to comprehend and use. It requires multiple user manuals and countless hours to understand what input the program is requesting. FireLib consists of libraries of data (formulas, fuel models, ect). The authors created FireSim to utilize the information stored in these libraries. The program uses the formulas and standard data to calculate the desired outputs. This makes the program more user-friendly and applicable to the presented problem.

2.0 Project Proposal

2.1 Description of Project

The program takes the slope of terrain, composition of the fuel, wind speed, and climate (such as temperature and rain fall) to produce a prediction on how the fire will behave: the temperature the fire will obtain, the total amount of the area burned, and the size of the flame. Determining the way a wildfire will behave in advance, giving the firefighters a significant advantage over the fire. The crew will then act proactively rather than being reactive. They will see the fire act before it will and will be able to construct better, more effective ways to stop the fire rather than only slowing it.

For centuries people have been fighting wildfires using many different methods. Society has greatly advanced its technology in fighting fires due to the growing threat of them. Wildfires have been recently devastating the western part of United States. Some places that have burned are: Show Low, Arizona, Durango, Colorado, and Los Alamos, New Mexico. A wildfire is a sweeping and destructive conflagration especially in wilderness and rural areas. This program is constructed to evaluate the effect of wildfires burning in woodland areas; these are commonly referred to as forest fires. The program takes the moisture content of the trees, tree type, tree density, slope of terrain, composition of the forest litter (i.e. pine cones, pine needles, and leaves), wind speed, and climate(such as temperature and rain fall) to produce a prediction on how the fire will behave, the temperature the fire will obtain, the velocity with which the fire burns through the area, the total amount of the burnt area, and the size of the flame.

3.0 Analytical Methodology

3.1 Mathematical Bases

For predicating the behavior of a prescribed burn many analytical models and calculation techniques are used. Many of these calculations utilize formulas developed by the forest service prior to the development of a prediction fire behavior program. The following formula calculates the rate at which a fire spreads through a uniform fuel array that may contain fuel particles of mixed sizes. The formula (Rothermel 1972) is referred to as the Fire Spread Model.

$$R = \frac{I_r \; \xi \, (1 + \Phi_W + \Phi_S)}{Pb^{\varepsilon} \; Qig}$$

R -- is the forward rate of spread of flaming front (ft/min)
I_r – reaction intensity- Energy Release Rate of fire front (Btu/ft^2/min)
ξ – propagating flux ratio; measure proportion of reaction intensity, which heats adjacent fuel particle to ignition
Φ_W – a dimensionless multiplier that accounts for effect of wind in increasing the flux ratio
Φs – dimensionless multiplier that accounts for the effect of slope in increasing the flux ratio
P_b – amount of fuel (lb/ft^3)
ε – measure of the proportion of fuel partial that is heated to ignition temperatures at time flaming combustion starts
Qig – amount of heat required to ignite one pound of fuel (Btu/lb)
Numerator: amount of heat actually received by potential fuel
Denominator: heat required to bring the fuel to ignition temperatures

A complete set of all formulas used in the program can be found in Appendix 3: Formulas and Computer Calculations

3.2 Computer Applications

The following flow chart demonstrates the processing of the inputs and the calculations preformed by the program it is marked Figure 2-1.

