Fisheries and Environment Canada
Pêches et Environnement Canada

Forest Fires in the Computer

A model to predict the perimeter location of a forest fire

Information Report FF-X-65 December 1977

Peter Kourtz
Shirley Nozaki
William O'Regan

FOREST FIRES IN THE COMPUTER - A MODEL TO PREDICT THE PERIMETER LOCATION OF A FOREST FIRE

by

Peter Kourtz
Shirley Nozaki
William G. O'Regan

Résumé en français

Information Report FF-X-65

December 1977

The authors are, respectively, Research Officer, Forest Fire Research Institute, Canadian Forestry Service; Research Assistant, University of California, Berkeley; Mathematical Statistician, Pacific Southwest Forest and Range Experiment Station, U.S.D.A. Forest Service, Berkeley (Retired).

Metric (SI) terminology has been used throughout this report. However, the convention of omitting the comma (used as a separator in Canada) from large numerical values, or replacing the comma with a blank in large numerical values has not been followed. The reader will also note that the terms hectare and ha (the SI symbol for hectare) are used interchangeably. This "bending" of the SI rules is for the benefit of those not completely familiar with the SI symbols and convention. One hectare is equal to 2.47 acres, approximately.

ABSTRACT

A model and corresponding computer program have been developed to predict the perimeter location of a forest fire. The model assumes that the forest floor is partitioned into a grid of two-hectare cells each containing one of a small number of homogeneous fuel types. The rate-of-spread of fire across a particular cell depends on the cell's fuel type and moisture content as well as the wind conditions. Canadian and U.S. Forest Service rate-of-spread equations are used to obtain these values. The model keeps account of the times required for fire to spread from cell to cell and thus the perimeter can be plotted by the computer at any specified time. Fires up to 7,000 hectares have been modeled using a small "mini" computer.

Verification of the model is about to begin. This will involve many years of field trials under operational conditions during which the model should undergo an evolution designed to bring its predictions closer to reality. Our initial goal is to obtain more accurate perimeter estimates than those produced by existing procedures. This goal should be achieved rather easily once fuel type data from the Forest Fire Research Institute's LANDSAT program is made available to the fire model.

RESUME

On a conçu un modèle et un programme d'ordinateur correspondant afin de prévoir le périmètre d'un feu de forêt. Ce modèle prend pour acquis que la surface de la forêt est divisée en une grille dont chaque cellule, en forme de carré, représente deux hectares et que chacune d'elles contient un petit nombre de types de combustibles homogènes. La vitesse de propagation du feu à l'intérieur d'une cellule déterminée dépend du type de combustible présent, de sa teneur en humidité ainsi que de la vélocité du vent. Pour obtenir le taux de propagation du feu, on se sert des équations utilisées à cette fin par les Services forestiers canadien et américan. Le modèle tient compte du temps nécessaire au feu pour se propager d'une cellule à l'autre. De cette façon l'ordinateur peut, à n'importe quel moment spécifique, relever le tracé du périmètre. En se servant d'un "mini" ordinateur, on a pu établir le modèle de feux atteignant jusqu'à 7,000 hectares.

La vérification du modèle est sur le point de commencer. Ceci exigera plusieurs années d'essais et d'opérations sur le terrain, durant lesquelles le modèle devra subir une évolution de façon à ce que ses prédictions correspondent le plus possible à la réalité. Notre but premier est d'obtenir une estimation plus précise du périmètre d'un incendie que peuvent le faire les moyens actuels. Ce but devrait être atteint assez facilement, lorsque nous pourrons incorporer au modèle les données sur les types de combustibles, fournies par le programme Landsat de l'Institut de recherches sur les feux de forêts.

CONTENTS

FOREST FIRES IN THE COMPUTER - A MODEL TO PREDICT THE PERIMETER LOCATION OF A FOREST FIRE

INTRODUCTION

The ability to predict the behavior of a forest fire in terms of its perimeter location at any future time has been a goal of forest fire behavior research for the last 50 years. In Canada, work has concentrated on methods to predict how fast fire will spread through some of our more important general fuel types (Van Wagner, 1973). The rate-of-spread across a few of these fuels now can be approximated given an estimate of the fuel's moisture content, the speed and direction of the wind and the slope of the terrain. If the fuel type and factors governing spread rates remain uniform, the fire theoretically should spread in an elliptical pattern. Its perimeter location can be easily predicted by a simple elliptical growth model such as that of Van Wagner (1969). Indeed, operational trials of a similar "geometric" model by the U.S. Forest Service and the Bureau of Land Management have produced encouraging results (Van Gelder, 1976).

This report describes a more complex model that is designed to consider either homogeneous or changing fuel types and hourly changes in wind speed and direction and fuel moisture. The model is in the form of a Fortran computer program designed to be run on a small "mini" computer. A user must enter forecasted hourly temperatures, relative humidities, precipitations and wind conditions, a general fuel type in which the fire is to burn, the desired ignition pattern, and the locations of fire breaks. Output is in the form of hourly burned-area estimates and corresponding perimeter maps. Because fuel types and terrain are considered to be uniform, a fire ignited from a point with a constant wind direction and no fire breaks will be elliptical in shape.

Although the model is designed for changing fuel types, such data are not readily available. It is hoped that, in the near future, data from the LANDSAT satellite will provide this necessary input (Kourtz, 1977). Work has begun at the Forest Fire Research Institute (FFRI) to construct a digital database containing approximately 10 broad forest cover types across an 8-million hectare area on a 1/2-hectare resolution. The database will be constructed using data from the LANDSAT satellite and classification methods and equipment of the Canada Centre for Remote Sensing.

The model described in this report is not new. As early as 1967, research began by the U.S. Forest Service on a related problem; that of predicting the target size associated with a small smouldering fire for a high altitude infrared fire detector (Kourtz and O'Regan, 1971). In 1971, the authors redesigned the model so that fires up to 15,000 hectares could be processed

using a large computer (O'Regan et al., 1973). Considerable effort has been spent by the U.S. Forest Service, studying the theoretical errors associated with the model (O'Regan et al., 1976) and the performance of the model under actual field conditions (Offensend et al., 1975).

Recently, the FFRI modified the U.S. Forest Service version for operational field use, assuming homogeneous fuel and flat terrain. This modification included the following:

a) Adaptation of the program to run on a small mini computer.

b) Redesign of the main storage algorithm resulting in improved efficiency in the computer code, allowing larger fires to be processed in the same internal memory space.

c) A rate-of-spread calculation program designed to supply the fire model with hourly spread rates for 17 fuel types, given forecasted fuel moisture and wind conditions. This program uses eight fire spread equations for common eastern Canadian fuel types as well as nine U.S. Forest Service standard fuel models and their corresponding spread equations (Van Wagner, 1973 and Rothermel, 1972).

d) The internal computer code was modified to allow quick and easy coupling of the model to a large fuel database such as the one currently under development at FFRI.

e) A format suitable for operational field trials, including conversational input and error checking was developed to permit use by noncomputer oriented people.

