# Computer Programs for Avalanche Runout Prediction

Prepared by,

THEODORE E. LANG

For the,

National Research Center for Disaster Prevention

Shinjo Branch

Shinjo

Japan

* * * * *

20 July 1983

## Contents

FIGURES AND TABLES

Page

# COMPUTER PROGRAMS FOR AVALANCHE RUNOUT PREDICTION

by

THEODORE E. LANG

SHINJO BRANCH, NATIONAL RESEARCH CENTER FOR DISASTER PREVENTION

SHINJO, YAMAGATA-KEN, JAPAN 996

## ABSTRACT

This report summarizes the function and application of three computer programs, made operational on the Shinjo Branch computer system, for use in analysis of snow avalanche runout. In the use of fluid mechanics principles to model snow flow, two of the codes are based upon equilibrium hydrodynamic equations, while a third incorporates transient, viscous effects in a two-dimensional incompressible boundary layer formulation. Of the two hydrodynamic based codes, one which was previously developed, has constant frictional and viscous material coefficients that vary significantly with different avalanche types, which makes it difficult to apply, except by experienced persons. The second code, developed at Shinjo Branch, brings in a flow depth dependence and a material locking property in definition of the material coefficients, which reduces significantly the range on the coefficient values for different avalanche types. Listing of each code is included in this reporting, as is the format and order in input data preparation. Comparison of predicted velocity profiles and runout distances from each code is made for one avalanche path (Ironton Park, Colorado USA). While velocity profiles are different for each code, runout distance can be matched by selective choice of parameters.

## Introduction

In this report is summarized the results of an investigation having the following dual purposes:

1. To make operational on the Shinjo Branch computer, certain codes that are used in analysis of snow avalanche runout.
2. To develop and checkout a modified version of a computer code, that is based upon equilibrium flow dynamics, but incorporates recent developments in the mechanics of avalanche flow.

The codes that have been made operational on the Shinjo Branch computer (Melcom 70 Computer System) are:

1. Program AVALNCH — This program models the two-dimensional transient flow of a viscous fluid. The code has been used to analyze numerous avalanche paths and different avalanche types. The code was developed by imposing restrictive conditions, unique to avalanche dynamics, upon a general purpose fluids code. The general purpose code can be used for a wide range of transient viscous fluid problems, including impact dynamics. Two versions of program AVALNCH are considered in this reporting. One version, operational since 1978, uses a so called "fast-stop" option to model the slow-down of avalanches at low-speed terminal flow. This modeling is necessary because of thixotropic character of flowing snow, which has a tendency to lockup as the flow speed reduces to a stop condition. The fast-stop algorithm is an empirical representation of the locking property. A modified version of AVALNCH, which incorporates a biviscous modeling of snow, is reported also. The program was developed during the course of this reported work. The biviscous representation of snow, approaching the Bingham fluid idealization of a locking material,is a more physically based approximation

of the snow locking property than fast-stop. Both versions of program AVALNCH use two parameters to represent the fluid state of flowing snow.

2. Program ACCEL — This program, developed by Cheng and Perla (1979), is based upon an equilibrium viscous fluid modeling of avalanching snow. The program uses hydrodynamic equations of two-dimensional flow, in which two parameters relating to surface friction and viscous drag are selected for an avalanche analysis.
3. Program BIEQ — This program, developed during the course of this work, is based upon the fluid dynamics of viscous equilibrium flow. It is a modification of program ACCEL in which recent mechanics principles of avalanching snow are incorporated, at least to some degree of approximation. The program incorporates a two-parameter material representation, but also accounts for material locking, explicitely.

In the following reporting of these computer codes, some details on the use of the codes are given, as well as results of analysis of avalanching snow. Program results are compared, and conclusions drawn from these comparisons.

## COMPUTER PROGRAM AVALNCH

Computer program AVALNCH is a specialized version of a general purpose program called SOLA-SURF, which was developed by Hirt, Nichols and Romero (1975) at Los Alamos Scientific Laboratories. Program SOLA-SURF models the 2 dimensional transient flow of an incompressible viscous fluid which may have a free surface. The programs are based upon numerical integration, using finite difference methodology, of the 2 dimensional equations of motion of a viscous fluid. The equations are

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = g_x - \frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\nabla^2 u$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = g_y - \frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\nabla^2 v$$

where $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$

In addition to these equations, the equation of conservation of mass, namely

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

also enters the computations. Computations are carried out in two steps within the computer code. At each cycle (CYCLE) the fluid is advanced in the grid, based upon the gravitational driving force and the frictional drag force acting on each cell that contains fluid. Following this calculation, the fluid is redistributed by one or more iterations (ITER) in order that the total mass of material not change within a specified limit of accuracy. At the start of each cycle Eqs (1) are solved, whereas in the iterative phase simple linear equations are used, since the change needed in order to conserve mass, is small.

Program AVALNCH has been specialized to model the flow of a snow avalanche over an extended path by several simplifying assumptions that significantly reduce computer running time, compared to that of a general fluid modeling of the problem. The basic simplifications are:

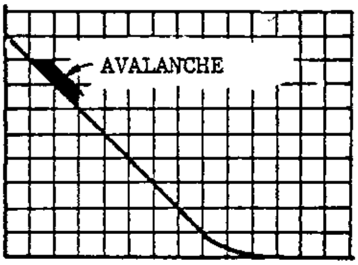
1. Specialization of the avalanche flow to that of bounday layer

flow, for which a single vertical cell is used to represent the depth of the flow.

2. Representation of the actual avalanche path profile by a horizontal grid of elements, for which slope-parallel and slope-normal gravity components are specified.

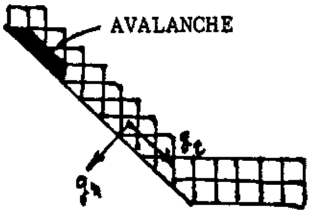
These simplifications can be demonstrated graphically by a sequence of diagrams (Figure 1). It is representation (c) of Figure 1 that is used in program AVALNCH. This approximation excludes only impulses impressed upon the flowing material as the profile slope decreases. However, the effect of this has been shown to be negligible for ordinary avalanche paths (Cheng, Perla, 1979). The reduction to two cells in the vertical direction, has resulted in significant reduction in computer running time; yet the results show accurate prediction of avalanche speeds and runout distances along the path. In using only two vertical cells, the representation of any vertical variation in the flow parameters is excluded, so that modeling of flow depth should not be expected to be accurate. To represent vertical effects more accurately, additional vertical cells should be used (and may be used in program AVALNCH), but, with the long runout distances of avalanches, the computer cost may become excessive.

A number of avalanches have been modeled with program AVALNCH, so that the range in the basic parameters of the code have been established (Lang, etal., 1979), (Martinelli, etal., 1980). The two parameters are viscosity, $\nu$, and friction coefficient, $f$. In the case of high altitude mid-winter, strongly sintered, dry snow avalanches, as occur in the Rockies, the values of $f$ and $\nu$ that model the flow are numerically equal at $f$=0.45 and $\nu$=0.45m$^2$s$^{-1}$. For low altitude coastal wet snow avalanches, values as high as 0.6 to 0.8 have been used. For weakly sintered dry snow avalanches values as low as 0.4 have yielded adequately modeled results.



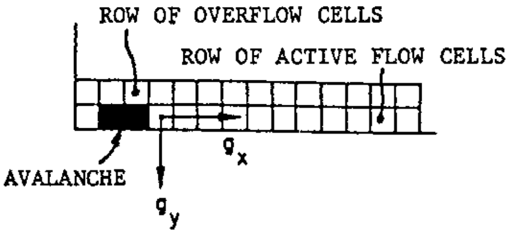
(a)

HERE THE AVALANCHE PROFILE IS SUPERIMPOSED UPON A RECTANGULAR GRID OF CELLS - THE SETUP FOR THE ORDINARY APPROACH TO NUMERICAL FLOW PROBLEMS



(b)

HERE IF SHOWN A GRID SETUP FOR A BOUNDARY LAYER FLOW PROBLEM IN WHICH ONE GRID DIMENSION IS LONG - BUT WITH THIS SETUP BOUNDARY CONDITIONS ARE COMPLEX



(c)

HERE IS SHOWN A HORIZONTAL GRID REPRESENTATION OF THE FLOW PROBLEM. GRAVITY COMPONENTS $g_x$ AND $g_y$ ACCOUNT FOR THE DRIVING AND CONTACT FORCES OF THE MOTION, WHICH MAY VARY WITH EACH CELL

FIGURE 1: FLOW DOMAIN REPRESENTATIONS

It is suggested that if the program is to be used in site specific applications that test cases be run to determine the range of values of $f$ and $\nu$.

## AVALANCHE ANALYSIS USING PROGRAM AVALNCH

Several steps are involved in setting up a problem to be run with program AVALNCH. Once an avalanche path is selected, the first step is to draw a profile of the path using the same scale in the vertical and horizontal directions. Generally, data from which a profile is drawn is taken from topographic maps of the avalanche region. A typical profile plot of an avalanche path in Colorado (Ironton Park avalanche path) plotted from a 1:25,000 topographic map is shown in Figure 2. The profile may be approximated by a continuous curve as in Figure 2, or by a series of straight line segments, which is computationally easier. Having drawn the profile, the next step is to lay off a uniform grid along the slope, selecting the grid dimension so that less than 200 grid lines are used along the path. In the case of Ironton Park 110 grid lines were used, separated by 10.0m increments. By some measurement or calculation technique the change in elevation from one end of each 10.0m element (or cell) to the other end must be determined. For example, for Ironton Park the elevation change of each element was determined by rule measurements on the profile (Table 1). Accuracy of measurement of each element elevation change is not as important as having the total elevation change equal that of the profile, and this should be checked each time a profile is set up. For example, in the case of Ironton Park the elevation changes in Table 1 could be rounded off to whole numbers and not significantly change the velocity along the path, provided the total elevation change remains the same.

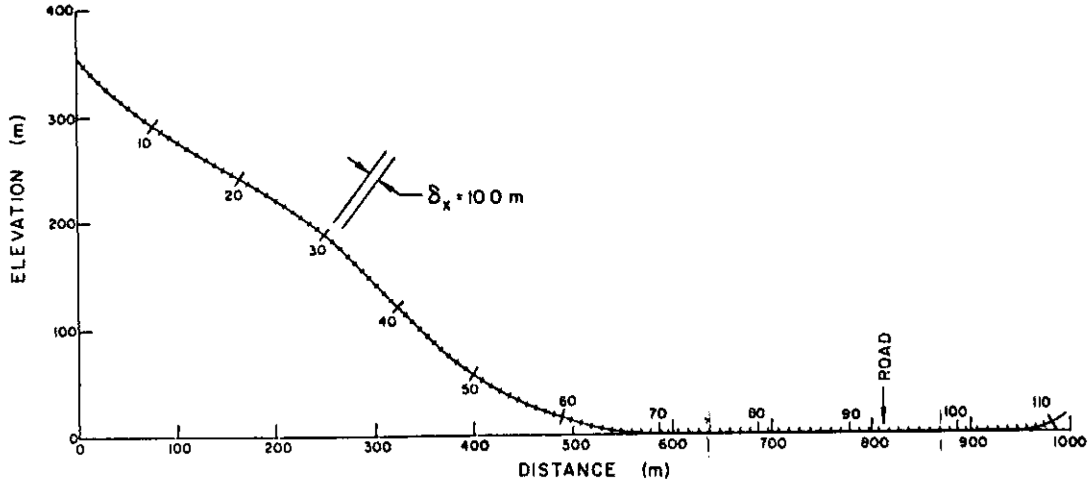


FIGURE 2: IRONTON PARK AVALANCHE PATH PROFILE

**Table 1: Ironton Park element elevation change**

| ELEMENT | ELEVATION CHANGE (m) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 to 8 | 6.5 | 6.2 | 6.2 | 6.2 | 6.5 | 6.2 | 6.2 | 6.2 |
| 9 to 16 | 6.2 | 6.5 | 6.5 | 6.2 | 5.8 | 6.0 | 5.8 | 5.6 |
| 17 to 24 | 5.6 | 5.2 | 4.6 | 4.2 | 4.2 | 4.6 | 5.2 | 4.6 |
| 25 to 32 | 4.8 | 5.0 | 5.2 | 5.4 | 5.6 | 5.8 | 6.2 | 6.2 |
| 33 to 40 | 6.2 | 6.2 | 6.5 | 6.8 | 7.4 | 7.2 | 7.2 | 7.1 |
| 41 to 48 | 7.0 | 7.0 | 7.0 | 6.6 | 6.2 | 6.2 | 6.0 | 6.0 |
| 49 to 56 | 5.6 | 5.2 | 5.2 | 5.0 | 4.6 | 4.2 | 4.2 | 4.0 |
| 57 to 64 | 4.0 | 3.6 | 3.6 | 3.4 | 3.4 | 3.0 | 2.8 | 2.5 |
| 65 to 72 | 2.2 | 1.8 | 1.6 | 1.4 | 1.2 | 2.0 | 1.3 | 0.8 |
| 73 to 80 | 0.4 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 81 to 88 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 89 to 96 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 97 to 104 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 105 to 110 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | | |

In the Ironton Park example there are no elements exhibiting what is termed adverse slope, which is slope that the avalanche must climb. In specifying adverse slope the elevation change (as in Table 1) is listed with negative values. Having determined the elevation change for each element of the profile, the remaining steps involve preparation of the computer input data.

## AVALNCH INPUT FORMAT

Input in the line sequence given below may be submitted either as a data file, or as a sequence of cards depending upon user preference.

### Line 1: FORMAT (40A2)

Columns 1-80: Title and identification information

### Line 2: FORMAT (2I10, 5F10.0, I10)

Columns 1-10: IBAR - number of cells in the slope-parallel direction; maxiumum is 200, unless program is changed.

Columns 11-20: JBAR - number of cells normal to the path; maximum is 2, unless program is changed.

Columns 21-30: DX - dimension of cell along path (m).

Columns 31-40: DY - dimension of cell normal to path (m).

Columns 41-50: YU - kinematic viscosity ($m^2s^{-1}$).

Columns 51-60: FK - friction coefficient; if given zero value here, then must input an array of friction coefficients for each cell (Line 5).

Columns 61-70: TF - avalanche flow time (s).

Columns 71-80: NP - number of cycles between extended printouts.

### Line 3: FORMAT (8F10.0)

Columns 1-10: thickness of avalanche slab in cell #1 (m).

Columns 11-20: thickness of avalanche slab in cell #2 (m).

Columns 21-30: thickness of avalanche slab in cell #3 (m).

Columns 31-40: thickness of avalanche slab in cell #4 (m).

Columns 41-50: thickness of avalanche slab in cell #5 (m).

Columns 51-60: thickness of avalanche slab in cell #6 (m).

Columns 61-70: thickness of avalanche slab in cell #7 (m).

Columns 71-80: thickness of avalanche slab in cell #8 (m).

Must continue this listing on succeeding lines until IBAR entries are specified, including zero-thickness cells.

### Line 4: FORMAT (8F10.0)

Columns 1-10: change in elevation of cell #1 (m).

Columns 11-20: change in elevation of cell #2 (m).

Columns 21-30: change in elevation of cell #3 (m).

Columns 31-40: change in elevation of cell #4 (m).

Columns 41-50: change in elevation of cell #5 (m).

Columns 51-60: change in elevation of cell #6 (m).

Columns 61-70: change in elevation of cell #7 (m).

Columns 71-80: change in elevation of cell #8 (m).

Must continue this listing on succeeding lines until IBAR entries are specified.

### Line 5: FORMAT (8F10.0)

(This set of data is required if FK=0 on Line 2)
Columns 1-10: friction coefficient for cell #1
Columns 11-20: friction coefficient for cell #2
Columns 21-30: friction coefficient for cell #3
Columns 31-40: friction coefficient for cell #4
Columns 41-50: friction coefficient for cell #5
Columns 51-60: friction coefficient for cell #6
Columns 61-70: friction coefficient for cell #7
Columns 71-80: friction coefficient for cell #8

Must continue this listing on succeeding lines until IBAR entries are specified.

This completes specification of input data for program AVALNCH. Examples of input data, and of program output for the Ironton Park avalanche path are published (Lang, etal. 1979).

## PROGRAM AVALNCH INTERNAL LOGIC

The original developers of program SOLA-SURF (Hirt etal., 1975) have a complete discussion on the logic of the program, which will not be repeated herein. In summary, the logic is shown by a flow chart of the program (Figure 3). Distinction between a cycle (CYCLE) and an iteration (ITER) is indicated on the flow chart. Section 1000, which pertains to a CYCLE, contains the complete Navier-Stohes equations. Section 3000, which pertains to ITER, contains simplified linear equations for small perturbation of parameters in order to achieve conservation of mass. The boundary condition section 2000, is basically the only section that must be changed in order to apply the program to different problem types. For example, for impact problems, the velocities in cells that represent a barrier must be zeroed in section 2000. However, if forces on a barrier are to be computed, then equations for this should be placed in section 4280 just prior to outputing data for the current CYCLE. In the case of impact problems it is also necessary to work

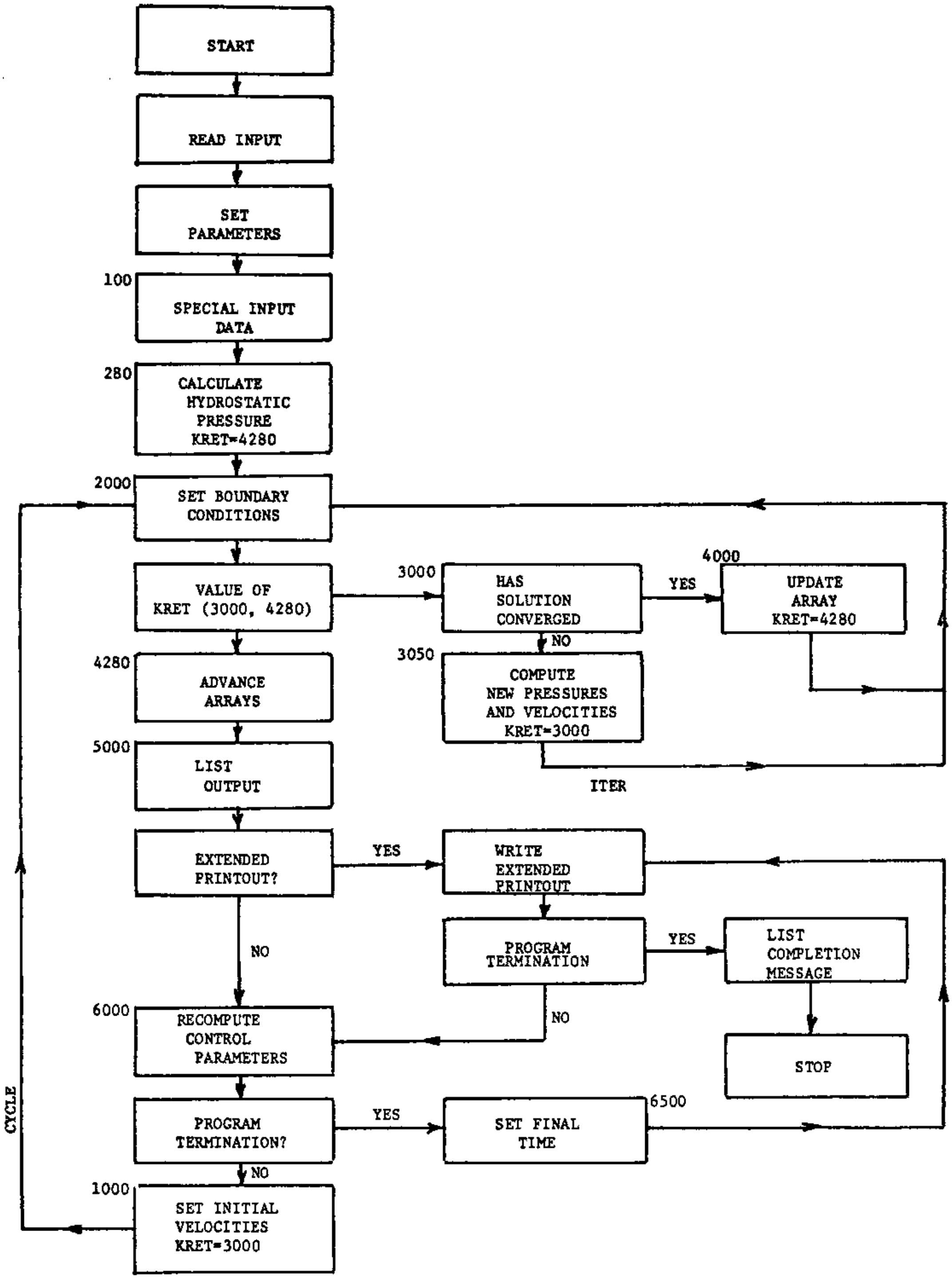


FIGURE 3: PROGRAM AVALNCH FLOW CHART

with more rectangular arrays, than with AVALNCH, in order to account for vertical variations in the flow. To specify a more rectangular array , not only must IBAR and JBAR be changed, but also the array specifications in the COMMON block at the start of the program must be changed. Application of program AVALNCH (in modified form) to impact analysis is reported by Lang and Brown, 1980. A listing of program AVALNCH that operates on a MELCOM 70 Computer System (Mitsubishi Electric, Japan) is given in Table 2.