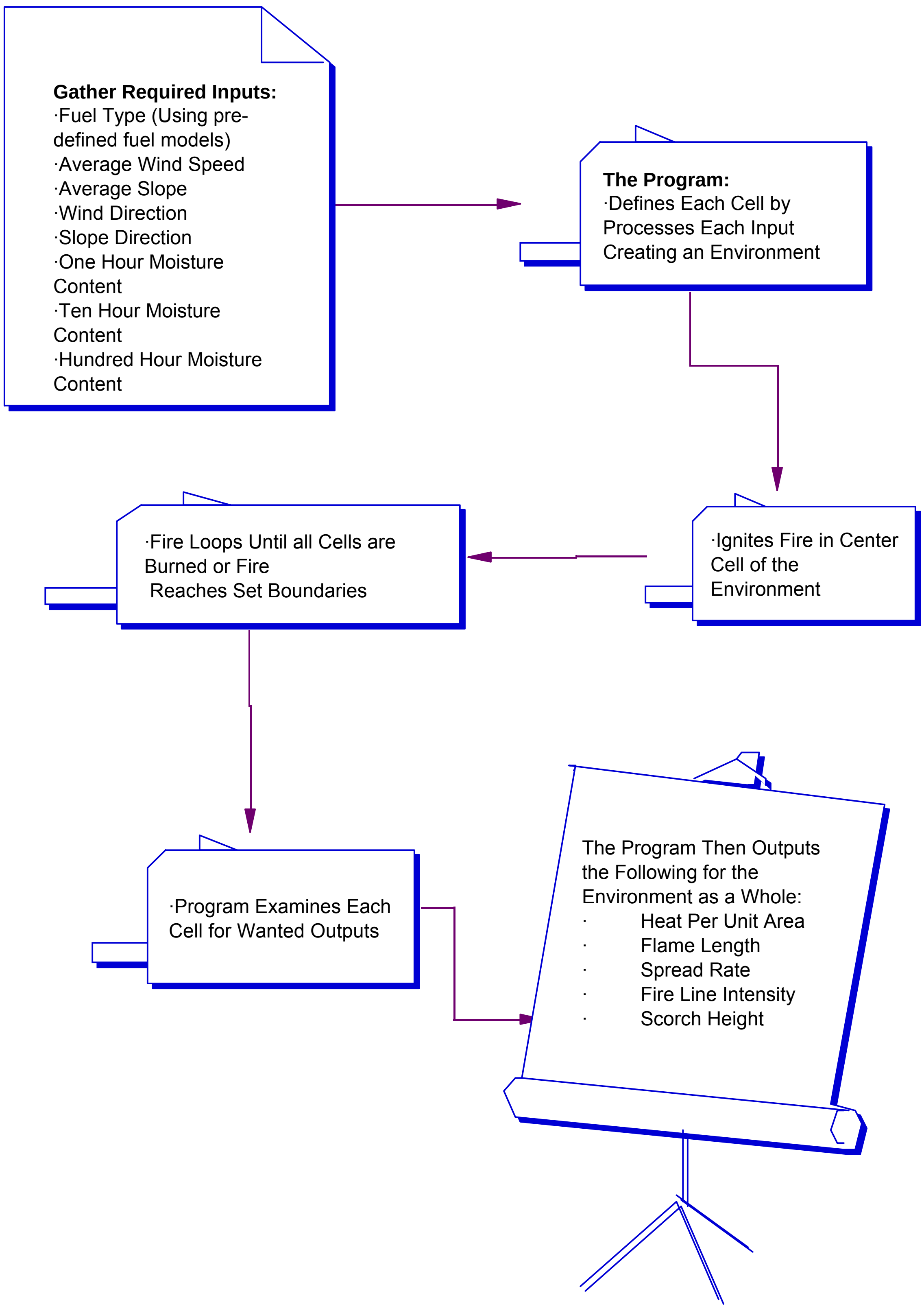
Gather Required Inputs:
·Fuel Type (Using pre-defined fuel models)
·Average Wind Speed
·Average Slope
·Wind Direction
·Slope Direction
·One Hour Moisture Content
·Ten Hour Moisture Content
·Hundred Hour Moisture Content
The Program:
·Defines Each Cell by Processes Each Input Creating an Environment
·Ignites Fire in Center Cell of the Environment
·Fire Loops Until all Cells are Burned or Fire Reaches Set Boundaries
·Program Examines Each Cell for Wanted Outputs
The Program Then Outputs the Following for the Environment as a Whole:
· Heat Per Unit Area
· Flame Length
· Spread Rate
· Fire Line Intensity
· Scorch Height

Figure 2-1

Wildfires spread based on the type, quantity, and quality of fuel surrounding it. The fuel load is the amount of flammable material that surrounds a fire. It is measured by the amount of fuel per unit area (tons per acre). A small load will spread slowly and with low intensity. The fire will spread quickly and intensely through a dense and deep fuel bed. When the fuel load is very dry, compared to being soaked with rainwater, it also increases the consummation rate. Some basic fuel characteristics are:

- size,
- shape,
- arrangement (such as slash, litter) , and
- moisture content of the wood and/or fuel.