Potential Uses for the Fire Model

Fire control agencies rely upon available personnel to evaluate the growth potential of newly reported small fires and the likely behavior of large fires. Both of these functions require the subjective integration of fire behavior knowledge, experience with current and future fuel and weather conditions as well as existing and planned suppression activities. The outcome of this exercise is usually expressed in terms of rather vague estimates of future fire size and perimeter location. These estimates are rarely recorded for later verification in spite of the fact that, each year in Canada, decisions costing millions of dollars are based on these estimates. These decisions involve everything from speed and strength of initial attack to control and community evacuation strategies for large fires.

Given our state of knowledge concerning fire behavior, fuel moisture/weather interactions, and our ability to forecast wind

and humidity conditions for short time periods, there seems to be little reason why, in the near future, a computerized model cannot forecast fire growth more accurately and easier than existing personnel. The process involved in making such a prediction can be viewed, as a large, but routine information processing problem. Data on fuel type, terrain, fuel moisture, and predicted weather must be assembled and quickly processed -- an ideal task for a computer. At the very least, such a model could assist and supplement the existing forecast procedure, thus providing more time for decision-makers to concentrate on costly resource utilization decisions.

The model offers the potential to test different fire line construction techniques and control strategies to see if suitable fire lines could be constructed and burned out before arrival of the fire front. More informed decisions regarding the evacuation of communities ahead of large fires might also be possible. In summary, the model should allow fire control experts to better anticipate growth potential of both small and large fires and hopefully more informed decisions will result.

The model may be useful for prescribed burn planning and execution. Different ignition tactics could be tested in different wind, fuel moisture and fire line situations in order to obtain a range of conditions that might attain the objectives of the burn. Actual fuel moisture and wind conditions could be used to indicate likely consequences of igniting a fire at a specific time.

The model has interesting research considerations. It can be viewed as a tool that would enable researchers to integrate much of what is known concerning fire behavior, fuel type, fuel moisture, terrain, and wind. Rate-of-spread equations for various conditions could be tested by comparisons to actual fire perimeters. In a similar manner, the effectiveness of different fuel classification schemes and weather conditions could be tested. Weaknesses in our knowledge will quickly become apparent, perhaps leading to new research efforts. One glaring deficiency, already so obvious, is our lack of knowledge concerning the conditions necessary for crown fires and their rate-of-spread. Currently, the model can only predict the growth of ground fires.

How the Model Works

The model assumes that the forest is made-up of adjoining square cells or patches each containing one of a small number of fuel types. Fire spreads from midcell to adjacent midcell in a time interval determined by the fuel types of the two cells and their corresponding rate-of-spread at that specific time of day and their orientation relative to the wind direction. The special optimizing algorithm used by the model merely ensures that the fire front proceeds from cell to cell in a proper chronological fashion. At the same time, it keeps track of the

arrival time of the fire front at each midcell point. This bookkeeping function makes it possible to construct a sequence of maps showing the perimeter position at various times after ignition.

At first glance, the task of modeling the growth of a fire does not appear difficult. All that is required is to duplicate the "natural" growth of the fire just as it must progress in the real-world. With the passage of each small unit of time, the perimeter must have the opportunity to advance. But, a problem arises in trying to do this because fire can reach a cell's center from many different directions. There is no guarantee that the fire will first reach a cell's midpoint by way of the closest burning cell. Keeping a proper account of the timing of fire arrival at cell centers becomes a difficult task, especially since "fingers" associated with faster burning fuels will always be present.

Another interpretation of the "natural" fire spread phenomenon was discovered that lead to several solutions to the problem of assigning correct midcell burning times. For the situation where fire from one cell eventually causes the ignition of another cell, whether the two cells be close or widely separated, fire must proceed along a minimum time-path. That is "nature" must "select" the path between two cells that requires the least time to traverse.

Initial Algorithm

The first solution to the problem of determining the minimum time required for fire to travel between any cell's midpoint and an ignition point, involved a dynamic programing algorithm. It directly or indirectly examined all possible paths from an ignition cell to all other cells. During this process, the shortest time-path and corresponding arrival time were identified.

An essential part of this procedure was the definition of a "direct" route and corresponding time to burn "directly" between any two cell pairs. This "direct" route involved a series of horizontal and vertical midcell moves as close as possible to a line joining the centers of the two cells of interest. The time required to travel this "direct" route was calculated knowing the fuel types along the route, their rate-of-spread, and the path's direction relative to the wind. Each successive dynamic programming stage of the algorithm involved calculating for each cell (say cell I):

a) For all other cells (i.e., including cell J), the sum of the "direct" travel time from J to I plus the previous stage's minimum cumulated arrival time at cell J.

b) The minimum of these sums (defined to be the stage's minimum cumulated arrival time for cell I).

This recursive scheme was repeated for as many stages as cells in the grid. In practice, however, the minimum times stabilized usually within four or five stages.

A plot of the minimum time-path routes at each stage quickly revealed an interpretation of the formulation. Each successive stage represented a potential additional "bend" in the path between the ignition cell and the cell of interest. Stage 1 represented the "direct" route from the ignition cell to all other cells. Each of stage N's final state values represented the minimum time to travel to the point of interest with N-1 or fewer "bends" along the route.

Although the "bends in the path" algorithm correctly calculated the minimum time for a fire to reach each midcell point and, therefore, provided a solution to the problem of "natural" fire spread, its solution was not economical. A small problem involving only 100 cells required a calculation time in excess of one minute on a moderately large computer.

Dijkstra Labelling Algorithm

The search for a more efficient procedure to determine shortest time-paths continued until the Dijkstra labelling algorithm was found (Dijkstra, 1959). This procedure assigns two types of labels to each cell center. The value of a temporary label on a cell represents an upper bound on the minimum time to go from the ignition cell to that cell. The value of a permanent label on a cell center represents the minimum time to go from the ignition cell to that cell. Through an iterative process, temporary cell values are reassigned and made permanent. The process terminates when either the node of interest is labelled permanent (in the case where the minimum time path to a specific cell is required) or when all cells of the grid are permanently labelled (in the case where perimeter contours are required). The steps of the algorithm are:

a) Assign to the ignition cell the permanent value of zero and to the remaining cells the temporary label value of infinity.

b) Reassign the temporary label values of those cells adjacent to the permanently labelled ignition cell. These values will be the times required to burn to the midpoints of these cells from the midpoint of the ignition cell.

c) Scan all cell values (temporary values) adjacent to the permanent cell values and relabel the smallest of these as permanent. Call this cell I.

d) For each temporary labelled cell value adjacent to I, calculate the time to burn from I to that adjacent cell and add this value to the value of I. Compare this total to the previously assigned temporary value and make the smaller of these the new temporary value.

e) Repeat steps (c) and (d) until the specific cell of interest or all cells have been permanently labelled (Kourtz and O'Regan, 1971).