Program AVALNCH has been used to model not only snow avalanches, but also other natural phenomona that involve transient fluid dynamics. For example, the mud flows associated with the 1980 eruption of Mt. St. Helens have been evaluated using the program (Lang, Dent, 1983). Also, a large volume rockslide that occurred in southwestern Montana following an earthquake in 1959, has been successfully modeled (Trunk, Dent, Lang, 1983).

**TABLE 2: Listing of Program AVALNCH for operation on a MELCOM 70 Computer System.**

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890101

C   *  *  *  *  PROGRAM AVALNCH  *  *  *  *
      DIMENSION U(202,4),V(202,4),UN(202,4),VN(202,4),P(202,4),
     1H(202),HN(202),FR(202),JT(202),GX(202),GY(202),NAME(40)
      READ(7,1) NAME
      WRITE(6,2)
      WRITE(6,3) NAME
C   *  *  *  *  READ INITIAL DATA  *  *  *  *
      READ(7,19) IBAR,JBAR,DX,DY,YU,FK,TF,NP
      WRITE(6,5) IBAR,JBAR,DX,DY,YU,FK,TF,NP
    1 FORMAT(40A2)
    2 FORMAT(1H1)
    3 FORMAT(5X,40A2)
    4 FORMAT(8F10.0)
    5 FORMAT(1H ,1X,'IBAR=',I4,2X,'JBAR=',I3,2X,'DX=',F6.2,2X,'DY=',
     1F5.2,2X,'YU=',F4.2,2X,'FK=',F4.2,2X,'TF=',F5.0,2X,'NP=',I4)
    6 FORMAT(1H0,35X,'FLOW HEIGHT')
    7 FORMAT(8F10.3)
    8 FORMAT(1H0,25X,'ELEVATION CHANGE FOR EACH CELL')
    9 FORMAT(1H0,25X,'BOUNDARY FRICTION COEFFICIENTS')
   10 FORMAT(1H0,25X,'SLOPE-PARALLEL GRAVITY COMPONENTS')
   11 FORMAT(1H0,25X,'SLOPE-NORMAL GRAVITY COMPONENTS')
   12 FORMAT(1H0,30X,'END OF INPUT DATA')
   13 FORMAT(2X,'CYCLE=',I4,2X,'ITER=',I3,2X,'DELT=',1PE9.2,2X,
     1'TIME=',E9.2,2X,'FVOL=',E9.2,2X,'UMAX=',E9.2,2X,'UEDG=',
     2E9.2,2X,'LDEG=',I3)
   14 FORMAT(6X,'I',7X,'J',8X,'U',13X,'V',13X,'P',13X,'H',9X,
     1'SUR CELL')
   15 FORMAT(4X,I3,5X,I3,4(4X,1PE10.3),6X,I2)
   16 FORMAT(5X,'PROBLEM RUNNING TIME EXCEEDED-CALCULATIONS STOPPED')
   17 FORMAT(5X,'AVALANCHE AT END OF GRID-CALCULATIONS TERMINATED')
   18 FORMAT(5X,'FLOW VELOCITY NEGLIGIBLE-CALCULATIONS TERMINATED')
   19 FORMAT(2I10,5F10.0,I10)
C   *  *  SET PARAMETERS  *  *
      IMAX=IBAR+2
      JMAX=JBAR+2
      RDX=1.0/DX
      RDY=1.0/DY
      IM1=IBAR+1
      JM1=JBAR+1
      DM=DY/100.
      DT=1.0
      T=0.0
      FLG=0.0
      UEDG1=0.0
      AB=1.0
      NC=0
      ITER=0
      IND=0
      LDEG=0
      G=9.806
```

```
         1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901
      OMG=1.7
      EPSI=.001
      ALPHA=0.1
      GAMMA=0.1
      DZRO=1.0
      BETA=OMG/(2.*DT*(RDX**2+RDY**2))
      IF(NP.EQ.1) ICP=2
      DO 100 I=1,IMAX
      H(I)=0.0
      HN(I)=0.0
      FR(I)=0.0
      JT(I)=0
      GX(I)=0.0
      GY(I)=0.0
      DO 100 J=1,JMAX
      U(I,J)=0.0
      V(I,J)=0.0
      UN(I,J)=0.0
      VN(I,J)=0.0
      P(I,J)=0.0
  100 CONTINUE
C   *   *   *   *  SPECIAL INPUT DATA  *   *   *   *
C
C   *   * FLOW HEIGHT   *   *
      READ(7,4)(H(I),I=2,IM1)
      WRITE(6,6)
      WRITE(6,7)(H(I),I=2,IM1)
C   *   * ELEVATION CHANGE FOR EACH CELL  *   *
      READ(7,4)(HN(I),I=2,IM1)
      WRITE(6,8)
      WRITE(6,7)(HN(I),I=2,IM1)
C   *   * FRICTION COEFFICIENTS *   *
      IF(FK.GT.0.0) GO TO 120
      READ(7,4)(FR(I),I=2,IM1)
      GO TO 130
  120 DO 125 I=2,IM1
  125 FR(I)=FK
  130 CONTINUE
      WRITE(6,9)
      WRITE(6,7)(FR(I),I=2,IM1)
C   *   * GRAVITY COMPONENTS  *   *
      DO 150 I=2,IM1
      SI=HN(I)/DX
      CO=SQRT(1.0-SI*SI)
      GX(I)=G*SI
  150 GY(I)=-G*CO
      WRITE(6,10)
      WRITE(6,7)(GX(I),I=2,IM1)
      WRITE(6,11)
      WRITE(6,7)(GY(I),I=2,IM1)
```

```
         1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901

C   *  *  END OF INPUT DATA   *   *
      WRITE(6,12)
C   *  *  SET CELL NUMBER OF FLOW HEIGHT  *   *
      DO 240 I=2,IM1
      JT(I)=INT(H(I)*RDY+0.001)+2
      IF(JT(I).GT.JM1) JT(I)=JM1
  240 HN(I)=0.0
      H(1)=H(2)
      H(IMAX)=H(IM1)
      JT(1)=JT(2)
      JT(IMAX)=JT(IM1)
C   *  *  CALCULATE HYDROSTATIC PRESSURE  *   *
      DO 280 I=2,IM1
      JT1=JT(I)
      DO 280 J=2,JT1
  280 P(I,J)=-GY(I)*(H(I)-(FLOAT(J)-1.5)*DY)
      ASSIGN 4280 TO KRET
      GO TO 2000
C   *  *  START CYCLE OF COMPUTATIONS  *   *
 1000 CONTINUE
      ITER=0
      FLG=1.0
      ASSIGN 3000 TO KRET
C   *  *  COMPUTE TEMPORARY U AND V VELOCITIES  *   *
      DO 1100 I=2,IM1
      JT1=JT(I)
      DO 1100 J=2,JT1
      FUX=((UN(I,J)+UN(I+1,J))*(UN(I,J)+UN(I+1,J))+ALPHA*ABS(UN(I,J)
     1+UN(I+1,J))*(UN(I,J)-UN(I+1,J))-(UN(I-1,J)+UN(I,J))*(UN(I-1,J)
     2+UN(I,J))-ALPHA*ABS(UN(I-1,J)+UN(I,J))*(UN(I-1,J)-UN(I,J)))/(4
     3.0*DX)
      FUY=((VN(I,J)+VN(I+1,J))*(UN(I,J)+UN(I,J+1))+ALPHA*ABS(VN(I,J)
     1+VN(I+1,J))*(UN(I,J)-UN(I,J+1))-(VN(I,J-1)+VN(I+1,J-1))*(UN(I,
     2J-1)+UN(I,J))-ALPHA*ABS(VN(I,J-1)+VN(I+1,J-1))*(UN(I,J-1)-UN(I
     3,J)))/(4.0*DY)
      FVX=((UN(I,J)+UN(I,J+1))*(VN(I,J)+VN(I+1,J))+ALPHA*ABS(UN(I,J)
     1+UN(I,J+1))*(VN(I,J)-VN(I+1,J))-(UN(I-1,J)+UN(I-1,J+1))*(VN(I-
     21,J)+VN(I,J))-ALPHA*ABS(UN(I-1,J)+UN(I-1,J+1))*(VN(I-1,J)-VN(I
     3,J)))/(4.0*DX)
      FVY=((VN(I,J)+VN(I,J+1))*(VN(I,J)+VN(I,J+1))+ALPHA*ABS(VN(I,J)
     1+VN(I,J+1))*(VN(I,J)-VN(I,J+1))-(VN(I,J-1)+VN(I,J))*(VN(I,J-1)
     2+VN(I,J))-ALPHA*ABS(VN(I,J-1)+VN(I,J))*(VN(I,J-1)-VN(I,J)))/(4
     3.0*DY)
      VISX=YU*((UN(I+1,J)-2.*UN(I,J)+UN(I-1,J))/DX**2+(UN(I,J+1)-2.*
     1UN(I,J)+UN(I,J-1))/DY**2)
      VISY=YU*((VN(I+1,J)-2.*VN(I,J)+VN(I-1,J))/DX**2+(VN(I,J+1)-2.*V
     1N(I,J)+VN(I,J-1))/DY**2)
      U(I,J)=UN(I,J)+DT*((P(I,J)-P(I+1,J))*RDX+GX(I)-FUX-FUY+VISX)
 1100 V(I,J)=VN(I,J)+DT*((P(I,J)-P(I,J+1))*RDY+GY(I)-FVX-FVY+VISY)
C   *  *  SET BOUNDARY CONDITIONS  *   *
```

```
         1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901
 2000 CONTINUE
      HN(1)=HN(2)
      HN(IMAX)=HN(IM1)
      JT(1)=JT(2)
      JT(IMAX)=JT(IM1)
C   *   *  LEFT WALL RIGID AND SLIP FREE  *   *
C   *   *  RIGHT WALL CONTINUOUS OUTFLOW  *   *
      DO 2200 J=1,JMAX
      U(1,J)=0.0
      V(1,J)=V(2,J)
      IF(ITER.GT.0) GO TO 2200
      U(IM1,J)=U(IBAR,J)
 2200 V(IMAX,J)=V(IM1,J)
C   *   *  TOP WALL CONTINUOUS OUTFLOW  *   *
C   *   *  BOTTOM WALL RIGID WITH FRICTION  *   *
      DO 2500 I=1,IMAX
      IF(ITER.GT.0) GO TO 2400
      V(I,JM1)=V(I,JBAR)
      U(I,JMAX)=U(I,JM1)
 2400 V(I,1)=0.0
 2500 U(I,1)=U(I,2)*(1.0-2.0*FR(I))
C   *   *  FREE SURFACE BOUNDARY CONDITIONS  *   *
      DO 2700 I=2,IM1
      JT1=JT(I)
      IF(JT(I+1).LT.JT(I)) U(I,JT1)=U(I,JT1-1)
      V(I,JT1)=V(I,JT1-1)-DY*RDX*(U(I,JT1)-U(I-1,JT1))
 2700 U(I,JT1+1)=U(I,JT1)
      GO TO KRET,(3000,4280)
 3000 CONTINUE
C   *   *  CHECK FOR CONVERGENCE  *   *
      IF(FLG.EQ.0.) GO TO 4000
      ITER=ITER+1
      IF(ITER.LT.500) GO TO 3050
      T=1.E+10
      GO TO 4000
 3050 FLG=0.0
C   *   *  COMPUTE UPDATED CELL PRESSURE AND VELOCITIES  *   *
      JB1=2
      DO 3500 I=2,IM1
      JT1=JT(I)
      DO 3500 J=2,JT1
      IF(JT1.EQ.JB1) GO TO 3060
      IF(J.NE.JB1.AND.J.NE.JT1) GO TO 3200
      IF(J.EQ.JT1) GO TO 3100
      GO TO 3200
 3060 CONTINUE
      F=V(I,J)+DY*RDX*(U(I,J)-U(I-1,J))
      DFDP=DT*RDY*(1.0+2.0*DY**2*RDX**2)
      DP1=-F/DFDP
 3100 ETA=DY/(HN(I)-(FLOAT(JT1)-2.5)*DY)
```

```
         1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901

      DP=(1.0-ETA)*P(I,JT1-1)-P(I,JT1)
      GO TO 3300
 3200 D=RDX*(U(I,J)-U(I-1,J))+RDY*(V(I,J)-V(I,J-1))
      IF(ABS(D/DZRO).GE.EPSI) FLG=1.0
      DP=-BETA*D
 3300 P(I,J)=P(I,J)+DP
      U(I,J)=U(I,J)+DT*RDX*DP
      U(I-1,J)=U(I-1,J)-DT*RDX*DP
      V(I,J)=V(I,J)+DT*RDY*DP
 3500 V(I,J-1)=V(I,J-1)-DT*RDY*DP
      GO TO 2000
 4000 CONTINUE
C   *   *  COMPUTE NEW POSITION OF TOP SURFACE  *   *
      DO 4100 I=2,IM1
      JT1=JT(I)
      HV=RDY*(HN(I)-FLOAT(JT1-2)*DY)
      UAV=0.5*(U(I-1,JT1)+U(I,JT1))
      H(I)=HN(I)*FV1/FV+DT*(HV*V(I,JT1)+(1.0-HV)
     1*V(I,JT1-1)-0.5*RDX*(UAV*HN(I+1)+GAMMA*ABS(UAV)
     2*(HN(I)-HN(I+1))-UAV*HN(I-1)-GAMMA*ABS(UAV)
     3*(HN(I-1)-HN(I))))
 4100 CONTINUE
C   *   *  CALCULATE CELL  IN WHICH SURFACE IS LOCATED  *   *
C   *   *  AND UPDATE ARRAY  *   *
      DO 4250 I=2,IM1
      IF(H(I).LT.DM) H(I)=0.0
      JT(I)=INT(H(I)*RDY+0.001)+2
      IF(JT(I).GT.JM1) JT(I)=JM1
 4250 CONTINUE
      ASSIGN 4280 TO KRET
      GO TO 2000
 4280 CONTINUE
C   *   *  CALCULATE TOTAL FLUID VOLUME  *   *
      FV=0.0
      DO 4300 I=2,IM1
 4300 FV=FV+H(I)*DX
      IF(NC.EQ.0) FV1=FV
C   *   *  FIND LEADING AND TRAILING EDGES OF AVALANCHE  *   *
      LDEG1=LDEG
      I=IBAR
 4400 IF(H(I).GT.DM) GO TO 4500
      I=I-1
      GO TO 4400
 4500 LDEG=I
      I=2
 4600 IF(H(I).GT.DM) GO TO 4700
      I=I+1
      GO TO 4600
 4700 KTEG=I
C   *   *  COMPUTE LEADING EDGE VELOCITY  *   *
```

```
         1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901

      IF(LDEG.EQ.LDEG1) GO TO 4800
      IF(NC.GT.0) UEDG=DX/TC
      IF(NC.EQ.0) UEDG=5.0
      TC=DT
      INFLO=1
      IF(UEDG.GT.UEDG1) UEDG1=UEDG
      GO TO 4910
 4800 TC=TC+DT
      INFLO=INFLO+1
C   *  *  ADVANCE U,V,H ARRAYS  *   *
 4910 UM=0.0
      VM=0.0
      DO 4900 I=1,IMAX
      DO 4900 J=1,JMAX
      UA=ABS(U(I,J))
      VA=ABS(V(I,J))
      PA=ABS(P(I,J))
      IF(UA.GT.1.0E+04) U(I,J)=0.0
      UN(I,J)=U(I,J)
      IF(VA.GT.1.0E+04) V(I,J)=0.0
      VN(I,J)=V(I,J)
      IF(PA.LT.1.0E-16) P(I,J)=0.0
 4900 HN(I)=H(I)
      DO 4950 I=KTEG,LDEG
      DO 4950 J=2,JM1
      UT=ABS(UN(I,J))
      VT=ABS(VN(I,J))
      IF(UT.GT.UM) UM=UT
 4950 IF(VT.GT.VM) VM=VT
C   *  *  LIST VELOCITY,PRESSURE,AND SURFACE POSITION  *   *
 5000 WRITE(6,13)NC,ITER,DT,T,FV,UM,UEDG,LDEG
      IF(NC.EQ.ICP) GO TO 5030
      GO TO 6000
 5030 ICP=ICP+NP
 5060 CONTINUE
      WRITE(6,14)
      DO 5250 I=1,IMAX
      JT1=JT(I)
      JT2=JT1+1
      DO 5250 J=1,JT2
      WRITE(6,15)I,J,U(I,J),V(I,J),P(I,J),H(I),JT1
 5250 CONTINUE
      GO TO (6000,6520,6530,6540),IND
C   *  *  RECOMPUTE CONTROL PARAMETERS  *   *
 6000 IF(NC.EQ.0) GO TO 6300
      DTX=DX/UM
      DTY=DY/VM
      DT=AMIN1(DTX,DTY)/3.0
      IF(ITER.LT.10) DT=1.5*DT
      YU1=YU-1.0E-06
```

```
         1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901

      IF(YU1.LT.0.0) GO TO 6300
      DET=(DX*DY)**2/(2.*YU*(DX**2+DY**2))
      IF(DT.LT.DET) GO TO 6300
      DT=0.9*DET
 6300 T=T+DT
      IF(NC.EQ.0) GO TO 6400
      DAX=UM*DT/DX
      DAY=VM*DT/DY
      ALPHA=1.35*AMAX1(DAX,DAY)
      IF(ALPHA.GT.1.0) ALPHA=0.95
      GAMMA=ALPHA
      BETA=OMG/(2.*DT*(RDX**2+RDY**2))
C    *   *  TEST FOR PROGRAM TERMINATION  *   *
 6400 IF(T.GT.TF) IND=2
      IF(H(IBAR).GT.DM) IND=3
      IF(INFLO.EQ.50) IND=4
      UEDG2=0.05*UEDG1
      IF(UEDG.LT.UEDG2) IND=4
      IF(IND.GT.1) GO TO 6500
      IF(NC.LT.3) GO TO 6440
      AA=1.0+20.0*EXP(-1.25*UEDG)
      DO 6430 I=2,IM1
 6430 FR(I)=FR(I)*AA/AB
      AB=AA
 6440 NC=NC+1
      GO TO 1000
 6500 T=T-DT
      GO TO 5060
 6520 WRITE(6,16)
      GO TO 6600
 6530 WRITE(6,17)
      GO TO 6600
 6540 WRITE(6,18)
 6600 STOP
      END