“Small fuel materials also called flashy fuels such as dry grass, pine needles, dry leaves, twigs, and other dead brush, burn faster than large logs or stumps”(Bonsor 2). As a fire grows it dries the fuel ahead of it, as a result, making the fuel easer to ignite and burn. Spread out fuel burns faster and the fire consumes more because there is more oxygen available and the fire will not suffocate, that is self extinguish. Fuels that are more tightly packed maintain more moisture, hence, absorbing the heat of the fire (Bonsor 2). Figure 2-3 demonstrates the average depth area for the shrubs. Average grass or shrub depth is about seventy percent of the maximum leaf or stalk height. Figure 2-4 demonstrates the difference between slash (S layer), litter (L layer), fermentation (F layer), and humus (H layer).

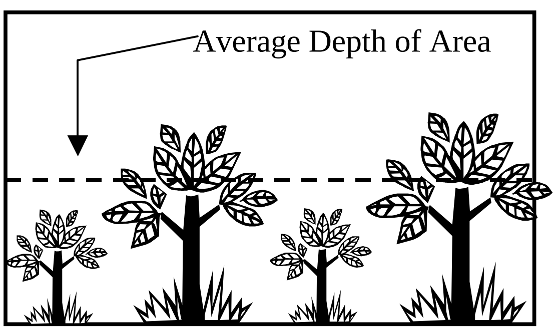

Figure 2-3

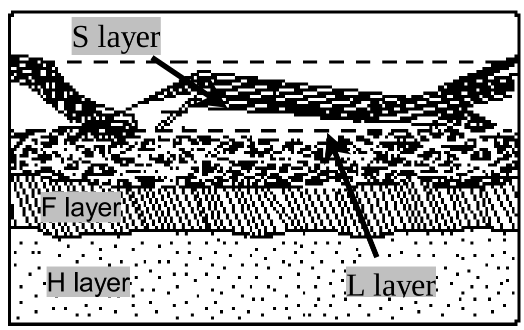

Figure 2-4

Weather plays a major role played in the ignition, growth, and extinguishment of the fire. Drought promotes extremely favorable conditions for a wildfire. Three of the most important factors, which determine the behavior of the wildfire, are: temperature, wind, and moisture. Temperature of the environment, fuel, and of the fire effects the ignition of the fire. The higher the temperature the faster the fuel will burn. Wind has the biggest impact on the way a fire burns. The following figures (figure 2-5 and figure 2-6) demonstrate the affect the wind has in respects to the slope.

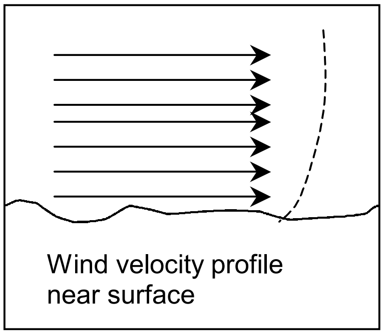

Figure 2-5

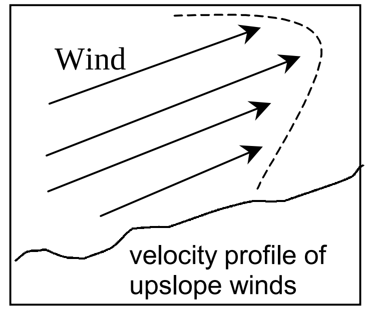

Figure 2-6

Moisture obviously effects how much, how long, and how rapid it will blaze. The relationship between the fire and wind is known as coupled fire at mosphere dynamics. The fire can produce its own winds and weather patterns. Fire whirls are the winds which large violent fires generate. These fire whirls act like tornados, the whirls are the part of the fire which throw flaming logs and burning derbies in front of the fire creating what is known as spot fires(Bonsor 5).

The lay of the land is the last major influence in the behavior of the fire. It can either help or hurt the fire. Fires travel much faster uphill than downhill. If the slope is steeper than the fire travels at a greater velocity. When the fire reaches the top of the hill it will most likely burn itself out because it cannot pre-heat the fuel on the reverse slope on the other side of the mountain (Bonsor 5-6).

4.0 Results

4.1 Computer Calculations

The following inputs are processed into the program:

- fuel Type (using pre-defined fuel models)
- average Wind Speed
- average Slope
- wind Direction
- slope Direction
- one Hour Moisture Content
- ten Hour Moisture Content
- hundred Hour Moisture Content

The program then takes each individual input and processes it through each individual cell in the given area, thus creating an environment and landscape for the fire to burn on. After this environment is created the program will ignite a “fire” in the center cell of the environment. The fire will then burn until either all cells are burnt or the fire reaches the boundaries of the environment area. After all this is complete, the program evaluates each individual cell to determine the outputs. The following outputs collected and processed as a whole and printed by the computer program:

- flame length (length of flame from top to where it leaves the ground),
- spread rate (the rate at which the fire spreads throughout the cells,
- heat generated by the fire per square foot of fuel, and the
- fire line intensity at the current azimuth.

