NASA/CR–1998–206854

LOOPREF: A Fluid Code for the Simulation of Coronal Loops

Rosalinda de Fainchtein, Raythoen STX, Lanham, Maryland

Spiro Antichos, Naval Research Laboratory, Washington, District of Columbia

Daniel Spicer, NASA Goddard Space Flight Center, Greenbelt, Maryland

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

April 1998

TABLE OF CONTENTS

1 INTRODUCTION

The understanding of solar phenomena has long been a topic of intense research among the Astrophysics community, in spite of the difficulties of obtaining useful observational measurements. NASA's production and "installment" of solar observing satellites such as SOHO, as well as the ever increasing awareness of the direct influence of solar phenomena and emission on earth's weather and communications disruptions has greatly fueled the interest on solar research.

Simultaneously, due to the fast increase of computational power available to the science community, computational simulations have been added to the arsenal of tools available for solar study. Three dimensional Magneto-hydrodynamic simulation codes that study both local and global solar phenomena are now being used in conjunction with the satellite observations available, not only to interpret the data available from observations, but in some instances serving as the motivation for observations.

As MHD simulations become increasingly sophisticated, added tools such as Adaptive Mesh Refinement (AMR) are proving very effective in maximizing the computer capabilities to simulate phenomena that require extremely high resolution. AMR is particularly helpful in simulating phenomena where the high resolution requirement [e.g. of the order of 10^5 times smaller than the length of the whole domain] occur on very small and time varying regions of the domain.

Despite great improvements in the quality of MHD simulation codes, some problems still remain beyond the capabilities of current 3D MHD codes. One important and challenging example is the simulation of Coronal Loop Phenomena where the presence of very steep temperature gradients determine the local energy balance and must accurately be resolved. The problem of solar coronal loop phenomena is in fact quite complex. It involves the balance of extremely large energy terms, some with steep gradients, such as radiative heat flux, thermal conduction, gravity, and external heating. Because of this, 3D simulations appropriate to simulate solar coronal loops are not yet available. In fact 1D simulations that incorporate all the energy terms mentioned above are themselves quite challenging. While examples of 1D fluid codes to simulate Coronal Loop Phenomena abound, none of them includes a full implementation of AMR.

The code we present here, LOOPREF, is a fully adaptive semi-one di-

mensional (variable loop cross section) fluid code that incorporates all the necessary energy terms mentioned above. LOOPREF is thus especially well suited to simulate challenging solar coronal loop phenomena such as prominence formation. During a prominence formation, some of the plasma gets condensed at the top of the loop, producing a narrow region (ΔS) at the edge of the condensation where the temperature gradient becomes very steep. This temperature gradient must be accurately represented, thus requiring very high grid resolution. Furthermore, as the region of condensation expands along the loop, it causes ΔS to travel. By using AMR, LOOPREF is able to "follow" ΔS, providing the necessary high resolution ONLY where it is needed.

LOOPREF is a versatile code. It is written in a way that makes it easy for the user to include the terms of interest and the necessary tools for a given problem, either through switches or small code changes. It is also equipped with Flux Limiting (FCT) in order to handle shocks and other discontinuities.

2 EQUATIONS OF MOTION

The equations of motions represent a cylindrically symmetric fluid in a tube of variable cross section $S(x)$. The fluid is subject to nonlinear heat conduction, gravitational forces, radiative energy loss, and external heating. (When gravity, radiative loss, heating, and heat conduction are all set to zero, the equations represent nozzle flow in a semi-1D tube [2]).

$$\partial_t(\rho S) + \partial_s(\rho v S) = 0$$

$$(\partial_t(\rho v S) + \partial_s(\rho v^2 S) + \partial_s(p S) = p \partial_s S + \rho S g_s$$

$$\partial_t(\rho e S) + \partial_s[(\rho e S + p S)v] = \partial_s[A T^b \partial_s T] - \rho v S g_s + R + H$$

where, as usual, ρ is the mass density, v is the velocity, ρE is the internal plus kinetic energy density, p is the pressure, T is the temperature, and s is the arc length.

Other variable definitions are as follows,

$S = S(s) =$ cross sectional area

$g_s =$ gravitational component along the loop

$R = -\rho^2 \Lambda(T) =$ radiative energy loss

$H =$ external heating

$K \equiv A T^b =$ Non-linear heat conduction coefficient.