The time of arrival of the fire at each cell's midpoint in the grid can be found using this algorithm. The actual path that the fire takes through the grid from the ignition cell to any other cell can be easily identified if the necessary direction-of-travel information were coded along with each cell's burn time (Elsner et al., 1975). However, knowledge of these actual fire paths is not necessary to establish perimeters. A fire perimeter can be sketched for a specific time using linear interpolation between those adjacent cells with burn times spanning the time of interest. Alternatively, the Dijkstra procedure can be stopped when the last permanently labelled value exceeds the time of interest. In this case, the perimeter is somewhere between the outside permanent cell's midpoints and their adjacent temporary cell's midpoints.

The Dijkstra solution simultaneously solves both problems associated with the two ways of viewing the fire spread phenomenon. With each iteration, a new permanent label is assigned, representing the "natural" or proper chronologically ordered growth of the fire perimeter. In this context, the algorithm can be viewed as a simple and efficient procedure to manage the bookkeeping-time problem associated with a spreading fire. As we have already seen, it is also a procedure to identify minimum-time paths and associated correct burn times for each cell.

Main Features of the Computer Program[1]

The present version of the model uses either a cell size of 142 meters square or approximately 2 hectares or a cell size of 71 meters square or approximately 1/2 hectare. The proposed LANDSAT fuel database is being designed with resolution elements of 71 meters square or approximately 1/2 hectare. The fire model can use either the 1/2-hectare cells or a grouping of four database cells depending upon the size and detail of the fire that the user wishes to model.

---

[1] A detailed discussion of the storage and processing structure used in the program can be seen in Appendix I.

A study examining the theoretical accuracy of the model pointed out the importance of the number of cells the algorithm considers to be adjacent to a particular cell (O'Regan, et al., 1976). In theory, a fire starting from a point on flat terrain with no wind and burning in a homogeneous fuel, should spread in a circular pattern. The model under the same conditions, however, results in a square pattern if the fire is only allowed to spread to adjacent north-south and east-west cells. The error amounts to a 36% underestimate compared to the circular pattern. The fire shape is octagonal if it is allowed to spread across main diagonals in addition to the north-south and east-west directions. The resulting underestimate is 10% -- an amount considered acceptable given our ability to predict rates-of-spread and our most elementary fuel classification scheme.[2] Thus, the present version assumes that fire can spread to eight surrounding cells.

Spread rates for fires in 17 different fuel types for eight directions corresponding to those used by the model are calculated by a separate program. This program begins by asking the user for moisture contents of the medium and heavy fuels. Next, for each hour that the user wishes to run the fire model, he or she must enter the anticipated temperature, relative humidity, precipitation and wind speed, and direction. Hourly fine fuel moistures are estimated using Van Wagner's hourly fine fuel moisture code system (Van Wagner, 1972). The program prints out to the user and stores for the fire model, the rates-of-spread (in meters per hour) for all fuels, directions, and hours.

Spread rates for eight of these fuel types are based on Canadian Forestry Service equations and the remaining nine are based on the U.S. Forest Service general fuel models and corresponding equations. The U.S. equations were developed using a combination of fire physics, laboratory modeling and empirical observations (Rothermel, 1972). The Canadian approach depends less on laboratory modeling and more on field observations (Van Wagner, 1975). Appendix III presents an example of the use of this program.

---

[2] Appendix II contains a more detailed discussion of this error.

Because the model's bookkeeping scheme keeps track of the time of arrival of the fire front at each cell's midpoint, it becomes possible to vary the fire's rate-of-spread each hour. This is done even as the fire progresses between midcell pairs. Predicted wind changes and hourly changes in fuel moisture may have a significant influence on the fire's shape. For example, as evening approaches, fine fuel moisture increases and winds usually drop. Under such conditions, the model will predict a corresponding slowdown or halting of fire growth.

The flexibility of the Dijkstra algorithm also allows the fire to be started from multiple ignition lines or points. This feature will be useful for modeling large fires starting from existing perimeter sections. It also has interesting prescribed burn and backfiring applications. Eventually, this feature may be used to predict the consequences of spot fires.

Another feature of the present program allows the fuel type of any cell to be reclassified to a non-burnable type. This feature is useful when used in combination with the line ignition feature for backfiring. Eventually, this feature will permit the evaluation of various fire line construction activities.

To start the program, the user must answer questions concerning the following:

a) The cell size (1/2 or 2 hectares).

b) Starting time and forecast period of the fire.

c) Whether or not maps at periodic time intervals are required showing the fire's shape or are only area estimates required.

d) The fuel type in which the fire is burning.

e) The coordinates of the ignition line or point[3] (only end points of horizontal or vertical lines need be read in).

f) The number and locations of barrier lines.

Figure 1 presents an example of the input and output for a run.

---

[3] The coordinate scheme is described in Appendix I.