4.2 Graphs, Tables, and Figures

A complete table of the pre-defined fuel models can be found in Appendix 1: Mathematical Models and Inputs under Table 1-1. Fuel models consist of the type of fuel located on the prescribed fire area. For examples, fuel model one consists of short grass(one foot). It has 0.74 tons per acre of one-hour fuel loading, a bed depth of one foot, and a 12 percent moisture content of the dead fuels. The following figure (figure 4-1) demonstrates the measurements of flame depth, and flame length in relation to a fire.

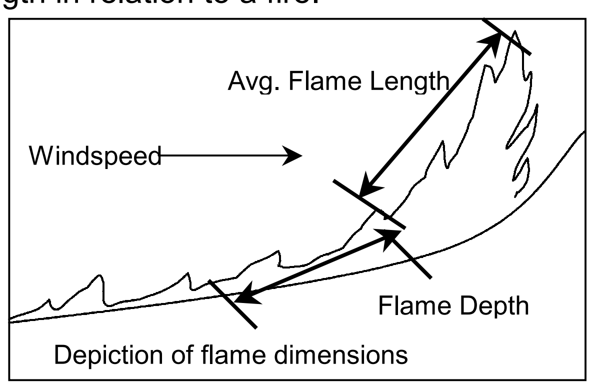

Figure 4-1

5.0 Conclusions

5.1 Computer Program

The computer program is created to create useful data on the behavior of a prescribed burn. The behavior is calculated by using certain characteristics that are gathered before the prescribed fire is to be ignited. Because wildfires are becoming more and more common as the years progress it is necessary for the forest service office and other organizations (such as Natural Resource Conservation Service) to have an accurate program to simulate the behavior of wildfires in all situations. The program is specially developed to be a program that is easy to use and modify to individual preference. It has also been constructed to produce an accurate prediction of the behavior of a wildfire.

The program successfully calculates and outputs the fire's flame length, spread rate, wind speed and direction, fuel model used, heat generated by the fire per square foot of fuel, and the fire line intensity at the current azimuth. All of these were calculated using the formulas found in Appendix 3: Formulas and Computer Calculations.

5.2 Word Processor Program

Word is used in the writing of the program. This particular word processor is program applicable because it has special features. Ever helpful comments and help icons are pop up; there are page breaks, and spelling and grammar checks. These are advantages because while programming it is always helpful to get tips. The grammar and spelling are especially nice because when programming it is difficult and frustrating to have to return to the error and find a misspelled word.

5.3 Recommendations

It is recommended that further work be done to predict the behavior of devastating crown fires. Fierce winds, which push the flames to the top of the trees, create what are known as Crown Fires. Crown fires are the most dangerous, hardest to control, and destructive of all the wildfires. They leap form treetop to treetop. In reality, they are impossible to fight. They have been known to hurtle not only firebreaks but also rivers hundreds of feet wide. These fires shoot above four hundered feet in the air an usually reah up to two thousand degrees Fahrenheit. They generally burn up to thirty-fice tons of fuel an acre in just an hour; the winds thay reate exceed one humdered miles per hour. Satellite and aircraft do all research on Croun Fires because it is virtually impossible to collect data and study them. It is said, "Forest fires are not willing research subjects" (Gantenbein 87). These enormous fires completely devastate forests of Penderosa Pine. Consequently, this slays dozens of species of birds, mammals and insects. And in the end the Ponderosa finds it hard to make a comeback for foreign weeds and trees that did not exist there before choke them (Gantenbein84-87).

References

Bevins, Collin D.2002, fireLib User Manual and Technical Reference. <http://www.fire.org/firelb/doc.html>

Bonsor, Kevin. 2002,“How Wildfires Work.” How Stuff Works. <http://www.howstuffworks.com/wildfire.htm/printable>

Burgan, Robert E., Richard C. Rothermel,1986,BEHAVE: Fire Behavior Prediction and Fuel Modeling System – FUEL Subsystem. NWCG Prescribed Fire and Fire Effects Working Team.