5

3   NUMERICAL IMPLEMENTATION

The equations of motion are integrated by the finite element method. The unstructured nature of this method makes it especially well suited for supporting adaptive mesh refinement. The algorithm is explicit and uses a two step marching scheme with Galerkin discretization by piecewise linear(full step) and piecewise constant (half step) functions. [3] This choice of basis functions insures that stability is maintained much in the same manner of the two-step Richtmyer variant of the Lax-Wendroff method [4].

Both a low order and high order scheme are available. The low order scheme uses the lumped mass matrix [2] and adds numerical diffusion of the type suggested by Rusanov [5]. The high order scheme solves the consistent mass matrix by iteration [6]

Another tool available in LOOPREF is a flux limiter, which allows the treatment of shocks and strong discontinuities in the flow. The flux limiter used in LOOPREF has the format derived by Zalesak [7], and expanded for finite elements by Löhner [8]. Both high and low order schemes are necessary when the flux corrector limiter is turned on.

While the code is equipped to solve the full loop equations expressed on the previous section, it is implemented in a way that makes it easy for the user to modify or delete terms or forces from the equations as needed, in some cases by just modifying an input parameter.

The adaptive mesh refinement routines allow the user to maintain high resolution in the regions that require it, and to lower the resolution on the regions that no longer require it. As the simulation evolves, elements are marked for refinement or de-refinement according to a pre-defined criterion: A local discretization *error indicator function* ϵ, is evaluated at each element. Elements with ϵ larger than a refinement threshold are refined, elements with ϵ smaller than a de-refinement threshold are de-refined. The function ϵ is essentially a normalized local second derivative of a relevant variable chosen by the programmer (e.g. density, temperature). The values of ϵ range between zero and one. The thresholds for refinement and de-refinement are also set on input by the programmer.

The AMR routines in LOOPREF are an improved version of AMR1D, a simple instructive code that has been posted on the World Wide Web: http://sdcd.gsfc.nasa.gov/ESS/exchange/contrib/de-fainchtein/ adaptive_mesh_refinement.html, and documented in NASA's Contractor Re-

port 4718 [1]. In LOOPREF, all point and element arrays are compressed after each call to the AMR routines.

4   TEST CASE RESULTS

LOOPREF has been successfully tested against a series of problems whose analytical solution is known. The results of some of these tests are displayed in this section.

Figure 1 shows results produced by running LOOPREF with initial conditions set for the Sod Shock Tube Problem with AMR enabled and no Flux Limiting [9]. Figures 2 and 3 show results produced by running LOOPREF with the same initial conditions set for the Sod Shock Tube Problem with no AMR and synchronized and characteristic Flux Limiting, respectively. For reference, Figure 4 shows results produced by running LOOPREF with no AMR nor Flux Limiting. One observes that the contact discontinuity has diffused completely in this case. Figure 5 shows a comparison to the analytical solution for steady flow on a nozzle [10]. The cross section $S(s)$ is

$$S(s) = 1 + s/10$$

The results of a steady state calculation of a non-linear heat conduction test, contrasted to the analytical solution, are shown on Figures 6 and 7. Both figures show good agreement with the analytical solution However, it is apparent from Figure 6 that additional resolution is needed next to the left boundary in order to reproduce the very steep Temperature there. Figure 7 shows how AMR provides increased resolution where it was needed. Resolution of these very steep temperature gradients is crucial for accurate solar coronal loop simulations.
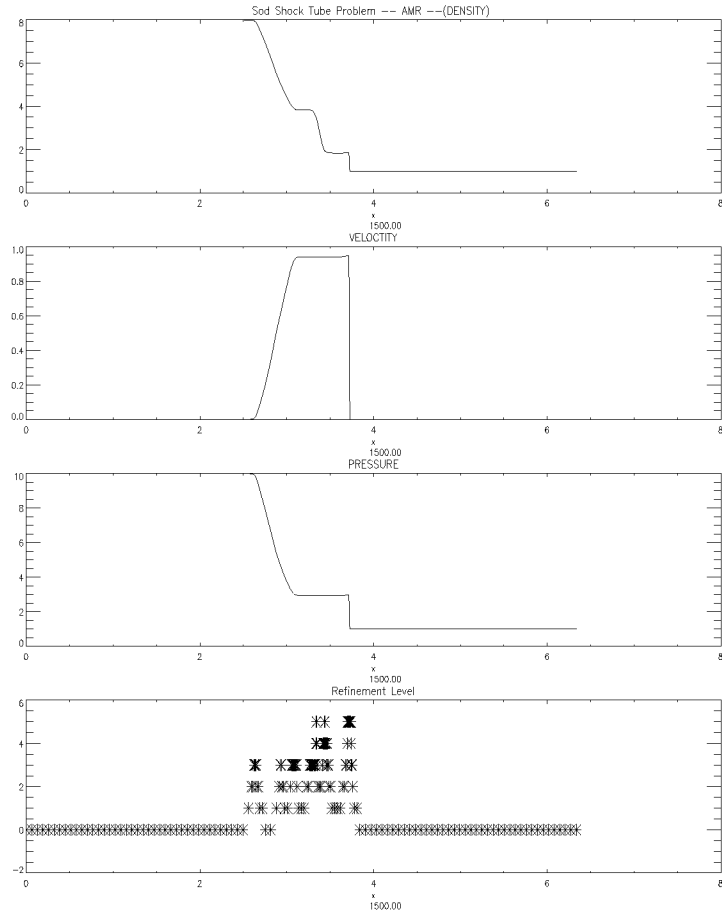
7