*** PRINT  END ***
```

## PROGRAM AVALNCH WITH BI-VISCOUS OPTION

From observations of avalanche flows it has long been recognized that avalanches decelerate at increasing rates as they come to a stop. Thus, disaggregated snow has the general fluid property of thixotropy, and in order to numerically model avalanche flow it is necessary to account for this thixotropic condition in some way. In program AVALNCH this was accomplished by increasing the friction coefficient, $f$ , as the flow slowed down. To do this it was necessary to prescribe a speed below which the increase starts, and a rate of increase of $f$ as the speed continues to decrease. From observations by Schaerer (1975) on a number of avalanches at Rodgers' Pass, Canada, a transition in the speed range $5-10 \mathrm{ms}^{-1}$ was noted, so a transition speed of $5.0 \mathrm{ms}^{-1}$ was used in the program. No data existed from which the rate of increase of $f$ with decrease in speed could be established. So numerical experimentation was carried out using different rates, until the stopping distances of several avalanches were matched with site measurements. It was determined that a geometric-progression type increase in $f$ as speed decreased below the transition speed of $5 \mathrm{ms}^{-1}$ was needed. The resulting expression for $f$ that is used in program AVALNCH is

$$f = f_o(1+20e^{-1.25v})$$

where $f_o$ is the nominal high speed value of the friction coefficient. In program AVALNCH this mechanism is referred to as the "fast-stop" option.

Recognizing, physically, that the surface friction is unlikely to increase in a geometric progression with decrease in speed, a more rational expression of the fast-stop or material locking property was saught. The physical process of fluid locking is known as the Bingham fluid property, or that the fluid is a "Bingham material". With regard to snow, the effort to apply the Bingham material concept was two-fold.

One effort was to computer model small volume snow flows that had been run experimentally, in order to determine the motion of decelerating snow (Dent, Lang 1982). Results from this work showed that a biviscous rather than a Bingham representation of the snow flow best fit the experimental data. The basic difference between the two mechanisms is that the Bingham fluid has infinite viscosity below the cutoff shear stress, $\tau_o$ , whereas the biviscous model has finite, but larger viscosity below the cutoff shear stress (Figure 4). In computer modeling the controlled volume snow flows kinematic viscosities of $\nu = 0.002 m^2 s^{-1}$ and $\nu' = 0.10 m^2 s^{-1}$ gave good correspondance between the data.

A second effort was to independently measure the cutoff shear stress, $\tau_o$ , that separates the two regions of the biviscous model. Results from the controlled volume snow flow tests gave $\frac{\tau_o}{\rho} = 2.2 m^2 s^{-2}$. Simple laboratory tests on similar type snow gave corresponding values for $\frac{\tau_o}{\rho}$ (Lang, Dent, 1983). Although the velocities of the two types of experiments were vastly different, the velocity gradients were of the same magnitude, which attributes to the close correspondance between the results of the two experiments.

From the controlled volume snow flow results we note a factor of 50 between the values of $\nu$ and $\nu'$. These findings indicate that it is the viscosity of the snow that changes value with speed, rather than the surface friction. So the question is raised if program AVALNCH can be modified to incorporate viscosity rather than friction change, as the flow velocity decreases, and yet retain the flow characteristics of snow.

To incorporate the biviscous option in program AVALNCH was a relatively simple process. The friction mechanism was changed from a partial-slip to a no-slip boundary condition at the lower boundary. This eliminates the need to input a value for friction. In its' place

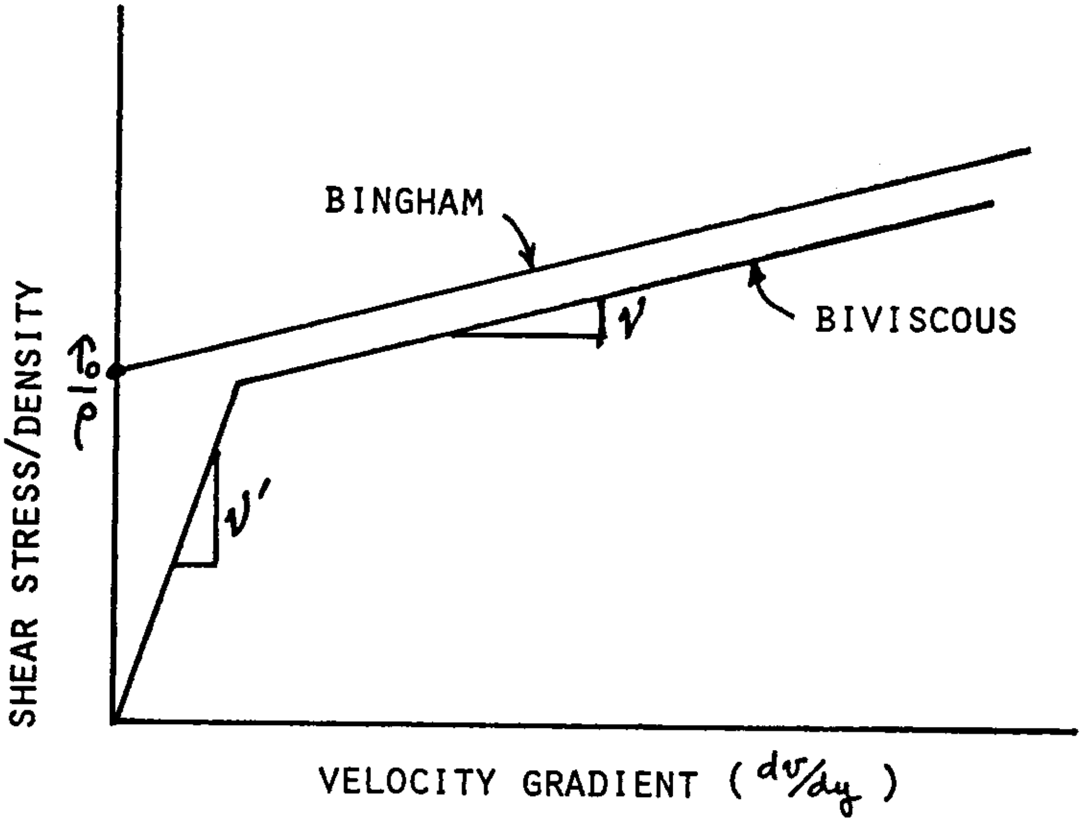


FIGURE 4: BINGHAM AND BIVISCOUS FLUID REPRESENTATION

on the input statement the multiplicative factor between $\nu$ and $\nu'$ is specified, and is designated XYU. The high stress viscosity $\nu$ is input as before, and is designated YU. The fast-stop instructions previously used were replaced by a set of instructions that test the speed of fluid in each cell along the avalanche path. If the cell has a velocity less than $5ms^{-1}$, then the viscosity is set at the low stress value (XYU times YU). If the velocity is greater than $5ms^{-1}$, then the viscosity is set at the high stress value (YU). These few changes were what were needed to modify the code to the biviscous option. What remained was to determine the values of $\nu$ and $\nu'$ in order to model an avalanche flow.

To evaluate the viscosities the Ironton Park avalanche was selected. In increasing the friction from $f=0.5$ (partial-slip) to $f=1.0$ (no-slip), the viscosity must be decreased from $\nu=0.5m^2s^{-1}$ for the partial-slip (fast-stop) case. In addition the multiplicative factor between $\nu$ and $\nu'$ needs to be specified. Although a factor of 50 between $\nu$ and $\nu'$ was noted earilier for the small volume snow flow experiments, this factor is known not to be sensitive, so only a factor of 10 was used in the Ironton Park evaluation. From runs of Ironton Park, it was determined that $\nu=0.23m^2s^{-1}$ yielded a duplication in runout distance and maximum velocity between the fast-stop and biviscous options. A comparison of velocities along the path is also necessary in order to establish correspondance (Figure 5), which is also achieved to a close approximation.

The corrections described for the biviscous version of AVALNCH are sufficient if the modeling is with constant values of friction and viscosity along the avalanche path. However, for some avalanches the viscosity should be variable along the path, as in the case of coastal avalanches, where part of the runout may be on dry snow, and another part on wet snow. With the no-slip boundary condition, which is the usual boundary condition in fluid dynamics, it must be assumed that

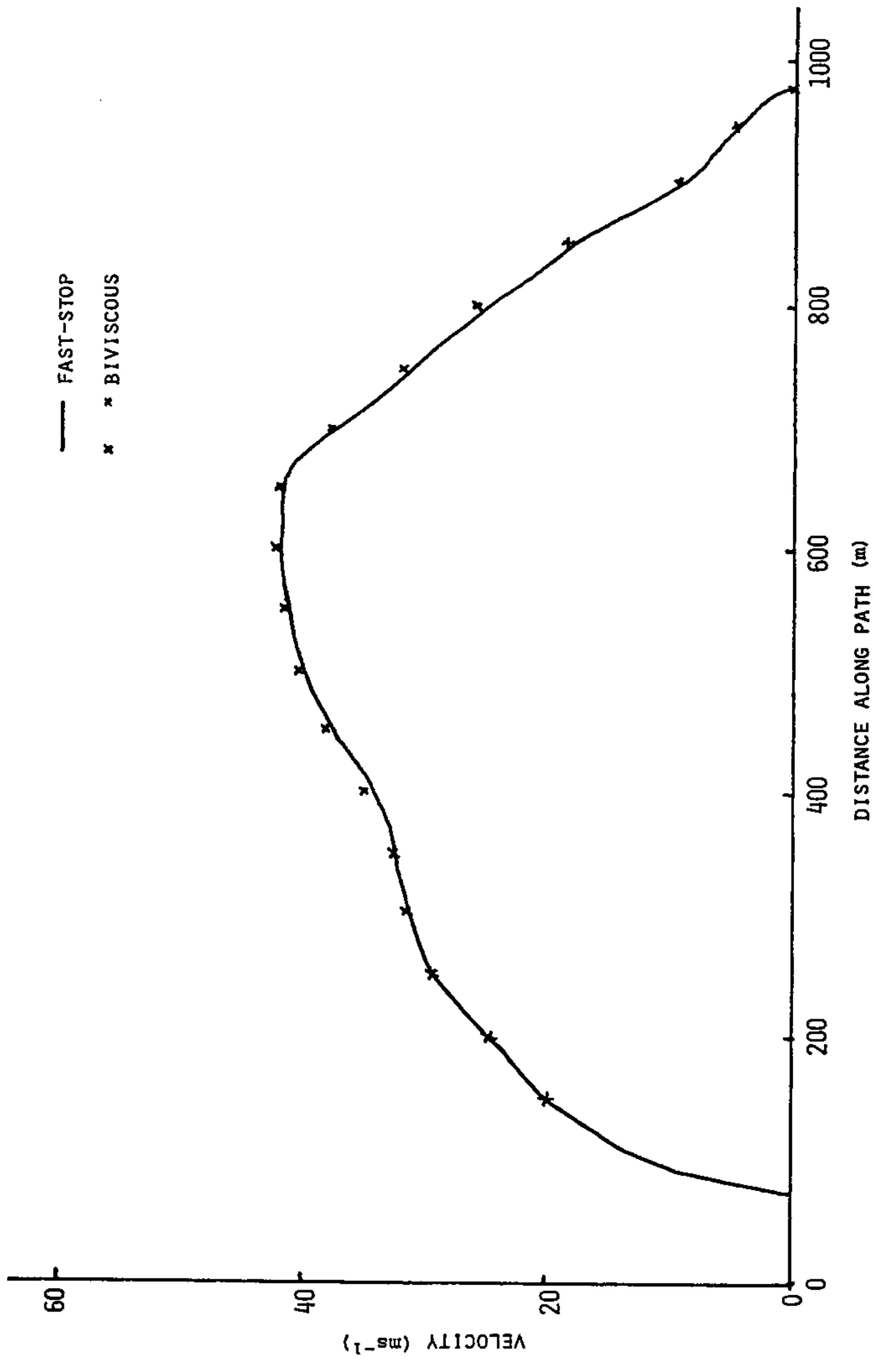


FIGURE 5: IRONTON PARK AVALANCHE VELOCITY PROFILES USING FAST-STOP AND BIVISCOUS OPTIONS.

the increased resistance of the wet snow produces additional internal mixing in the moving snow, and hence an increase in viscous dissipation. To incorporate a variable viscosity into the code it is necessary to set up viscosity as a one-dimensional array so that values of $\nu$ may be assigned to each cell along the path. This was done in a way similar to the variable friction option in the fast-stop version of the code. The change is reflected in the modified input format for the biviscous case, which is listed below.

## BIVISCOUS-AVALNCH INPUT FORMAT

Input may be by a data file or by a sequence of cards depending upon user preference.

### Line 1: FORMAT (40A2)

Columns 1-80: title and identification information

### Line 2: FORMAT (2I10, 5F10.0, I10)

Columns 1-10: IBAR - number of cells in the slope-parallel direction; maximum is 200, unless program is changed.

Columns 11-20: JBAR - number of cells normal to the path; maximum is 2, unless program is changed.

Columns 21-30: DX - dimension of cell along path (m).

Columns 31-40: DY - dimension of cell normal to path (m).

Columns 41-50: YU - high shear stress kinematic viscosity ($m^2s^{-1}$).

Columns 51-60: XYU - multiplicative factor for low shear stress viscosity.

Columns 61-70: TF - avalanche flow time (s).

Columns 71-80: NP - number of cycles between extended printouts.

### Line 3: FORMAT (8F10.0)

The thickness of the avalanche slab at initial release is listed in the same format as on page 9.

### Line 4: FORMAT (8F10.0)

The elevation change of each cell along the avalanche path is listed in the same format as on page 9.

### Line 5: FORMAT (8F10.0)

If YU=0 on Line 2, then viscosity must be specified for each cell along the path. This listing of viscosity is in the same format as for friction coefficients on page 10.

A listing of the Biviscous version of program AVALNCH is given in Table 3. The flow chart for the Biviscous version is the same as for the fast-stop version, as listed on page 11. Both the fast-stop and biviscous coefficient selection statements are in the 6000 section of the program.

**TABLE 3 : Listing of Biviscous version of AVALNCH.**

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890

C  *  *  BIAV: PROGRAM AVALNCH WITH BIVISCOUS OPTION  *  *
      DIMENSION U(202,4),V(202,4),UN(202,4),VN(202,4),P(202,4),
     1H(202),HN(202),XU(202,2),JT(202),GX(202),GY(202),NAME(40)
      READ(7,1) NAME
      WRITE(6,2)
      WRITE(6,3) NAME
C  *  *  *  *  READ INITIAL DATA  *  *  *  *
      READ(7,19) IBAR,JBAR,DX,DY,YU,XYU,TF,NP
      WRITE(6,5) IBAR,JBAR,DX,DY,YU,XYU,TF,NP
      READ(7,4) DTZ,UZ
      WRITE(6,20) DTZ,UZ
    1 FORMAT(40A2)
    2 FORMAT(1H1)
    3 FORMAT(5X,40A2)
    4 FORMAT(8F10.0)
    5 FORMAT(1H ,1X,'IBAR=',I4,2X,'JBAR=',I3,2X,'DX=',F6.2,2X,'DY=',
     1F5.2,2X,'YU=',F4.2,2X,'XYU=',F4.1,2X,'TF=',F5.0,2X,'NP=',I4)
    6 FORMAT(1H0,35X,'FLOW HEIGHT')
    7 FORMAT(8F10.3)
    8 FORMAT(1H0,25X,'ELEVATION CHANGE FOR EACH CELL')
    9 FORMAT(1H0,25X,'VISCOSITY VALUES FOR EACH CELL')
   10 FORMAT(1H0,25X,'SLOPE-PARALLEL GRAVITY COMPONENTS')
   11 FORMAT(1H0,25X,'SLOPE-NORMAL GRAVITY COMPONENTS')
   12 FORMAT(1H0,30X,'END OF INPUT DATA')
   13 FORMAT(2X,'CYCLE=',I4,2X,'ITER=',I3,2X,'DELT=',1PE9.2,2X,
     1'TIME=',E9.2,2X,'FVOL=',E9.2,2X,'UMAX=',E9.2,2X,'UEDG=',
     2E9.2,2X,'LDEG=',I3)
   14 FORMAT(6X,'I',7X,'J',8X,'U',13X,'V',13X,'P',13X,'H',9X,
     1'SUR CELL')
   15 FORMAT(4X,I3,5X,I3,4(4X,1PE10.3),6X,I2)
   16 FORMAT(5X,'PROBLEM RUNNING TIME EXCEEDED-CALCULATIONS STOPPED')
   17 FORMAT(5X,'AVALANCHE AT END OF GRID-CALCULATIONS TERMINATED')
   18 FORMAT(5X,'FLOW VELOCITY NEGLIGIBLE-CALCULATIONS TERMINATED')
   19 FORMAT(2I10,5F10.0,I10)
   20 FORMAT(5X,'DTZ=',F6.3,5X,'UZ=',F6.3)
C  *  *  SET PARAMETERS  *  *
      IMAX=IBAR+2
      JMAX=JBAR+2
      RDX=1.0/DX
      RDY=1.0/DY
      IM1=IBAR+1
      JM1=JBAR+1
      DM=DY/100.
      DT=DTZ
      T=0.0
      FLG=0.0
      UEDG1=0.0
      ICP=NP
      NC=0
      ITER=0
```

```
                 1         2         3         4         5         6         7
LINE.   1234567890123456789012345678901234567890123456789012345678901234567890

              IND=0
              LDEG=0
              G=9.806
              OMG=1.7
              EPSI=.001
              ALPHA=0.1
              GAMMA=0.1
              DZRO=1.0
              BETA=OMG/(2.*DT*(RDX**2+RDY**2))
              IF(NP.EQ.1) ICP=2
              DO 100 I=1,IMAX
              H(I)=0.0
              HN(I)=0.0
              XU(I,1)=0.0
              XU(I,2)=0.0
              JT(I)=0
              GX(I)=0.0
              GY(I)=0.0
              DO 100 J=1,JMAX
              U(I,J)=0.0
              V(I,J)=0.0
              UN(I,J)=0.0
              VN(I,J)=0.0
              P(I,J)=0.0
          100 CONTINUE
        C   *   *   *   *  SPECIAL INPUT DATA  *   *   *   *
        C
        C   *   *  FLOW HEIGHT  *   *
              READ(7,4)(H(I),I=2,IM1)
              WRITE(6,6)
              WRITE(6,7)(H(I),I=2,IM1)
        C   *   *  ELEVATION CHANGE FOR EACH CELL  *   *
              READ(7,4)(HN(I),I=2,IM1)
              WRITE(6,8)
              WRITE(6,7)(HN(I),I=2,IM1)
        C   *   *  INITIALIZE VISCOSITY  *   *
              IF(YU.EQ.0.0) GO TO 125
              DO 120 I=2,IM1
              XU(I,1)=YU
          120 XU(I,2)=YU
              GO TO 130
          125 READ(7,4)(XU(I,1),I=2,IM1)
              DO 127 I=2,IM1
          127 XU(I,2)=XU(I,1)
              WRITE(6,9)
              WRITE(6,7)(XU(I,1),I=2,IM1)
          130 CONTINUE
        C   *   *  GRAVITY COMPONENTS  *   *
              DO 150 I=2,IM1
              SI=HN(I)/DX
```

