# Illinois Transient Model (V. 1.1) Programmer's Manual


**Arturo S. León,**
**Christopher Martin,**
**Nils Oberg,**
**Arthur R. Schmidt**
and
**Marcelo H. García**


August 20, 2009

# Contents

# List of Figures

# Chapter 1

# Introduction

This manual presents a brief description, hierarchy and flowcharts of the main routines of the Illinois Transient Model (ITM) with the aim of helping the user's of this model for modifying the source code according with their needs. This manual also presents a brief overview of the steps for identifying a successful simulation. The ITM is a Finite Volume (FV) model intended to analyze transient and non-transient flows in closed-conduit systems ranging from dry-bed flows, to gravity flows, to partly gravity-partly surcharged flows (mixed flows) to fully pressurized flows (waterhammer flows). The ITM model, which was originally developed at the University of Illinois at Urbana-Champaign, is a Finite Volume (FV) model that can handle complex boundary conditions such as dropshafts, reservoirs, closing and opening of gates as function of time, and junctions with any number of connecting pipes and any type of horizontal and vertical alignment. The ITM model is an open source code that is in constant development and its releases are made available at both Boise State University (BSU) and the University of Illinois at Urbana-Champaign (UIUC). For the user's manual of the ITM model the reader is referred to León et al. (2009a).

## Disclaimer:

ITM is a free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful,

# Chapter 2

# Description of Subroutines

This section presents a brief description of the main routines of the ITM model.

## Area_from_H

This routine computes the hydraulic area from the piezometric head.

## boundaries

This routine is the main module for computing fluxes at boundaries.

## CFL_time_step

This routine computes the time step according with the Courant criteria. This routine is also used for relaxing the time step when there are no inflows.

## const_bound

This routine computes the flux when using a constant boundary (flow discharge or piezometric head).

## converg

This routine is used for determining if convergence has been achieved when solving non-linear equations.

## Dropshaft_general

This routine is used for computing fluxes at a drop-shaft boundary.

## ERP_Combined

This routine is used for performing reconstruction of the flow variables to achieve second-order accuracy.

## Freesurface_flowregime

This routine is used for determining if a junction/reservoir/dropshaft boundary is surcharged or not.

## H_from_Area

This routine computes the piezometric head from the hydraulic area.

## interf_speed

This routine computes the interface speed of a mixed flow interface.

## itm_conser_volume

This routine compares the water entered to the system and the water stored in the system.

## junct2pipes_same_diam

This routine computes fluxes at a junction boundary of two pipes that have the same diameter with no inflow hydrograph.

## junction_general

This routine computes fluxes at a junction boundary of $N$ connecting pipes with or without inflow hydrographs.

# MAIN PROGRAM

MAIN PROGRAM is the main module of the ITM model and contains the main structure of the model.

## negint1

This routine computes fluxes at a negative mixed flow interface moving downstream.

## negint2

This routine computes fluxes at a negative mixed flow interface moving upstream.

## posint1

This routine computes fluxes at a positive mixed flow interface moving downstream.

## posint2

This routine computes fluxes at a positive mixed flow interface moving upstream.

## Pressure_Pho

This routine is used for computing $\frac{A\bar{p}}{\rho_{ref}}$ and $\frac{A_f p}{\rho_{ref}}$ in León et al. 2009b.

## Reservoirs

This routine is used for computing fluxes at a reservoir boundary.

## Riemann_open

This routine is used for computing fluxes at a pure free surface flow interface.

## Riemann_pressur

This routine is used for computing fluxes at a pure pressurized flow interface.

## Riemann_pressur_open

This routine is used for computing fluxes at an open channel-pressurized flow interface (mixed flow interface).

## Source_term

This routine is used for incorporating source terms (friction and gravity terms) into the solution through time splitting.

## Ycrit

This routine is used for computing the critical depth.

## Ynormal

This routine is used for computing the normal depth.

## Yconjugate

This routine is used for computing the conjugate depth (supercritical flow).

# Chapter 3

# Hierarchy and Flowcharts of Subroutines

This section presents the hierarchy and the flowcharts for the main routines of the ITM model.

## 3.1 Hierarchy

1. **Area_from_H**

    - Calls:
    - Called by:
        (a) Conjug_iter
        (b) const_bound
        (c) Crit_iter
        (d) dropsh
        (e) Freesurface_flowregime
        (f) ITM_EXEC_STEP
        (g) junction_general
        (h) junction_solver
        (i) negint1
        (j) negint2
        (k) Norm_iter

(l) Reser
(m) Reservoirs
(n) Riemann_open
(o) Riemann_pressur_open
(p) Source_term

2. **boundaries**
   - Calls:
     (a) const_bound
     (b) Dropshaft_general
     (c) Reservoirs
     (d) junction_general
     (e) junct2pipes_same_diam
   - Called by:
     (a) ITM_EXEC_STEP

3. **CFL_time_step**
   - Calls:
   - Called by:
     (a) ITM_EXEC_STEP

4. **const_bound**
   - Calls:
     (a) Area_from_H
     (b) Pressure_Pho
     (c) const_depth
     (d) const_disch
     (e) converg
     (f) Riemann_open
     (g) Riemann_pressur
     (h) Riemann_pressur_open
     (i) H_from_Area

- Called by:
  (a) boundaries

5. **const_depth**
   - Calls:
   - Called by:
     (a) const_bound
6. **const_disch**
   - Calls:
   - Called by:
     (a) const_bound
7. **converg**
   - Calls:
   - Called by:
     (a) air_pocket
     (b) const_bound
     (c) Dropshaft_general
     (d) H_from_Area
     (e) junction_general
     (f) Reservoirs
     (g) Riemann_pressur_open
     (h) Ycrit
     (i) Ynormal
8. **Crit_iter**
   - Calls:
     (a) Area_from_H
   - Called by:
     (a) Ycrit

9. **der_Phi**
   - Calls:
   - Called by:
     (a) H_from_Phi
10. **dropsh**
   - Calls:
     (a) Area_from_H
   - Called by:
     (a) Dropshaft_general
11. **Dropshaft_general**
   - Calls:
     (a) get_inflow
     (b) Freesurface_flowregime
     (c) Pressure_Pho
     (d) Area_from_H
     (e) dropsh
     (f) converg
   - Called by:
     (a) boundaries
12. **ERP_Combined**
   - Calls:
     (a) H_from_Area
     (b) Pressure_Pho
   - Called by:
     (a) ITM_EXEC_STEP
     (b) junct2pipes_same_diam
     (c) junction_general
     (d) Reservoirs

13. **Freesurface_flowregime**
    - Calls:
      (a) Area_from_H
      (b) Riemann_pressur_open
    - Called by:
      (a) Dropshaft_general
      (b) Reservoirs
14. **H_from_Area**
    - Calls:
      (a) H_Area
      (b) converg
    - Called by:
      (a) const_bound
      (b) ERP_Combined
      (c) Freesurface_flowregime_point_junction
      (d) junction_general
      (e) Riemann_open
      (f) Source_term
15. **H_Area**
    - Calls:
    - Called by:
      (a) H_from_Area
16. **interf_speed**
    - Calls:
    - Called by:
      (a) Riemann_pressur_open
      (b) Source_term
17. **itm_conser_volume**

- Calls:
  (a) get_storage
  (b) get_curve_area
- Called by:
  (a) ITM_EXEC_STEP

18. **itm_conser_volume**
    - Calls:
      (a) init
    - Called by:
      (a) MAIN PROGRAM

19. **ITM_EXEC_STEP**
    - Calls:
      (a) CFL_time_step
      (b) itm_conser_volume
      (c) boundaries
      (d) ERP_combined
      (e) Riemann_open
      (f) Riemann_pressur
      (g) Riemann_pressur_open
      (h) Source_term
      (i) Area_from_H
      (j) report_all
      (k) Plot_several_pipes
      (l) itm_step
    - Called by:
      (a) MAIN PROGRAM

20. **junct2pipes_same_diam**
    - Calls:

(a) ERP_combined
(b) Riemann_open
(c) Riemann_pressur
(d) Riemann_pressur_open

- Called by:

(a) boundaries

21. **junction_general**

- Calls:

(a) get_inflow
(b) ERP_combined
(c) Pressure_Pho
(d) Phi1
(e) Area_from_H
(f) Riemann_open
(g) H_from_Area
(h) junction_solver
(i) converg

- Called by:

(a) boundaries

22. **junction_solver**

- Calls:

(a) Area_from_H
(b) Phi1

- Called by:

(a) junction_general

23. **MAIN PROGRAM**

- Calls:

(a) getarg
(b) ITM_DLL_INIT

(c) ITM_EXEC_STEP
(d) ITM_DLL_END

- Called by:

24. **negint1**
    - Calls:
      (a) Area_from_H
    - Called by:
      (a) Riemann_pressur_open

25. **negint2**
    - Calls:
      (a) Area_from_H
    - Called by:
      (a) Riemann_pressur_open

26. **Norm_iter**
    - Calls:
      (a) Area_from_H
    - Called by:
      (a) Ynormal

27. **Phi1**
    - Calls:
    - Called by:
      (a) H_from_Phi
      (b) junction_general
      (c) junction_solver
      (d) Riemann_open

28. **posint1**
    - Calls:

- Called by:
  (a) Riemann_pressur_open

29. **posint2**
    - Calls:
    - Called by:
      (a) Riemann_pressur_open

30. **Pressure_Pho**
    - Calls:
    - Called by:
      (a) const_bound
      (b) Dropshaft_general
      (c) ERP_Combined
      (d) junction_general
      (e) Reservoirs
      (f) Riemann_open
      (g) Riemann_pressur_open

31. **Reser**
    - Calls:
      (a) Area_from_H
      (b) get_storage
    - Called by:
      (a) Reservoirs

32. **Reservoirs**
    - Calls:
      (a) ERP_combined
      (b) Freesurface_flowregime
      (c) Pressure_Pho
      (d) Area_from_H

(e) get_storage
(f) Reser
(g) converg

- Called by:

(a) boundaries

33. **Riemann_open**

- Calls:

(a) Area_from_H
(b) Pressure_Pho
(c) Phi1
(d) Linearized
(e) H_from_Area

- Called by:

(a) const_bound
(b) ITM_EXEC_STEP
(c) junct2pipes_same_diam
(d) junction_general
(e) Riemann_pressur_open

34. **Riemann_pressur**

- Calls:

(a) Linearized

- Called by:

(a) const_bound
(b) ITM_EXEC_STEP
(c) junct2pipes_same_diam

35. **Riemann_pressur_open**

- Calls:

(a) Area_from_H
(b) Pressure_Pho

(c) interf_speed
(d) converg
(e) Riemann_open
(f) negint1
(g) negint2
(h) posint2
(i) posint1

- Called by:
  (a) air_pocket
  (b) const_bound
  (c) Freesurface_flowregime
  (d) ITM_EXEC_STEP
  (e) junct2pipes_same_diam

36. **Source_term**

- Calls:
  (a) interf_speed
  (b) H_from_Area
  (c) Area_from_H
- Called by:
  (a) ITM_EXEC_STEP

37. **Ycrit**

- Calls:
  (a) Crit_iter
  (b) converg
- Called by:
  (a) Freesurface_flowregime_point_junction

38. **Ynormal**

- Calls:
  (a) Norm_iter

(b) converg

- Called by:

  (a) Freesurface_flowregime_point_junction

39. **Yconjugate**

- Calls:
- Called by:

  (a) Freesurface_flowregime_point_junction

## 3.2 Flowcharts

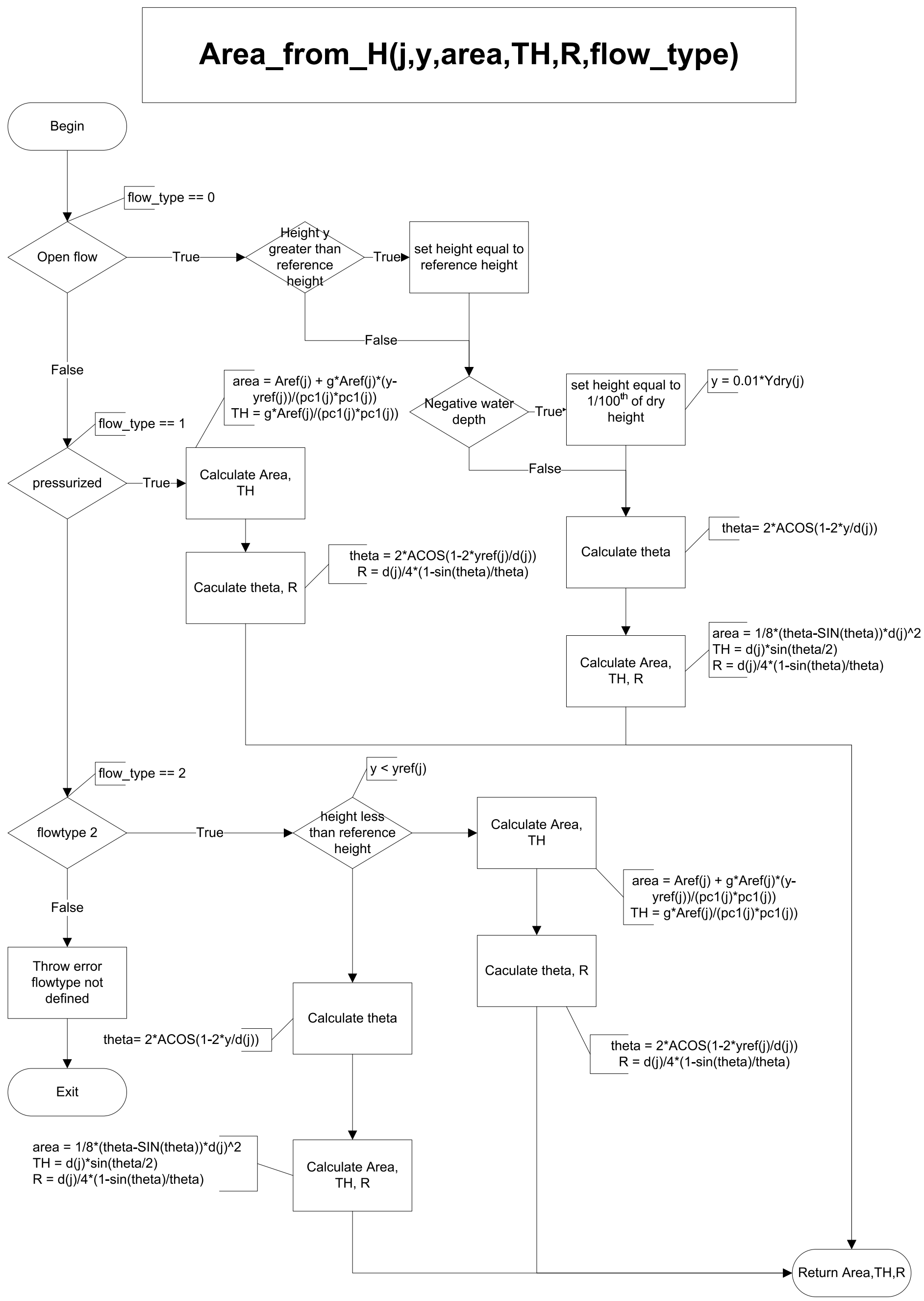


Figure 3.1: Flowchart for routine Area_from_H

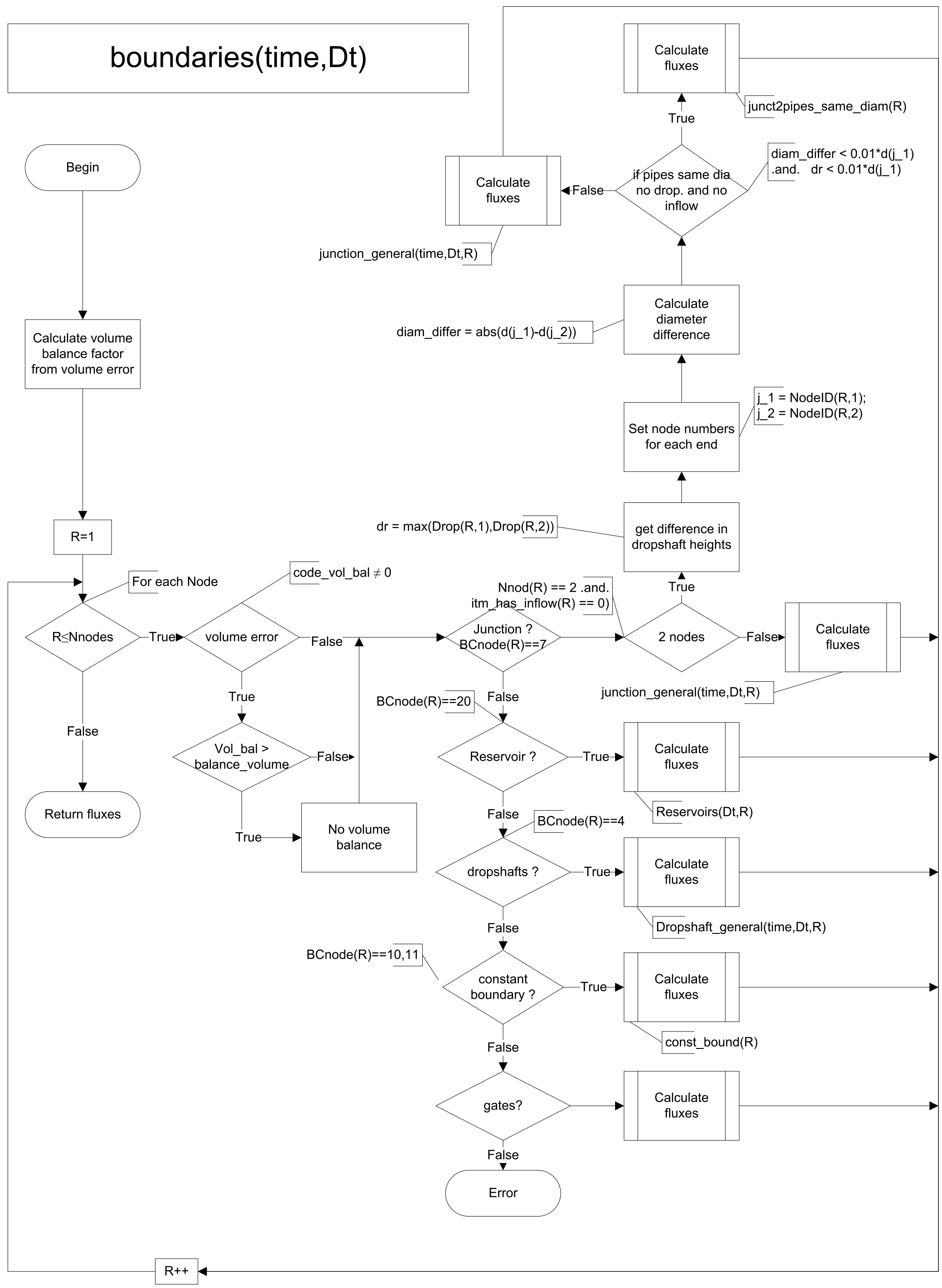


Figure 3.2: Flowchart for routine boundaries

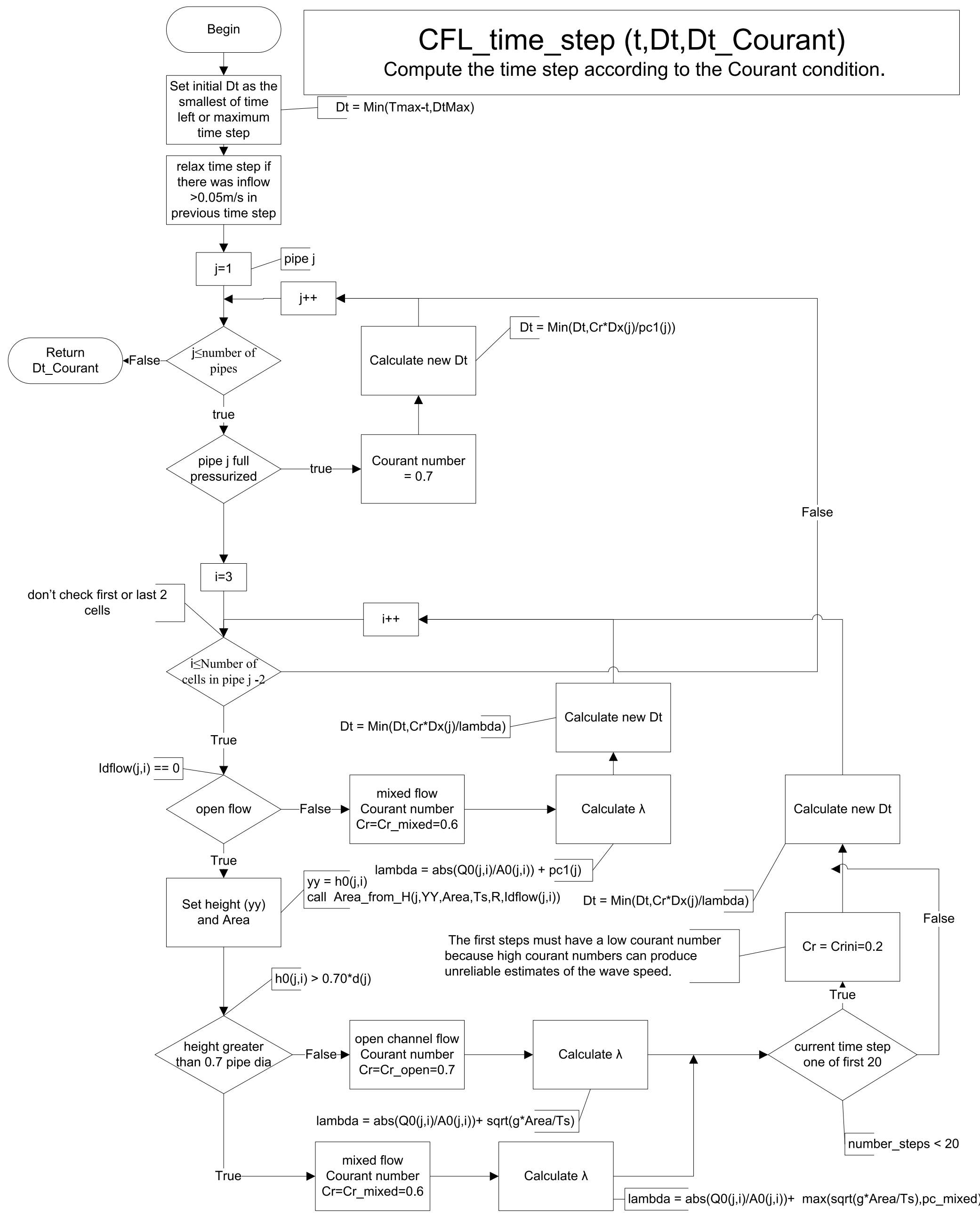


Figure 3.3: Flowchart for routine CFL_time_step

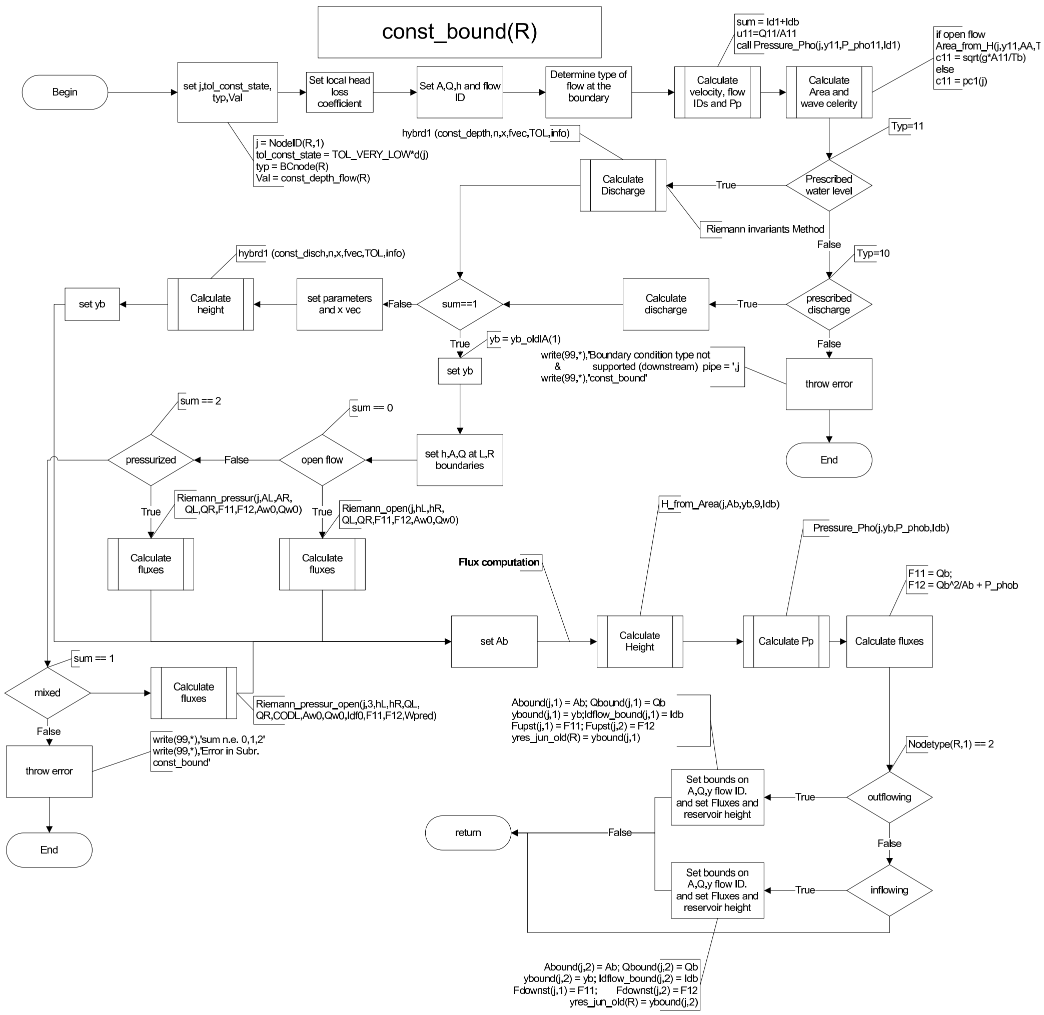