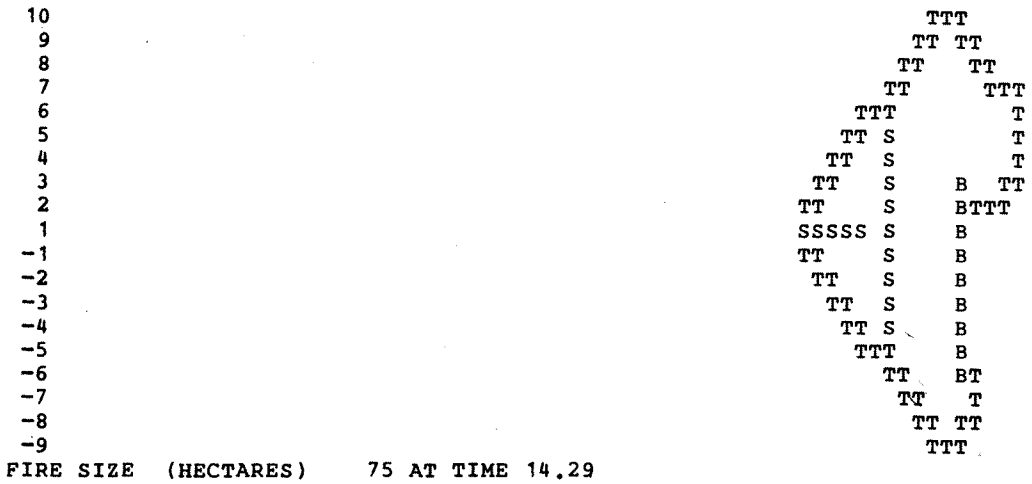
Figure 1

An Example of Input and Corresponding Output Produced by the Fire Model Program

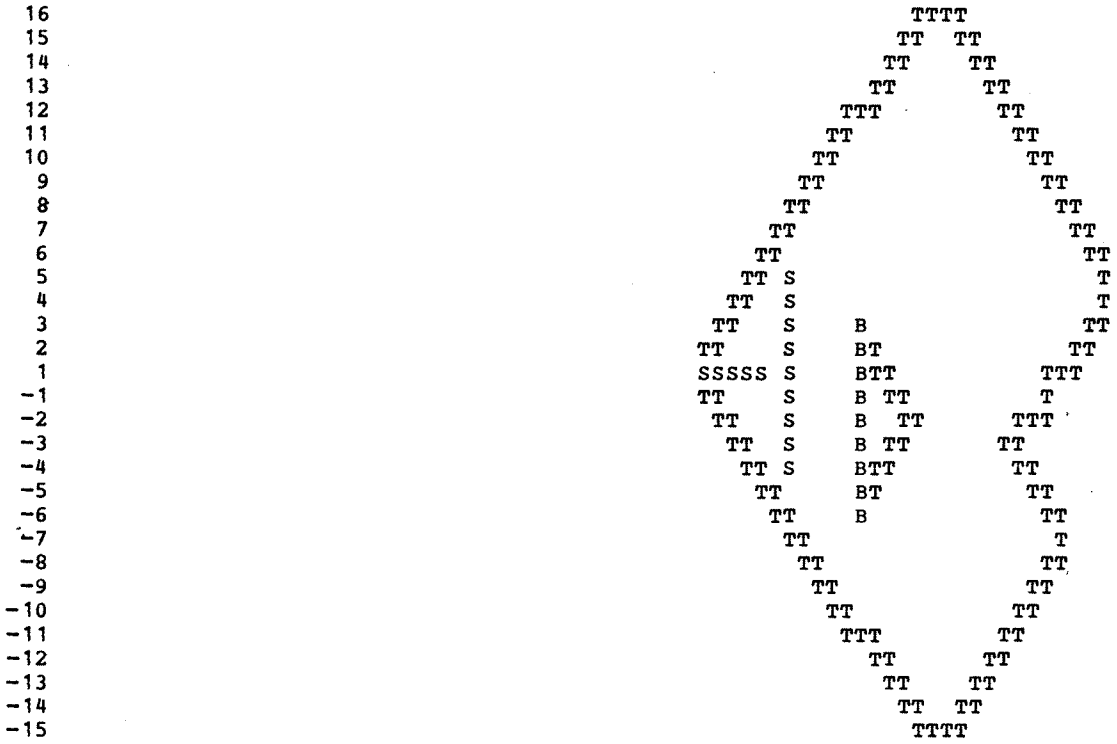
(Spread Rates for this example can be seen in Appendix III)

```
 RUN DK1:FIRE
STANDARD CELL SIZE IS 2 HECTARES
DO YOU WANT A CELL SIZE OF 1/2 HECTARE? Y
DO YOU WANT MAPS OF THE FIRE? Y
 STARTING HOUR OF FIRE? 13
UNTIL WHAT HOUR DO YOU WANT TO FORECAST? 15
DO YOU WANT TO SEE A FUEL TYPE LIST? Y

         HERE IS A LIST OF THE POSSIBLE FUEL TYPES
              USFS SHORT GRASS [1]
              USFS LONG GRASS [2]
              USFS TIMBER WITH GRASS AND UNDERSTORY [5]
              USFS TIMBER LITTER [6]
              USFS TIMBER WITH LITTER AND UNDERSTORY [7]
              USFS HARDWOOD LITTER [8]
              USFS LIGHT LOGGING SLASH [9]
              USFS MEDIUM LOGGING SLASH [10]
              USFS HEAVY LOGGING SLASH [11]
              CFS JACKPINE SLASH [12]
              CFS BLACK SPRUCE [13]
              CFS ASPEN [14]
              CFS MIXED HW & SW [15]
              CFS MATURE JACKPINE [16]
              CFS PARTIAL CUT [17]

WHAT TYPE IS THE FIRE BURNING IN?12
STARTING COORDINATES OF FIRE 500 500
HOURLY DOWN WIND SPREAD RATES 316.1 371.7 436.9 799.2 987.71070.4 726.5 700.3 664.7 623.6
DOES THE FIRE START AT A LINE OR A POINT? LINE
HOW MANY POINTS DEFINE THE IGNITION LINE? 4
GIVE X-Y COORDINATES OF  4CORNER POINTS
POINT PAIRS MUST FORM HORIZONTAL OR VERT LINES
POINT 1 X-Y CORDS 500,504
POINT 2 X-Y CORDS 500,496
POINT 3 X-Y CORDS 495,500
POINT 4 X-Y CORDS 499,500
ARE THERE ANY FIRE BARRIER LINES? (YES OR NO)Y
HOW MAY BARRIER CORNER POINTS?2
GIVE X-Y COORDINATES OF
POINT PAIRS MUST FORM HORIZONTAL OR VERT LINES
POINT 1 X-Y CORDS 505,502
POINT 2 X-Y CORDS 505,494
FIRE SIZE  (HECTARES)     25 AT TIME 13.48
FIRE SIZE  (HECTARES)     50 AT TIME 13.94
```

```
CURRENT SIZE OF FIRE IN HECTARES AT HOUR 14 IS     54
EACH T AND S IS 0.5 HECTARES
*=BURNED CELL  T=BURNING CELL    S=STARTING CELL
      B=BARRIER CELL   C=CENTER
```



```
FIRE SIZE  (HECTARES)     75 AT TIME 14.29
FIRE SIZE  (HECTARES)    100 AT TIME 14.48
FIRE SIZE  (HECTARES)    125 AT TIME 14.64
FIRE SIZE  (HECTARES)    150 AT TIME 14.78
FIRE SIZE  (HECTARES)    175 AT TIME 14.91

CURRENT SIZE OF FIRE IN HECTARES AT HOUR 15 IS    197
EACH T AND S IS 0.5 HECTARES
*=BURNED CELL  T=BURNING CELL    S=STARTING CELL
      B=BARRIER CELL   C=CENTER
```



```
TT2  --  STOP
>
```

Output maps may be produced showing the location of the fire's perimeter relative to the fuel grid's coordinate system.