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890
      CO=SQRT(1.0-SI*SI)
      GX(I)=G*SI
  150 GY(I)=-G*CO
      WRITE(6,10)
      WRITE(6,7)(GX(I),I=2,IM1)
      WRITE(6,11)
      WRITE(6,7)(GY(I),I=2,IM1)
C   *   *  END OF INPUT DATA  *    *
      WRITE(6,12)
C   *   *  SET CELL NUMBER OF FLOW HEIGHT  *   *
      DO 240 I=2,IM1
      JT(I)=INT(H(I)*RDY+0.001)+2
      IF(JT(I).GT.JM1) JT(I)=JM1
  240 HN(I)=0.0
      H(1)=H(2)
      H(IMAX)=H(IM1)
      JT(1)=JT(2)
      JT(IMAX)=JT(IM1)
C   *   *  CALCULATE HYDROSTATIC PRESSURE  *   *
      DO 280 I=2,IM1
      JT1=JT(I)
      DO 280 J=2,JT1
  280 P(I,J)=-GY(I)*(H(I)-(FLOAT(J)-1.5)*DY)
      ASSIGN 4280 TO KRET
      GO TO 2000
C   *   *  START CYCLE OF COMPUTATIONS  *   *
 1000 CONTINUE
      ITER=0
      FLG=1.0
      ASSIGN 3000 TO KRET
C   *   *  COMPUTE TEMPORARY U AND V VELOCITIES  *   *
      DO 1100 I=2,IM1
      JT1=JT(I)
      DO 1100 J=2,JT1
      FUX=((UN(I,J)+UN(I+1,J))*(UN(I,J)+UN(I+1,J))+ALPHA*ABS(UN(I,J)
     1+UN(I+1,J))*(UN(I,J)-UN(I+1,J))-(UN(I-1,J)+UN(I,J))*(UN(I-1,J)
     2+UN(I,J))-ALPHA*ABS(UN(I-1,J)+UN(I,J))*(UN(I-1,J)-UN(I,J)))/(4
     3.0*DX)
      FUY=((VN(I,J)+VN(I+1,J))*(UN(I,J)+UN(I,J+1))+ALPHA*ABS(VN(I,J)
     1+VN(I+1,J))*(UN(I,J)-UN(I,J+1))-(VN(I,J-1)+VN(I+1,J-1))*(UN(I,
     2J-1)+UN(I,J))-ALPHA*ABS(VN(I,J-1)+VN(I+1,J-1))*(UN(I,J-1)-UN(I
     3,J)))/(4.0*DY)
      FVX=((UN(I,J)+UN(I,J+1))*(VN(I,J)+VN(I+1,J))+ALPHA*ABS(UN(I,J)
     1+UN(I,J+1))*(VN(I,J)-VN(I+1,J))-(UN(I-1,J)+UN(I-1,J+1))*(VN(I-
     21,J)+VN(I,J))-ALPHA*ABS(UN(I-1,J)+UN(I-1,J+1))*(VN(I-1,J)-VN(I
     3,J)))/(4.0*DX)
      FVY=((VN(I,J)+VN(I,J+1))*(VN(I,J)+VN(I,J+1))+ALPHA*ABS(VN(I,J)
     1+VN(I,J+1))*(VN(I,J)-VN(I,J+1))-(VN(I,J-1)+VN(I,J))*(VN(I,J-1)
     2+VN(I,J))-ALPHA*ABS(VN(I,J-1)+VN(I,J))*(VN(I,J-1)-VN(I,J)))/(4
     3.0*DY)
```

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890

      VISX=XU(I,1)*((UN(I+1,J)-2.*UN(I,J)+UN(I-1,J))/DX**2+(UN(I,J+1)
     1-2.*UN(I,J)+UN(I,J-1))/DY**2)
      VISY=XU(I,1)*((VN(I+1,J)-2.*VN(I,J)+VN(I-1,J))/DX**2+(VN(I,J+1)
     1-2.*VN(I,J)+VN(I,J-1))/DY**2)
      U(I,J)=UN(I,J)+DT*((P(I,J)-P(I+1,J))*RDX+GX(I)-FUX-FUY+VISX)
 1100 V(I,J)=VN(I,J)+DT*((P(I,J)-P(I,J+1))*RDY+GY(I)-FVX-FVY+VISY)
C   *   *  SET BOUNDARY CONDITIONS  *   *
 2000 CONTINUE
      HN(1)=HN(2)
      HN(IMAX)=HN(IM1)
      JT(1)=JT(2)
      JT(IMAX)=JT(IM1)
C   *   *  LEFT WALL RIGID AND SLIP FREE  *   *
C   *   *  RIGHT WALL CONTINUOUS OUTFLOW  *   *
      DO 2200 J=1,JMAX
      U(1,J)=0.0
      V(1,J)=V(2,J)
      IF(ITER.GT.0) GO TO 2200
      U(IM1,J)=U(IBAR,J)
 2200 V(IMAX,J)=V(IM1,J)
C   *   *  TOP WALL CONTINUOUS OUTFLOW  *   *
C   *   *  BOTTOM WALL RIGID WITH FRICTION  *   *
      DO 2500 I=1,IMAX
      IF(ITER.GT.0) GO TO 2400
      V(I,JM1)=V(I,JBAR)
      U(I,JMAX)=U(I,JM1)
 2400 V(I,1)=0.0
 2500 U(I,1)=-U(I,2)
C   *   *  FREE SURFACE BOUNDARY CONDITIONS  *   *
      DO 2700 I=2,IM1
      JT1=JT(I)
      IF(JT(I+1).LT.JT(I)) U(I,JT1)=U(I,JT1-1)
      V(I,JT1)=V(I,JT1-1)-DY*RDX*(U(I,JT1)-U(I-1,JT1))
 2700 U(I,JT1+1)=U(I,JT1)
      GO TO KRET,(3000,4280)
 3000 CONTINUE
C   *   *  CHECK FOR CONVERGENCE  *   *
      IF(FLG.EQ.0.) GO TO 4000
      ITER=ITER+1
      IF(ITER.LT.500) GO TO 3050
      T=1.E+10
      GO TO 4000
 3050 FLG=0.0
C   *   *  COMPUTE UPDATED CELL PRESSURE AND VELOCITIES  *   *
      JB1=2
      DO 3500 I=2,IM1
      JT1=JT(I)
      DO 3500 J=2,JT1
      IF(JT1.EQ.JB1) GO TO 3060
      IF(J.NE.JB1.AND.J.NE.JT1) GO TO 3200
```

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890

      IF(J.EQ.JT1) GO TO 3100
      GO TO 3200
 3060 CONTINUE
      F=V(I,J)+DY*RDX*(U(I,J)-U(I-1,J))
      DFDP=DT*RDY*(1.0+2.0*DY**2*RDX**2)
      DP1=-F/DFDP
 3100 ETA=DY/(HN(I)-(FLOAT(JT1)-2.5)*DY)
      DP=(1.0-ETA)*P(I,JT1-1)-P(I,JT1)
      GO TO 3300
 3200 D=RDX*(U(I,J)-U(I-1,J))+RDY*(V(I,J)-V(I,J-1))
      IF(ABS(D/DZRO).GE.EPSI) FLG=1.0
      DP=-BETA*D
 3300 P(I,J)=P(I,J)+DP
      U(I,J)=U(I,J)+DT*RDX*DP
      U(I-1,J)=U(I-1,J)-DT*RDX*DP
      V(I,J)=V(I,J)+DT*RDY*DP
 3500 V(I,J-1)=V(1,J-1)-DT*RDY*DP
      GO TO 2000
 4000 CONTINUE
C   * *    COMPUTE NEW POSITION OF TOP SURFACE  *   *
      DO 4100 I=2,IM1
      JT1=JT(I)
      HV=RDY*(HN(I)-FLOAT(JT1-2)*DY)
      UAV=0.5*(U(I-1,JT1)+U(I,JT1))
      H(I)=HN(I)*FV1/FV+DT*(HV*V(I,JT1)+(1.0-HV)
     1*V(I,JT1-1)-0.5*RDX*(UAV*HN(I+1)+GAMMA*ABS(UAV)
     2*(HN(I)-HN(I+1))-UAV*HN(I-1)-GAMMA*ABS(UAV)
     3*(HN(I-1)-HN(I))))
 4100 CONTINUE
C   * *    CALCULATE CELL IN WHICH SURFACE IS LOCATED  *   *
C   * *    AND UPDATE ARRAY  *   *
      DO 4250 I=2,IM1
      IF(H(I).LT.DM) H(I)=0.0
      JT(I)=INT(H(I)*RDY+0.001)+2
      IF(JT(I).GT.JM1) JT(I)=JM1
 4250 CONTINUE
      ASSIGN 4280 TO KRET
      GO TO 2000
 4280 CONTINUE
C   * *    CALCULATE TOTAL FLUID VOLUME  *   *
      FV=0.0
      DO 4300 I=2,IM1
 4300 FV=FV+H(I)*DX
      IF(NC.EQ.0) FV1=FV
C   * *    FIND LEADING AND TRAILING EDGES OF AVALANCHE  *   *
      LDEG1=LDEG
      I=IBAR
 4400 IF(H(I).GT.DM) GO TO 4500
      I=I-1
      GO TO 4400
```

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890

 4500 LDEG=I
      I=2
 4600 IF(H(I).GT.DM) GO TO 4700
      I=I+1
      GO TO 4600
 4700 KTEG=I
C   *   *  ADVANCE U,V,H ARRAYS  *   *
 4910 UM=0.0
      VM=0.0
      DO 4900 I=1,IMAX
      DO 4900 J=1,JMAX
      UA=ABS(U(I,J))
      VA=ABS(V(I,J))
      PA=ABS(P(I,J))
      IF(UA.GT.1.0E+04) U(I,J)=0.0
      UN(I,J)=U(I,J)
      IF(VA.GT.1.0E+04) V(I,J)=0.0
      VN(I,J)=V(I,J)
      IF(PA.LT.1.0E-16) P(I,J)=0.0
 4900 HN(I)=H(I)
      DO 4950 I=KTEG,LDEG
      DO 4950 J=2,JM1
      UT=ABS(UN(I,J))
      VT=ABS(VN(I,J))
      IF(UT.GT.UM) UM=UT
 4950 IF(VT.GT.VM) VM=VT
C   *   *  COMPUTE LEADING EDGE VELOCITY  *   *
      IF(LDEG.EQ.LDEG1) GO TO 4800
      IF(NC.GT.0) UEDG=DX/TC
      IF(NC.LT.10) UEDG=UM
      IF(NC.EQ.0) UEDG=UZ
      TC=DT
      INFLO=1
      IF(UEDG.GT.UEDG1) UEDG1=UEDG
      GO TO 4810
 4800 TC=TC+DT
      INFLO=INFLO+1
 4810 CONTINUE
C   *   *  LIST VELOCITY,PRESSURE,AND SURFACE POSITION  *   *
 5000 WRITE(6,13)NC,ITER,DT,T,FV,UM,UEDG,LDEG
      IF(NC.EQ.ICP) GO TO 5030
      GO TO 6000
 5030 ICP=ICP+NP
 5060 CONTINUE
      WRITE(6,14)
      DO 5250 I=1,IMAX
      JT1=JT(I)
      JT2=JT1+1
      DO 5250 J=1,JT2
      WRITE(6,15)I,J,U(I,J),V(I,J),P(I,J),H(I),JT1
```

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890:

 5250 CONTINUE
      GO TO (6000,6520,6530,6540),IND
C   *  *   RECOMPUTE CONTROL PARAMETERS  *   *
 6000 IF(NC.EQ.0) GO TO 6300
      DTX=DX/UM
      DTY=DY/VM
      DT=AMIN1(DTX,DTY)/3.0
      IF(ITER.LT.10) DT=1.5*DT
      YU=XU(LDEG,1)
      YU1=YU-1.0E-06
      IF(YU1.LT.0.0) GO TO 6300
      DET=(DX*DY)**2/(2.*YU*(DX**2+DY**2))
      IF(DT.LT.DET) GO TO 6300
      DT=0.9*DET
 6300 T=T+DT
      IF(NC.EQ.0) GO TO 6400
      DAX=UM*DT/DX
      DAY=VM*DT/DY
      ALPHA=1.35*AMAX1(DAX,DAY)
      IF(ALPHA.GT.1.0) ALPHA=0.95
      GAMMA=ALPHA
      BETA=OMG/(2.*DT*(RDX**2+RDY**2))
C    *   *   TEST FOR PROGRAM TERMINATION  *   *
 6400 IF(T.GT.TF) IND=2
      IF(H(IBAR).GT.DM) IND=3
      IF(INFLO.EQ.50) IND=4
      UEDG2=0.05*UEDG1
      IF(UEDG.LT.UEDG2) IND=4
      IF(IND.GT.1) GO TO 6500
      IF(NC.LT.10) GO TO 6440
      DO 6430 I=KTEG,LDEG
      IF(U(I,2).LT.5.0) XU(I,1)=XYU*XU(I,2)
 6430 IF(U(I,2).GE.5.0) XU(I,1)=XU(I,2)
 6440 NC=NC+1
      GO TO 1000
 6500 T=T-DT
      GO TO 5060
 6520 WRITE(6,16)
      GO TO 6600
 6530 WRITE(6,17)
      GO TO 6600
 6540 WRITE(6,18)
 6600 STOP
      END