Figure 3.4: Flowchart for routine const_bound

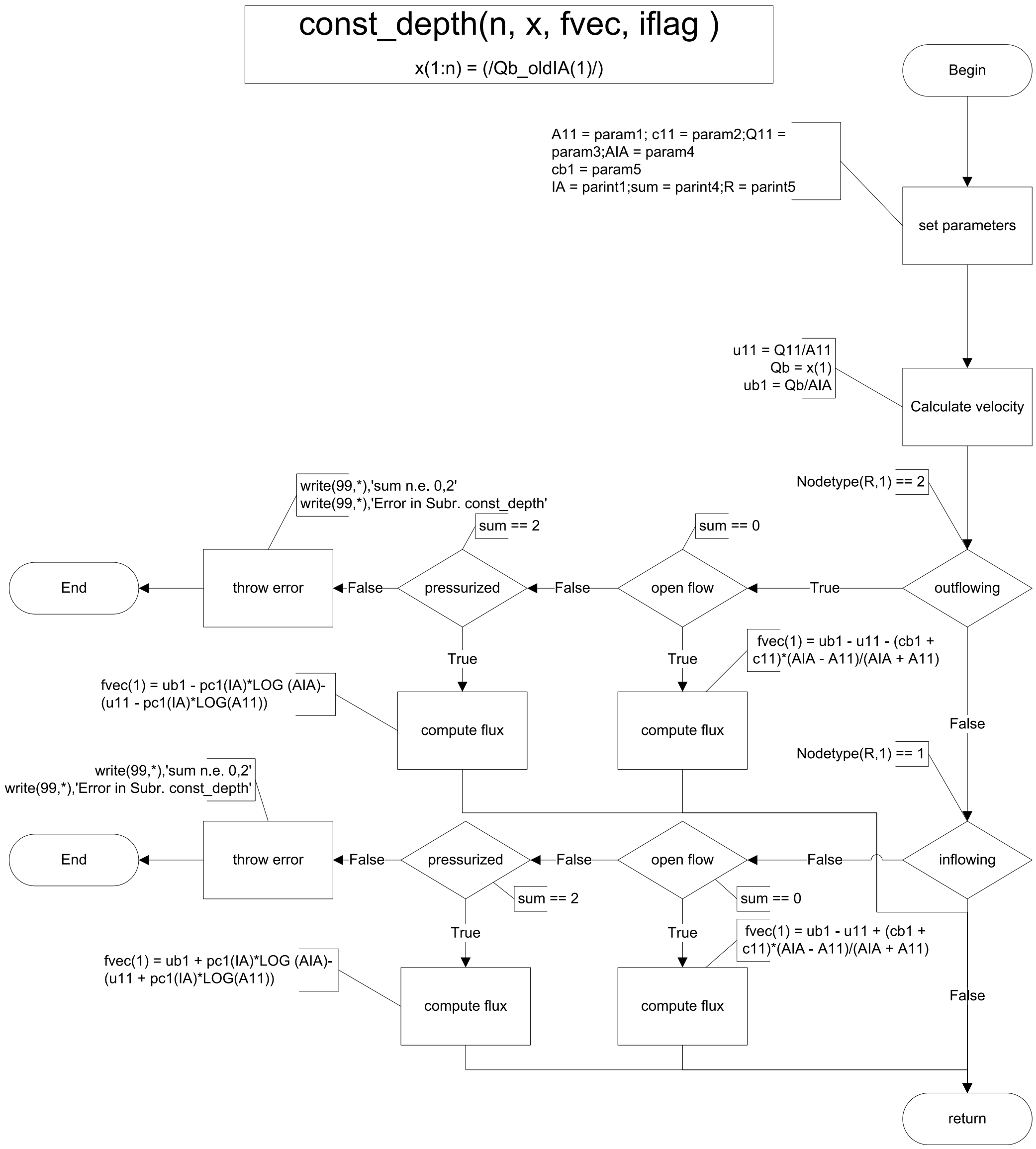


Figure 3.5: Flowchart for routine const_depth

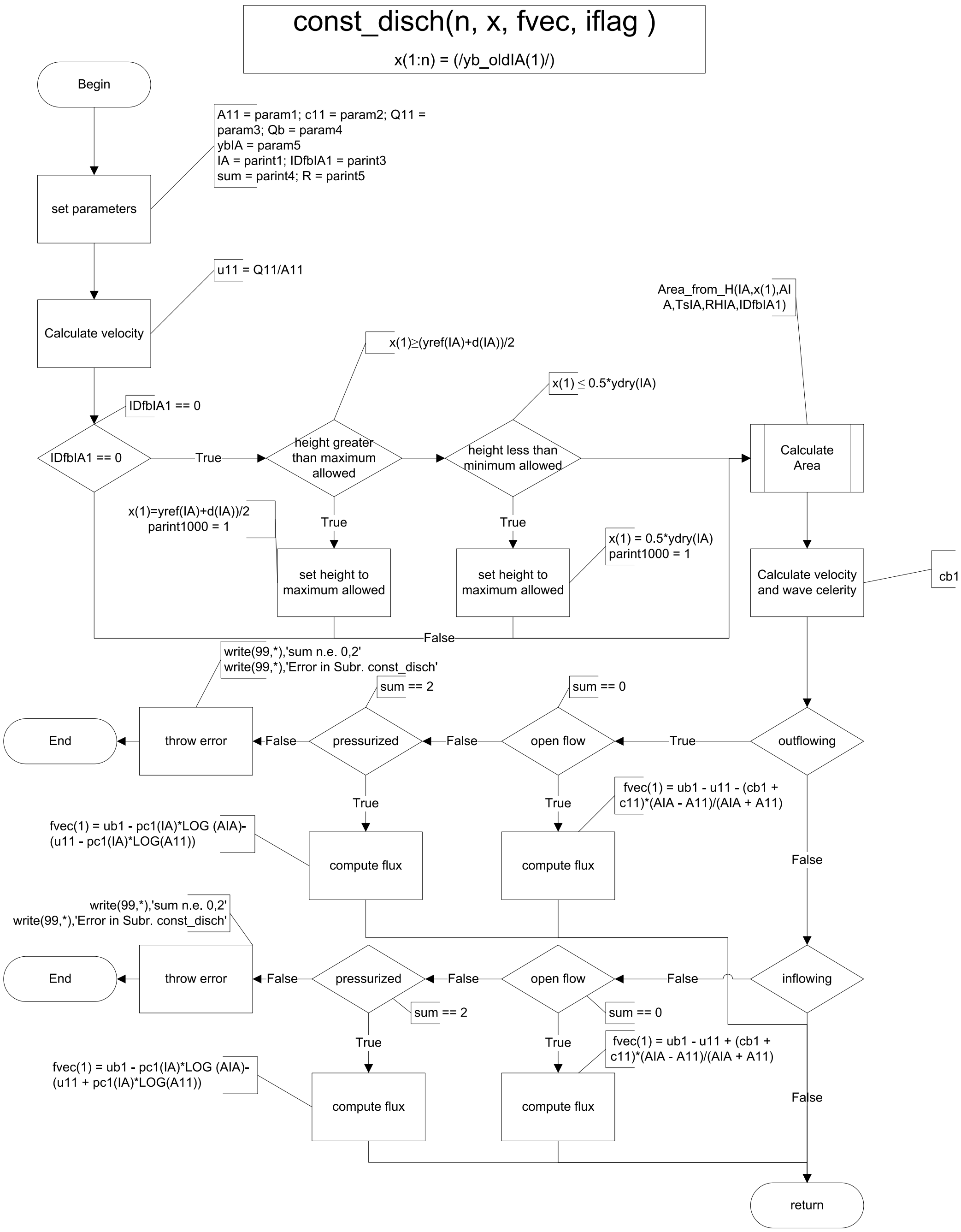


Figure 3.6: Flowchart for routine const_disch

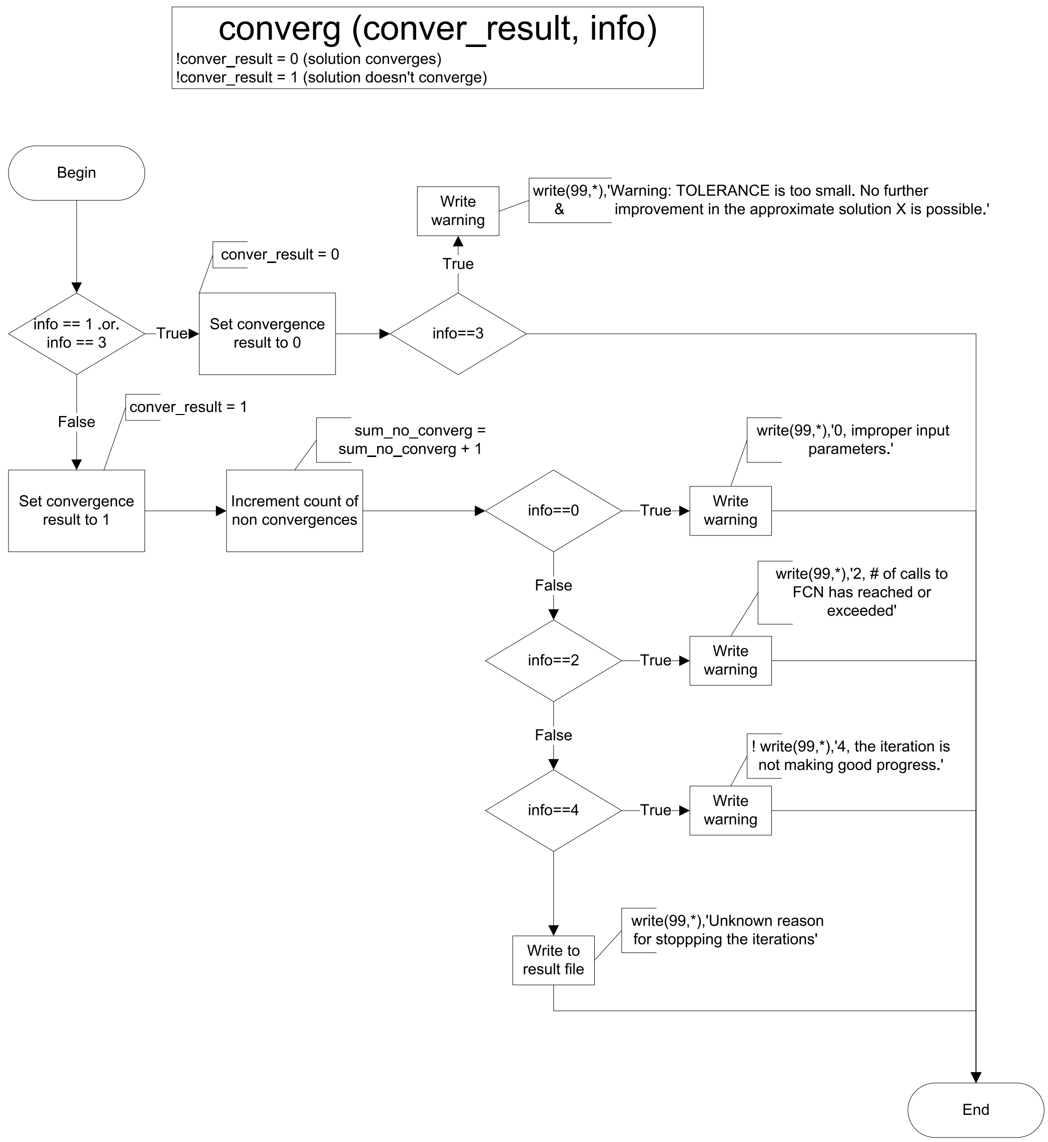


Figure 3.7: Flowchart for routine converg

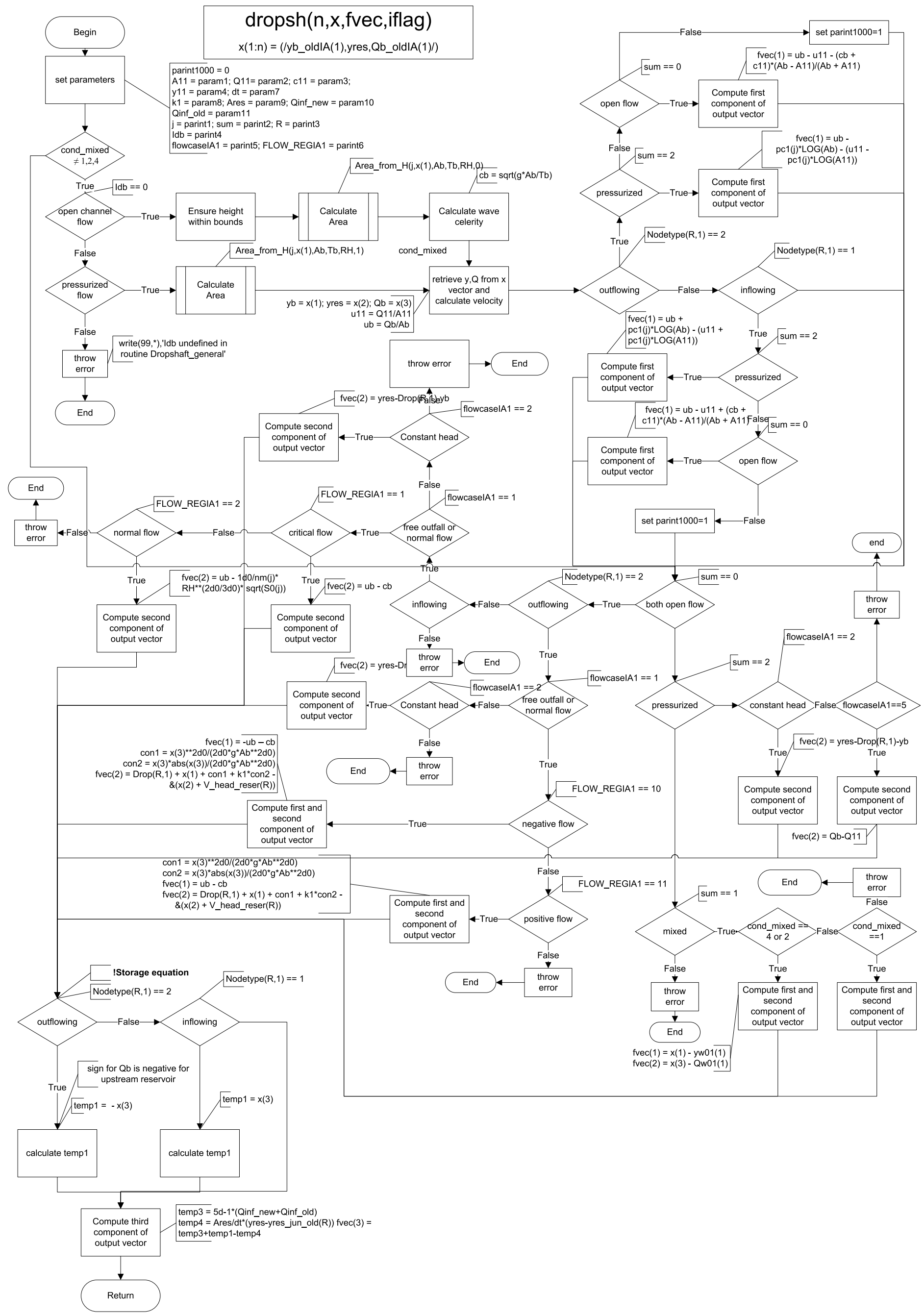

Figure 3.8: Flowchart for routine dropsh