Figure 1: *Results Produced by running LOOPREF with AMR and no Flux Limiting, for the Sod Shock Tube Problem. The plot at the bottom corresponds to the refinement level of each element.*
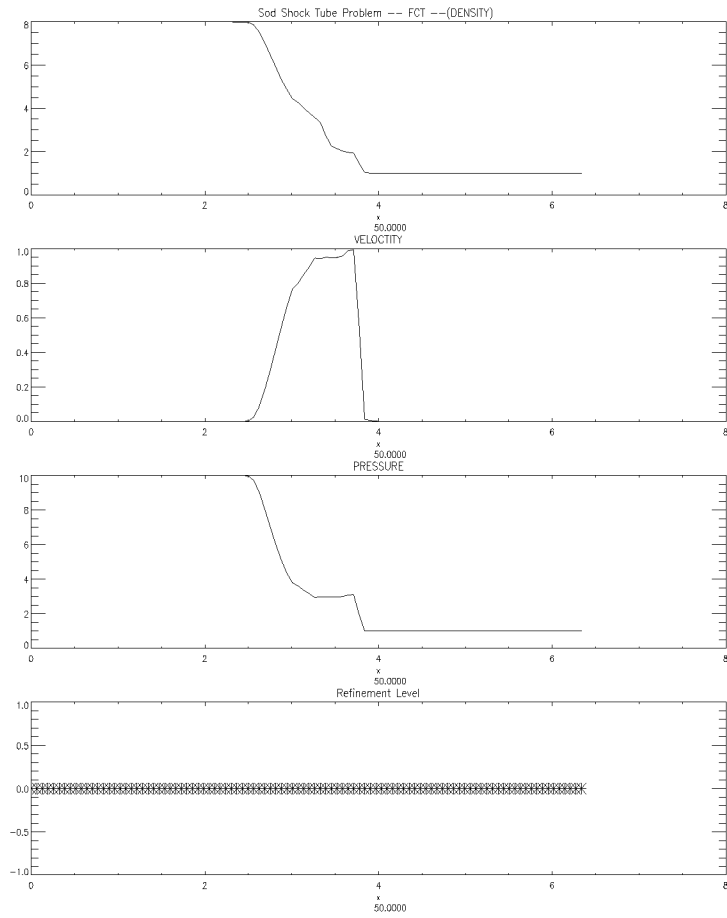
8