Gantenbein,2002,Douglas.“Burning Questions.” Scientific American.

Rothermel, Richard C.,1986,How to Predict the Spread and Intensity of Forest and Range Fires. NWCG Prescribed Fire and Fire Effects Working Team.

Scott, Chris. US Forest Service, Silver City, New Mexico

Appendices

Appendix 1: Input Tables

Appendix 2: Computer Code

Appendix 3: Formulas and Computer Calculations

Appendix 1: Input Tables

Table 1-1 **Standard Input Fuel Models Used in Fire Behavior:**

Fuel Model	Typical Fuel Complex	Fuel Loading (Tons/Acre)				Bed Depth (feet)	Dead Moisture Content (Percent)
		1 hr	10hrs	100 hrs	live*		
Grass and Grass-Dominated							
1	Short Grass (1ft)	0.74	0.00	0.00	0.00	1.0	12
2	Timber(grass)	2.00	1.00	.50	.50	1.0	15
3	Tall Grass (2.5ft)	3.01	.00	.00	.00	2.5	25
Chaparral and Shrub Fields							
4	Chaparral (6ft)	5.01	4.01	2.00	5.01	6.0	20
5	Brush (2 ft)	1.00	.50	.00	2.00	2.0	20
6	Slash (hardwood)	1.50	2.50	2.00	.00	2.5	25
7	Southern Rough	1.13	1.87	1.50	.37	2.5	40
Timber Litter							
8	Close Timber Litter	1.50	1.00	2.50	0.00	0.2	30
9	Hardwood Litter	2.92	.41	.15	.00	0.2	25
10	Timber (litter)	3.01	2.00	5.01	2.00	1.0	25
Slash							
11	Light. Log Slash	1.50	4.51	5.51	0.00	1.0	15
12	Med Log Slash	4.01	14.03	16.53	.00	2.3	20
13	Heavy Log Slash	7.01	23.04	28.05	.00	3.0	25

*Amount of live fuel to be burned in each fuel model.

Table 1-2: **Beaufort's Scale for Estimating Wind Speed at 20ft Elevation**

Wind Class	**Speed (mi/h)**	**Nomenclature**
1	≤ 3	*Very Light*-- smoke rises nearly vertical. Leaves of quaking aspen in constant motion; small branches of bush sway slender branchlets and twigs of trees move gently; tall grasses and weeds sway and bend with wind, wind vane barely moves
2	4-7	trees in the open sway gently; wind felt distinctly on face; loose scraps of d flutters small flag.
3	8-12	*Gentle Breeze*-- pole size trees in open sway very noticeably; large branches of pole-size trees in the open toss; tops of trees in dense stands sway; wind extends small flag; few crested waves form on lakes
4	13-18	*Moderate Breeze*-- pole size trees in the open violently sway; whole trees in dense area sway noticeably; dust is raised on road
5	19-24	*Fresh*-- branchlets are broken from trees; inconvenience is felt in walking against wind;
6	25-31	*Strong*-- tree damage increases with occasional breaking of exposed tops and branches; progress hampered when walking against wind; light structural damage to buildings.
7	32-38	*Moderate Gale*-- severe damage to tree tops; very difficult to walk into wind; significant structural damage occurs
8	≥ 39	*Fresh Gale*-- intense stress on all exposed objects, vegetation, buildings; canopy offers virtually no protection; wind flows is systematic in disturbing everything in its path.