*** PRINT  END ***
```

## EQUILIBRIUM FLOW MODELS WITH MATERIAL LOCKING

The equations for equilibrium flow modeling of snow avalanche runout, originally developed by Voellmy (1955), were later adapted to computer simulation by Cheng and Perla (1979). In the computer representation the avalanche path is divided into straight-line segments of varying length and integrated forms of Voellmy's acceleration equation are applied to the flow in each segment. In the computer based formulation the Voellmy acceleration equation is expressed by

$$a = g(\sin\theta - \mu\cos\theta) - \frac{D}{M}v^2$$

In this equation $\theta$ is the slope angle, $\mu$ is the coefficient of sliding friction, $g$ is the gravitational constant ($9.806 \mathrm{ms}^{-2}$), and $D/M$ is a drag coefficient. The stated ranges on the friction and drag parameters are:

$$0.1 \le \mu \le 0.5$$
$$10 \le M/D \le 10^4 \quad (\mathrm{m})$$

which must be selected based upon the site specific conditions of each avalanche path. This broad range in parameter selection has long been a difficulty in application of the equations to different types of avalanche flow. Also inherent in this formulation is an instability that occurs when the slope angle equals the assumed friction angle of the snow.

In light of recent developments pertaining to snow avalanche flow, we consider modification of the Cheng-Perla computer program by attempting to incorporate the following conditions:

1. At low shear stress values in flowing snow the snow has a tendency to lock up; a characteristic of a thixotropic fluid.
2. Flow of avalanches involves bulk flow of the major portion of the moving material riding upon a high-velocity-gradient boundary layer of granularized snow. In the boundary layer the basic mechanism of energy dissipation is by viscous effects, as is typical of fluid processes in general.

The objective in incorporating these physical effects into the equilibrium flow model is to reduce the variation in parameter selection for different avalanche cases, and to eliminate the instability condition of the Voellmy equation.

## EQUATION FORMULATION

Using recently developed mechanics of snow flow (Dent, Lang. 1982) a model for flow incorporates a viscous boundary layer that supports the material bulk flow. We designate the depth of the bulk material by $h$, depth of the boundary layer by $\lambda$, and velocity of the bulk material by $v$ (Figure 6). Forces acting on the bulk material are gravitational, viscous, and assumed frictional (Figure 7), where $\tau$ is the viscous shear stress, and A is the area of contact between the mass segment and the boundary layer. With these forces acting the equation of motion of the mass segment is:

$$\Sigma F = ma = mg\sin\theta - \mu N - \tau A$$

Dividing by m and setting $N = mg\cos\theta$, an acceleration equation similar to that of Voellmy is obtained.

$$a = g(\sin\theta - \mu\cos\theta) - \frac{\tau A}{m}$$

The term $\frac{\tau A}{m}$ can be rewritten $\frac{\tau}{\rho} \cdot \frac{\rho A h}{m} \cdot \frac{1}{h}$ where $\rho$ is the density of the snow. But $\rho A h = m$, the mass of the bulk material, so for the equation of motion we have

$$a = g(\sin\theta - \mu\cos\theta) - \frac{1}{h}\frac{\tau}{\rho}$$

If we set $\frac{\tau}{\rho} = \frac{g}{\xi} v^2$ we have the Voellmy equation, were $\xi$ is a coefficient of dynamic resistance.

If the boundary layer is assumed to be a Newtonian fluid in a laminar flow regime, then

$$\frac{\tau}{\rho} = \nu \frac{dv}{dy}$$

and the velocity gradient is approximated by $\frac{dv}{dy} = \frac{v}{\lambda}$ to obtain

$$\frac{\tau}{\rho} = \nu \frac{v}{\lambda}$$

where $\nu$ is the kinematic viscosity. In this formulation $\lambda$ is the depth of the active boundary layer upon which the bulk of flowing snow

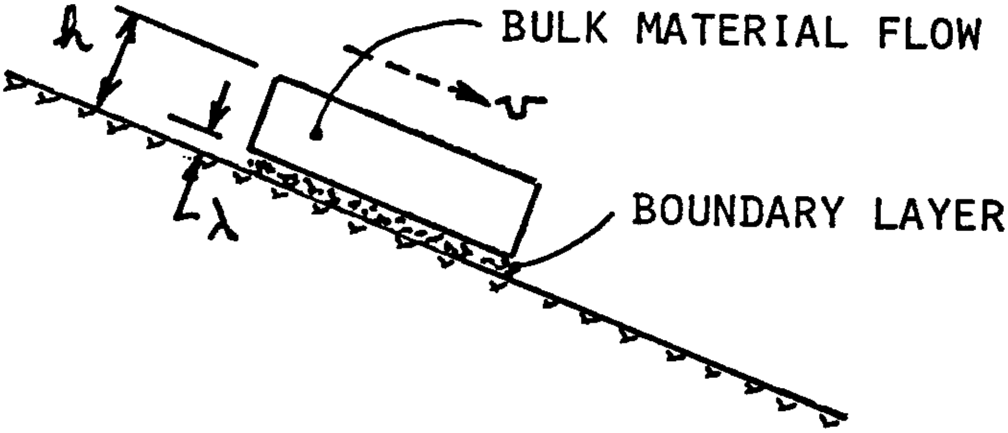


FIGURE 6: FLOWING SNOW CONFIGURATION

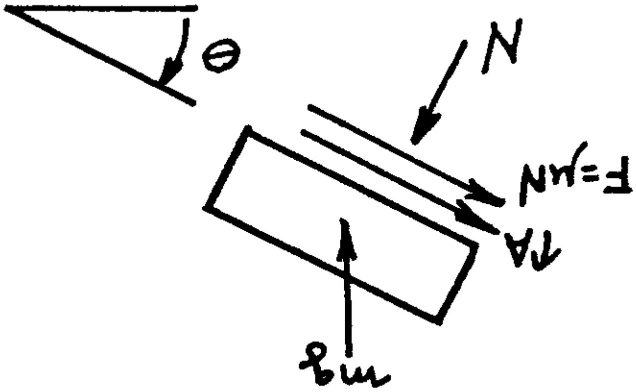


FIGURE 7: ELEMENT FORCES

rides. While the leading edge of an avalanche may exhibit strong mixing, the bulk of the dense flowing material is expected to be smooth, because of path smoothing at the leading edge. Presently, we have no basis for specifying $\lambda$, the depth of the boundary layer. In tests with small volumes of snow in controlled releases the depth observed was about 2cm for a flow depth of about 20cm. Depth of the boundary layer should depend upon the roughness over which the flow advances. For flow in pipes the depth of the boundary layer is related directly to wall roughness. In the case of avalanche flow the definition of roughness is not known. Is roughness to be related to undulations in the runout path, or to grain size in the boundary layer, or to fluxuations in the boundary layer profile? We might exclude path undulations on the basis of leading edge smoothing. However, to go beyond this to consider boundary layer profile fluxuations is more complex, and warrants experimental investigation, as has been done in the case of pipe flow. At the present time, we will express the boundary layer thickness as a fraction of the flow depth in the form

$$\lambda = r h$$

The acceleration equation under these assumptions becomes

$$a = g(\sin\theta - \mu\cos\theta) - \frac{\nu^*}{h^2} v$$

when $\nu^* = \frac{\nu}{r}$, is one of the basic parameters that must be evaluated. Setting the acceleration $a = v\frac{dv}{ds}$ and integrating, we obtain the following equation relating velocity to distance of travel, $s$,

$$\frac{1}{\beta^2}(\alpha - \beta v) - \frac{\alpha}{\beta^2}\ln(\alpha - \beta v) = s + C_1$$

where $C_1$ is a constant of integration, and

$$\alpha = g(\sin\theta - \mu\cos\theta) \quad , \quad \beta = \frac{\nu^*}{h^2}$$

Imposing the constraint condition that at $s=0, v=v_A$ leads to

$$s = \frac{1}{\beta^2}\left[\beta(v_A - v) + \alpha\ln\left(\frac{\alpha - \beta v_A}{\alpha - \beta v}\right)\right]$$

then, selecting at $S=L, v=v_B$, we obtain

$$\beta L = (v_A - v_B) + \frac{\alpha}{\beta}\ln\left(\frac{\alpha - \beta v_A}{\alpha - \beta v_B}\right)$$

which is a transcendental equation for velocity $v_B$, assuming that in a segment analysis that $v_A$ would be specified from the previous segment analysis, and L is the length of the segment under evaluation. Thus, to solve for $v_B$ in the above equation requires the use of a numerical methods algorithm.

We note that in the laminar flow assumption that the viscous drag term in the acceleration equation is proportional to velocity to the first power. This deviates from the original Voellmy assumption, which is based upon a turbulent assumption. To investigate the turbulent flow assumption, the Boussinesq formulation (Shames, 1982) is usually cited for the relationship between shear stress in the boundary layer and the velocity gradient, namely

$$\frac{\tau}{\rho} = \nu' \left(\frac{dv}{dy}\right)^2$$

where $\nu'$ is designated the kinematic eddy viscosity, which may be related to the fluid mixing length. Making the previously defined approxinations for the velocity gradient, the acceleration equation has the form

$$a = g(\sin\theta - \mu\cos\theta) - \frac{\nu^*}{h^3} v^2$$

where $\nu^* = \nu'/\kappa^2$. Here the viscous dissipation term is proportional to $v^2$, which is the same as in the Voellmy formulation. However, a difference here is that the viscous term is also proportional to $h^{-3}$ rather than to $h^{-1}$, as in the Voellmy equation. Writing for the acceleration $a = \frac{1}{2}\frac{dv^2}{ds}$, integrating, and imposing the constraint conditions, as before, the following segment equations are obtained:

$$v_B = \left[ \frac{\alpha}{\beta}(1 - e^{-2\beta L}) + v_A^2 e^{-2\beta L} \right]^{1/2}$$

Thus, in this case it is possible to express $v_B$ explicitely in terms of $v_A$ and L, the length of the segment. In a segment in which the flow stops ($v_B$=0) the runout distance is expressed by

$$s_f = \frac{1}{2\beta} \ln\left(1 - \frac{\beta}{\alpha} v_A^2\right)$$

In these equations

$$\alpha = g(\sin\theta - \mu\cos\theta) \quad , \quad \beta = \frac{\nu^*}{h^3}$$

so that $S_f$ is not finite for the case $\alpha = 0$, which occurs if $\mu = \tan\theta$, the afore mentioned singularity condition.

## MATERIAL LOCKING MECHANISM

It has long been recognized that flowing snow exhibits a thixotropic property that produces an accelerated slow-down as the avalanche comes to a stop. This locking property has been approximated by a biviscous model (Figure 8) in computer studies with multi-celled configurations (Dent, Lang, 1983). In these models the viscosity in all cells do not change simultaneously to produce a sudden change in the flow resistance. However, in the equilibrium flow models based upon Voellmy's equation only a single segment of material is used in each numerical step. Because of this simplification, it was decided that a simple biviscous approximation would be too abrupt, as the flow passed through the transition point. Instead a continuous variation in viscosity was opted for (dashed line, Figure 8), for which a functional representation is required. However, if a functional representation is selected for $\nu^*$, then the previously integrated equations are no longer applicable. Two possible recourses to this difficulty are: 1) to numerically integrate the acceleration equation in each segment analysis, or 2) to subdivide the segments into a step-wise linearized approximation. The first approach would be a radical divergence from the algorithm used by Cheng and Perla, so was discarded in favor of the second approach, which is only a modification to the Cheng-Perla program.

The next consideration is to decide upon an avalanche speed at which the viscosity transition is to occur. Data from Schaerer (1975), based upon observations of a number of avalanches at Rodger's Pass, Canada,

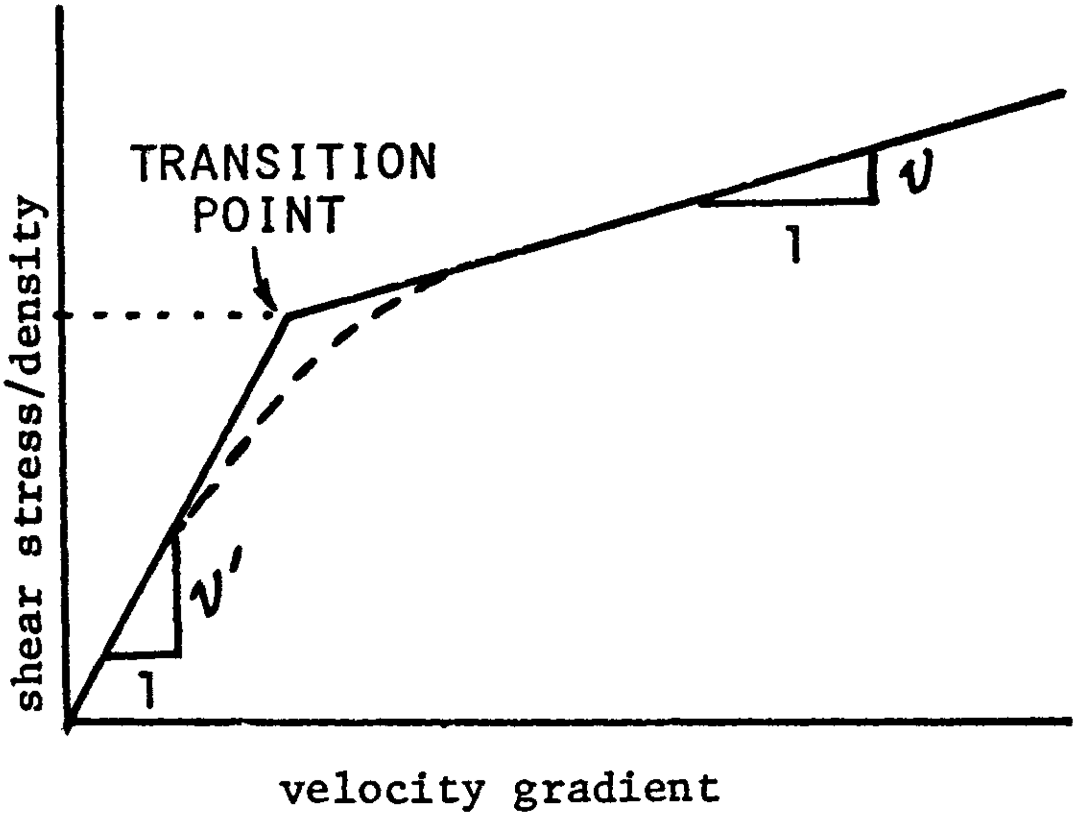


FIGURE 8: BIVISCOUS MATERIAL REPRESENTATION

place the transition in the speed range $10 > v > 5$ ms$^{-1}$. This speed range has also been used with success in the fast-stop option of program AVALNCH (Lang, Dawson, Martinelli, 1979). Based upon these observations the form of the function for $\nu^*$ is taken as

$$\nu^* = \nu_0 \left( 1 + C_0 e^{-1.25 v} \right)$$

The exponent coefficient 1.25 is sufficient for $\nu^* \to \nu_0$ for $v > 8.0$ ms$^{-1}$. The coefficients $\nu_0$ and $C_0$ must be determined, based upon numerical study of typical avalanche flows.

The other parameter, the friction coefficient $\mu$, in the Voellmy formulation, must be reconsidered in relation to the viscous mechanism that has now been defined. In granular material flow processes, the viscous (or velocity dependent) resistive processes are the dominant effect, and snow should be no exception. In the Voellmy formulation, since the viscous resistance is proportional to $v^2$, and hence decreases in effect at low speeds, the Coulomb type friction must become dominant at low speeds. However, in incorporating viscous intensification from material locking at low speeds, the importance of the $\mu$ type resistance should be less. In the case of avalanche flow the dominant effect of $\mu$ type friction should occur only at the start of the avalanche, when the motion is of a sliding type, until the snow blocks break up and a granular layer develops. From observations that avalanches seldom release on slopes less than 20 to 25°, a reasonable value of $\mu$ in the starting zone is $\mu = \tan 22° = 0.4$.

In the terminal flow regime, we expect that with the locking mechanism operating, that $\mu$ should have values smaller than those in the range specified in the original Voellmy case. Also, from measurements by Lang, Dent (1982), frictional resistance is noted to be a linear function of the overburden load, so that the modified form of the friction coefficient, designated, $\mu^*$, is

$$\mu^* = \mu_0 h$$

where $\mu_0$ must be determined from numerical evaluation of avalanche flows.

In summary, the acceleration equation we now use for numerical evaluation of segment kinematics is

$$a = g(\sin\theta - \mu^* \cos\theta) - \frac{\nu^*}{h^3} v^2$$

where $\mu^* = \mu_0 h$ and $\nu^* = \nu_0 (1 + C_0 e^{-1.25 v})$. In regions where $\nu^*$ is variable at low velocities, the segment length is decreased so that a slep-wise linear approximation can be made of the acceleration, so that velocity at the end of each segment may still be computed using the equation

$$v_B = \left[ \frac{\alpha}{\beta}\left(1 - e^{-2\beta L}\right) + v_A^2 e^{-2\beta L} \right]^{1/2}$$