Figure 2: *Results Produced by running LOOPREF with Flux Limiting and no AMR, for the Sod Shock Tube Problem. The plot at the bottom corresponds to the refinement level of each element.*
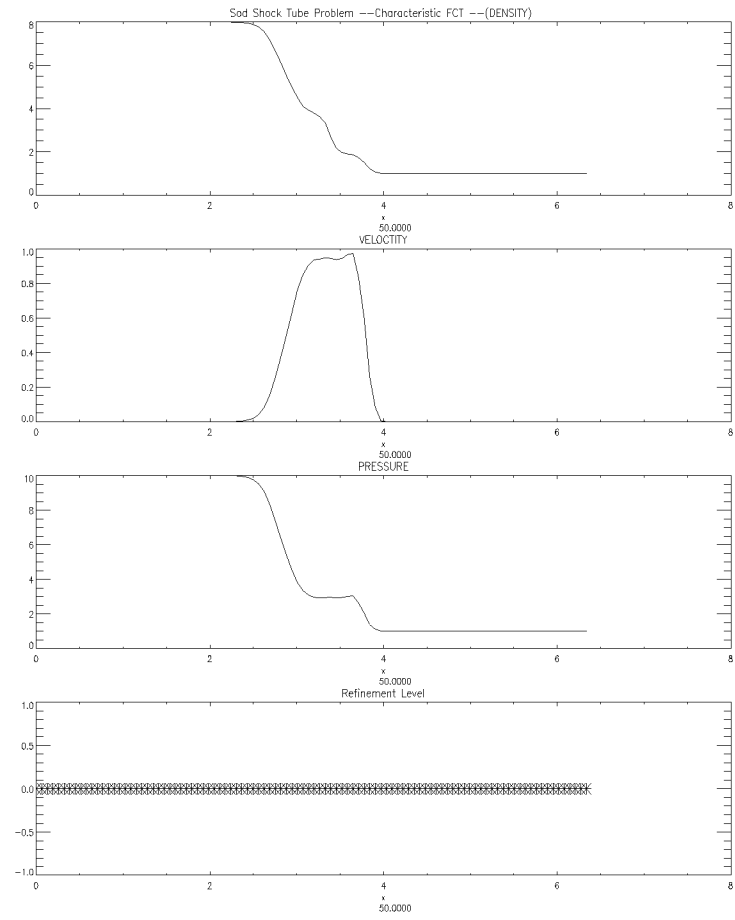
Figure 3: *Results Produced by running LOOPREF with Characteristic Flux Limiting and no AMR, for the Sod Shock Tube Problem.*
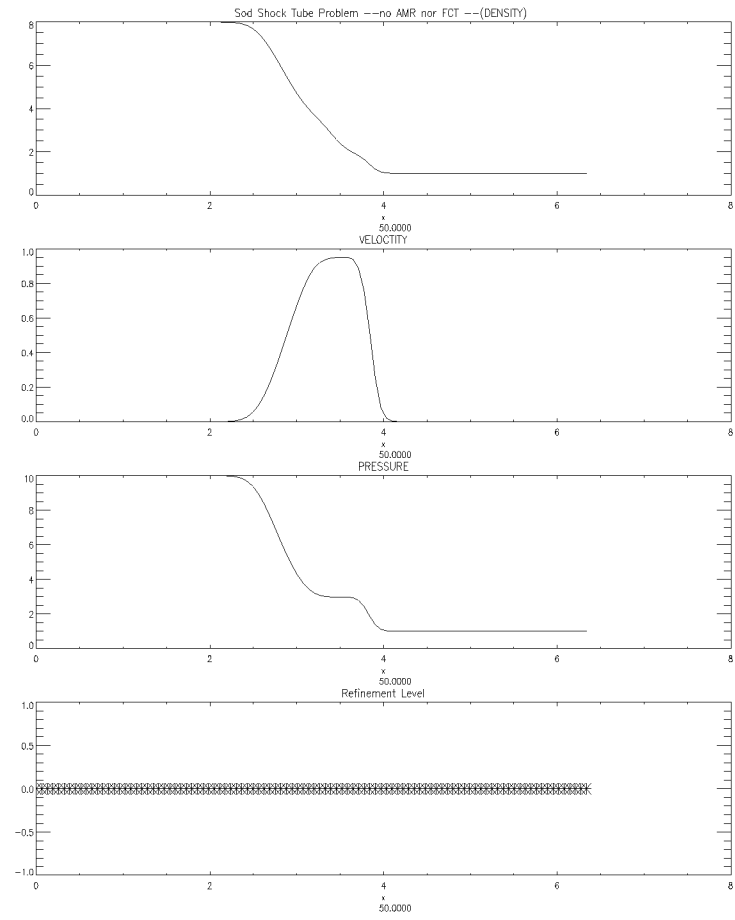
Figure 4: *Results Produced by running LOOPREF with no Flux Limiting nor AMR, for the Sod Shock Tube Problem.*
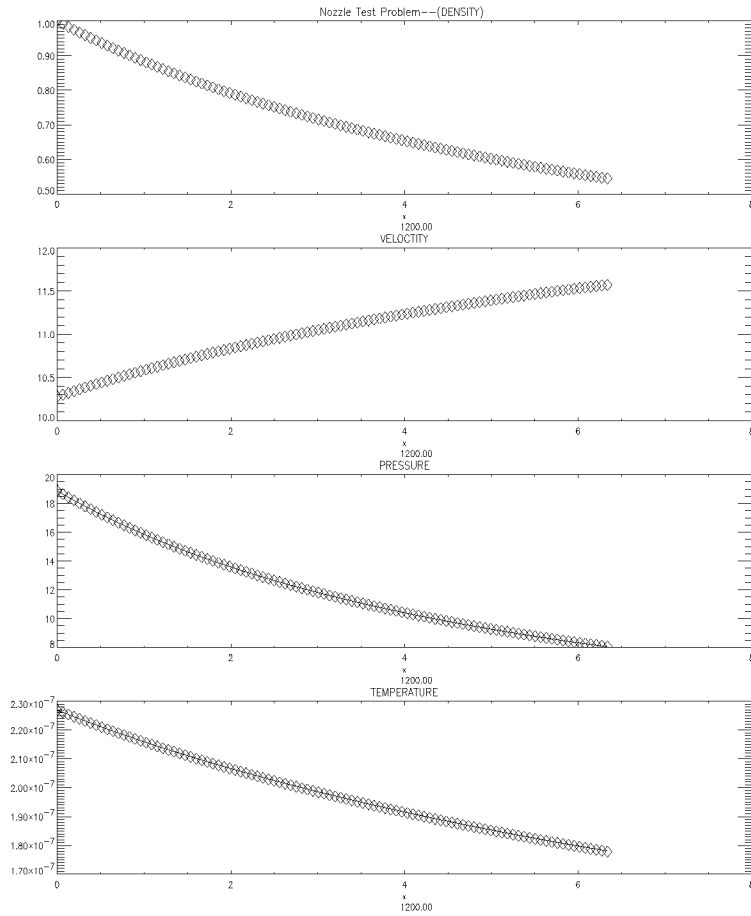
Figure 5: *Nozzle flow test. Comparison to the analytical solution for steady flow on a tube of variable cross section. The solid line represents the analytical solution.*