Appendix 2: Computer Code

```
#include "fireLib.h"- /* fireLib.h is the library containing all fuel models and
calculations*/

#define INFINITY 999999999.

/*  these modify map size & resolution. */
static int    Rows   = 10;     /* Number of rows in each map. */
static int    Cols   = 10;
;     /* Number of columns in each map. */
static double CellWd = 220.;    /* Cell width (E-W) in feet. */
static double CellHt = 220.;    /* Cell height (N-S) in feet. */

/* these set uniform burning conditions. */
static size_t Model   = 0;      /* NFFL 1 */
static double WindSpd = 0;     /* mph */
static double WindDir = 0;     /* degrees clockwise from north */
static double Slope   = 0.0;    /* fraction rise / reach */
static double Aspect  = 0.0;    /* degrees clockwise from north */
static double M1      = 0;    /* 1-hr dead fuel moisture */
static double M10     = 0;    /* 10-hr dead fuel moisture */
static double M100    = 0;    /* 100-hr dead fuel moisture */
static double Mherb   = 0;   /* Live herbaceous fuel moisture */
static double Mwood   = 0;   /* Live woody fuel moisture */

static int PrintMap _ANSI_ARGS_((double *map, char *fileName ));

int main ( int argc, char **argv )
{
   /* neighbor's address*/     /* N  NE   E  SE   S  SW   W  NW */
   static int nCol[8] =       {  0,  1,  1,  1,  0, -1, -1, -1};
   static int nRow[8] =       {  1,  1,  0, -1, -1, -1,  0,  1};
   static int nTimes = 0;      /* counter for number of time steps */
   FuelCatalogPtr catalog;     /* fuel catalog handle */
   double nDist[8];            /* distance to each neighbor */
   double nAzm[8];             /* compass azimuth to each neighbor (0=N) */
   double timeNow;             /* current time (minutes) */
   double timeNext;            /* time of next cell ignition (minutes) */
   int    row, col, cell;      /* row, col, and index of current cell */
   int    nrow, ncol, ncell;   /* row, col, and index of neighbor cell */
   int    n, cells;            /* neighbor index, total number of map cells */
   size_t modelNumber;         /* fuel model number at current cell */
   double moisture[6];         /* fuel moisture content at current cell */
   double fpm;                 /* spread rate in direction of neighbor */
   double minutes;             /* time to spread from cell to neighbor */
   double ignTime;             /* time neighbor is ignited by current cell */
```

```
int    atEdge;              /* flag indicating fire has reached edge */
size_t *fuelMap;            /* ptr to fuel model map */
double *ignMap;             /* ptr to ignition time map (minutes) */
double *flMap;              /* ptr to flame length map (feet) */
double *slpMap;             /* ptr to slope map (rise/reach) */
double *aspMap;             /* ptr to aspect map (degrees from north) */
double *wspdMap;            /* ptr to wind speed map (ft/min) */
double *wdirMap;            /* ptr to wind direction map (deg from north) */
double *m1Map;              /* ptr to 1-hr dead fuel moisture map */
double *m10Map;             /* ptr to 10-hr dead fuel moisture map */
double *m100Map;            /* ptr to 100-hr dead fuel moisture map */
double *mherbMap;           /* ptr to live herbaceous fuel moisture map */
double *mwoodMap;           /* ptr to live stem fuel moisture map */
/*take user inputs*/

printf("Which fuel model(0-13) will the fire burn through? ");
scanf("%d", &Model);
printf("What is the current wind speed in mph? ");
scanf("%lf", &WindSpd);
printf("How many degrees clockwise from north is the wind coming from? ");
scanf("%lf", &WindDir);
printf("What is the average slope? ");
scanf("%lf", &Slope);
printf("How many degrees clockwise from north is the slope coming from? ");
scanf("%lf", &Aspect);
printf("What is the one hour mosture content? ");
scanf("%f", &M1);
printf("What is the ten hour mosture content? ");
scanf("%f", &M10);
printf("What is the hundred hour mosture content? ");
scanf("%f", &M100);
printf("What is the live fuel mosture content for the area? ");
scanf("%f", &Mherb);
printf("What is the dead fuel mosture content for the area? ");
scanf("%f", &Mwood);
printf("\n");


/* allocate all the maps. */
cells = Rows * Cols;
if ( (ignMap   = (double *) calloc(cells, sizeof(double))) == NULL
  || (flMap    = (double *) calloc(cells, sizeof(double))) == NULL
  || (slpMap   = (double *) calloc(cells, sizeof(double))) == NULL
  || (aspMap   = (double *) calloc(cells, sizeof(double))) == NULL
  || (wspdMap  = (double *) calloc(cells, sizeof(double))) == NULL
  || (wdirMap  = (double *) calloc(cells, sizeof(double))) == NULL
```

```
  || (m1Map    = (double *) calloc(cells, sizeof(double))) == NULL
  || (m10Map   = (double *) calloc(cells, sizeof(double))) == NULL
  || (m100Map  = (double *) calloc(cells, sizeof(double))) == NULL
  || (mherbMap = (double *) calloc(cells, sizeof(double))) == NULL
  || (mwoodMap = (double *) calloc(cells, sizeof(double))) == NULL
  || (fuelMap  = (size_t *) calloc(cells, sizeof(size_t))) == NULL )
{
   fprintf(stderr, "Unable to allocate maps with %d cols and %d rows.\n",
      Cols, Rows);
   return (1);
}

/*initialize all the maps */
for ( cell=0; cell<cells; cell++ )
{
   fuelMap[cell]  = Model;
   slpMap[cell]   = Slope;
   aspMap[cell]   = Aspect;
   wspdMap[cell]  = 88. * WindSpd;     /* convert mph into ft/min */
   wdirMap[cell]  = WindDir;
   m1Map[cell]    = M1;
   m10Map[cell]   = M10;