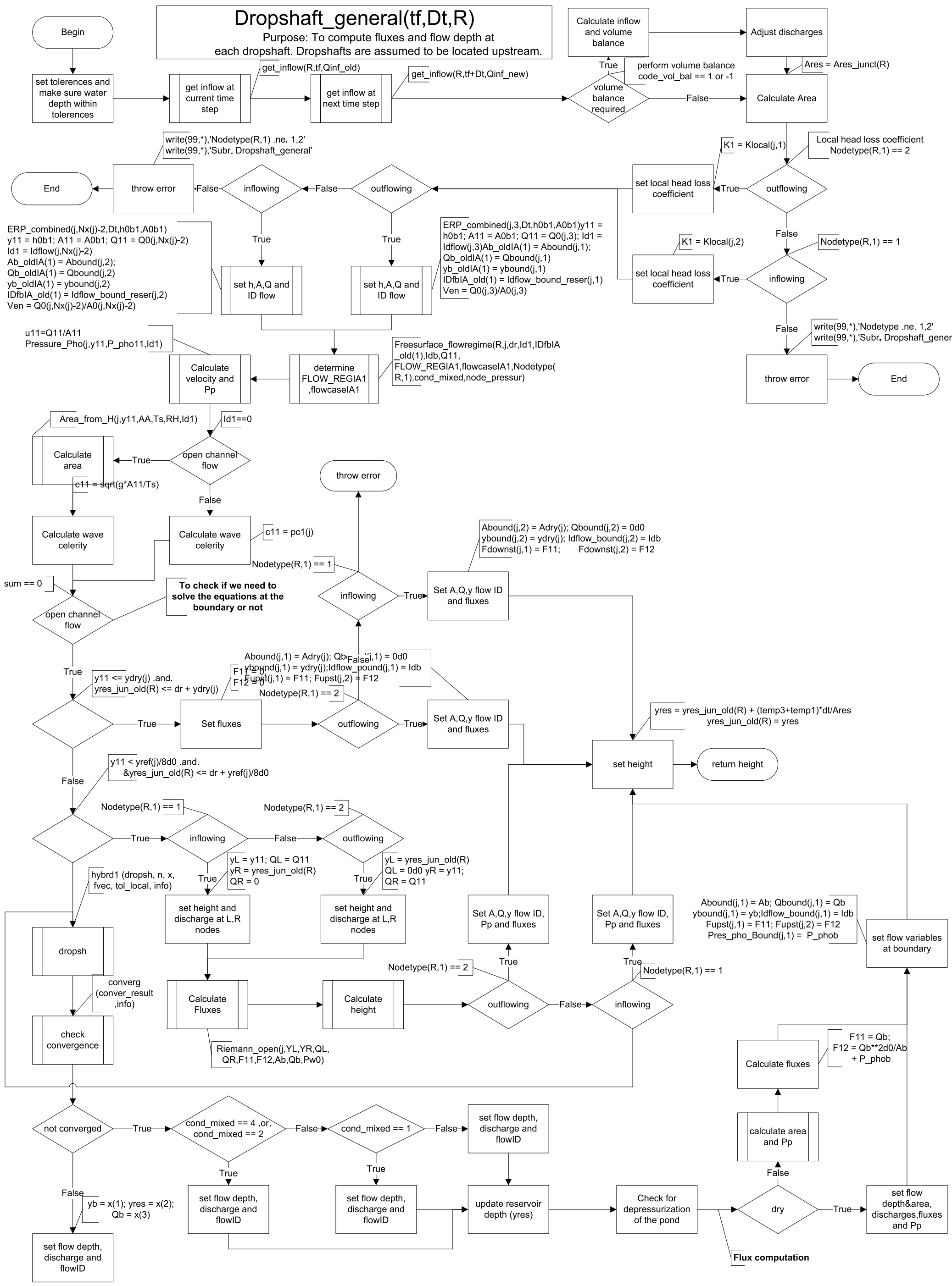


Figure 3.9: Flowchart for routine Dropshaft_general

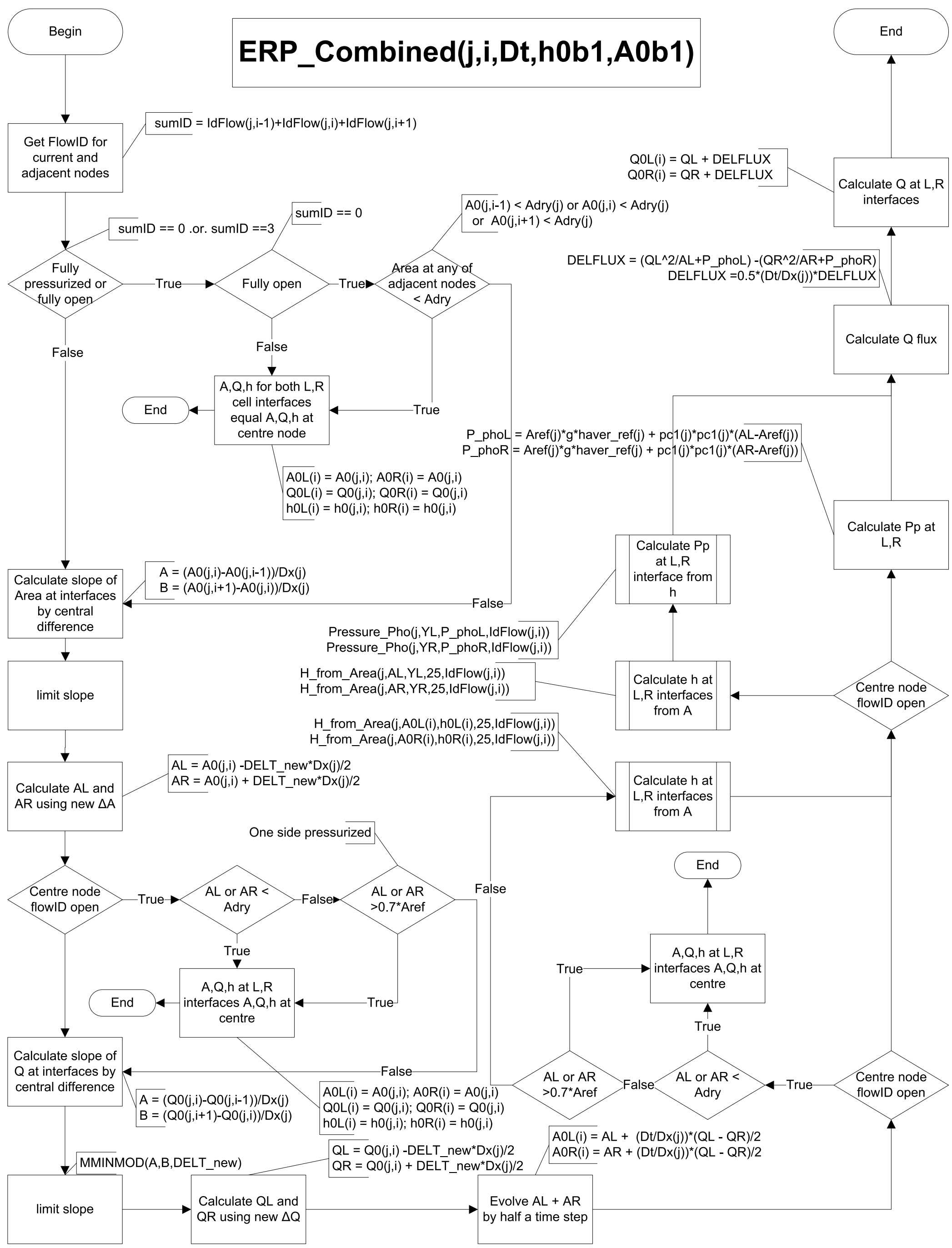


Figure 3.10: Flowchart for routine ERP_Combined

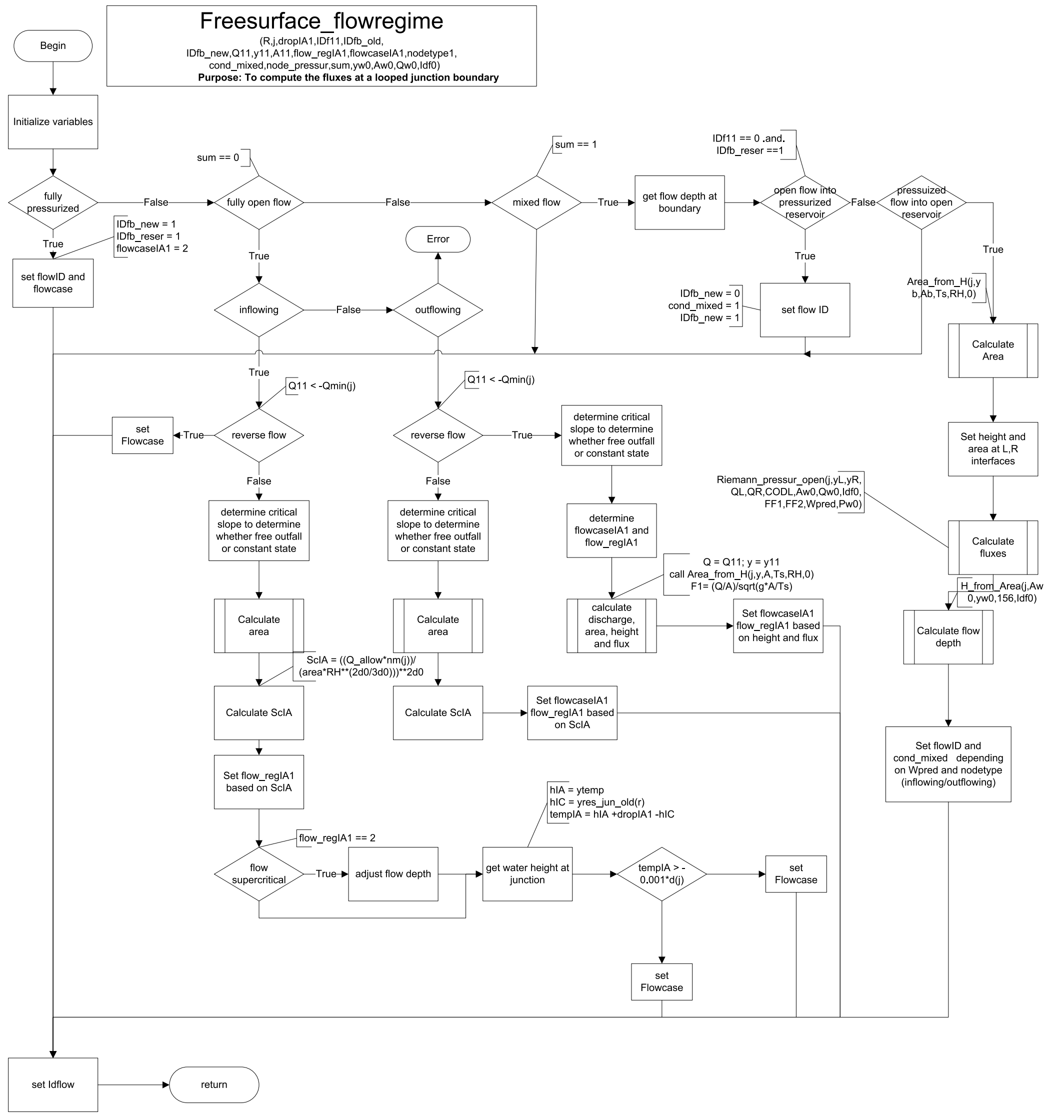


Figure 3.11: Flowchart for routine Freesurface_flowregime

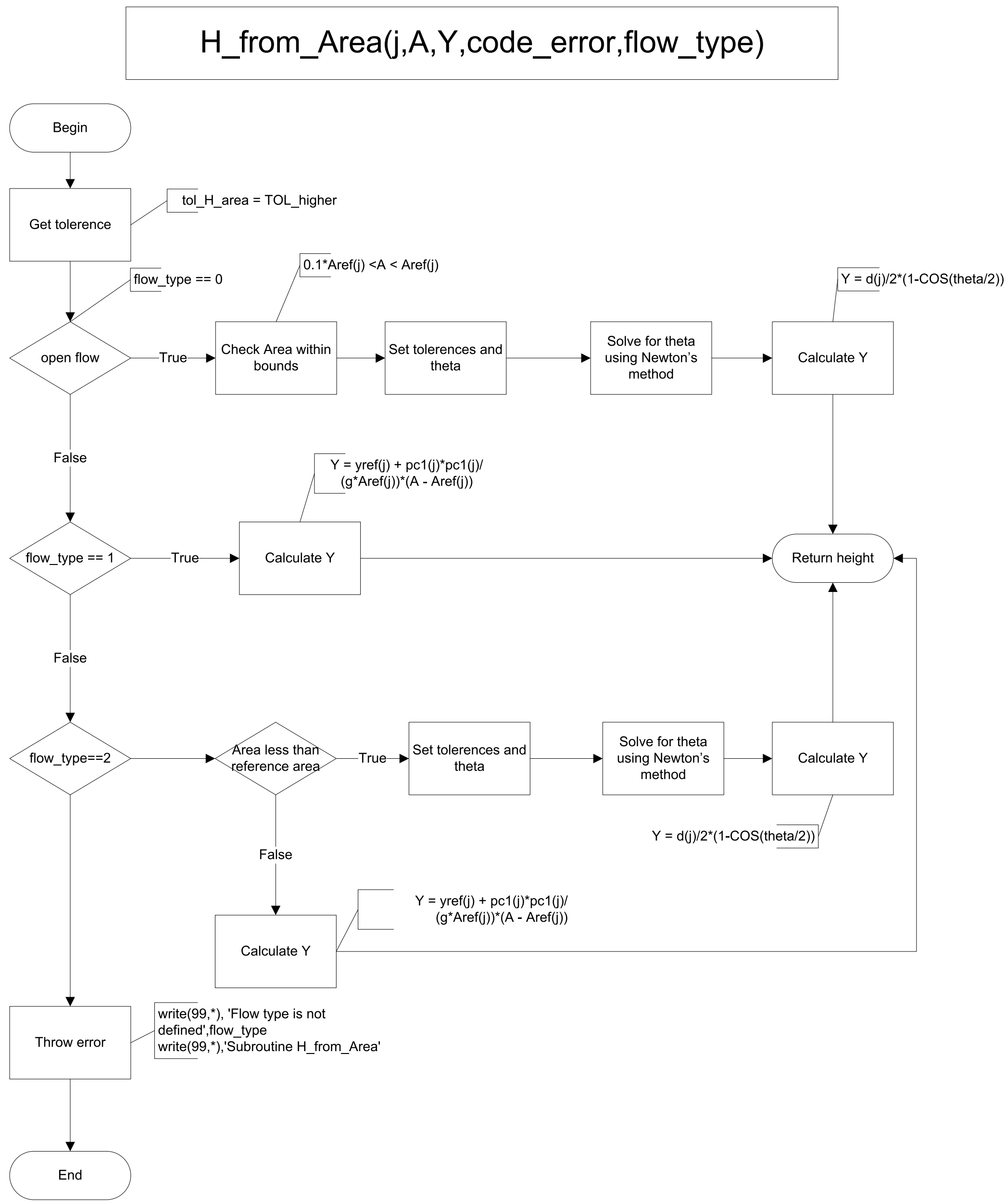


Figure 3.12: Flowchart for routine H_from_Area