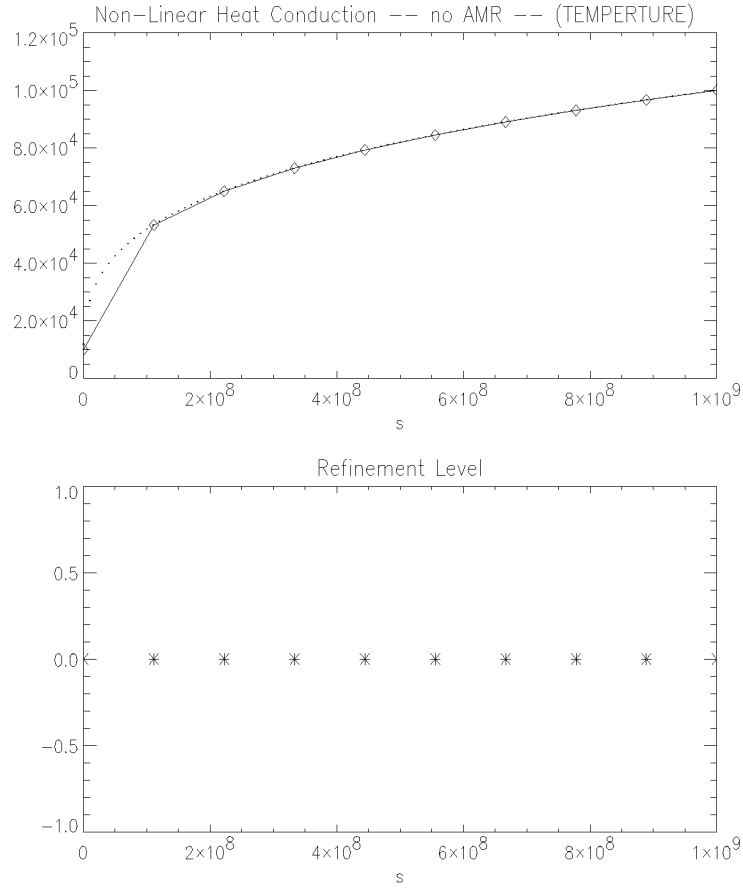
Figure 6: *Non-linear heat test. Comparison of the temperature profiles for the analytical and computed solutions of a uniform tube with non-linear heat conduction, hard wall boundary conditions, and fixed boundary temperatures. The dotted line corresponds to the analytical solution. No AMR was used for this computation. The two solutions differ only on the steep Temperature region next to the left boundary where the need for higher resolution is apparent.*
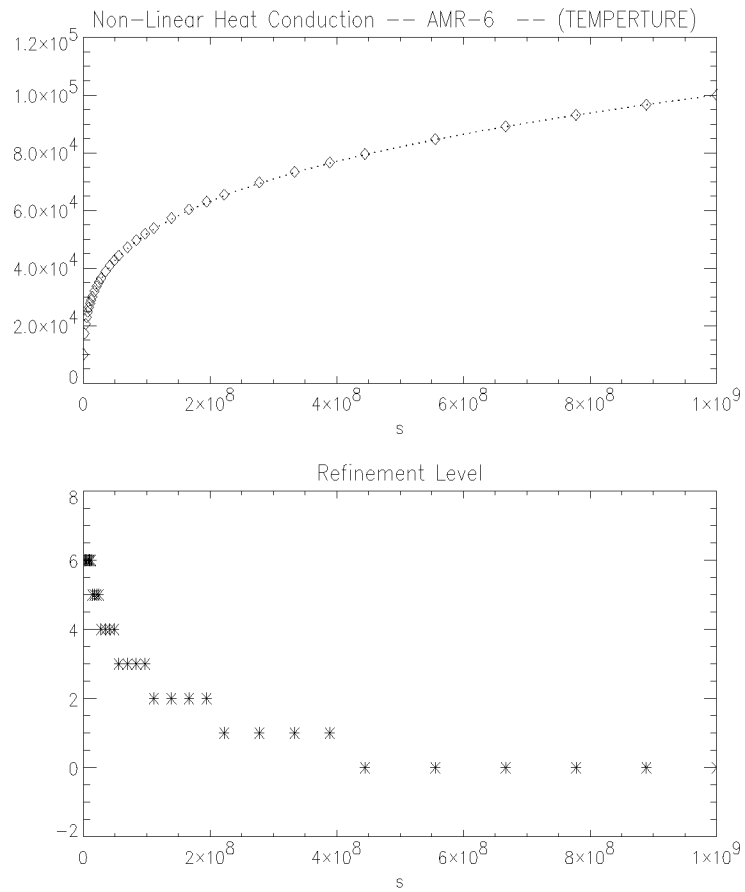
Figure 7: *Same calculation as on Figure 6 with AMR (6 levels of refinement).*