   m100Map[cell]  = M100;
   mherbMap[cell] = Mherb;
   mwoodMap[cell] = Mwood;
   ignMap[cell]   = INFINITY;
   flMap[cell]    = 0.;
}

/* set an ignition time & pattern (this ignites the middle cell and outer cells). */
cell = Cols/2 + Cols*(Rows/4);
ignMap[cell] = 1.0;
cell = 2*Cols/3 + Cols*(Rows/4);
ignMap[cell] = 1.0;

/* create a standard fuel model catalog and a flame length table. */
catalog = Fire_FuelCatalogCreateStandard("Standard", 13);
Fire_FlameLengthTable(catalog, 0, 0);

/* Calculate distance across cell to each neighbor and its azimuth. */
for ( n=0; n<8; n++ )
{
   nDist[n] = sqrt ( nCol[n] * CellWd * nCol[n] * CellWd
                   + nRow[n] * CellHt * nRow[n] * CellHt );
   nAzm[n] = n * 45.;
}
```

```
/* find the earliest (starting) ignition time. */
for ( timeNext=INFINITY, cell=0; cell<cells; cell++ )
{
   if ( ignMap[cell] < timeNext )
      timeNext = ignMap[cell];
}

/* loop until no more cells can ignite or fire reaches an edge. */
atEdge = 0;
while ( timeNext < INFINITY && ! atEdge )
{
   timeNow  = timeNext;
   timeNext = INFINITY;
   nTimes++;

   /* examine each ignited cell in the fuel array. */
   for ( cell=0, row=0; row<Rows; row++ )
   {
      for ( col=0; col<Cols; col++, cell++ )
      {
         /* Skip this cell if it has not ignited. */
         if ( ignMap[cell] > timeNow )
         {
            /* first check if it is the next cell to ignite. */
            if ( ignMap[cell] < timeNext )
               timeNext = ignMap[cell];
            continue;
         }

         /* flag if the fire has reached the array edge. */
         if ( row==0 || row==Rows-1 || col==0 || col==Cols-1 )
            atEdge = 1;

         /* determine basic fire behavior within this cell. */
         modelNumber = fuelMap[cell];
         moisture[0] = m1Map[cell];
         moisture[1] = m10Map[cell];
         moisture[2] = m100Map[cell];
         moisture[3] = m100Map[cell];
         moisture[4] = mherbMap[cell];
         moisture[5] = mwoodMap[cell];
         Fire_SpreadNoWindNoSlope(catalog, modelNumber, moisture);
         Fire_SpreadWindSlopeMax(catalog, modelNumber, wspdMap[cell],
            wdirMap[cell], slpMap[cell], aspMap[cell]);
```

```
            /*  examine each unignited neighbor. */
            for ( n=0; n<8; n++ )
            {
               /* First find the neighbor's location. */
               nrow = row + nRow[n];
               ncol = col + nCol[n];
               if ( nrow<0 || nrow>=Rows || ncol<0 || ncol>=Cols )
                  continue;
               ncell = ncol + nrow*Cols;

               /* Skip this neighbor if it is already ignited. */
               if ( ignMap[ncell] <= timeNow )
                  continue;

               /* Determine time to spread to this neighbor. */
               Fire_SpreadAtAzimuth(catalog, modelNumber, nAzm[n],
FIRE_FLAME | FIRE_BYRAMS);
               if ( (fpm = Fuel_SpreadAny(catalog, modelNumber)) > Smidgen)
               {
                  minutes = nDist[n] / fpm;

                  /* Assign neighbor the earliest ignition time. */
                  if ( (ignTime = timeNow + minutes) < ignMap[ncell] )
                  {
                     ignMap[ncell] = ignTime;
                     Fire_FlameScorch(catalog, modelNumber, FIRE_FLAME );
                     flMap[ncell] = Fuel_FlameLength(catalog,modelNumber);
                  }

                  /* Keep track of next cell ignition time. */
                  if ( ignTime < timeNext )
                     timeNext = ignTime;
               }
            }   /* next neighbor n */
         }   /* next source col */
      }   /* next source row */
   } /* next time */

    /*print fire outputs*/
   printf("The Flame Length is %2.2f feet.\n",
            Fuel_FlameLength(catalog, modelNumber));
   printf("The spread rate is %3.2f feet per minute.\n",
            Fuel_SpreadAny(catalog, modelNumber));
   printf("The wind speed is %3.2f miles per hour.\n",
            WindSpd);
   printf("The fuel model %d was used.\n",
```

```
            Fuel_Model(catalog, modelNumber));
    printf("The heat per square foot of fuel bed is %3.2f BTU.\n",
            Fuel_HeatPerUnitArea(catalog, modelNumber));
    printf("The fire line intensity at the current azimuth is %3.2f.\n",
            Fuel_ByramsIntensity(catalog, modelNumber));
    printf("The wind is coming form %3.0f degrees clockwise from north.\n",
            WindDir);
    PrintMap(flMap, "Flame Map.txt");
    PrintMap(ignMap, "Ign Map.txt");
   getch();
   /*  if requested, save the ignition & flame length maps. */
   if ( argc > 1 )
      PrintMap(ignMap, argv[1]);
   if ( argc > 2 )
      PrintMap(flMap, argv[2]);

   return (0);
}

static int PrintMap ( map, fileName )
   double *map;
   char   *fileName;
{
   FILE *fPtr;
   int cell, col, row;

   if ( (fPtr = fopen(fileName, "w")) == NULL )
   {
      printf("Unable to open output map \"%s\".\n", fileName);
      return (FIRE_STATUS_ERROR);
   }

   fprintf(fPtr, "north: %1.0f\nsouth: %1.0f\neast: %1.0f\nwest: %1.0f\nrows:
%d\ncols: %d\n",
      (Rows*CellHt), 0., (Cols*CellWd), 0., Rows, Cols);
   for ( row=Rows-1; row>=0; row-- )
   {
      for ( cell=row*Cols, col=0; col<Cols; col++, cell++ )
      {
         fprintf(fPtr, " %2.0f", (map[cell]==INFINITY) ? 0.0 : map[cell]);
      }
      fprintf(fPtr, "\n");
   }
   fclose(fPtr);
   return (FIRE_STATUS_OK);
```

```
}
```