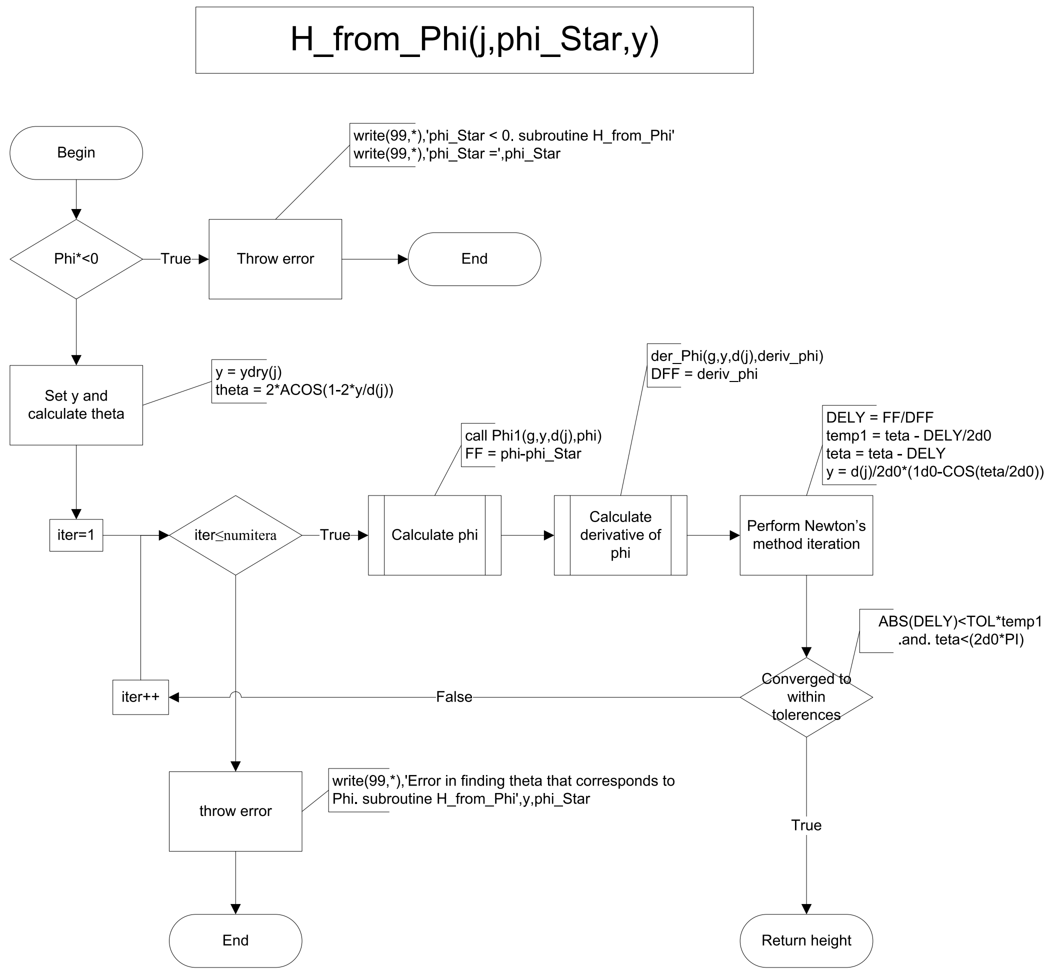


Figure 3.13: Flowchart for routine H_from_Phi

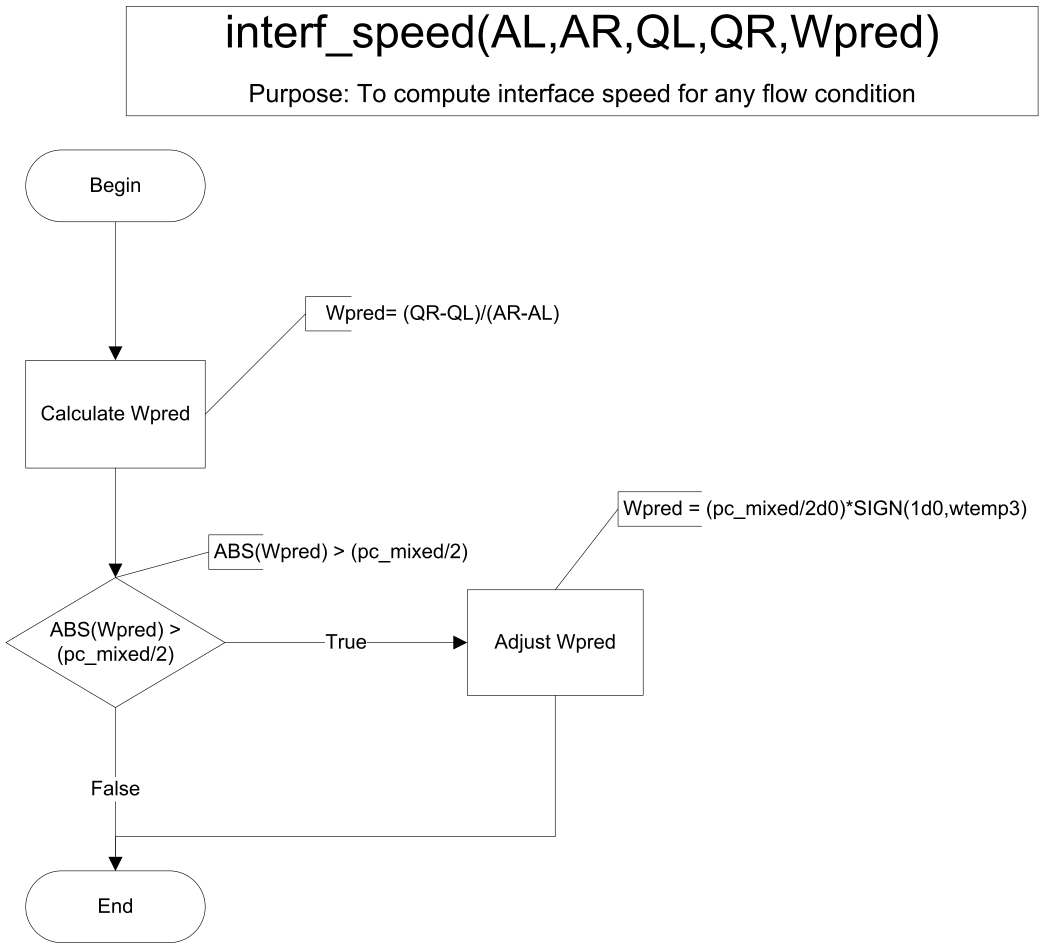


Figure 3.14: Flowchart for routine interf_speed

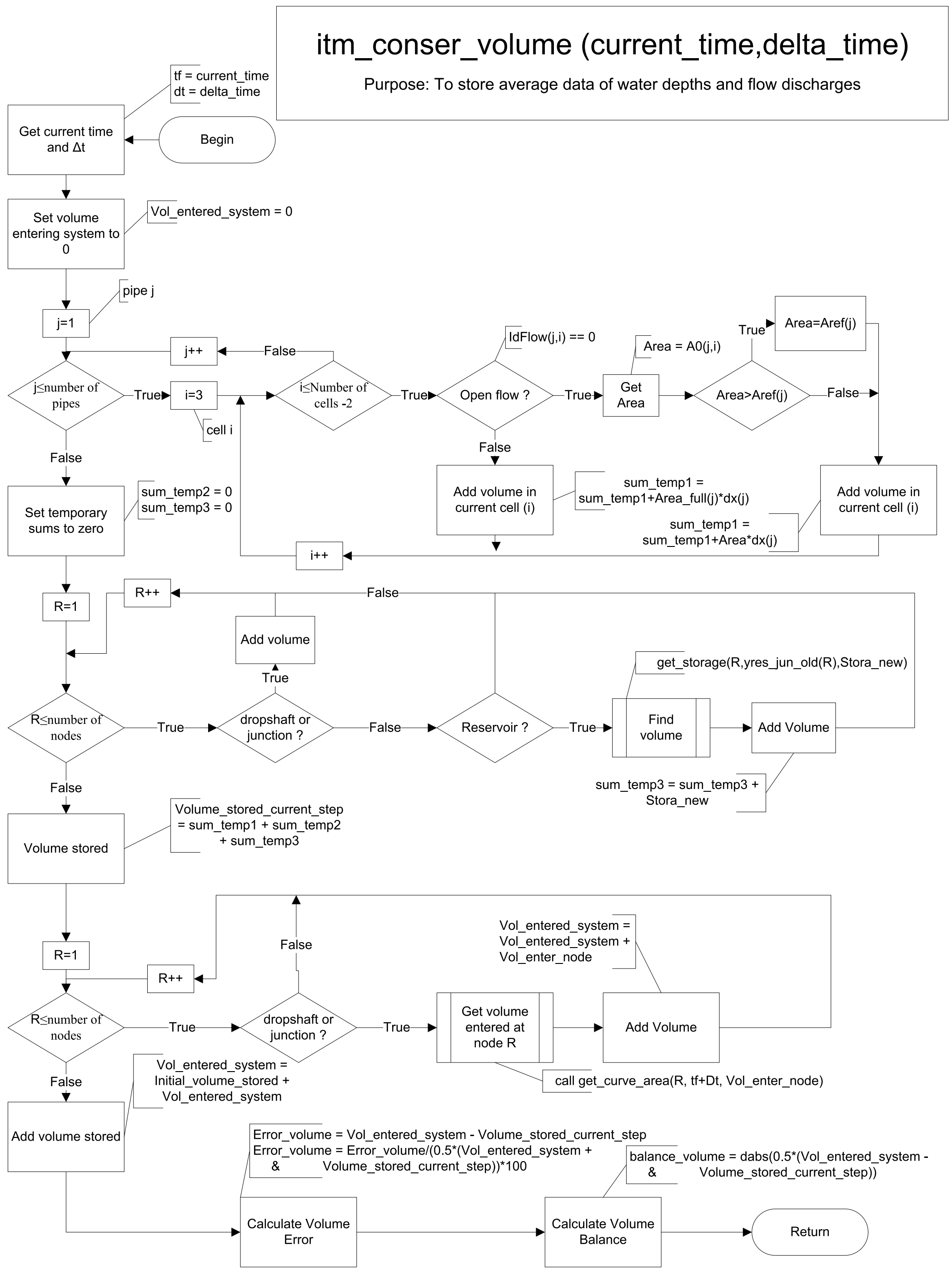


Figure 3.15: Flowchart for routine itm_conser_volume

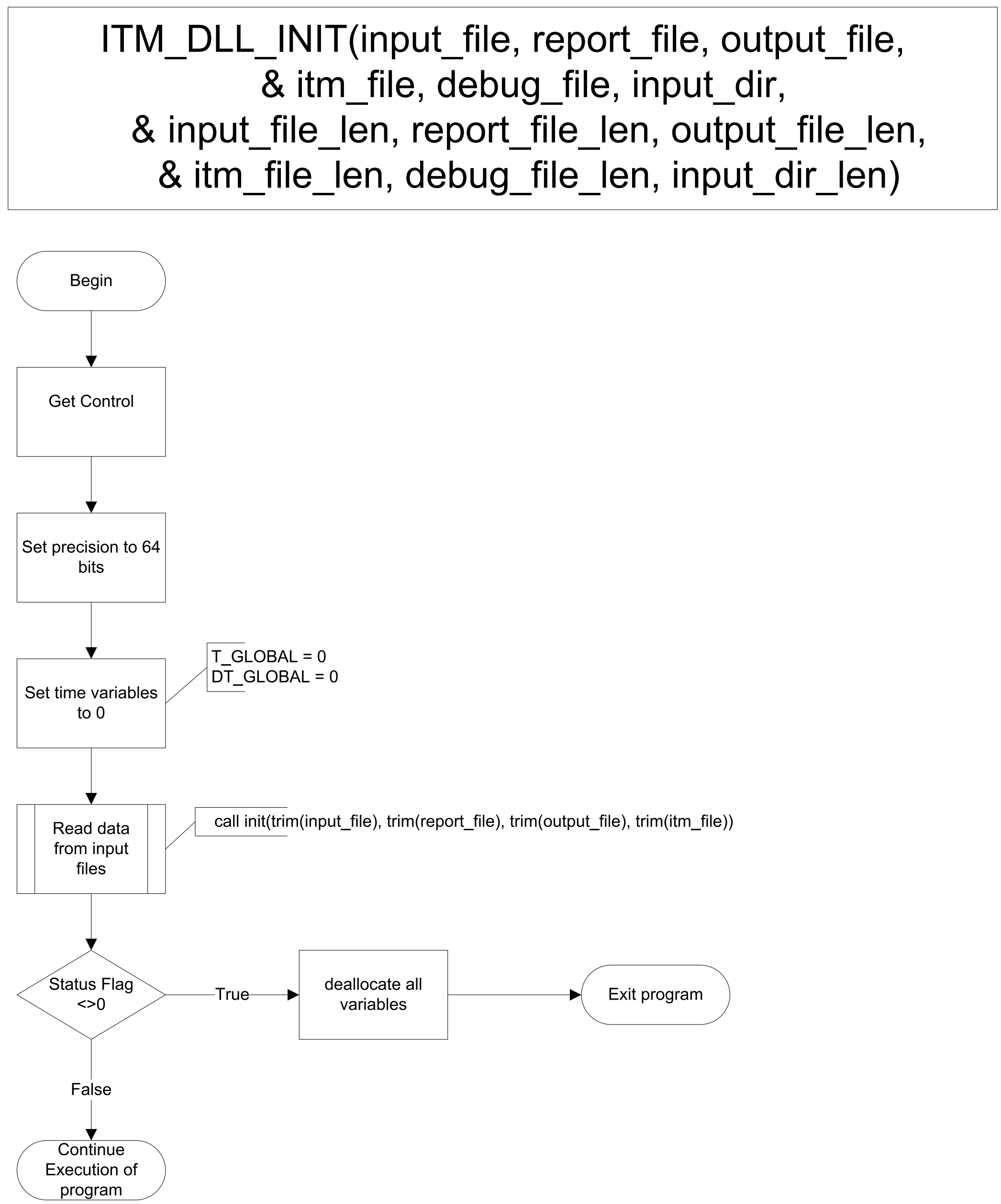


Figure 3.16: Flowchart for routine itm_dll_init

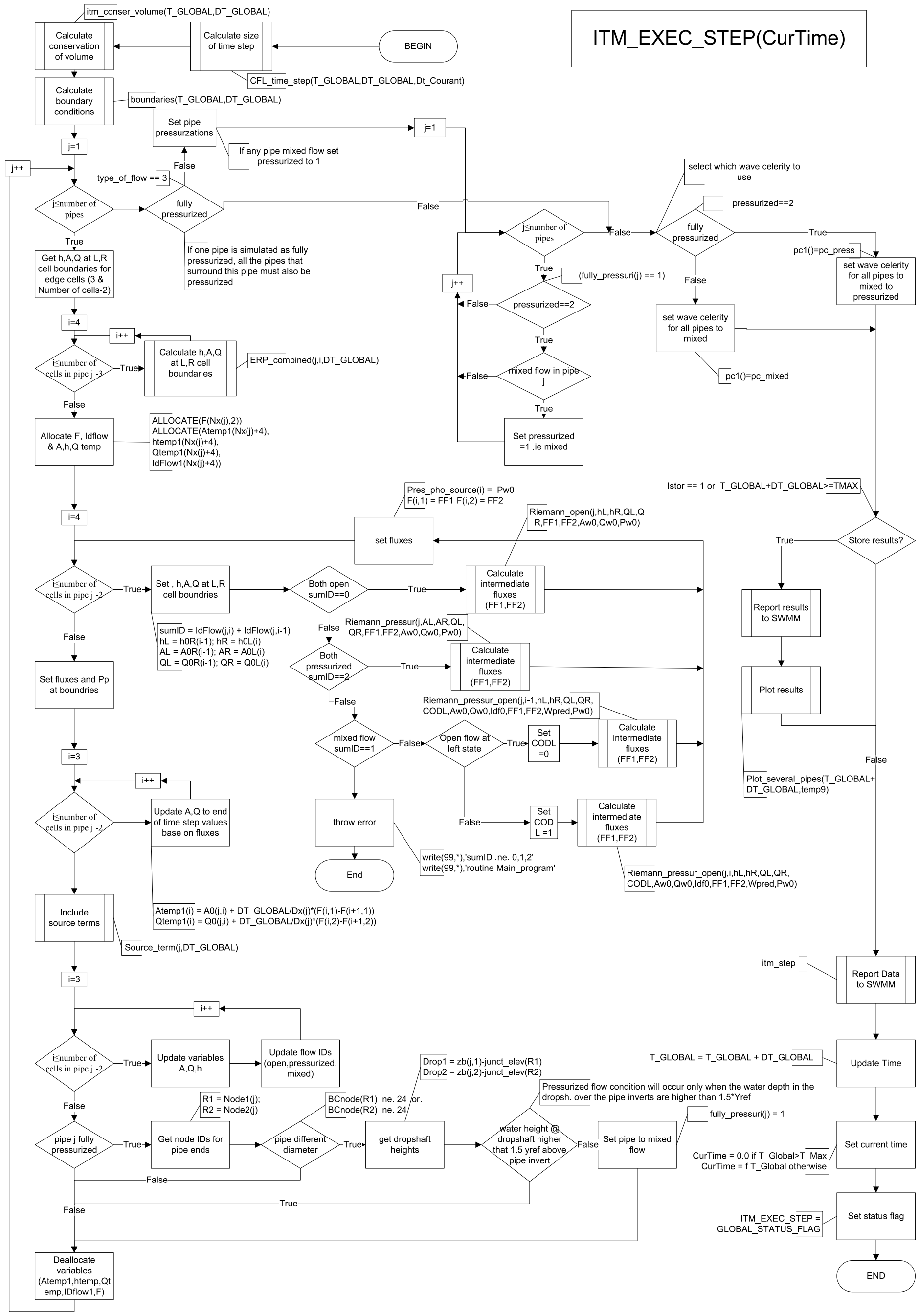