The computer program was written in Fortran IV for a Digital PDP-11,T-34 mini-computer for execution in a 32,000, 16-bit word partition without overlaying. Execution time for a 5,000 hectare fire is approximately 5 minutes. Much of this time is taken by the map printing section. Conversion of the program to run on another computer will not be an easy task chiefly because of the extensive internal use of word packing. A copy of the Fortran Computer code for the model is available from the Forest Fire Research Institute, Canadian Forestry Service, 240 Bank Street, Ottawa, Canada K1G 3Z6.

Verification of the Model

To date, in Canada, only one attempt has been made to determine the reliability of the model. Fuel conditions and weather, as best they were known, for the 1971 Thackeray Ontario fire were put into the model and the perimeter location was plotted each hour (Walker and Stocks, 1972). Comparisons of the predicted and actual perimeter locations were attempted. Only vague statements could be made about the model's accuracy, such as, its predictions were "reasonable" and within the "ball park."

This verification exercise merely reconfirmed the difficulty in testing any hypothesis concerning forest fire behavior using actual observations. First, it is nearly impossible to describe, with any degree of accuracy, the state of a large active forest fire at periodic time intervals. Second, it is just as impossible to obtain reliable estimates of wind and fuel conditions at such fires. Safety considerations, logistics problems and lack of sophisticated monitoring equipment, all contribute to this situation. It is very safe to say that the model is not accurate and that it is not likely to be so in the next 20 years. However, even the version presented in this report may predict fire growth better than existing methods.

Verification of this model will be a time-consuming process involving many years. During this time, the model must be used in the field, in conjunction with the conventional fire growth prediction procedures. Cooperation between researchers and field-operations people will be required so that weaknesses can be identified and corrected. It probably will take five to ten years before this process will evolve a version of the model that has sufficient reliability to deserve the confidence of field personnel.

Shortcomings of the Model and Future Work

As mentioned previously, our state of knowledge concerning fire spread and fuel classification is still quite primitive. There are people both in the fire research and fire control communities who will say that, because the problems associated

with fire spread and fuel typing are so complex, the model presented here is wishful thinking. In response, one might well ask, that if the situation appears so hopeless, what has been the purpose and achievements of forest fire behavior research of the last 40 years. Also, fire growth predictions are being made in the field every day by not-so-experienced field staff and expensive fire control decisions made based on them. When one examines the information used and thought processes involved in making these estimates and compares these to what the model uses, one cannot help concluding that it should not be too difficult for a model to do better.

The model described here is far from the final version. Major items yet to be considered are:

a) Nonhomogeneous fuel mixes (a problem hopefully to be corrected shortly using a Landsat database).

b) Crowning conditions and corresponding rate-of-spread.

c) Probabilistic inputs for fuel type, fuel moisture and wind with corresponding probability statements of future perimeter location. Considerable work has already been carried out in this area (O'Regan et al., 1976).

d) Terrain and its effect on fuel moisture and rate-of-spread.

e) Spotting behavior.

In addition, improved forest fuel classifications should be developed that realistically match our ability to automatically map huge areas using remote sensing methods. Of course, more accurate rate-of-spread equations corresponding to improved fuel classifications will be required.

REFERENCES

Dijkstra, E.W. 1959. A note on two problems in connection with graphs. Numerische Math. 1:269-271.

Elsner, G.H., M.R. Travis and P.H. Kourtz. 1975. Dynamic programming subroutines based on the Dijkstra algorithm for finding minimum cost paths in directed networks. Can. Forest. Serv. Inf. Rep. FF-X-51. Ottawa.

Gear, C.W. 1969. Computer organization and programming. McGraw-Hill, Inc.

Kourtz, P.H. and W.G. O'Regan. 1971. A model for a small forest fire ... to simulate burned and burning areas for use in a detection model. Forest Sci. 17(2):163-169.

Kourtz, P.H. 1977. An application of Landsat digital technology to forest fire fuel type mapping. Proceedings 11th Int. Sym. on Remote Sensing of Environment, April.

Offensend, F.L., P. Chatterjee and D.W. North. 1975. An economic assessment of Firescope based on historical fires. Internal Rep., U.S. Forest Serv. P.S.W. Grant No. 16, March. Prepared by Stanford Research Inst., Menlo Park, Calif.

O'Regan, W.G., S. Nozaki and P.H. Kourtz. 1973. A method for using directional rate of spread to predict forest fire configurations. WSS/CI Paper 73-17. Western States Section/The Combustion Inst. Spring Meeting. Tempe.

O'Regan, W.G., P.H. Kourtz and S. Nozaki. 1976. Bias in the contagion analog to fire spread. Forest Sci. 22(1):61-68.

O'Regan, W.G., S. Nozaki and N.E. Savin. 1976. Stochastic fire-front forecasting from reconnaissance - based estimates of rate-of-fire spread. Unpublished Report, USDA Forest Service, PSWFRES. Berkeley, Calif.

Rothermel, R.C. 1972. A mathematical model for predicting fire spread in wildland fuels. U.S. Forest Serv. Res. Paper INT-115.

Van Gelder, R.J. 1976. A fire potential assessment model for brush and grass fuels. Fire Management Notes. Summer.

Van Wagner, C.E. 1969. A simple fire growth model. Forestry Chronicle. April.

Van Wagner, C.E. 1972. A table of diurnal variation in the fine fuel moisture code. Can. For. Serv. Inf. Rep. No. PS-X-38. Chalk River.

Van Wagner, C.E. 1973. Rough prediction of fire spread rates by fuel type. Can. Forest. Serv., Inf. Rep. No. PS-X-42. Chalk River.

Van Wagner, C.E. 1975. A comparison of the Canadian and American forest fire danger rating systems. Can. Forest. Serv. Inf. Rep. No. PS-X-59. Chalk River.

Walker, J.D. and B.J. Stocks. 1972. Analysis of two 1971 fires in Ontario: Thackeray and Whistle Lake. Can. Forest. Serv. Inf. Rep. No. 0-X-166.

APPENDIX I

Details of Internal Program Structures

Word Packing

This version of the program is designed for a Digital PDP-11,T-34 mini-computer that allocates 16 binary storage locations (bits) for each stored integer number (word). The machine makes available approximately 18,000 words of fast-access internal storage to the program. This assumes that the program is loaded into a 32,000-word partition. Because so many numbers required for the model are small (i.e., X-Y coordinates and fuel types), the program makes extensive use of 8-bit half-words or "bytes." Digital Fortran IV allows the direct manipulation of these bytes just as if they were the full 16-bit words. Arrays containing 16,240 numbers are packed into half as many words using this feature.