However, the computation of the final runout distance is no longer based upon evaluation of $S_f$, as in the Voellmy approach. Instead, with the grid refinement that is used, runout distance is simply the grid location where the velocity of flow becomes negligible.

## PARAMETER EVALUATION: IRONTON PARK AVALANCHE

To proceed further, we evaluate $C_0$, $\nu_0$ and $\mu_0$ for a specific avalanche path for which the flow resistance is apparently constant over the entire path of runout. The Ironton Park path located in the San Juan Mountains in Colorado is used. Documentation of this path is given in detail by Lang, Dawson, Martinelli, 1979. For the Cheng-Perla evaluation (designated ACCEL hence forth) and the Biviscous-Equilibrium evaluation (designated BIEQ henceforth) an 11 segment approximation of the path was selected (Figure 9). The 11th segment is of a frozen lake bed, which is horizontal and extends for 350m, although only 50m of the segment is shown in Figure 9.

As a measure of comparison the ACCEL fit and the BIEQ fit are compared to corresponding results obtained from program AVALNCH. For example,

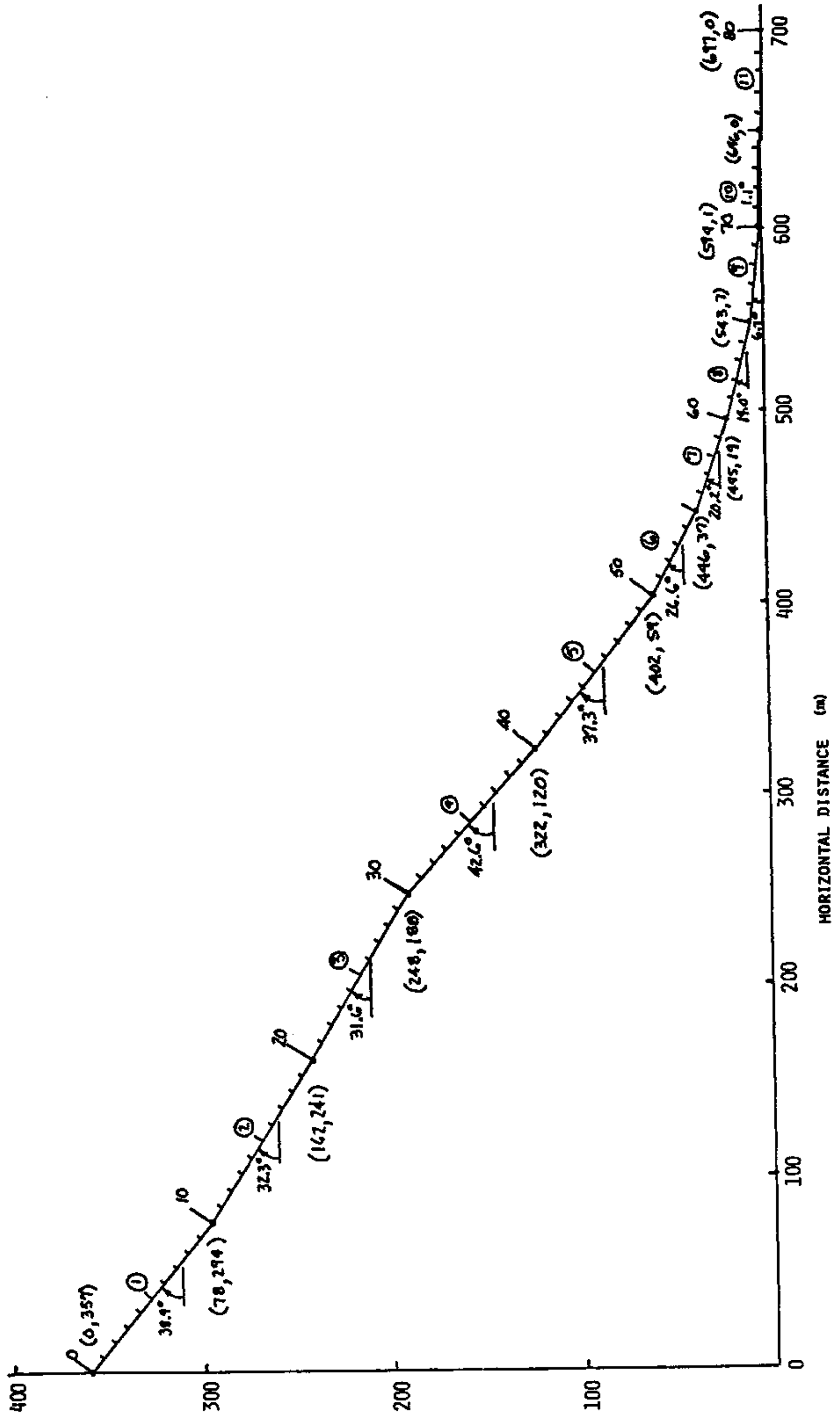


FIGURE 9: IRONTON PARK AVALANCHE PATH

for a 2.0m nominal depth starting zone of snow, program AVALNCH computes a runout distance 230m into segment 11, and a maximum speed of approximately $42 ms^{-1}$. This same level of performance is obtained with the ACCEL program with $\mu$=0.09 and $M/D$=400. Note, that with program AVALNCH the released snow was distributed in cells 1 through 7 with a leading edge taper (Figure 10). In the ACCEL program computation the first segment was taken as the last 30m of segment ①.

To evaluate the BIEQ parameters, the Ironton Park avalanche path was evaluated first with program AVALNCH for starting zone snow depths of $h$=0.5, 1.0, 1.5 and 2.0m. Parameterization with program AVALNCH, was $f$=0.5 and $\nu$=$0.5 m^2 s^{-1}$. The four starting zone depth cases were then run with BIEQ, adjusting the governing parameters in order to obtain a best-fit of all the cases. The intent was to evaluate $\mu_o$, $\nu_o$ and $C_o$ so that all the different depth cases were approximated by a single set of these parameters. With program AVALNCH the $h$=0.5m case is a sluff onto the bench of cells 20 to 30 (Figure 9). In program BIEQ the same type of effect was modeled, but without considering a detailed duplication between the programs. A single set of parameters that models all cases was found to be $\mu_o$=0.027, $\nu_o$=0.027 and $C_o$=500. Using these values the maximum velocities and corresponding runout distances are summarized in Table 4. Program ACCEL was run only for the $h$=2.0m case. Assuming that the primary function of these codes is to compute runout distance of the larger avalanches, then it is seen that all versions provide satisfactory results.

The distribution of velocity of the avalanche along the path can also be compared from the data that was obtained. The profiles are shown in Figures 11, 12 and 13 for $h$=1.0, 1.5 and 2.0m, respectively. It is seen that the BIEQ velocities are approximately 10 to 20 % higher than the AVALNCH values for the case $h$=1.0m, with runout distances

TABLE 4: Ironton Park Avalanche Path, computer program comparisons for different starting zone snow depths.

| STARTING ZONE NOMINAL SNOW DEPTH (m) | MAXIMUM VELOCITY ALONG PATH ($ms^{-1}$) | | | DISTANCE OF TRAVEL ALONG PATH (m) | | |
|---|---|---|---|---|---|---|
| | AVALNCH | BIEQ | ACCEL | AVALNCH | BIEQ | ACCEL |
| 2.0 | 42.0 | 39.2 | 42.9 | 980 | 970 | 975 |
| 1.5 | 27.8 | 27.4 | — | 800 | 810 | — |
| 1.0 | 13.0 | 15.4 | — | 710 | 720 | — |
| 0.5 | — | — | — | SLUFF | SLUFF | — |

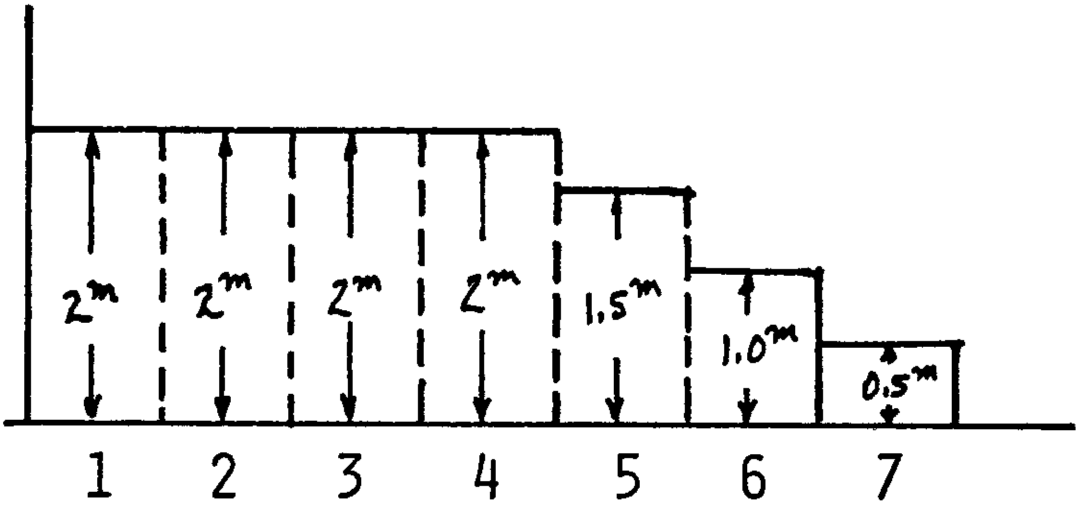


FIGURE 10: SNOW DISTRIBUTION IN THE STARTING ZONE OF THE IRONTON PARK AVALANCHE PATH.

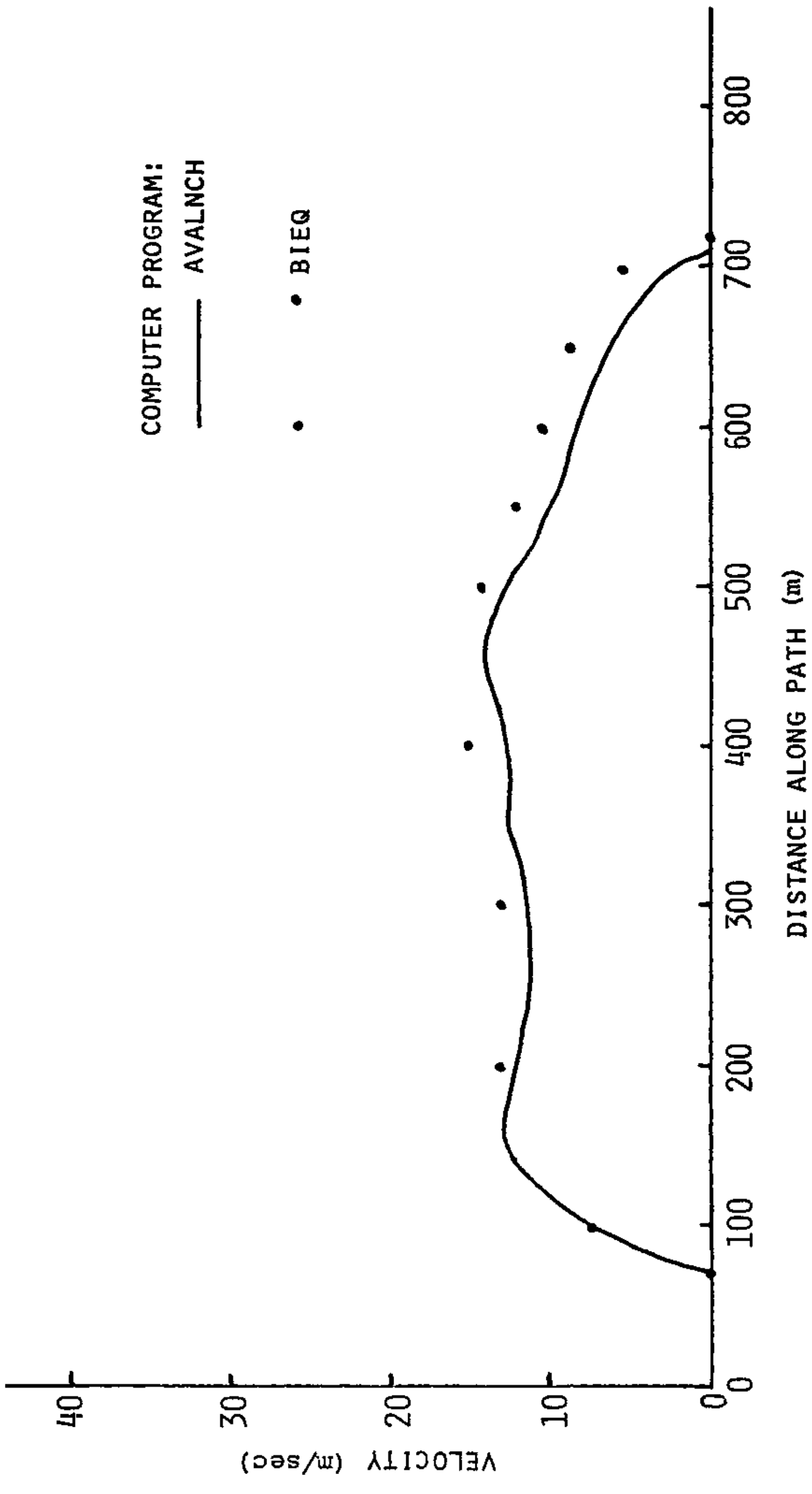


FIGURE 11: IRONTON PARK AVALANCHE PATH : VELOCITY PROFILE FOR STARTING ZONE SNOW DEPTH $h$=1.0m.

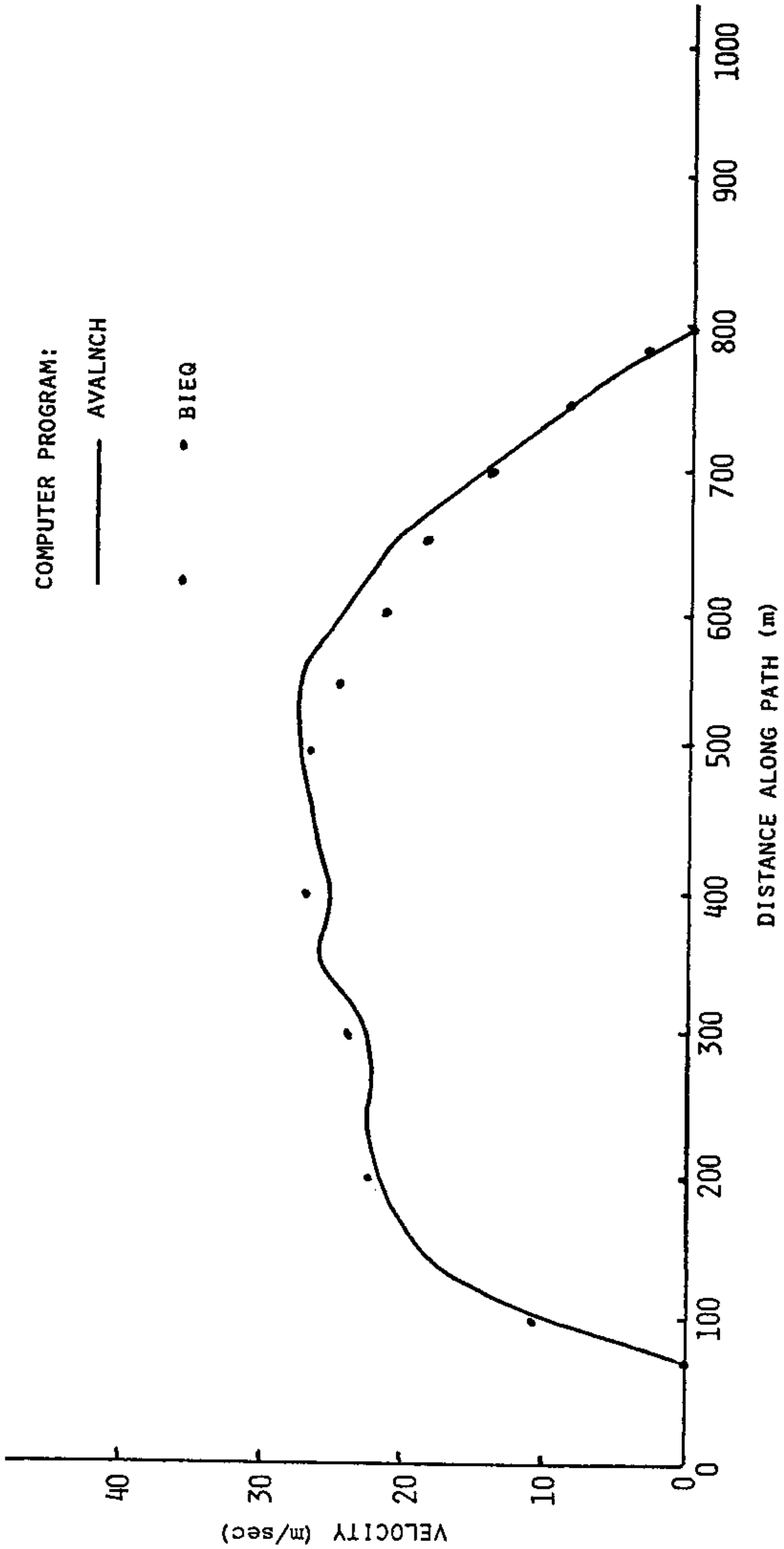


FIGURE 12: IRONTON PARK AVALANCHE PATH : VELOCITY PROFILE FOR STARTING ZONE SNOW DEPTH $h$ =1.5m.

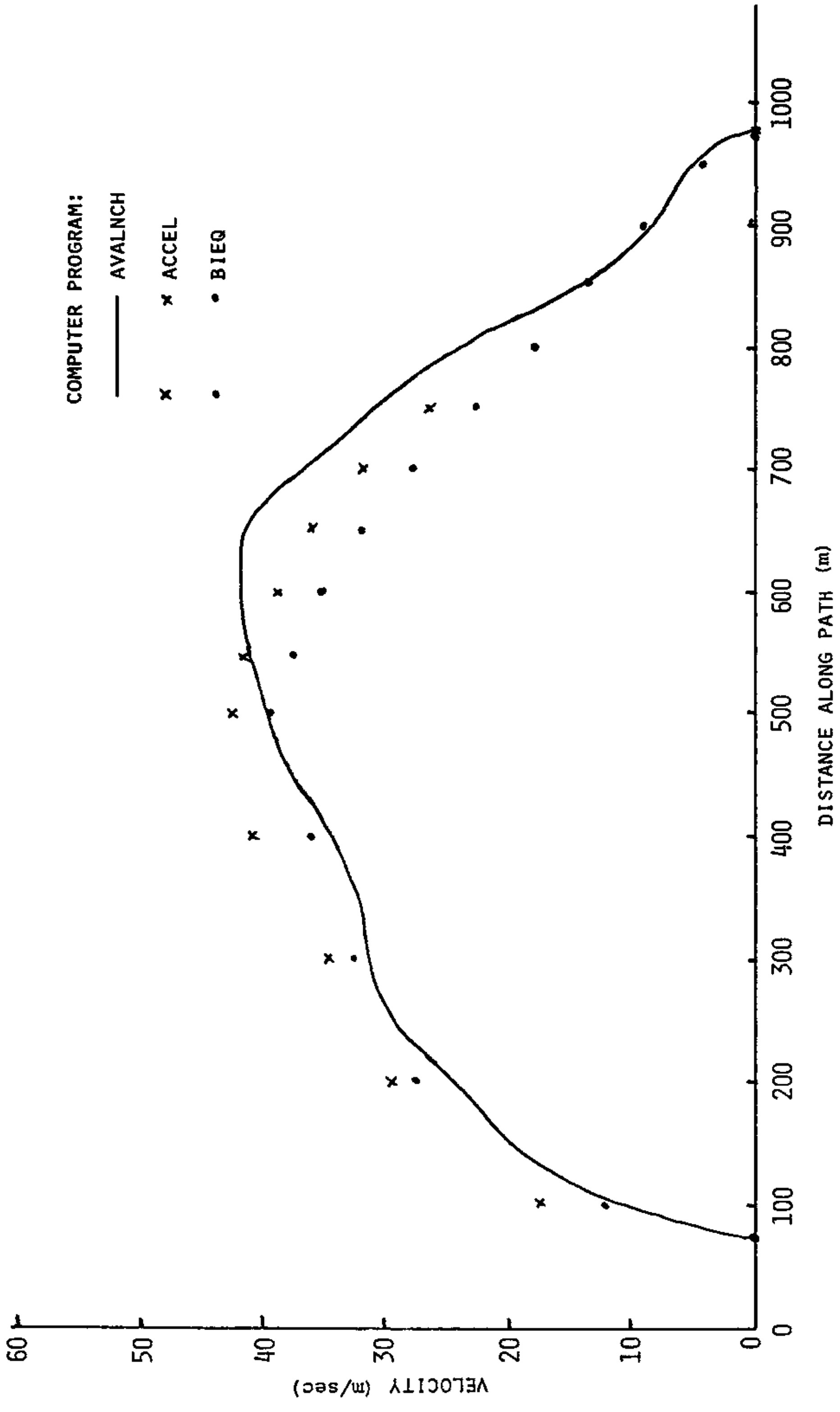


FIGURE 13: IRONTON PARK AVALANCHE PATH : VELOCITY PROFILE FOR STARTING ZONE SNOW DEPTH $h$ =2.0m.

approximately the same. The closest fit between the two sets of data is obtained for the $h$=1.5m case, where differences are less than 10%. For the case of a starting zone snow depth of $h$=2.0m, the shapes of the profiles are different, with attendant larger differences at some data points (up to 30%). Shown also for this case is a profile from program ACCEL. If modeling of the other cases is carried out with program ACCEL, different values of parameters $M/D$ and $\mu$ would be necessary, as no explicit dependence on flow depth is included in the parameterization of this program. Parameterization in all of these computer results is not unique, and it is likely that different shapes would be obtained if different, but equally valid, parameterization is used. In the case of program AVALNCH the parameterization used has been correlated with an actual experimental velocity profile, which, however, was not of the Ironton Park path (LaChapelle, Lang, 1980). Since program AVALNCH is based upon transient fluid motion, while BIEQ (and ACCEL) is based upon equilibruim fluid dynamics, it should not be expected that the profiles be in complete agreement. The numerical experimentation necessary in order to determine if other parameterization of BIEQ (or of ACCEL) produce better profile correspondance does not seem warranted at the present time, since detailed experimental data is first needed in order to establish an absolute basis of comparison.

With $\nu_0$ and $C_0$ now known for the Ironton Park avalanche path, the material locking viscous equation is

$$\nu^* = \nu_0 \left( 1 + 500\, e^{-1.25\, v} \right)$$

where $\nu_0$ =0.027m$^2$. From this equation, viscosity $\nu^*$ varies with velocity, as shown in Figure 14. Viscous drag, proportional to $\nu^* v^2$, increases from $v$=5ms$^{-1}$ to about $v$=2ms$^{-1}$, whereas the ordinary drag $\nu_0 v^2$ monotonically decreases with considerably smaller magnitude. From $v$=2ms$^{-1}$ to $v$=1ms$^{-1}$ the viscous drag $\nu^* v^2$ begins to decrease, as the $v^2$ factor

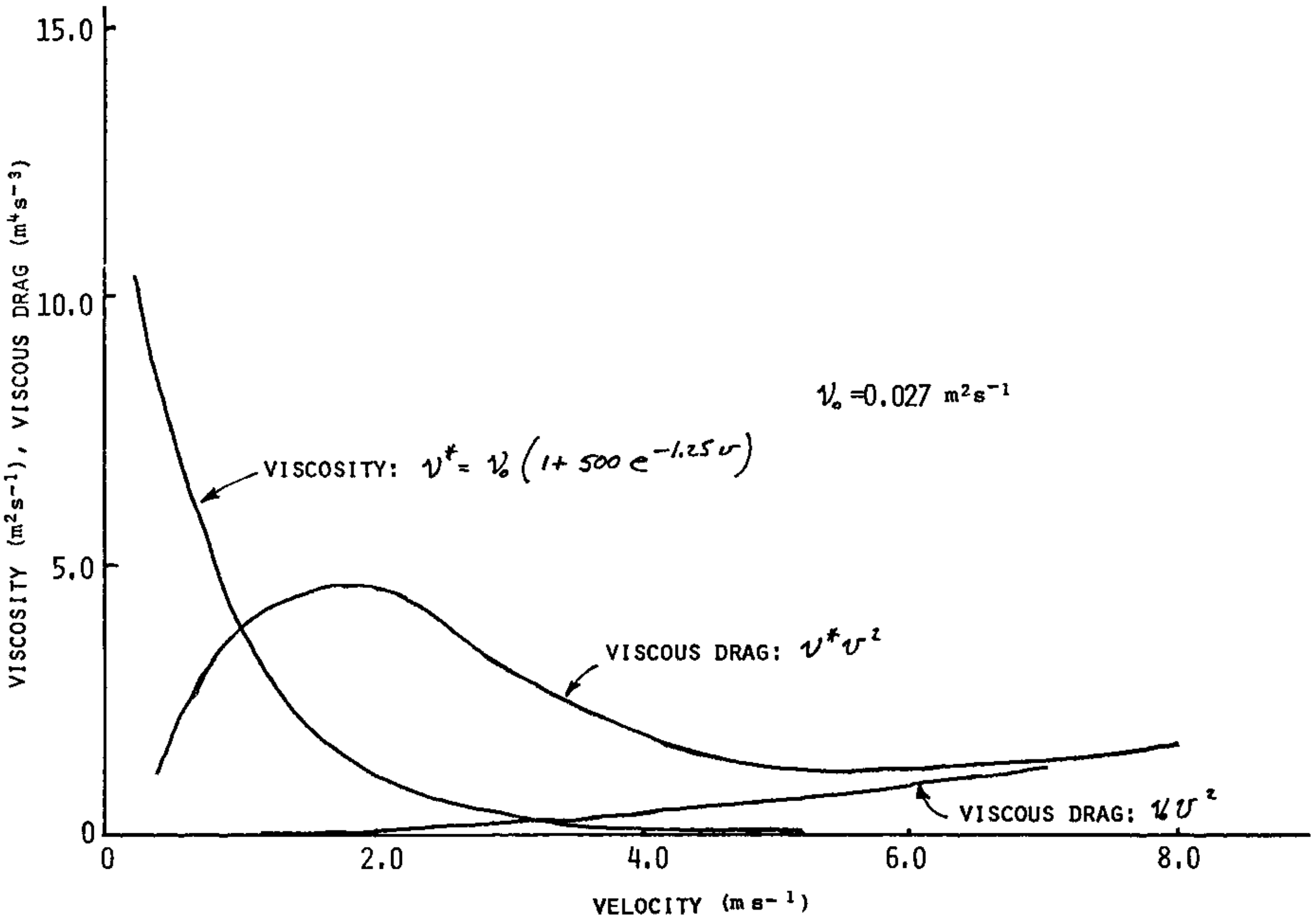


FIGURE 14: VISCOSITY AND VISCOUS DRAG VERSE FLOW VELOCITY

dominates over the exponential form of $v^*$. However, for the larger avalanches for which these computer programs are assumed applicable, $v < 1\text{ms}^{-1}$ is a small velocity of negligible order. Thus, in program BIEQ if the velocities $v_A$ and $v_B$ at the beginning and end of a segment are both less than $1.0\text{ms}^{-1}$, then computations are terminated.

## FLOW PARAMETER SENSITIVITIES

The basic parameters of program BIEQ are assumed to be the viscosity $v_0$, friction coefficient $\mu_0$, and flow depth $h$. We look now at the sensitivity on runout distance to small variations in these parameters. Likewise, the parameters of program ACCEL are $M/D$ and $\mu$, for which small variations are considered with respect to runout. In these calculations the Ironton Park avalanche path is used, and the reference avalanche configuration is taken as $h = 2.0\text{m}$, with a runout distance into segment ⑪ of 230m. Results are presented for small percentage variations in the basic parameters (Figure 15). With program BIEQ the depth of the snow release is the most sensitive, by a factor of 2, of the parameters considered. Next in order of sensitivity is viscosity, followed by friction at about half that of viscosity. Starting zone snow depth is not explicit in program ACCEL so no evaluation is given. However, for viscosity and friction the sensitivities are in general correspondence with those of BIEQ.

Although the Ironton Park avalanche path has no negative or adverse slope along its' length, to check sensitivity of this parameter, the flat runout segment ⑪ was given representative negative slope values. At a slope angle of -10° runout was 40m into segment ⑪. For small percentage changes in this angle the percent change in runout is of the same order as that for viscosity (Figure 15).

The results of this sensitivity study show that flow depth is a

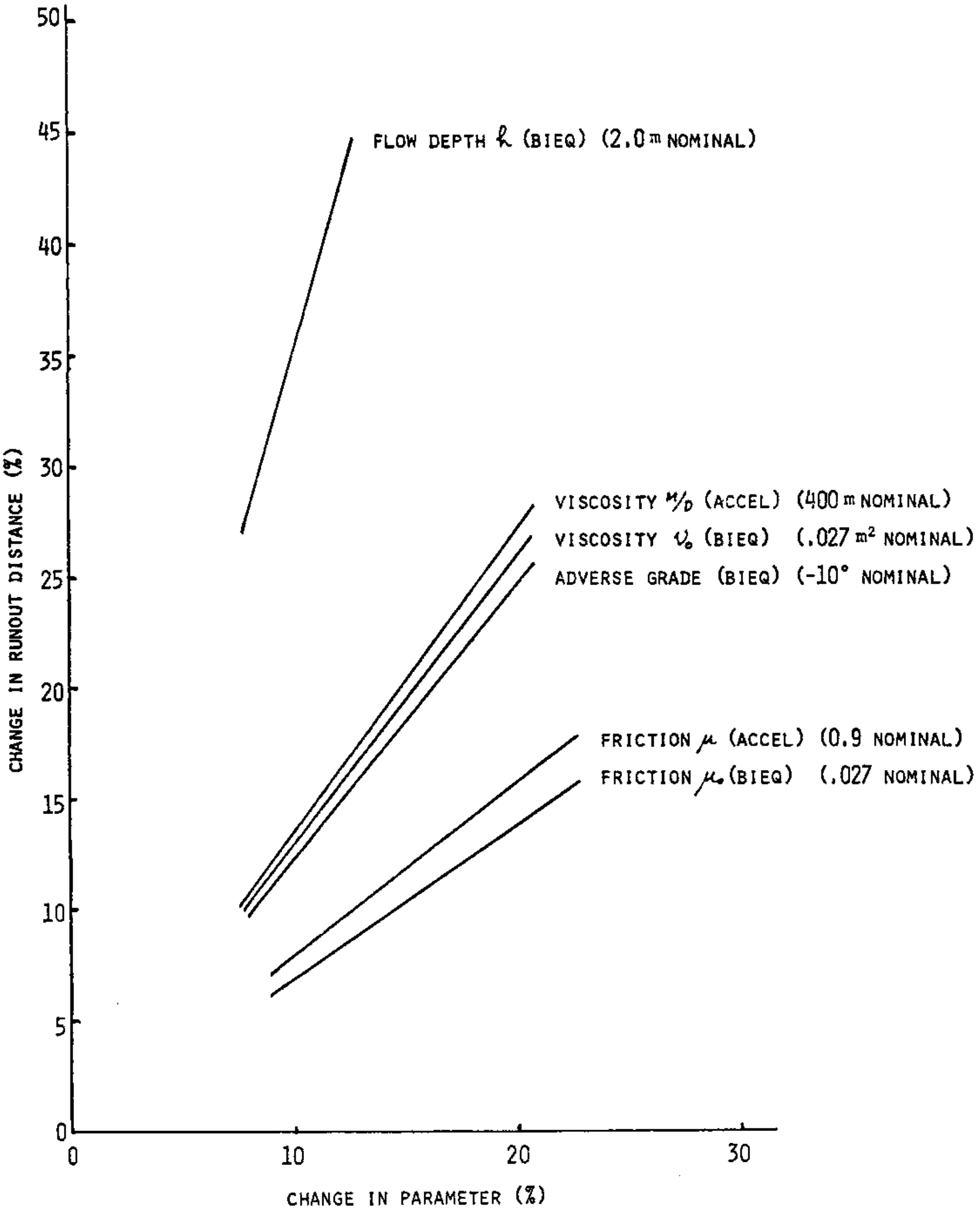


FIGURE 15: CHANGE IN AVALANCHE RUNOUT DISTANCE WITH CHANGE IN VARIOUS FLOW PARAMETERS FOR THE IRONTON PARK AVALANCHE.

variable of primary importance in avalanche runout prediction. With no explicit representation of $h$ in the form of the Voellmy equation used in ACCEL, it requires large variation of the parameters in this equation in order to model the varying conditions of different avalanches and different avalanche paths. With the explicit representation of $h$ in program BIEQ, the variation of the remaining coefficients in order to model different avalanche paths should be less. No claim is intended that in program BIEQ that the $h$ dependence is exact. However, because of the inherent sensitivity of $h$ in avalanche runout, it should be considered, and experimentation should be carried out to define more rational dependence of $h$ in the avalanche equations of motion.

## COMPUTER PROGRAM BIEQ

The version of computer program BIEQ listed in Table 5 has array dimensions that allow up to 100 slope segments to be input. The order and format of the input data is summarized as follows:

### Line 1: FORMAT (I10)

Columns 1-10: IC - Integer number of test cases that are to be run (right adjusted)

### Line 2: FORMAT (40A2)

Columns 1-80: Name and identification information for test case #1

### Line 3: FORMAT (4F10.0)

Columns 1-10: H - Snow depth in the starting zone (m).

Columns 11-20: YNU - high stress viscosity ($m^2$). If the value is set to zero here, it indicates to the computer that an array of values will be input (Line 4).

Columns 21-30: SML - sub-element length (m).

Columns 31-40: YMU - friction coefficient

Line 4: FORMAT (I10, 3F10.0)

Columns 1-10: IS - segment number

Columns 11-20: ANGLE - segment slope (°).

Columns 21-30: SEGL - segment length (m).

Columns 31-40: VNU - segment viscosity ($m^2$). If YNU=0.0 in line 3, then viscosity must be input for each segment. If YNU > 0.0 then any values input here are disregarded.

The format of line 4 is repeated for as many segments as are used to represent an avalanche path. After all segments have been listed a blank card should follow. Following the blank card a second set of data may be input in the same order and format as Lines 2 through 4, until a number of cases equal to the value of IC of Line 1 have been set up. If the program is to be used to run only one case at a time, then statements 5, 6, 21, 105 and 106 may be eliminated from the program (Table 5).

Regarding other parameters in the program, the value of friction in the first segment of flow of an avalanche is defined in statement 49 by YUU=0.4, which may be changed at user discretion. Statement 61 is the test for negligible flow speed, that if $\upsilon_A < 1.0\text{ms}^{-1}$ and $\upsilon_B < 1.0\text{ms}^{-1}$, the computations are terminated. Error messages following statement 92 account for the following conditions:

1) The avalanche does not stop, and calculations are terminated.
2) The number of segments exceeds IMAX=100, and calculations are terminated.
3) The flow velocity through a segment is negligible, and calculations are terminated.
4) Viscosity is not specified, by one of two possible input options, and calculations are terminated.

The segment mini-length parameter SML, input to the program by statement 11, was taken as 10.0m for the Ironton Park path analysis, which provided sufficient accuracy in runout distance prediction. For different avalanche paths, user option is to change the value of SML. When the program switches to the mini-segment analysis option, the velocity at the end of each mini-segment increment is printed to the right of the mainline printout. Thus, the user knows when the switch has been made in the program. The mainline output from the program consists of a listing of segment number, velocity $v_A$ at the start of the segment, and velocity $v_B$ at the end of the segment. If the avalanche stops within a segment, then the output is the segment number in which the avalanche stops, the velocity, $v_A$ , at the start of the segment, and the total runout distance measured along the path, which is the sum of all segment lengths up to the stopping point. The partial segment length of the segment in which the avalanche stops is also included in the sum. Also output by the program, for reference purposes, are the values of all input parameters.

**TABLE 5: Listing of Computer Program BIEQ.**