Figure 3.17: Flowchart for routine itm_exec_step

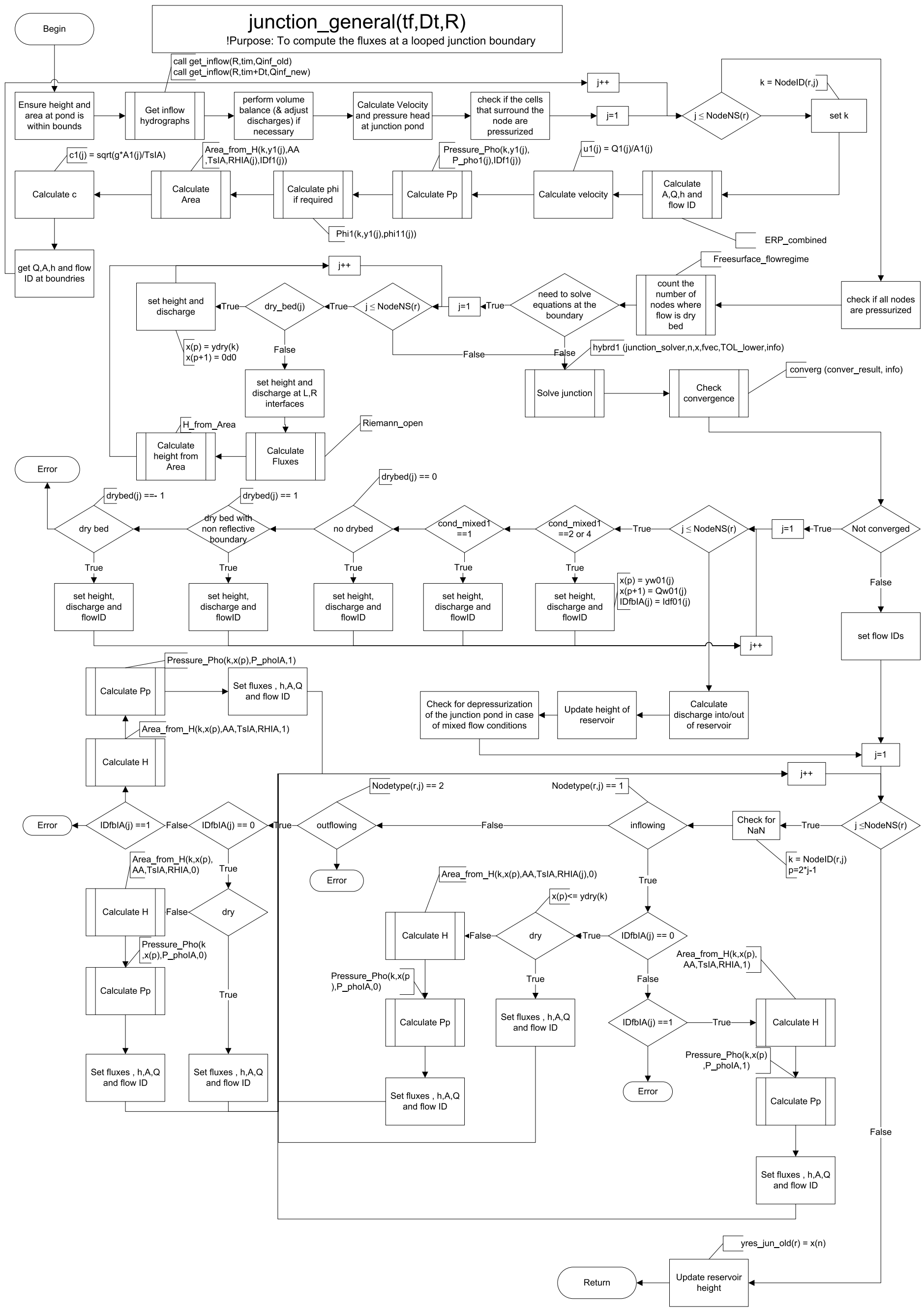

Figure 3.18: Flowchart for routine junction_general

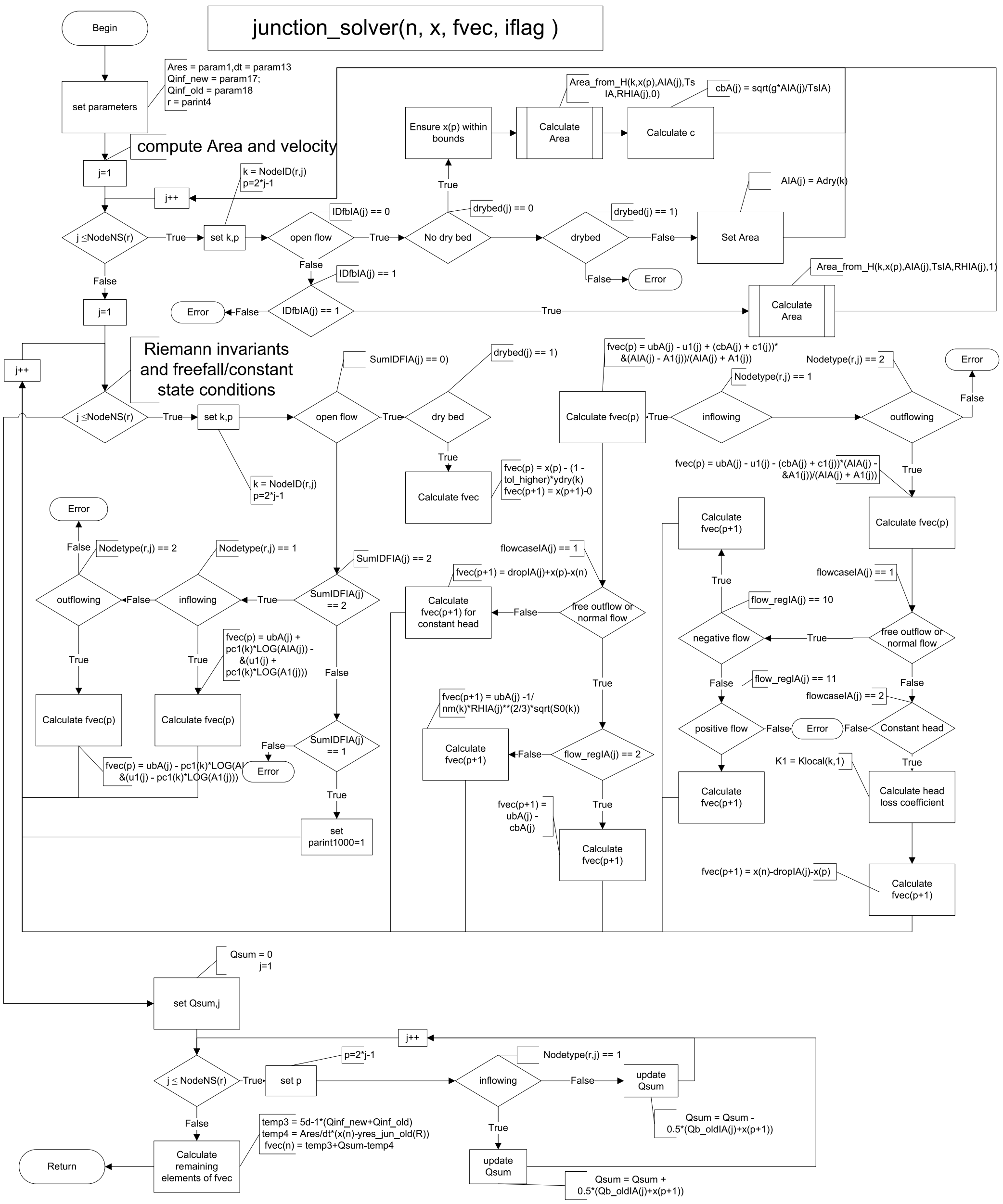


Figure 3.19: Flowchart for routine junction_solver

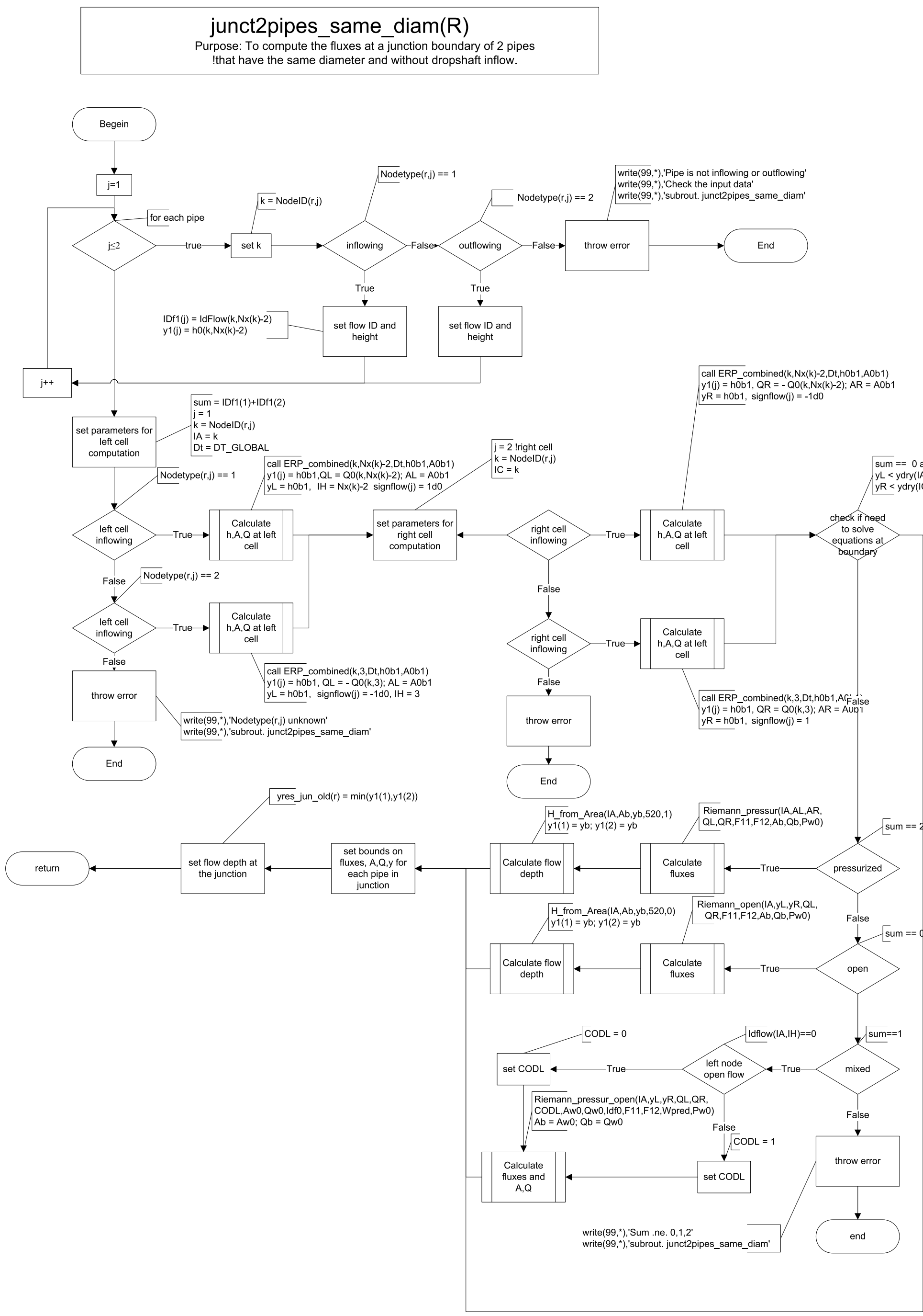


Figure 3.20: Flowchart for routine junct2pipes_same_diam

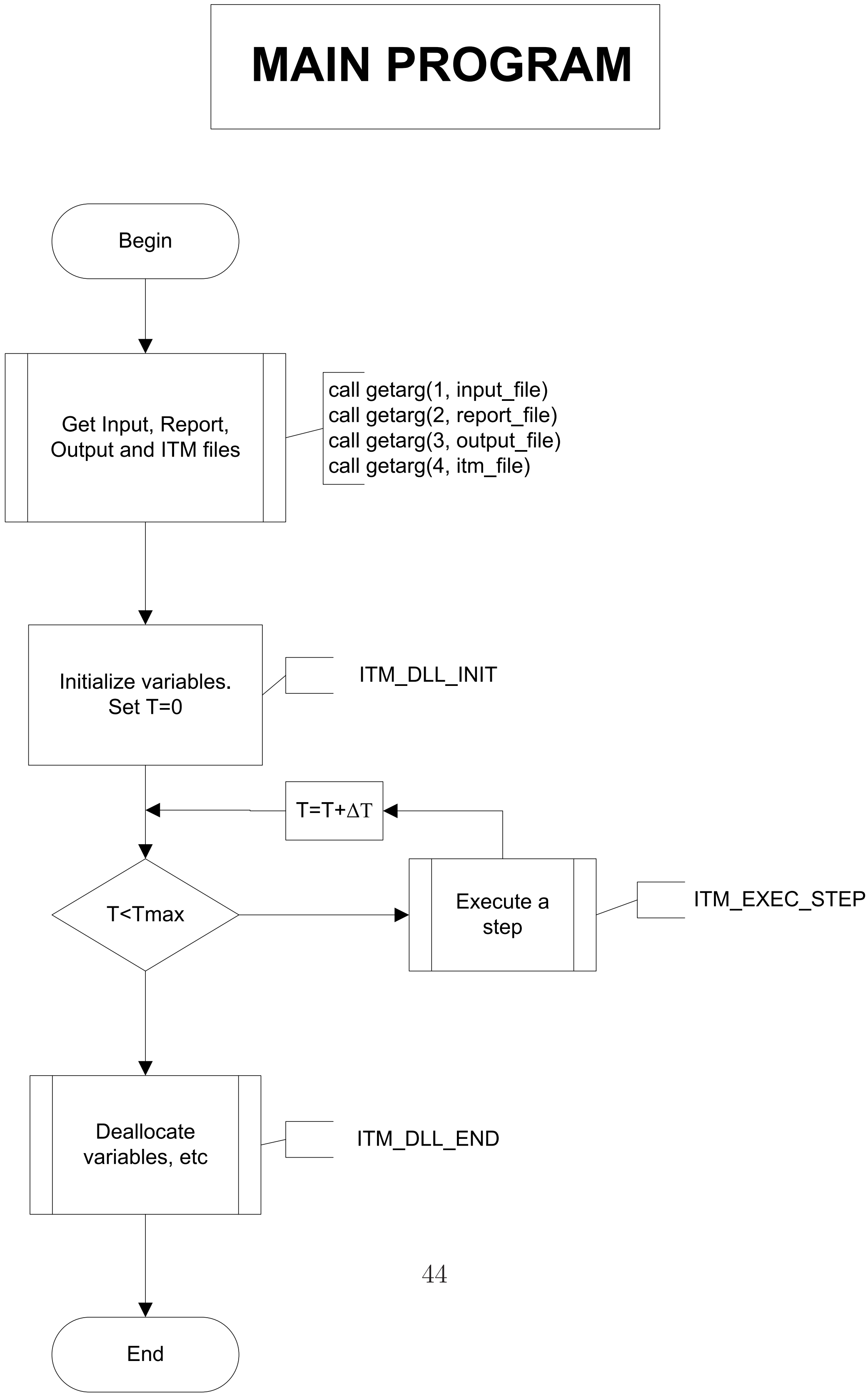