Hashing and Linking

The program uses two large sets of data -- namely the fuel cell and Dijkstra algorithm burn-time data. Both of these data sets could be stored and later located according to their X-Y location but this would require an enormous amount of internal storage. Only a very small proportion of these data sets are needed in the computer at any one time. The required values could be stored in relatively small lists, provided that they were not located within the lists according to their X-Y location. One storage scheme that would accomplish this result involves the storing of items in any order but packed with each item are the X-Y coordinates associated with it. The problem with this approach would be the large search time required to find the correct item. Many words in the list would have to be unpacked and the corresponding X-Y coordinates checked against the ones that are required. This process would continue until the correct item was found.

Hashing and linking offer a second alternative aimed at reducing the search time required to find a specific item in the list. Hashing involves the use of a "hash" function that points to a specific location in the list, given the X-Y location of the item of interest (Gear, 1969). For example, the hash function used by the Dijkstra algorithm points to a location in a list containing 1,000 items, given the X-Y coordinates from any cell in a grid of 64,500 cells.

The form of the hash function used in the fire model is subjectively determined and involves the use of an arithmetic manipulation of the X-Y coordinates followed by division by a number equal to the length of the storage list. The remainder associated with this division is used as the storage location in the list. Because such a large number of cell values are being forced into a small list, it is certain that there will be

conflicts. To overcome these, first, it is necessary to store the X-Y coordinates in association with the value of interest. Second, a "link" value is stored that is used when a conflict is encountered. This link value points to a location in a separate "free storage" area. This area can be viewed as an extension of the original list. In the free storage area, the value in conflict is stored according to the original list's format. The free storage area is used only when conflicts in the hash location of two values are identified. Each new item that is entered into the free storage area is added to the bottom of the free storage list and the link value of the hash location or previously referenced free storage location is defined to be this last location. Conflicts can occur within the free storage area and these are handled by chaining the link procedure further into the free storage area.

Information for the Dijkstra algorithm is stored in 5 vectors of half words, each 1,500 elements in length. The first 500 elements of each vector corresponds to free storage. Cell X-Y coordinates, midcell fire arrival time, hash link, and impossible spread directions (described later) are stored in these vectors for each cell of interest.

Fuel type data is stored in 4 vectors of half words, each 3,000 elements in length. The values stored are X-Y location, fuel type and hash link. The first 1,000 elements of each fuel vector are reserved as free storage.

This procedure seems rather cumbersome, however, the payoffs come when we wish to locate a previously stored value. Suppose we wish to locate the value at coordinate location (87, -43). Initially, the hash function might point to location 603. Upon decoding the X-Y coordinates at this location, it is found that they are not the ones of interest. The link value points to location 56 in the free storage area, but, on decoding these coordinates, the ones of interest are not found, but the link value now is 130. Finally, at location 130, we find the correct coordinates and corresponding value. The hashing and linking scheme has enabled us to locate a specific value contained in a list possibly as long as 1,500 items, with only three "looks." No extensive search was necessary.

Trials using this scheme and actual fire model data indicate that the average number of "looks" required to find the item of interest is less than three. Of course, the efficiency of this scheme depends on the programer's skill in defining the hash function and on the length of the hash list. In the search for an efficient function, many different hash functions were formulated and tested.

Improvements to the Algorithm

The original Dijkstra algorithm involves calculation of temporary and permanent cell values representing upper bounds on

the minimum burn time and the actual minimum burn time respectively for each cell. One easy method to program this algorithm is to establish two arrays representing these permanent and temporary times. But, since we wish to model rather large fires and the shape that the fire will take is not known, the number of two-hectare cells in the grid probably should be at least 50,000. Therefore, the simple approach of two time arrays would require at least 100,000 storage positions in the computer. Even the largest computers do not have this amount of fast storage available for a reasonable cost. In addition, the search for the minimum temporary cell value would require an unacceptable amount of computer time in such an array of temporary values with each new iteration.

The solution to this problem that eventually evolved makes use of only a small amount of computer storage and also eliminates the need for time-consuming "linear" searches for minimum temporary cell values. A dynamic, unstructured storage system was devised making use of word packing, hashing and linking techniques.

Close examination of the Dijkstra algorithm reveals that it is not necessary to maintain a list of all possible temporary cell values. Only those cells immediately in front of the fire perimeter need be searched for the next minimum temporary value.

In addition, it is not necessary to store the permanent cells representing areas already burned provided that information concerning possible spread directions for each temporary cell is kept current. Fire in temporary cells cannot be allowed to burn back into permanent cells.

A variable-sized vector is defined containing each temporary cell's hash storage location in the main storage array (this array contains the X-Y location, midcell fire arrival time, hashing links and binary-coded spread direction indicator). Immediately after a temporary cell is declared permanent (when its fire arrival time is lower than all other temporary times), its reference is erased from the temporary cell pointing vector and from the main storage array. With the pointing vector, this is accomplished by copying the last entry into the position to be erased and shortening the list by one. The process of deleting this cell's information from the main storage array is much more complex. Because multi-link chaining is possible, a complex trace-back and shuffling procedure is required.

As an indication of the efficiency of this procedure, trials indicate that the hash and free storage arrays of lengths 1,000 and 500 respectively are only filled after fire sizes of 3,500 cells. Improved efficiency is still likely through improved hashing functions and more optimal hash-to-free storage area ratios.

Perimeter location maps can be drawn when the fire reaches specific sizes or burns for a given time interval. This is accomplished by interrupting the algorithm's execution, and printing in map form, the locations of all temporary cells. The interior boundary of temporary cells represents the present fire perimeter. This method of processing perimeter location data prevents the need to store complete sets of permanent cell values.

Briefly summarizing, the word packing, hashing and linking scheme, combined with the efficiencies obtained by dealing only with those temporary cells associated with the fire perimeter results in the Dijkstra algorithm being extremely efficient in computer time and storage.

Fuel Data Processing

It is assumed that each cell is square and either of 1/2 hectare or of two hectares in area and that a fire can burn 127 cells in any direction from an ignition point.[1] This means that a square area of up to 64,516 hectares, each with a fuel type, must be available to the model.

At any one instant, the computer requires only fuel types associated with the temporary cells surrounding the latest permanent cell. Fuel types associated with interior permanent cells are no longer needed. Likewise, it is not necessary to use valuable storage space for fuel types associated with cells located a long way ahead of the fire front. But, a compromise is required with the amount of space used for high speed, internal computer storage and the number of times the slow, auxiliary memory device is referenced in search of new fuel data. Once again, the scheme that evolved makes excellent use of the packing, hashing and linking techniques.