```
REAL TIME FORTRAN VER.E00   PAGE  1       DATE 83  06  27

C        PROGRAM BIEQ
      DIMENSION NAME(40),SEGL(100),THETA(100),VNU(100)
      IMAX=100
      G=9.806
      READ(7,9) IC
      IP=1
    5 READ(7,20) NAME
      WRITE(6,30) NAME
      WRITE(6,40)
C       READ INPUT DATA
      READ(7,50) H,YNU,SML,YMU
      I=1
   70 READ(7,10) IS,ANGLE,SEGL(I),VNU(I)
      IF(YNU.GT.0.0001) VNU(I)=YNU
      IF(IS.EQ.0)  GO TO 100
      WRITE(6,80) IS,ANGLE,SEGL(I),VNU(I)
      THETA(I)=3.14159*ANGLE/180.0
      I=I+1
      IF(I.GT.IMAX) GO TO 444
      GO TO 70
    9 FORMAT(I10)
   10 FORMAT(I10,3F10.0)
   20 FORMAT(40A2)
   30 FORMAT(1H1,5X,40A2)
   40 FORMAT(1H0,15X,'INPUT DATA'//10X,'SEGMT   ANGLE   SEGL  VSCSTY')
   50 FORMAT(4F10.0)
   60 FORMAT(1H0,9X,'SNOW DEPTH=',F5.2/10X,'HIGH STRESS VISCSTY=',
     *F6.4/10X,'SEGMT MINI-LNGTH=',F6.2/10X,'FRICTION COEF=',F5.3)
   80 FORMAT(10X,I4,2X,F8.1,1X,F6.1,2X,F6.4)
  210 FORMAT(1H0,20X,'RESULTS'/10X,'SEGMT',5X,'VA',7X,'VB')
  240 FORMAT(5X,I9,F10.2,F9.2)
  260 FORMAT(5X,I9,F10.2,3X,'RUNOUT=',F10.2)
  270 FORMAT(50X,'V=',F9.3)
  334 FORMAT(10X,'AVALANCHE DOES NOT STOP')
  445 FORMAT(5X,'SEGMENT NUMBER EXCEEDS SPECIFIED IMAX')
  556 FORMAT(10X,'FLOW VELOCITY NEGLIGIBLE')
  667 FORMAT(10X,'VISCOSITY NOT SPECIFIED')
  888 FORMAT(10X,'COMPUTATIONS TERMINATED')
C
C     INITIAL COMPUTATIONS
  100 WRITE(6,60) H,YNU,SML,YMU
      IF(VNU(1).LE.0.0) GO TO 666
      VA=0.0
      IT=I-1
      I=1
  120 IM=1
      SUL=SML
      YUU=YMU
      IF(I.EQ.1) YUU=0.4
      IF(I.EQ.1) WRITE(6,210)
      A=G*SIN(THETA(I))-G*YUU*H*COS(THETA(I))
      B=VNU(I)*(1.0+500.*EXP(-1.25*VA))/H**3
      IF(I.EQ.1) A=G*SIN(THETA(I))-G*YUU*COS(THETA(I))
      IF(I.EQ.1) B=2.0*VNU(I)/H**3
```

```
REAL TIME FORTRAN VER.E00   PAGE  2       DATE 83  06  27
      E=EXP(-2.0*B*SEGL(I))
      P=VA*VA*E+A*(1.0-E)/B
      IF(P.LT.64.0) GO TO 150
  130 VB=SQRT(P)
      WRITE(6,240) I,VA,VB
      Q=THETA(I)-THETA(I+1)
      IF(VA.LT.1.0.AND.VB.LT.1.0) GO TO 555
      VA=VB*COS(Q)
      IF(Q.LE.0.0) VA=VB
      I=I+1
      IF(I.GT.IT) GO TO 333
      GO TO 120
C      SUB-SEGMENT COMPUTATIONS
  150 IN=INT(SEGL(I)/SUL)
      V=VA
      SL=SEGL(I)
      SEGL(I)=0.0
      IF(IN.LE.1) GO TO 180
  170 B=VNU(I)*(1.0+500.*EXP(-1.25*V))/H**3
      IF(VA.EQ.0.0) B=2.0*VNU(I)/H**3
      E=EXP(-2.0*B*SUL)
      P=V*V*E+A*(1.0-E)/B
      SEGL(I)=SEGL(I)+SUL
      IF(P.LE.0.0) GO TO 190
      IF(IM.EQ.IN+1) GO TO 130
      V=SQRT(P)
      WRITE(6,270) V
      IF(IM.EQ.IN) SUL=SL-SEGL(I)
      IM=IM+1
      GO TO 170
  180 SEGL(I)=SL
      IF(P) 190,190,130
  190 S=0.0
      DO 200 J=1,I
  200 S=S+SEGL(J)
      WRITE(6,260) I,VA,S
      GO TO 999
C       ERROR MESSAGES
  333 WRITE(6,334)
      WRITE(6,888)
      GO TO 999
  444 WRITE(6,445)
      WRITE(6,888)
      GO TO 999
  555 WRITE(6,556)
      WRITE(6,888)
      GO TO 999
  666 WRITE(6,667)
      WRITE(6,888)
  999 CONTINUE
      IP=IP+1
      IF(IP.NE.IC+1) GO TO 5
      STOP
      END
```

## Computer Program ACEL

Computer program ACCEL, developed by Cheng and Perla (1979), is listed with detailed explanations in their reference. A modification of their program, designated ACEL, that operates on the Melcom 70 Computer, is listed in Table 6. Since the ACCEL and ACEL programs are similar, no descriptive summary is included herein for program ACEL.

TABLE 6: Listing of Computer Program ACEL.

```
REAL TIME FORTRAN VER.E00   PAGE  1       DATE 83  06  24

C        PROGRAM ACEL
      DIMENSION THETA(100),SLENG(100),NAME(40),YMU(100),YMD(100)
      READ(7,80) IC
      IMAX=100
      G=9.806
      IP=0
    5 WRITE(6,10)
   10 FORMAT(1H1)
      READ(7,20) NAME
   20 FORMAT(40A2)
      WRITE(6,30) NAME
   30 FORMAT(5X,40A2)
      READ(7,40) FMU,FMD
   40 FORMAT(F10.0,F10.0)
      IF(FMU.GT.0.0.OR.FMD.GT.0.0) WRITE(6,50) FMU,FMD
   50 FORMAT(13X,'MU=',F5.2,5X,'M/D=',F8.0)
      WRITE(6,60)
   60 FORMAT(1H0,15X,'INPUT DATA')
      WRITE(6,70)
   70 FORMAT(1H0,10X,'SEGMT   ANGLE   LENG   MU    M/D')
  210 FORMAT(1H0,20X,'RESULTS'/10X,'SEGMT',5X,'VA',7X,'VB')
  240 FORMAT(5X,I9,F10.2,F9.2)
  300 FORMAT(5X,I9,F10.2,3X,'RUNOUT=',F10.2)
      I=1
   75 READ(7,80) IS,ANGLE,SLENG(I),YMU(I),YMD(I)
      IF(IS.EQ.0) GO TO 100
   80 FORMAT(I10,4F10.0)
      IF(YMU(I).LE.0.0) YMU(I)=FMU
      IF(YMD(I).LE.0.0) YMD(I)=FMD
      IF(YMD(I).EQ.0.0) GO TO 666
      WRITE(6,90) IS,ANGLE,SLENG(I),YMU(I),YMD(I)
   90 FORMAT(10X,I4,2X,F8.1,1X,F6.1,F6.2,1X,F6.0)
      THETA(I)=ANGLE*3.1416/180.
      I=I+1
      IF(I.GT.IMAX) GO TO 777
      GO TO 75
  100 II=I-1
      I=1
  215 VA=0.0
  220 ALPHA=G*SIN(THETA(I))-G*YMU(I)*COS(THETA(I))
      PA=EXP(-2.0*SLENG(I)/YMD(I))
      P=VA*VA*PA+ALPHA*YMD(I)*(1.0-PA)
      IF(P) 290,280,230
  230 VB=SQRT(P)
      IF(I.EQ.1) WRITE(6,210)
      WRITE(6,240) I,VA,VB
      Q=THETA(I)-THETA(I+1)
      IF(Q) 260,260,250
  250 VA=VB*COS(Q)
      GO TO 270
  260 VA=VB
  270 I=I+1
      IF(I.GT.II) GO TO 333
      GO TO 220
```

```
REAL TIME FORTRAN VER.E00   PAGE  2        DATE 83  06  24

         280 VB=0.0
             WRITE(6,240) I,VA,VB
             I=I+1
             IF(I.GT.II) GO TO 333
             GO TO 215
         290 IF(ALPHA.EQ.0.0) GO TO 555
             DD=1.0-(VA*VA)/(ALPHA*YMD(I))
             IF(DD.LT.0.0) GO TO 444
             S=0.5*YMD(I)*ALOG(DD)
             WRITE(6,300) I,VA,S
             GO TO 999
         333 WRITE(6,334)
         334 FORMAT(10X,'AVALANCHE DOES NOT STOP')
             WRITE(6,888)
             GO TO 999
         444 WRITE(6,445)
         445 FORMAT(5X,'ARGUMENT OF LOG CANNOT BE NEGATIVE')
             WRITE(6,888)
             GO TO 999
         555 WRITE(6,556) I
         556 FORMAT(5X,'ALPHA IS ZERO FOR SEGMENT',I4)
             WRITE(6,888)
             GO TO 999
         666 WRITE(6,667) I
         667 FORMAT(5X,'M/D IS ZERO FOR SEGMENT',I4)
             WRITE(6,888)
             GO TO 999
         777 WRITE(6,778)
         778 FORMAT(5X,'SEGMENT NUMBER EXCEEDS SPECIFIED IMAX')
             WRITE(6,888)
         888 FORMAT(10X,'COMPUTATIONS TERMINATED')
         999 IP=IP+1
             IF(IP.NE.IC) GO TO 5
             STOP
             END
```

## PROGRAM BIEQ option studies

Computer program BIEQ was changed in several ways in order to evaluate other possible options of the code. One change was to eliminate dry friction and to incorporate a true biviscous material representation. The ratio between the low stress and high stress viscosities was set at 10, and the transition velocity between the two viscosities was set at $U_T$ = 5 and 8 $ms^{-1}$ in different evaluations. Results are summarized in Table 7. While runout distances can be duplicated to errors less than 5% by appropiate selection of the high stress viscosity, the predicated maximum runout speeds in all cases are low by as much as 25%. This indicates that viscosity, when set for correct runout, is too large for sufficient runout speed. Increasing the low stress range from $U_T$ =5 to 8 $ms^{-1}$ reduced the maximum speed error to 18% or less. However, the largest error in max speed is with the 2.0m deep avalanche, and it is with the deeper avalanches that accurate modeling is wanted. These results were obtained with a low speed cutoff of $U$=1.0 $ms^{-1}$ for computer termination of calculations. In other program versions the low speed cutoff is expressed as a percentage of maximum speed of the avalanche. The next modification of BIEQ was to incorporate a percent cutoff, the value set at 7.0%. Thus, when the speed of each avalanche dropped below 7% of maximum speed over an entire segment, computations were terminated. Results based upon this program option are summarized in Table 8. It is noted that runout distances are better matched with the transition speed $U_T$=5.0 $ms^{-1}$ compared with $U_T$=8.0 $ms^{-1}$. Putting a percent low speed cutoff reduced the error in max. speed from 25% to 22%, but remains a large error. Also evaluated was increasing the viscosity ratio from 10 to 20; however the effect of this was insignificant in changing any of the kinematical data.

The results of these option studies with program BIEQ are not yielding

TABLE 7: Ironton Park avalanche study using program BIEQ with no dry friction and a true biviscous material representation.

| Depth of flow (m) | trans. sp: $\upsilon_T$=5.0ms$^{-1}$ viscosity: $\nu$=.05m$^2$ | | trans. sp: $\upsilon_T$=8.0ms$^{-1}$ viscosity: $\nu$=.04m$^2$ | | PROGRAM AVALNCH $\nu$=0.23m$^2$ | |
|---|---|---|---|---|---|---|
| | max vel. (ms$^{-1}$) | runout distance (m) | max vel. (ms$^{-1}$) | runout distance (m) | max vel. (ms$^{-1}$) | runout distance (m) |
| 1.0 | 11.5 | 650 | 12.9 | 650 | 12.0 | 640 |
| 1.5 | 21.0 | 720 | 23.4 | 700 | 28.0 | 730 |
| 2.0 | 31.3 | 900 | 34.4 | 910 | 42.0 | 910 |

TABLE 8: Ironton Park avalanche study using program BIEQ with no dry friction, a true biviscous material representation, and a 7% low speed cutoff option.

| Depth of flow (m) | trans. sp: $\upsilon_T$=5.0ms$^{-1}$ viscosity: $\nu$=0.045m$^2$ | | trans. sp: $\upsilon_T$=8.0ms$^{-1}$ viscosity: $\nu$=0.037m$^2$ | | PROGRAM AVALNCH $\nu$=0.23m$^2$ | |
|---|---|---|---|---|---|---|
| | max vel. (ms$^{-1}$) | runout distance (m) | max vel. (ms$^{-1}$) | runout distance (m) | max vel. (ms$^{-1}$) | runout distance (m) |
| 1.0 | 12.1 | 640 | 13.4 | 650 | 12.0 | 640 |
| 1.5 | 22.1 | 720 | 24.3 | 700 | 28.0 | 730 |
| 2.0 | 32.8 | 920 | 35.6 | 910 | 42.0 | 910 |

as satisfactory a fit to avalanche runout as was obtained with the version of BIEQ with small dry friction and a gradual biviscous transition at low speeds. An abrupt change in viscosity at a transition speed is unlikely to be physically accurate, and with only one segment modeling in BIEQ, velocity changes rapidly when the viscosity changes. However, the gradual viscosity change used previously in BIEQ, also is not physically based, and may not be generally applicable to different avalanche problems.

A listing of program BIEQ with no dry friction, true biviscous material representation, and 7% velocity cutoff options is given in Table 9. The program was given the code name BEAR with these option changes.

Another modification incorporates the improved results obtained from previous studies, and uses a true biviscous model coupled with low friction. The previous improved results were with a transition speed $\upsilon_T = 5.0 ms^{-1}$, and a velocity cutoff at 5% of maximum velocity. Findings of this evaluation are summarized in Table 10.

TABLE 10: Ironton Park avalanche study using program BIEQ with a true biviscous material representation, low dry friction, and a 5% low speed cutoff option.

| | trans. speed : $\upsilon_T = 5.0 ms^{-1}$<br>viscosity : $\nu = 0.025 m^2$<br>friction : $\mu = 0.030$ | | Program AVALNCH<br>$= 0.23 m^2$ | |
|---|---|---|---|---|
| Depth of flow ( m ) | maximum velocity ( $ms^{-1}$ ) | runout distance ( m ) | maximum velocity ( $ms^{-1}$ ) | runout distance ( m ) |
| 1.0 | 16.0 | 660 | 14.0 | 640 |
| 1.5 | 28.3 | 740 | 28.0 | 730 |
| 2.0 | 40.4 | 910 | 41.0 | 910 |

These results are comparable to the results obtained using a gradually changing viscosity through the transition speed range. The inclusion of small dry friction coupled with viscosity allows matching of both runout distance and max velocity for the different release depth cases.

**TABLE 9: Listing of computer program BIEQ under option changes of (a) no dry friction, (b) true biviscous material representation, and (c) 7% velocity cutoff.**