Figure 3.21: Flowchart for routine mainProgram

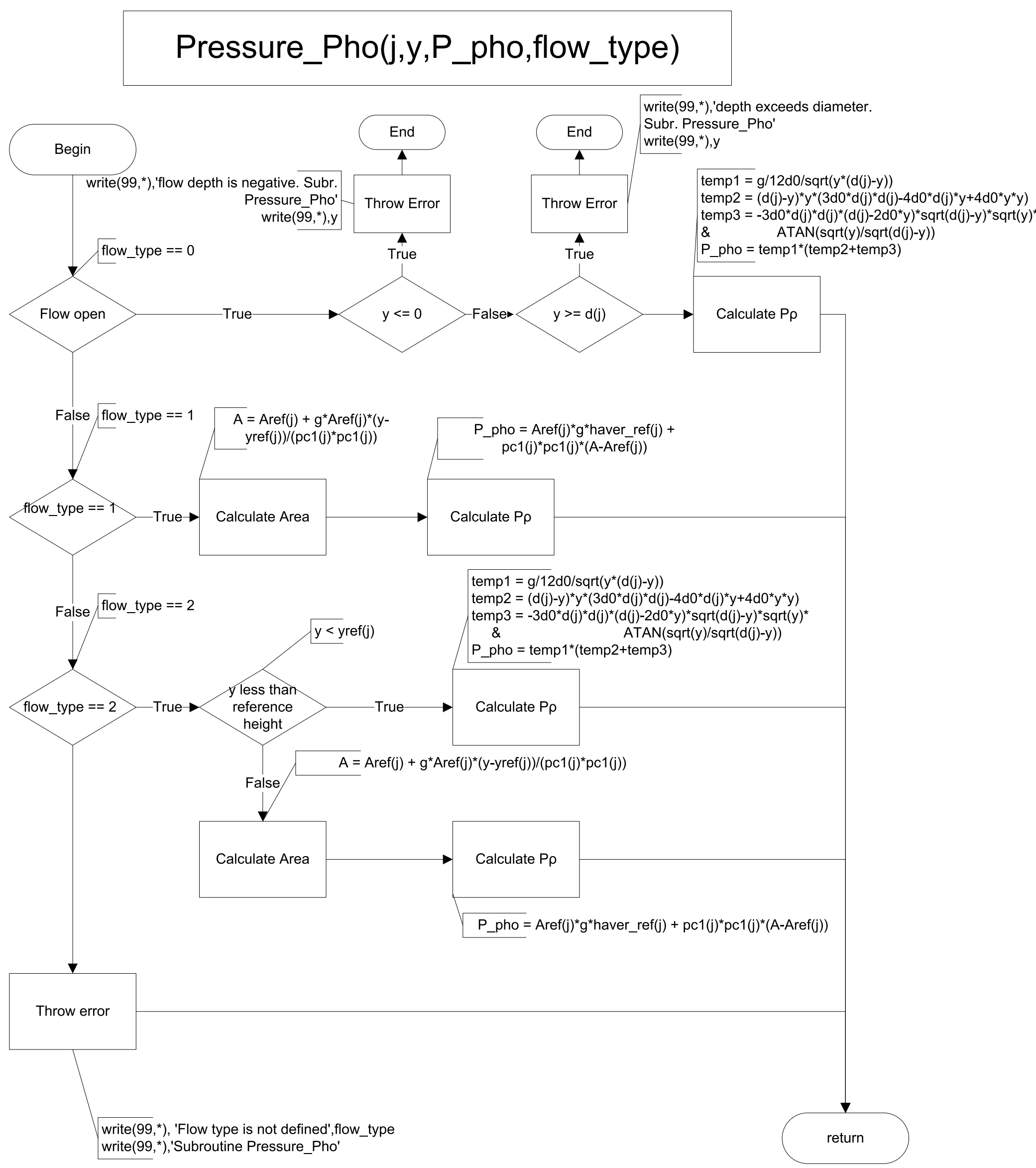


Figure 3.22: Flowchart for routine Pressure_Pho

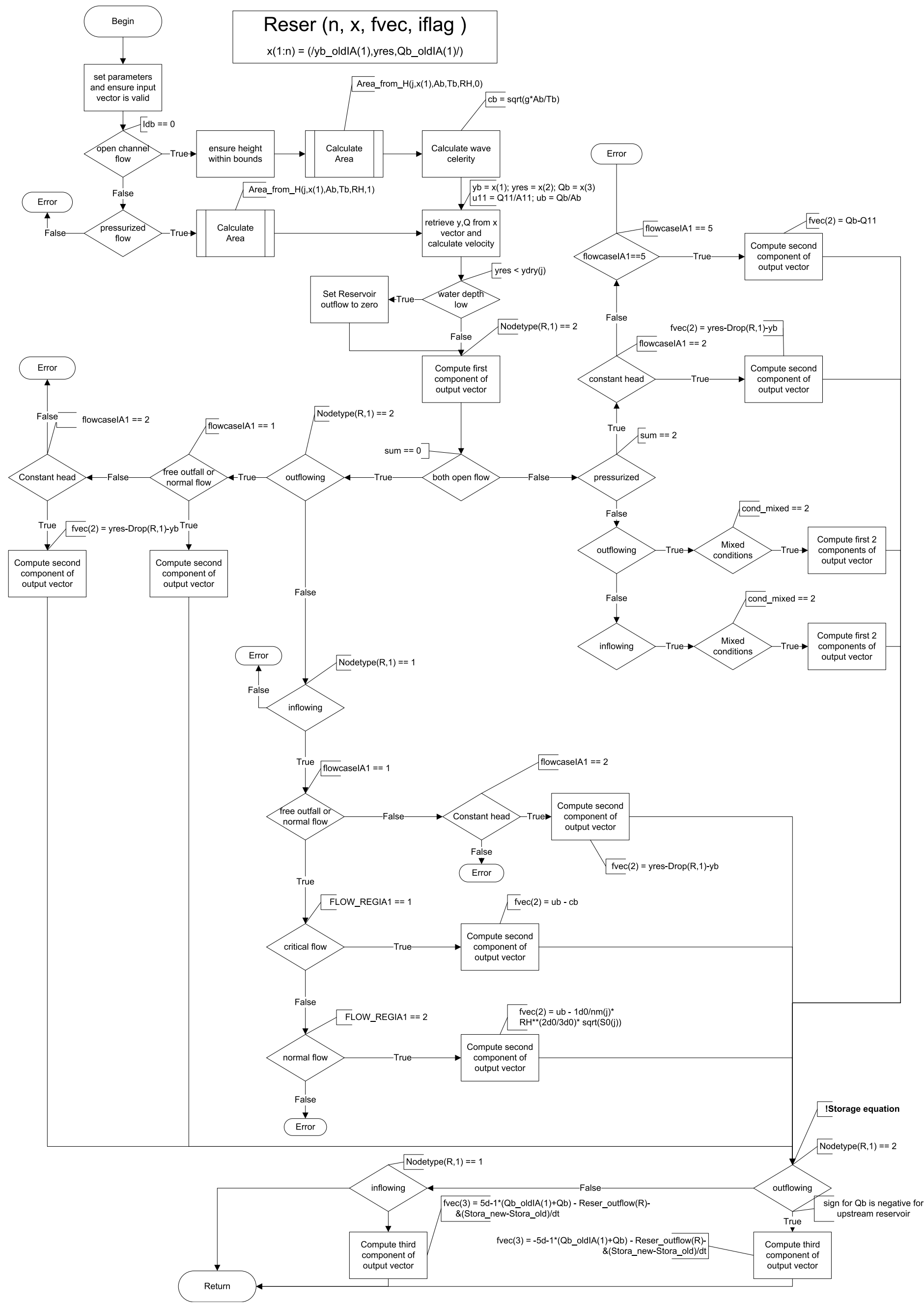


Figure 3.23: Flowchart for routine Reser

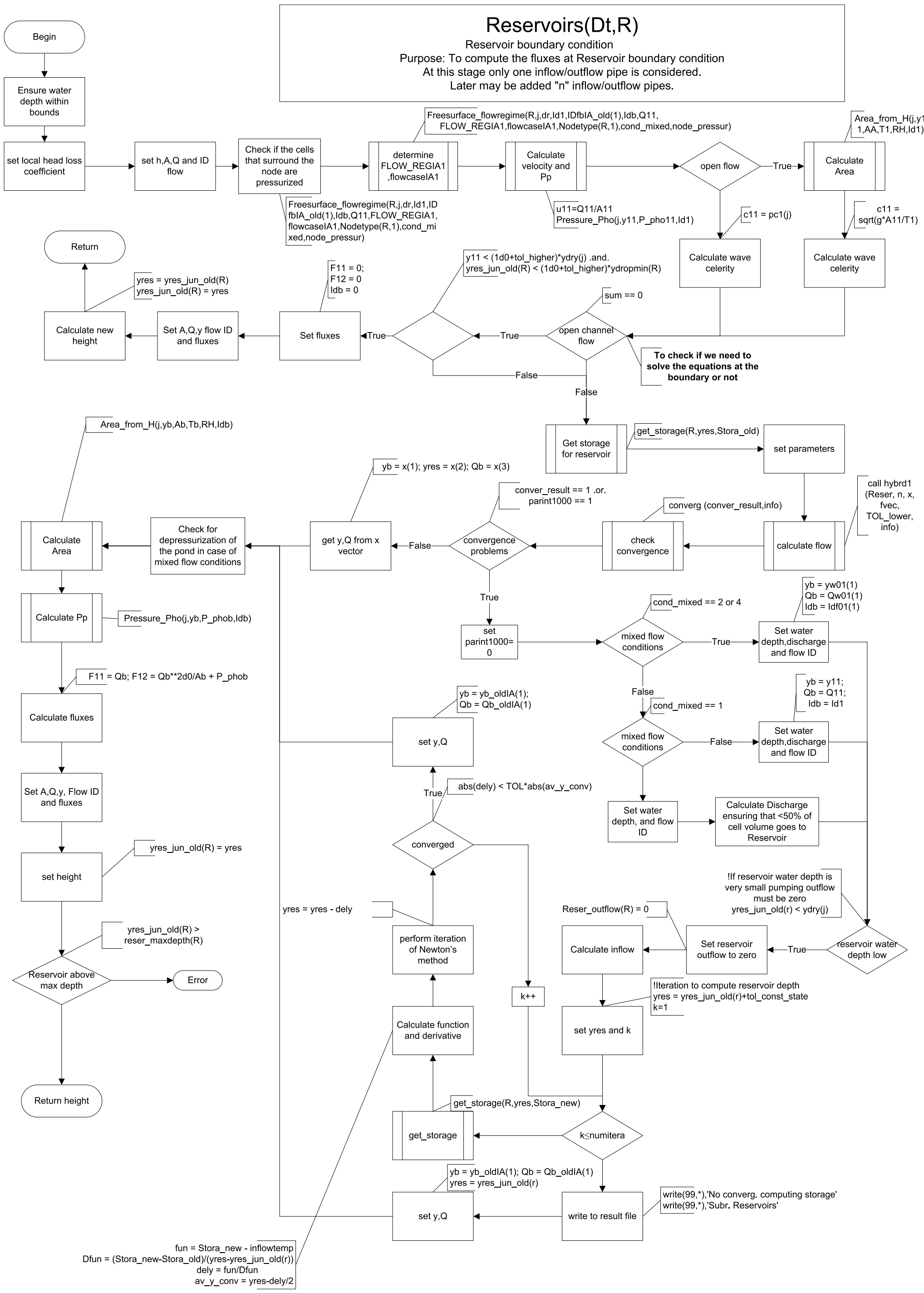


Figure 3.24: Flowchart for routine Reservoirs

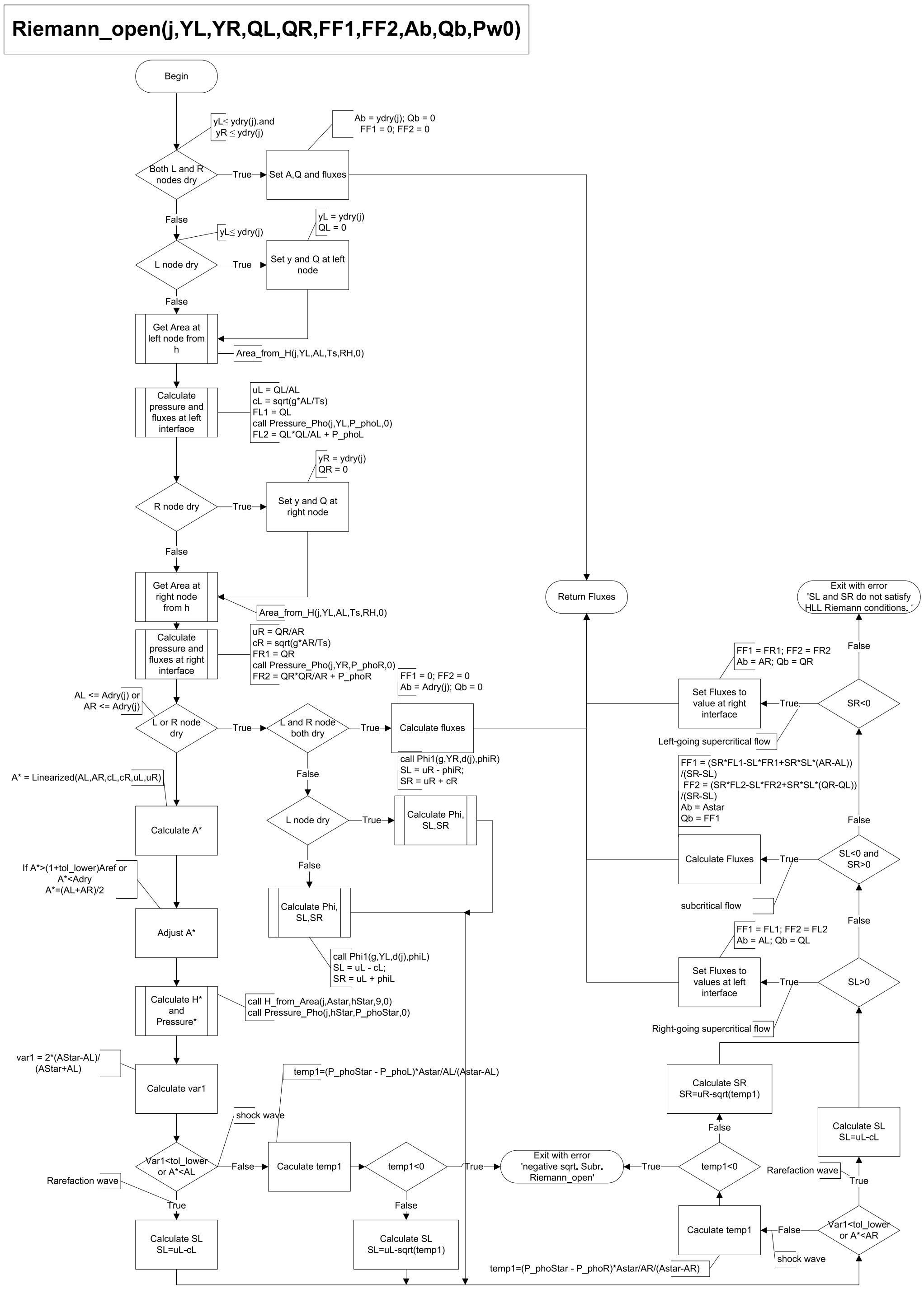


Figure 3.25: Flowchart for routine Riemann_open