The fuel management subroutine returns the fuel type corresponding to an X-Y coordinate pair to the main algorithm. If the location asked for is new to the hashed fuel storage array, the routine will go to the auxiliary storage device (a Digital RK05 2.4 megabyte disk) and retrieve a 15 by 15 matrix of fuel types that includes the X-Y location of interest. These 225 fuel types are hashed into the fuel array at a location determined by their X-Y location and a hash function. Also, this routine initializes the fuel array and deletes fuels already burned.

---

[1] Maximum coordinate location storable in an eight-bit word is ± 127.

The routine presently assumes that a large database is available consisting of 16 blocks of 250,000, two-hectare cells and that the correct block has been transferred to the disk. Thus, the disk is expected to have a 500 by 500 block of fuel types with the fire origin somewhere in the interior. These data will be stored in binary form in 1,089 records of 225 bytes (15 by 15). Records are numbered row-wise from the upper left corner. Row 1 ends with record 33 and row 2 starts with record 34. The last record is 1,089. Within a record, the 15 by 15 fuel matrix is stored column-wise.

The fire origin is assumed to be located at (1,1) using a relative coordinate scheme. This is the ignition point when the fire starts from a single point. The fuel subroutine contains all the necessary logic to map the relative coordinate location to the actual disk record containing that location, to retrieve it, and hash its elements into the fuel storage vector. In the future, elevation or slope data could be coupled with fuel type data and processed in a similar fashion.

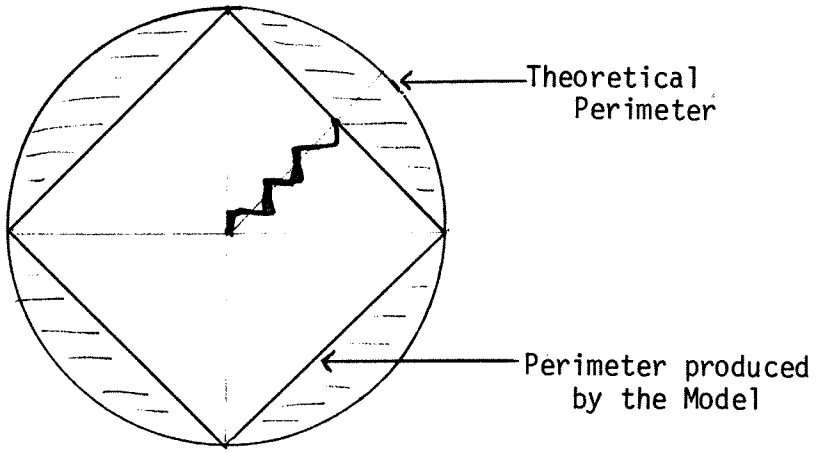
APPENDIX II

A Theoretical Error in the Model and the Correction

Trials of an early version of the model that used only horizontal and vertically adjacent cells under no wind, flat terrain and homogeneous fuel immediately revealed an interesting error. In this case, a fire should take the shape of a circle but the model produces a square shape. Figure 1 illustrates this error.

Figure 1

Theoretical and Model Perimeters Under the Assumption of Equal Spread Rates in All Directions



The percent error (E) is defined as:

$$E = \frac{T-A}{T} * 100$$

where: T is the circular area $T = \Pi * (\text{rate of spread} * \text{elapsed time})^2$
A is the actual area produced by the model
$A = 2 * (\text{rate-of-spread} * \text{elapsed time})^2$

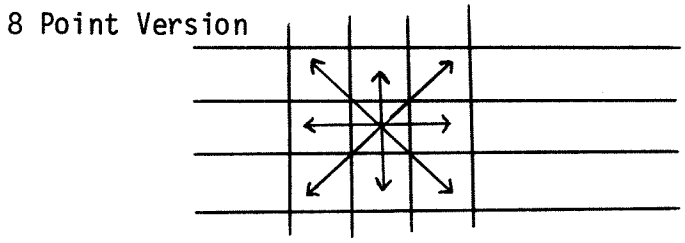
Note that (T-A) is represented in Fig. 1 by the shaded area and that it is always a positive value. This error was a significant 36%. It is caused by the fact that the fire is forced to spread only to four symmetrically located adjacent midcell points. Fig. 1 illustrates the "stepped" path required by the model compared with the direct theoretical path. Both paths are of equal length but the diagonal distance travelled by the modeled fire is considerably shorter than the theoretical distance. Only in the four major directions are both distances

equal. It is interesting to note that no matter how small the cell size is, the percent error remains constant.

The only way to reduce the percent error is to include more adjacent cells. The improvement associated with a hexagonal grid of cells in which fire was allowed to spread to six adjacent midcell points was examined. The theoretical error associated with it was 17%. No other cell shapes that permit burning directly to adjacent midcell points are possible. To further reduce the error, a new definition of adjacency was established that allowed diagonal crossings between nearby cells. Fig. 2 describes the eight and sixteen adjacent point situations associated with a grid of square cells. Table 1 gives an analysis of the error size associated with the various definitions of adjacency.

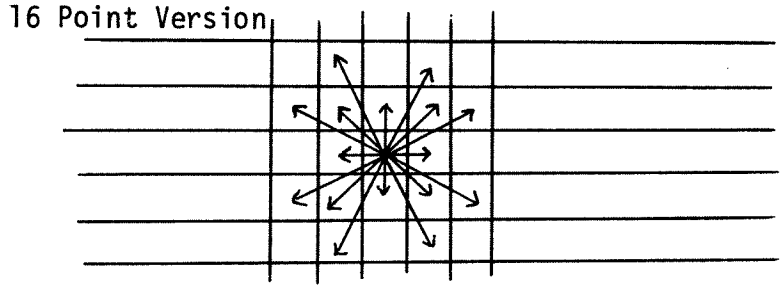
Figure 2

Definition of Adjacency for Eight and sixteen Adjacent Cell Versions



Arrows indicate the path taken by the fire to the eight "adjacent" cells.

Locations of Sixteen adjacent cells.



Note that the fire passes through four different cells on its way to the midpoints of the outer eight cells.

The elapsed time to spread from the point of interest to an adjacent cell midpoint is found by summing the individual times spent by the fire in each cell along the most direct route. This time is a function of the length of the direct route in the cell.