```
         1         2         3         4         5         6         7
1234567890123456789012345678901234567890123456789012345678901234567890123456789901

C       PROGRAM BEAR: VARIABLE TRANSITION VEL.: VEL. THRESHOLD 7%
      DIMENSION NAME(40),SEGL(100),THETA(100),VNU(100)
      IMAX=100
      G=9.806
    5 READ(7,20) NAME
      WRITE(6,30) NAME
      WRITE(6,40)
C      READ INPUT DATA
      READ(7,50) H,YNU,SML,XNU,VP
      I=1
   70 READ(7,10) IS,ANGLE,SEGL(I),VNU(I)
      IF(YNU.GT.0.0) VNU(I)=YNU
      IF(IS.EQ.0)  GO TO 100
      WRITE(6,80) IS,ANGLE,SEGL(I),VNU(I)
      THETA(I)=3.14159*ANGLE/180.0
      I=I+1
      IF(I.GT.IMAX) GO TO 444
      GO TO 70
    9 FORMAT(I10)
   10 FORMAT(I10,3F10.0)
   20 FORMAT(40A2)
   30 FORMAT(1H1,5X,40A2)
   40 FORMAT(1H0,15X,'INPUT DATA'//10X,'SEGMT   ANGLE   SEGL  VSCSTY')
   50 FORMAT(5F10.0)
   60 FORMAT(1H0,9X,'SNOW DEPTH=',F5.2/10X,'SEGMT MINI-LNGTH=',F6.2,
     */10X,'VSCSITY MULT FACTOR=',F5.1/10X,'VEL AT TRANSTION=',F5.1)
   80 FORMAT(10X,I4,2X,F8.1,1X,F6.1,2X,F6.4)
  210 FORMAT(1H0,20X,'RESULTS'/10X,'SEGMT',5X,'VA',7X,'VB',
     *9X,'S',9X,'T')
  240 FORMAT(5X,I9,F10.2,F9.2,F9.2,F9.2)
  260 FORMAT(5X,I9,F10.2,3X,'RUNOUT=',F10.2)
  270 FORMAT(53X,'V=',F9.3)
  334 FORMAT(10X,'AVALANCHE DOES NOT STOP')
  445 FORMAT(5X,'SEGMENT NUMBER EXCEEDS SPECIFIED IMAX')
  556 FORMAT(10X,'FLOW VELOCITY NEGLIGIBLE')
  667 FORMAT(10X,'VISCOSITY NOT SPECIFIED')
  888 FORMAT(10X,'COMPUTATIONS TERMINATED')
C
C    INITIAL COMPUTATIONS
  100 WRITE(6,60) H,SML,XNU,VP
      IF(VNU(1).LE.0.0) GO TO 666
      VA=0.0
      VMAX=0.0
      IT=I-1
      I=1
      S=0.0
      T=0.0
  120 IM=1
      SUL=SML
      IF(I.EQ.1) WRITE(6,210)
```

```
         1         2         3         4         5         6         7
12345678901234567890123456789012345678901234567890123456789012345678901
      A=G*SIN(THETA(I))
      B=VNU(I)/H**3
      E=EXP(-2.0*B*SEGL(I))
      P=VA*VA*E+A*(1.0-E)/B
      VP2=VP*VP
      IF(VA.EQ.0.0.AND.P.LT.VP2) GO TO 130
      IF(VA.NE.0.0.AND.P.LT.VP2) GO TO 150
  130 VB=SQRT(P)
      IF(VMAX.LT.VB) VMAX=VB
      S=S+SEGL(I)
      T=T+2.0*SEGL(I)/(VA+VB)
      WRITE(6,240) I,VA,VB,S,T
      Q=THETA(I)-THETA(I+1)
      VT=0.07*VMAX
      IF(VA.LT.VT.AND.VB.LT.VT) GO TO 190
      VA=VB*COS(Q)
      IF(Q.LE.0.0) VA=VB
      I=I+1
      IF(I.GT.IT) GO TO 333
      GO TO 120
C       SUB-SEGMENT COMPUTATIONS
  150 IN=INT(SEGL(I)/SUL)
      V=VA
      SL=SEGL(I)
      SEGL(I)=0.0
      IF(IN.LE.1) GO TO 180
  170 IF(V.LE.VP) B=XNU*VNU(I)/H**3
      IF(V.GT.VP) B=VNU(I)/H**3
      E=EXP(-2.0*B*SUL)
      P=V*V*E+A*(1.0-E)/B
      VN=SQRT(P)
      IF(VMAX.LT.VN) VMAX=VN
      VT=0.07*VMAX
      SEGL(I)=SEGL(I)+SUL
      IF(VN.LT.VT.AND.V.LT.VT) GO TO 190
      IF(IM.EQ.IN+1) GO TO 130
      V=VN
      WRITE(6,270) V
      IF(IM.EQ.IN) SUL=SL-SEGL(I)
      IM=IM+1
      GO TO 170
  180 SEGL(I)=SL
      GO TO 130
  190 S=0.0
      DO 200 J=1,I
  200 S=S+SEGL(J)
      WRITE(6,260) I,VA,S
      GO TO 555
C        ERROR MESSAGES
  333 WRITE(6,334)
```

```
                     1         2         3         4         5         6         7
.INE.       12345678901234567890123456789012345678901234567890123456789012345678901

  101             WRITE(6,888)
  102             GO TO 999
  103         444 WRITE(6,445)
  104             WRITE(6,888)
  105             GO TO 999
  106         555 WRITE(6,556)
  107             WRITE(6,888)
  108             GO TO 999
  109         666 WRITE(6,667)
  110             WRITE(6,888)
  111         999 CONTINUE
  112             STOP
  113             END

*** PRINT   END ***
```

For the three cases, maximum error in velocity is 14%, and maximum error in runout is 3%, both for the h=1.0m case. For the larger release depths the errors reduce, which is a desireable result, as the intention is to better model the deeper flows.

Table 11 is a listing of the computer program that incorporates the improvements described above.

TABLE 11: Listing of program BIEQ with options of (a) low dry friction, (b) true biviscous material representation, and (c) 5% velocity cutoff.

```
REAL TIME FORTRAN VER.E00  PAGE  1      DATE 83  08  24

C   PROGRAM BIEQ WITH TRUE BIVISCOUS, 5% CUTOFF VELOCITY OPTIONS
      DIMENSION NAME(40),SEGL(100),THETA(100),VNU(100)
      IMAX=100
      G=9.806
    5 READ(7,20) NAME
      WRITE(6,30) NAME
      WRITE(6,40)
C      READ INPUT DATA
      READ(7,50) H,YNU,SML,XNU,YMU
      I=1
   70 READ(7,10) IS,ANGLE,SEGL(I),VNU(I)
      IF(YNU.GT.0.0) VNU(I)=YNU
      IF(IS.EQ.0)  GO TO 100
      WRITE(6,80) IS,ANGLE,SEGL(I),VNU(I)
      THETA(I)=3.14159*ANGLE/180.0
      I=I+1
      IF(I.GT.IMAX) GO TO 444
      GO TO 70
   10 FORMAT(I10,3F10.0)
   20 FORMAT(40A2)
   30 FORMAT(1H1,5X,40A2)
   40 FORMAT(1H0,15X,'INPUT DATA'//10X,'SEGMT   ANGLE   SEGL  VSCSTY')
   50 FORMAT(5F10.0)
   60 FORMAT(1H0,9X,'SNOW DEPTH=',F5.2/10X,'SEGMT MINI-LNGTH=',F6.2,
     */10X,'VSCSITY MULT FACTOR=',F5.1/10X,'FRICTION COEF=',F5.3)
   80 FORMAT(10X,I4,2X,F8.1,1X,F6.1,2X,F6.4)
  210 FORMAT(1H0,20X,'RESULTS'/10X,'SEGMT',5X,'VA',7X,'VB',8X,'S',
     *7X,'T')
  240 FORMAT(5X,I9,F10.2,F9.2,F9.2,F9.2)
  260 FORMAT(5X,I9,F10.2,3X,'RUNOUT=',F10.2)
  270 FORMAT(53X,'V=',F9.3)
  334 FORMAT(9X,'AVALANCHE DOES NOT STOP'/9X,'CMPUTATNS TERMINATED')
  445 FORMAT(9X,'NO. OF SGMTS > IMAX'/9X,'COMPUTATIONS TERMINATED')
  556 FORMAT(9X,'FLO VELCTY NEGLIGBLE'/9X,'COMPUTATNS TERMINATED')
  667 FORMAT(9X,'VSCSITY NOT SPECIFD'/9X,'CMPUTATNS TERMINATED')
C
C    INITIAL COMPUTATIONS
  100 WRITE(6,60) H,SML,XNU,,YMU
      IF(VNU(1).LE.0.0) GO TO 666
      VA=0.0
      VMAX=0.0
      IT=I-1
      I=1
      S=0.0
      T=0.0
  120 IM=1
      SUL=SML
      YUU=YMU
      IF(I.EQ.1) WRITE(6,210)
      A=G*SIN(THETA(I))-G*YUU*H*COS(THETA(I))
      B=VNU(I)/H**3
      E=EXP(-2.0*B*SEGL(I))
      P=VA*VA*E+A*(1.0-E)/B
      IF(VA.EQ.0.0.AND.P.LT.25.0) GO TO 130
```

```
REAL TIME FORTRAN VER.E00  PAGE  2      DATE 83  08  24
      IF(VA.NE.0.0.AND.P.LT.25.0) GO TO 150
  130 VB=SQRT(P)
      IF(VMAX.LT.VB) VMAX=VB
      S=S+SEGL(I)
      T=T+2.0*SEGL(I)/(VA+VB)
      WRITE(6,240) I,VA,VB,S,T
      Q=THETA(I)-THETA(I+1)
      VT=0.05*VMAX
      IF(VA.LT.VT.AND.VB.LT.VT) GO TO 190
      VA=VB*COS(Q)
      IF(Q.LE.0.0) VA=VB
      I=I+1
      IF(I.GT.IT) GO TO 333
      GO TO 120
C       SUB-SEGMENT COMPUTATIONS
  150 IN=INT(SEGL(I)/SUL)
      V=VA
      SL=SEGL(I)
      SEGL(I)=0.0
      IF(IN.LE.1) GO TO 180
  170 IF(V.LE.5.0) B=XNU*VNU(I)/H**3
      IF(V.GT.5.0) B=VNU(I)/H**3
      E=EXP(-2.0*B*SUL)
      P=V*V*E+A*(1.0-E)/B
      VN=SQRT(P)
      IF(VMAX.LT.VN) VMAX=VN
      VT=0.05*VMAX
      SEGL(I)=SEGL(I)+SUL
      IF(VN.LT.VT.AND.V.LT.VT) GO TO 190
      IF(IM.EQ.IN+1) GO TO 130
      V=VN
      WRITE(6,270) V
      IF(IM.EQ.IN) SUL=SL-SEGL(I)
      IM=IM+1
      GO TO 170
  180 SEGL(I)=SL
      GO TO 130
  190 S=0.0
      DO 200 J=1,I
  200 S=S+SEGL(J)
      WRITE(6,260) I,VA,S
      GO TO 555
C        ERROR MESSAGES
  333 WRITE(6,334)
      GO TO 999
  444 WRITE(6,445)
      GO TO 999
  555 WRITE(6,556)
      GO TO 999
  666 WRITE(6,667)
  999 CONTINUE
      STOP
      END
```

## Summary

Three computer programs, useful in analysis of snow avalanche runout prediction, have been compared by analysis of The Ironton Park avalanche path. Two of the programs, AVALNCH with fast-stop, and ACCEL have been used previously in typical avalanche analyses. Program BIEQ, and a modified version of AVALNCH, referred to as the biviscous version, have been developed in the course of this work.

Program AVALNCH, with its' two versions, is the most versatile of the codes, since transient fluid processes can be modeled. The other codes, based upon the Voellmy theory of avalanche flow, incorporate fluid equilibrium-flow equations. Although program AVALNCH has greater versatility, it requires orders-of-magnitude more time to run a path analysis, compared to programs BIEQ and ACCEL. Both of the programs BIEQ and ACCEL use the Voellmy equations; however, program BIEQ incorporates parameter definitions that are based upon recent findings on the mechanics of flowing avalanches. In taking account of these mechanics processes in writing program BIEQ, a reduction has been obtained in the variation of parameters in order to model different avalanche cases. As determined from numerical evaluation, the parameter that has strongest influence on avalanche runout is the snow depth. The primary change in program BIEQ is to represent the effects of friction and viscous drag as functions of the snow depth. Friction is made proportional to the depth $h$, and viscous drag to $h^{-3}$, based upon physical arguments. Then by selection of one set of values of the proportionality coefficients the Ironton Park avalanche runout for snow depths of 2.0, 1.5, 1.0 and 0.5m was approximated. This is in contrast to the need for different valued coefficients for each of these cases if use is made of previously developed versions of the Voellmy equations. No claim is intented that the depth dependence selected is correct. However, the observed

effect of setting up these relationships in $k$, in that the variation in parameters in order to match different avalanche runouts is greatly reduced, is encouraging. Pending further checkout of program BIEQ, less sensitivity in parameter selection is expected compared with former versions of analysis methods based upon the Voellmy equations. Obvious improvement of the algorithm would be to incorporate snow depth changes as the avalanche advances along its' path. Program AVALNCH does this, but since it is a 2-dimensional code, lateral expansion or contraction of a flow is not accounted for. The only redeeming aspect of this flow depth variation problem is the tendency in viscous fluid dynamics that as the flow increases in depth friction effects decrease and viscous effects increase, and visa versa as flow depth decreases. Thus, if account is taken of both friction and viscous processes, then they have an interactive balancing effect with changing flow depths. In the case of the Ironton Park avalanche path the width of avalanche runout is nearly constant, which simplifies the modeling problem.

Incorporated in program BIEQ is a version of the physical condition of material locking, which has been observed with snow flow. The representation used for material locking is that viscosity of the flow begins to increase at a flow speed of $\mathcal{V} = 8.0 \mathrm{ms}^{-1}$, and exponentially increases as the speed decreases. The material locking algorithm that is used in BIEQ is selected based upon a single segment equilibrium modeling of snow flow, and is only one of many that currecntly could be selected. With further check-out of program BIEQ, more rational representation of the material locking algorithm should become evident.

## REFERENCES

Hirt, C.W., Nichols, B.D. and N.C. Romero. 1975. 'SOLA: a numerical soluation algorithm for transient fluid flow', Los Alamos Sci. Lab., Los Alamos, N. Mex., La 5852, 50p.

Cheng, T.T. and Perla R., 1979. 'Numerical computation of avalanche motion', Ottawa, Environment, Canada. Inland Waters Directorate. National Hydrology Research Institute. (NHRI Paper No.5)

Lang T.E., Dawson, K.L., and Martinelli, M. Jr., 1979. 'Numerical simulation of snow avalanche flow', U.S.D.A.-Forest Service, R.M 205.

Martinelli, M.Jr., Lang,T.E., and Mears, A.I., 1980. 'Calculations of avalanche friction coefficients from field data', J. of Glaciology, 26, 94, p.109-19.

Lang, T.E., and Dent, J.D., 1983. 'Evaluation of the fluid dynamic properties of mud flows on Mt. St. Helens', Dept. of Interior, Bureau of Reclamation, Washington, D.C., final report of contract #14-4-0001-1471.

Trunk, F.J., Dent, J.D., and Lang, T.E., 1983. 'Development of computer modeling of large volume rockslides (Sturzstrom)', Montana State University, Bozeman, Mt., submitted for publication.

Lang, T.E. and Brown, R.L. 1980. 'Snow avalanche impact on structures', J. of Glaciology, 25, 93, p.445-55.

Schaerer, P.A., 1975. 'Friction coefficients and speed of flowing avalanches', Proceedings of the International symposium on Mechanics of Snow, Erindelwald, Switzerland, p.425-32.

Dent, J.D. and Lang, T.E. 1982. 'Experiments on mechanics of flowing snow', Cold Regions Sci and Tech, 5, p.253-58.

Dent, J.D. and Lang, T.E. 1983. 'A biviscous modified Bingham model of snow avalanche motion', J. of Glaciology, proceedings of Applied Glaciology Conference, Hanover, N.H., in press.

Lang, T.E. and Dent, J.D. 1983. 'Basal surface layer properties in flowing snow', J. of Glaciology, proceedings of Applied Glaciology Conference, Hanover, N.H., in press.

Voellmy, A. 1955. 'Uber die Zerstorungskraft von Lawinen', Schweizeriche Bauzeitung, Jahrg. 73, HE.12, p.159-42; HE.15, P.212-17 HE.17, p.246-49; HE.19, p.280-85.

Shames, I.N. 1982. ' Mechanics of fluids', McGraw-Hill, New York, N.Y.

LaChapelle, E.R., and Lang, T.E. 1980. 'A comparison of observed and calculated avalanche velocities', J. of Glaciology 25, 92, p.309-14.

(Manuscript received February 13, 1984)

624.14：681.326

# 雪崩走出予知用コンピュータープログラム

セオドール・イー・ラング*

国立防災科学技術センター新庄支所

この報告書は雪崩走出の解析用として，新庄支所の計算機システムに使えるように書き換えられた計算機用プログラム三種の機能と応用とをとりまとめたものである．雪の流動現象のモデル化の際の流体力学原理としては，このコンピューターコード三つのうちの二つは平衡流体動力学方程式に基づいたものを利用している．三番目のものは二次元非圧縮性境界層理論に基づいた過渡的粘性効果に準拠している．上述の二つの流体動力学に基づいたコードのうち，一つは既に開発されていたものであるが，それには一定範囲の摩擦係数と粘性係数とが含まれており，これらは個々の雪崩の型により大きく異なるため経験を積んだ者以外にはその利用は仲々困難であった．二番目のコードは新庄支所で開発されたものであり，それは流れの厚さ依存性を含むと同時にまたその流れる物質の性質を表わす諸係数を明確化する際に，閉塞の性質をも含んでいるものである．それぞれのコードのリスト，すなわち計算機へのデータインプット操作のフォーマットと順序がこの報告書に含まれている．あらかじめ予想された速度プロファイルと走出距離とそれぞれの計算コードを用いて求められたものとの比較が，一つの雪崩走路（アメリカ・コロラド州のアイロントン公園）に対してなされた．速度プロファイルはそれぞれのコンピューターコードで異なりはしたものの，走出距離については適当なパラメーターを選ぶことによりうまく適合させることができた．

---

* 昭和58年度科学技術庁外国人研究者招へい制度で来日　アメリカ・モンタナ州立大学工学部土木工学科・機械力学科教授