Appendix 3: Formulas and Computer Calculations

$$R = \frac{I_r \ \xi \ (1 + \Phi_W + \Phi_s)}{Pb^{\varepsilon} \ Qig}$$

R -- is the forward rate of spread of flaming front (ft/min)

I_r – reaction intensity- Energy Release Rate of fire front (Btu/ft^2/min)

ξ – propagating flux ratio; measure proportion of reaction intensity, which heats adjacent fuel particle to ignition

Φ_W – a dimensionless multiplier that accounts for effect of wind in increasing the flux ratio

Φs – dimensionless multiplier that accounts for the effect of slope in increasing the flux ratio

P_b – amount of fuel (lb/ft^3)

ε – measure of the proportion of fuel partial that is heated to ignition temperatures at time flaming combustion starts

Qig – amount of heat required to ignite one pound of fuel (Btu/lb)

Numerator: amount of heat actually received by potential fuel

Denominator: heat required to bring the fuel to ignition temperatures

$I_B = H_A R$ $\quad$ $H_A = I_R t_r$ $\quad$ $t_r = \frac{384}{\sigma}$ $\quad$ $F_L = 0.45 \, I_B^{\,0.46}$

H_A – heat per unit area (Btu/ft^2)

I_B – fireline intensity – heat released per min for ft of fuel (Btu/ft/min)

R – rate of spread (ft/min)

I_R – reaction intensity - heat released per min per ft^2(Btu/ft^2/min)

t_r – flame residence time – amount of time for fuel to burn (min)

F_L – flame length (ft)

σ -- characteristic surface-area-to-volume ratio of the fuel array (ft^2/ft^3)