5 THE CODE SUBROUTINES

In this section the subroutines will be grouped by their functionality into: Dynamics, Geometry, Refinement, Heat and Radiation, and Limiting routines. All parameters and initial conditions are read through subroutine init.

5.1 Dynamics Subroutines

The program has a core of dynamics routines: flux1d, deltau, half, full, advance.

5.2 Geometry Subroutines

The geometry of the grid is determined in 3 subroutines:

- geom – which is generic enough to be good for any loop.

- cross – a routine that is to be modified by the user to define the cross section of the loop as a function of arc length.

- height – another routine to be modified by the user. Height defines the shape of the loop through the variable hp. The variable hp determines the height within the loop as a function of arc length.

These routines are called at initialization, and after each call to the refinement routines.

5.3 Flux Limiting Subroutines

Flux limiting is provided, as an option. The synchronized flux limiter is invoked when the input parameter "ifct" is set to 1. The minimum limiting coefficient is computed after limiting on all the unknowns, $\rho, \rho v, \rho E$. The limiting routines are:

- reglim – This routine is the driver for synchronized flux limiting.

- flofct —This routine computes the limiting coefficient with respect to a variable defined on the calling routine (reglim or charlim).

If limiting on a different combination of variables is desired, the subroutine reglim can be easily modified.

The "Characteristic Flux Limiter" is invoked when ifct is set to 2. This option for Flux Limiting is an adaptation for finite element calculations of Steven T. Zalesak original set of routines for Characteristic Flux Limiting. The characteristic limiting routines are:

- charlim –This routine is the driver for Characteristic Flux Limiting.

- avg –This routine computes element averages of the unknowns.

- rmatrix –This routine computes the characteristic transformation matrix.

- sinv – This routine transforms the low order solutions to characteristic space.

- sinvt –This routine transforms the anti-diffusion to characteristic space.

- flofct –Same routine as used for synchronized limiting.