Table 1

Percent Error Associated with Different Definitions of Adjacency

No. of Adjacent Cells	Approximate Theoretical Error (%)	Actual Error in Trials (%)	Computer Time Required (sec.) for a 5,000 cell fire
4	36.3	34.0	2.9
6	17.3	18.0	3.9
8	10.0	10.0	5.0
16	2.5	2.5	9.2
32	0.006	1.0	17.9

1) The approximate theoretical error was $\frac{\Pi - A}{\Pi} * 100$

where: $A = C * SIN (\frac{180}{C}) * C\emptyset S (\frac{180}{C})$ and C represents the number of "adjacent" cells.

2) Based on a fire of 5,000 cells in size in uniform fuel with no slope or wind.

3) Test carried out on a UNIVAC 1108 computer. From Table 1 it can be seen that the eight-cell definition of adjacency reduces the error to about 10% but at a cost of doubling the computer time required. An earlier version of the computer program, written for a large computer allowed the user to select from four up to thirty-two adjacent cells.

APPENDIX III

An Example of the Input and Output Associated with the Program to Calculate Spread Rates

```
 RUN DK1:FLAME

!!!!!!!! ALL INPUT IS IN METRIC UNITS !!!!!!!!!
            ALL MOISTURE CONTENTS (MC) ARE %

DO YOU WANT CONVERSATIONAL INPUT CAPABILITY FOR WX  DATA ?
( NO IMPLIES YOU WILL CREATE AN  EDIT FILE PK:DIURWXDAT )NO

INPUT FUEL MC FOR...

MEDIUM DEAD FUEL  30

HEAVY DEAD FUEL  30

FINE GREEN FUEL  120


INPUT DEAD FUEL MC (R) FOR HOUR 11....20

WX INPUT DATA AND CALCULATIONS FOR MC'S AND FFMC

DAY HR   T RH   W WD     R      EC     MC   FFMC
  1 11  20 35   8270   0.00   10.7   18.7   85.6
  1 12  21 30   8270   0.00    9.5   17.3   86.7
  1 13  22 28   8270   0.00    8.9   16.0   87.7
  1 14  25 26  16270   0.00    7.9   14.4   88.9
  1 15  28 24  16270   0.00    7.0   12.8   90.2
  1 16  22 30  16270   0.00    9.3   12.2   90.7
  1 17  17 35   8270   0.00   11.3   12.1   90.8
  1 18  18 60   8270   0.00   14.2   12.4   90.6
  1 19  16 65   8270   0.00   15.5   12.7   90.2
  1 20  16 70   8270   0.00   16.6   13.2   89.9

DO YOU WANT TO CHANGE YOUR ANSWERS? NO

DO YOU WANT OUTPUT TO THE TERMINAL ? Y

DO YOU WANT ENTIRE 10 HR RATE OF SPREAD OUTPUT?...NO

START TIME...14

END TIME....15
```

RATE OF SPREAD FOR EACH FUEL AND DIRECTION

USFS SPREAD RATES (METERS/HR) FOR HOUR 14 BY 8 DIRECTIONS

DIRECTION ANGLE	90.0	270.0	0.0	180.0	315.0	45.0	225.0	135.0
SHORT GRASS	3715.1	62.4	62.4	62.4	62.4	2785.6	62.4	2785.6
LONG GRASS	3944.0	62.7	62.7	62.7	62.7	2534.7	62.7	2534.7
OPEN CONIFER	1845.7	31.5	31.5	31.5	31.5	998.2	31.5	998.2
CLOSED CONIFER	45.8	3.0	3.0	3.0	3.0	40.4	3.0	40.4
SEMIOPEN CONIFER	124.4	4.6	4.6	4.6	4.6	77.8	4.6	77.8
HARDWOOD LITER	332.4	10.4	10.4	10.4	10.4	188.5	10.4	188.5
LIGHT SLASH	187.4	8.2	8.2	8.2	8.2	128.7	8.2	128.7
MEDIUM SLASH	383.2	19.2	19.2	19.2	19.2	265.6	19.2	265.6
HEAVY SLASH	450.4	25.2	25.2	25.2	25.2	312.4	25.2	312.4

CFS RATES OF SPREAD (METERS/HR)

DIRECTION ANGLE	90.0	270.0	0.0	180.0	315.0	45.0	225.0	135.0
JACKPINE SLASH	799.2	35.7	35.7	35.7	35.7	556.1	35.7	556.1
BL SPRUCE LITTER	20.4	0.3	0.3	0.3	0.3	10.6	0.3	10.6
ASPEN	90.8	4.0	4.0	4.0	4.0	63.1	4.0	63.1
MIXED HW & SW	174.8	4.6	4.6	4.6	4.6	104.4	4.6	104.4
MATURE JACKPINE	50.0	1.1	1.1	1.1	1.1	28.5	1.1	28.5
PARTIAL CUT	143.1	4.8	4.8	4.8	4.8	91.7	4.8	91.7

USFS SPREAD RATES (METERS/HR) FOR HOUR 15 BY 8 DIRECTIONS

DIRECTION ANGLE	90.0	270.0	0.0	180.0	315.0	45.0	225.0	135.0
SHORT GRASS	4039.4	66.1	66.1	66.1	66.1	2951.1	66.1	2951.1
LONG GRASS	4178.3	66.4	66.4	66.4	66.4	2685.3	66.4	2685.3
OPEN CONIFER	1925.1	32.9	32.9	32.9	32.9	1041.2	32.9	1041.2
CLOSED CONIFER	49.1	3.1	3.1	3.1	3.1	42.7	3.1	42.7
SEMIOPEN CONIFER	127.7	4.7	4.7	4.7	4.7	79.9	4.7	79.9
HARDWOOD LITER	352.0	11.0	11.0	11.0	11.0	199.6	11.0	199.6
LIGHT SLASH	198.5	8.7	8.7	8.7	8.7	136.3	8.7	136.3
MEDIUM SLASH	406.3	20.4	20.4	20.4	20.4	281.7	20.4	281.7
HEAVY SLASH	477.5	26.7	26.7	26.7	26.7	331.1	26.7	331.1

CFS RATES OF SPREAD (METERS/HR)

DIRECTION ANGLE	90.0	270.0	0.0	180.0	315.0	45.0	225.0	135.0
JACKPINE SLASH	987.7	44.1	44.1	44.1	44.1	687.3	44.1	687.3
BL SPRUCE LITTER	32.9	0.5	0.5	0.5	0.5	17.1	0.5	17.1
ASPEN	112.5	5.0	5.0	5.0	5.0	78.1	5.0	78.1
MIXED HW & SW	247.9	6.6	6.6	6.6	6.6	148.1	6.6	148.1
MATURE JACKPINE	74.0	1.7	1.7	1.7	1.7	42.1	1.7	42.1
PARTIAL CUT	190.5	6.4	6.4	6.4	6.4	122.1	6.4	122.1