5.4   Heat Subroutines

There are two routines called when the heat conduction term is enabled (nheat=1):

- dtheat – This routines computes the temperature and heat conduction coefficient K at each point, according to the formula,

$$K = AheatT^{hexp}$$

where Aheat and hexp are read from an input file. Subroutine dtheat also computes the maximum time step that will satisfy the stability condition

$$dtT = \frac{2R\rho\Delta x^2}{4K(\gamma - 1)}$$

where R is the gas constant: $8.31451 \times e^7$ erg $mole^{-1}$ K^o

- hheat — this subroutine computes the heat conduction contribution to the energy equation

16

5.5 Radiation Subroutines

Likewise, there are two routines called when radiative cooling (R) is present (irad=1),

- dtrad –this subroutine, computes the radiative cooling at each point (rad). It calls RADN, a subroutine to be provided by the user, that defines $\Lambda(T)$, dtrad then multiplies $\Lambda(T)$ by the square of the density. If terms such as uniform heating are to be added, they are added in this subroutine.

- radiat – this subroutine adds the contribution of radiative cooling to the energy equation.

- heating —a subroutine is also provided to allow the addition of source terms to the energy equation, such as spatially dependent heating.

5.6 AMR subroutines

Finally, the Adaptive Mesh Refinement routines are invoked when the input parameter nrmax is NOT set to zero. The AMR routines are as follows,

- meshref — this routine is the driver of the rest of the AMR routines

- grerror — mark the elements for refinement and de-refinement by computing the error indicator function ϵ at each element, and comparing it to the thresholds for refinement (ctore) and de-refinement (ctode) chosen on input.

- unref — de-refine the elements marked for de-refinement in grerror.

- refine —refine the elements marked for refinement in grerror.

- compres — compress the element arrays, getting rid of the holes left by de-refinement.

- compresp — compress the point arrays, getting rid of the holes left by de-refinement.

The output routine, OUTFEM, writes out the data after it is sorted in routine SORTOUT.

6 INPUT PARAMETER FILE

The run parameters are read by LOOPREF form an input file (fort.10).
Following is an example fort.10 file. The parameters are defined below

```
gamma           rn              cour
0.16667E+01    0.20000E+01    0.25000E+00
   ibcon(1)    ibcon(2)    nheat
       1           1          1
Aheat           hexp           T0
1.e-06        0.25000E+01    25000.
     intrans    irad       nfct       icmm
       1          1         0          0
     ntime      nwrite     npoin
   2000000      10000       100
di              di4              dmp
0.50000E+00    0.00000E+00        0.00
ctore           ctode           epsil
0.14000E+00    0.50000E-01    0.50000E-02
     nbuff      ntref      iref    nrmax
       3          10         4        8
iconti      iplot      iheat
   1          0          0
```

- gamma = specific heats ratio.

- rn = parameter used in the state equation:

$$p = rn(\gamma - 1)[rhoE - (1/2)\rho v^2]$$

- cour = Courant number

- ibcon(1) = left boundary condition parameter (1 = hard wall, 0=supersonic inflow)

- ibcon(2) = right boundary condition parameter.

- nheat = Compute heat term? (1=yes)

- Aheat = Heat conduction parameter

- hexp = Heat conduction parameter

- T0 = Temperature at the left boundary.

- itrans: 1=Transient solution 0=Steady state solution

- irad = Radiation source term? (0=no)

- nfct = limiting parameter.

    - 0 = no limiting - use low order scheme.
    - 1 = synchronized limiting
    - 2 = characteristic limiting

- icmm = choice of mass matrix for finite element computations of the high order scheme.

    - = 0 Lumped Mass Matrix
    - = 1 Consistent Mass Matrix (computed by iteration)

- ntime = Total number of time steps.

- nwrite = Number of time steps between outputs.

- di = 2nd-Order Numerical Diffusion coefficient.

- di4 = 4th Order diffusion coefficient

- ctore= Threshold of error fcn. to refine

- ctode= Threshold to de-refine

- epsil = Refinement filter coefficient

- nbuff= No. of buffer layers

- ntref= No. of time steps between refinements.

- nrmax= Max. No. of Refinement Levels.

- iref = variable chosen for computation of (discretization) error function.

  - 1=rho
  - 2=rhov
  - 3=rhoE
  - 4=Temperature

- iplot = real time graphics (0 = no – works only on SGI workstations)

- iheat = Start adding the spatially dependent heating source term after "iheat" time steps.

7 INITIAL CONDITIONS FILE

LOOPREF reads the initial data form an input file (fort.40). This input file can be generated using the program initgen. Initgen reads the run parameters from the file fort.10, calls a subroutine to be modified by the user, that determines the initial data values (x=loop coordinate, rho=density, v=velocity, rhoE=kinetic + internal energy density), and finally prints the input data into fort.40.

Below is a sample of the form of fort.40,

```
initial conditions
          it        npoin         nelem
           0          100            99
          xp            rho             v            rhoE
  0.650000000E+09  0.967760687E-13  0.000000000E+00  0.603454020E+00
  0.679797980E+09  0.955547450E-13  0.000000000E+00  0.601687968E+00
  0.709595960E+09  0.920505958E-13  0.000000000E+00  0.596522659E+00
  0.739393939E+09  0.866953990E-13  0.000000000E+00  0.588328510E+00
           .
           .
           .
           .
```

8 ACKNOWLEDGMENTS

References

[1] de Fainchtein, R., 1996, NASA Contractor Report 4718, A USERS GUIDE TO AMR1D: An Instructional Adaptive Mesh Refinement Code for Unstructured Grids.

[2] Hirsch, C., *Numerical Computation of Internal and External Flows* Vol. II, 1988, J. Wiley.

[3] Löhner, R., Morgan, K., and Zienkiewicz, O.C., *Computer Methods Appl. Mech. Eng.*, **51** (1985) p.441.

[4] Richtmyer, R.D. and Morton K.W., *Difference Methods for Initial-Value Problems*, Second Edition, 1967, Inter-science Publishers, New York.

[5] Rusanov V.V., 1961, *The Calculation of the Interaction of Non-Stationary Shock Waves and Obstacles*, Zh. vych. mat. 1: No.2, pp.267-279, USSR. [English Translation in: Rusanov V.V., 1962, J. Comp. Math. Math. Phys., No.2, USSR].

[6] Donea, J., 1984, *Int. J. Numer. Methods Eng.*, **20** p.101.

[7] Zalesak, S.T., 1973, "Fully Multidimensional Flux-Corrected Transport Algorithms for Fluids", *J. Comp. Phys.*, **31**, pp. 335-362.

[8] Löper, R., Morgan, K., Peraire, J., and Vahdati, M., 1987, "Finite Element Flux-Corrected Transport (FEM-FCT) for the Euler and Navier-Stokes Equations", *Int. J. for Num. Methods in Fluids*, **7** pp.1093-1109.

[9] Sod, Gary A., 1978, "A Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws", *J. Comp. Phys.*, **27**, pp. 1-31.

[10] Binder, R.C., *Advanced Fluid Mechanics*, Vol. I, pp. 73-76, 1958, Prentice Hall

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE April 1998	3. REPORT TYPE AND DATES COVERED Contractor Report

4. TITLE AND SUBTITLE

LOOPREF: A Fluid Code for the Simulation of Coronal Loops

5. FUNDING NUMBERS

Code 931
NAS5–32350

6. AUTHOR(S)

Rosalinda de Fainchtein, Spiro Antiochos, and Daniel Spicer

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS (ES)

Goddard Space Flight Center
Greenbelt, Maryland 20771

8. PEFORMING ORGANIZATION REPORT NUMBER

98B00038

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS (ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

NASA/CP–1998–206854

11. SUPPLEMENTARY NOTES

R. de Fainchtein: Raytheon STX, 4400 Forbes Blvd., Lanham, MD 20706
S. Antiochos:Naval Research Laboratory, Code 7675, Washington, District of Columbia 20375
D. Spicer: NASA Goddard Space Flight Center, Code 930, Greenbelt, Maryland 20771

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Unclassified–Unlimited
Subject Category: 92
Report available from the NASA Center for AeroSpace Information,
7121 Standard Drive, Hanover, MD 21076-1320. (301) 621-0390.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report documents the code LOOPREF. LOOPREF is a semi-one dimensional finite element code that is especially well suited to simulate coronal-loop phenomena. It has a full implementation of adaptive mesh refinement (AMR), which is crucial for this type of simulation. The AMR routines are an improved version of AMR1D, an AMR code that is posted on the World Wide Web: http://sdcd.gsfc.nasa.gov/ESS/exchange/contrib/de-fainchtein/adaptive_mesh_refinement.html, and documented in NASA's Contractor Report 4718 [1]. LOOPREF's versatility makes is suitable to simulate a wide variety of problems. In addition to efficiently providing very high resolution in rapidly changing regions of the domain, it is equipped to treat loops of variable cross section, any non-linear form of heat conduction, shocks, gravitational effects, and radiative loss.

14. SUBJECT TERMS

Solar Coronal Loops, Computational Techniques, Adaptive Mesh Refinement, unstructed Mesh

15. NUMBER OF PAGES
21

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39.18
298-102