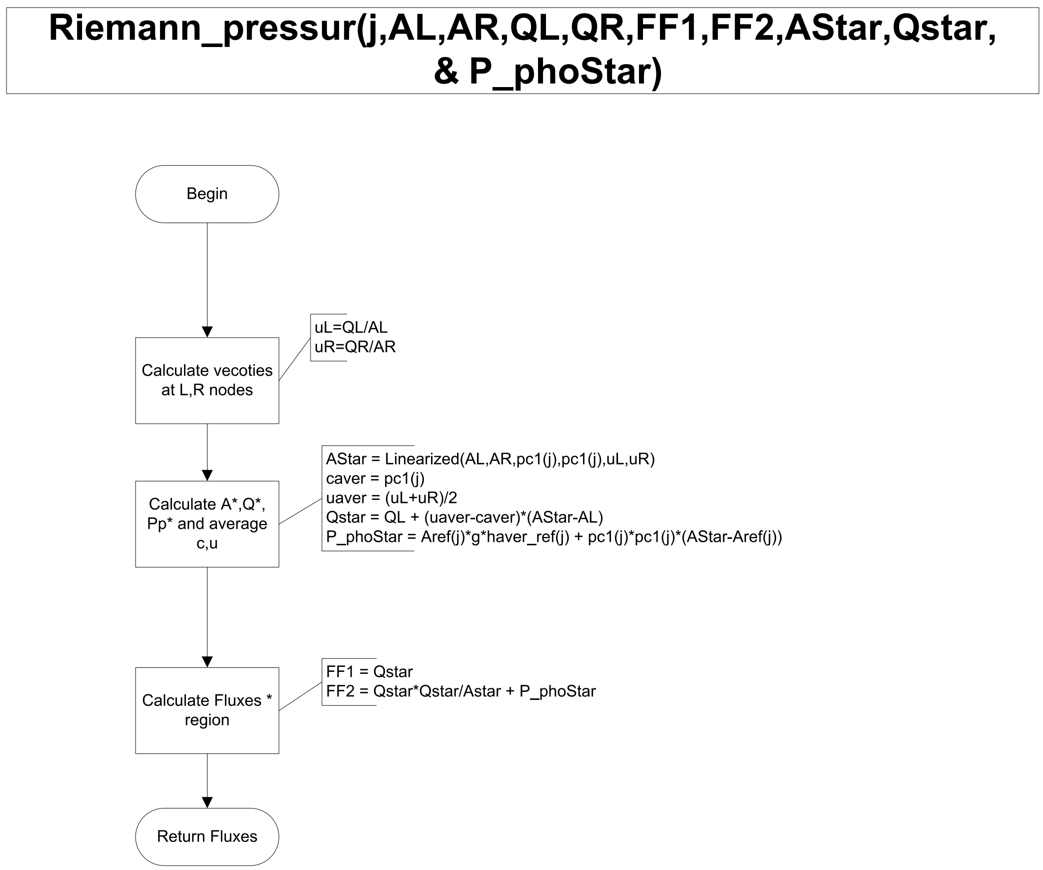


Figure 3.26: Flowchart for routine Riemann_pressure

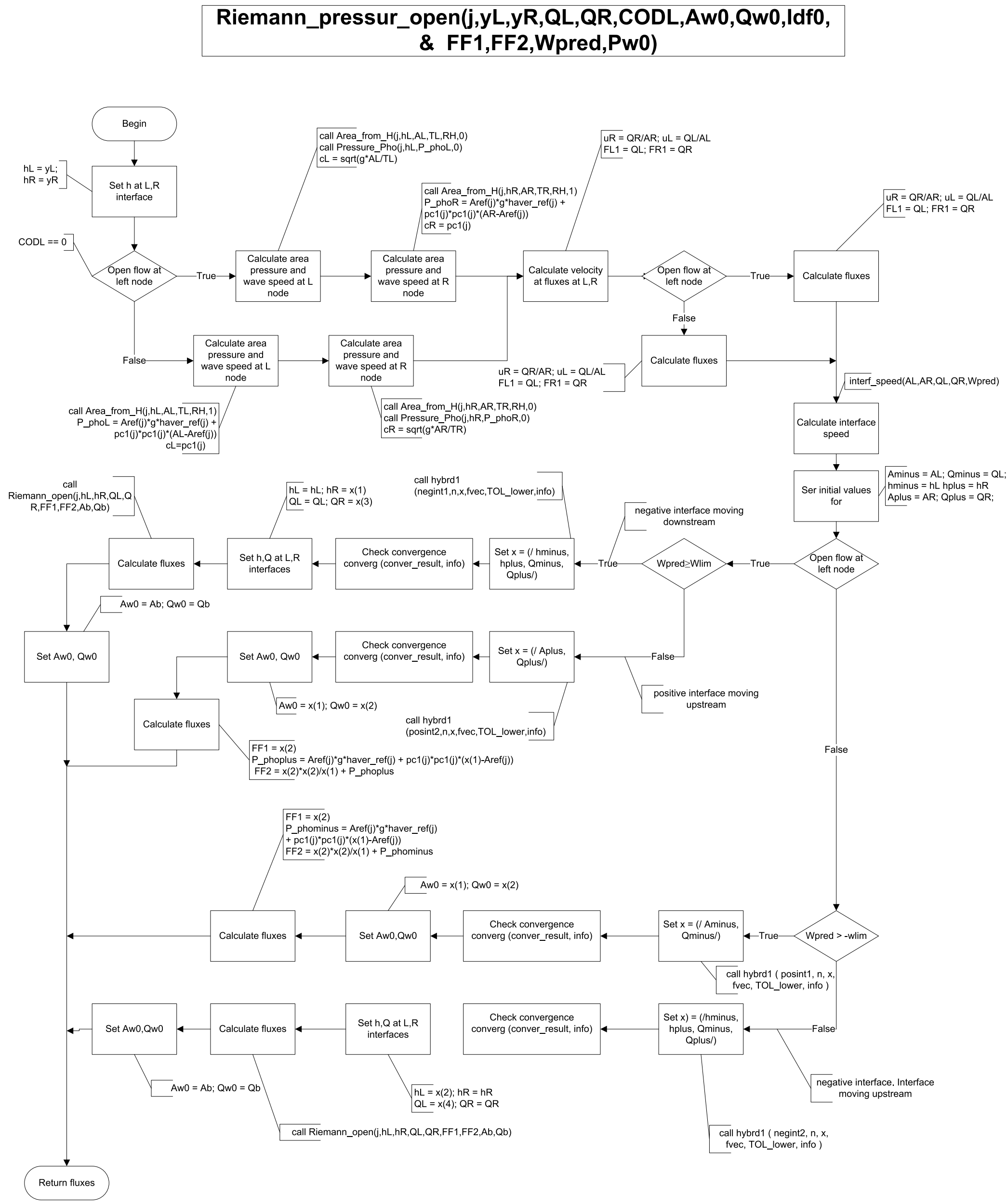


Figure 3.27: Flowchart for routine Riemann_pressure_open

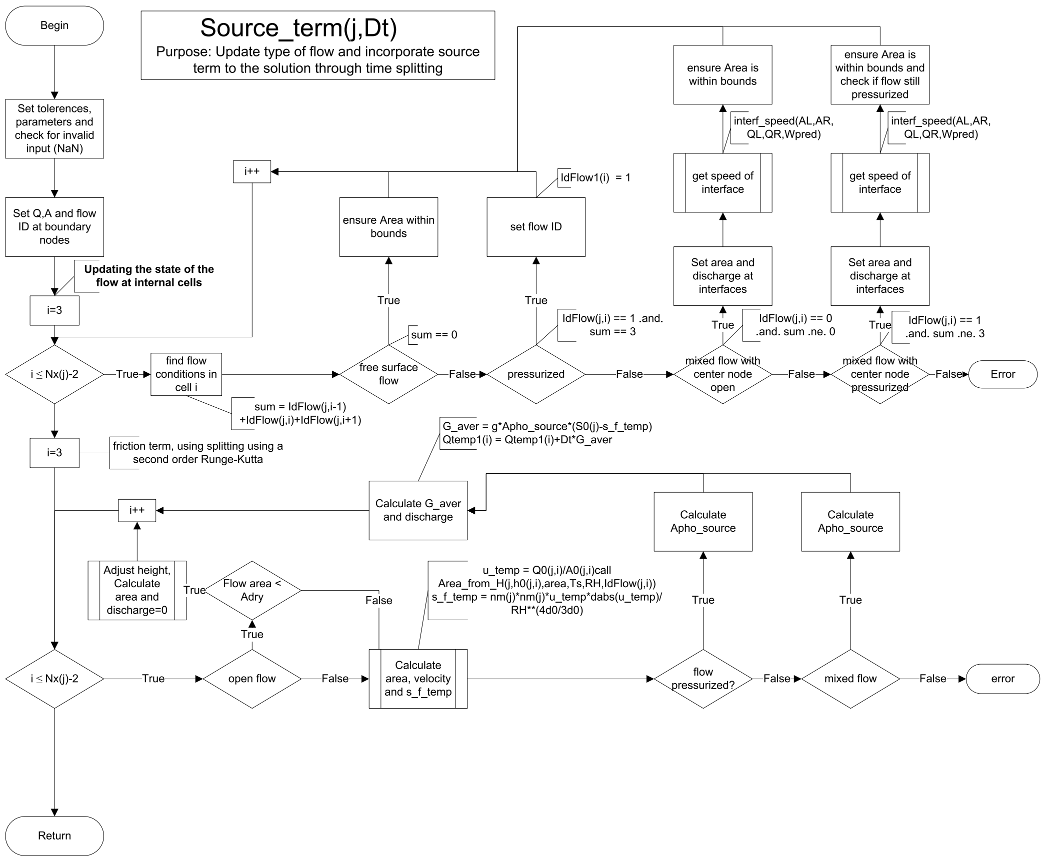


Figure 3.28: Flowchart for routine Source_term

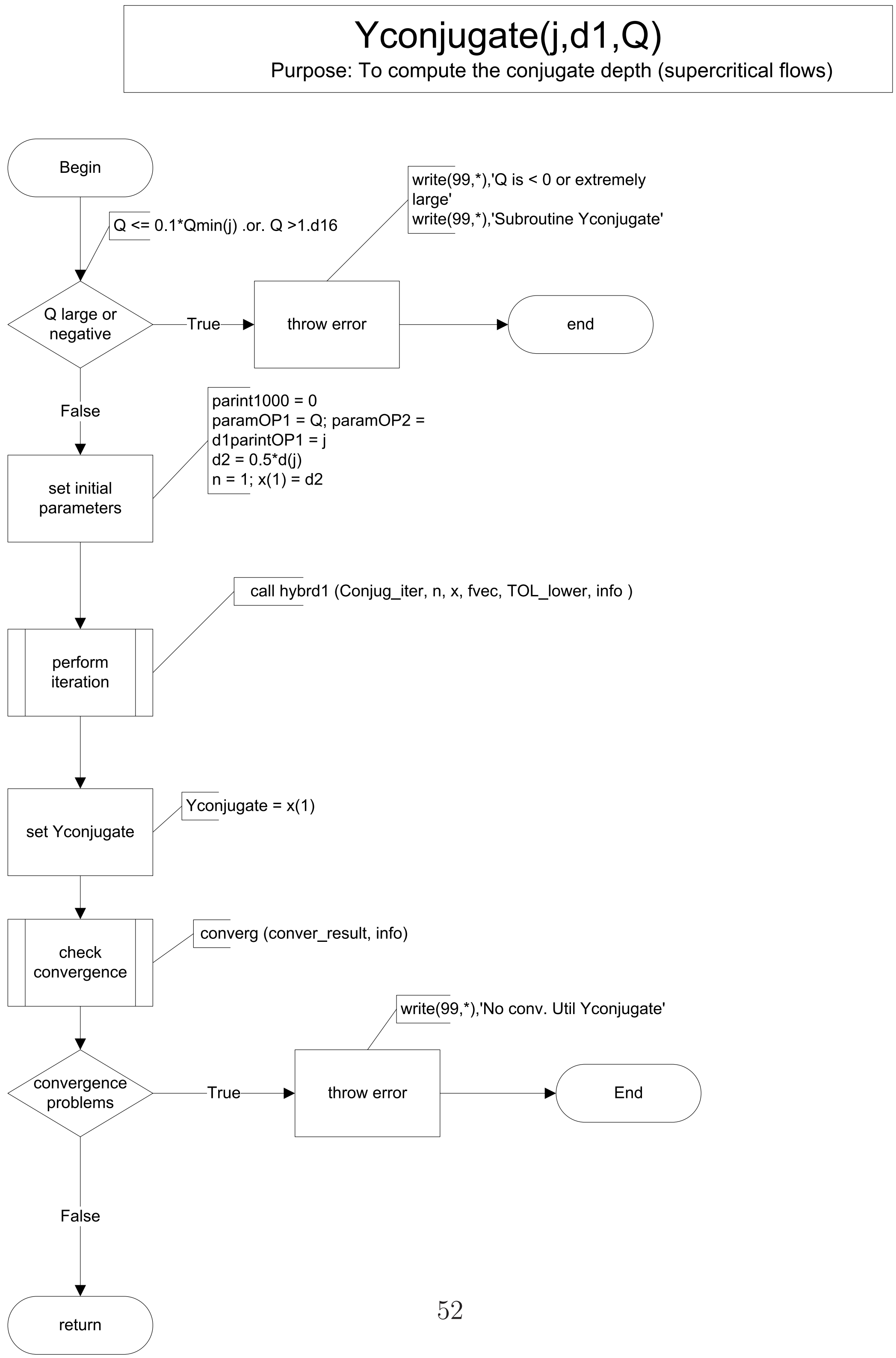




Figure 3.29: Flowchart for routine Yconjugate

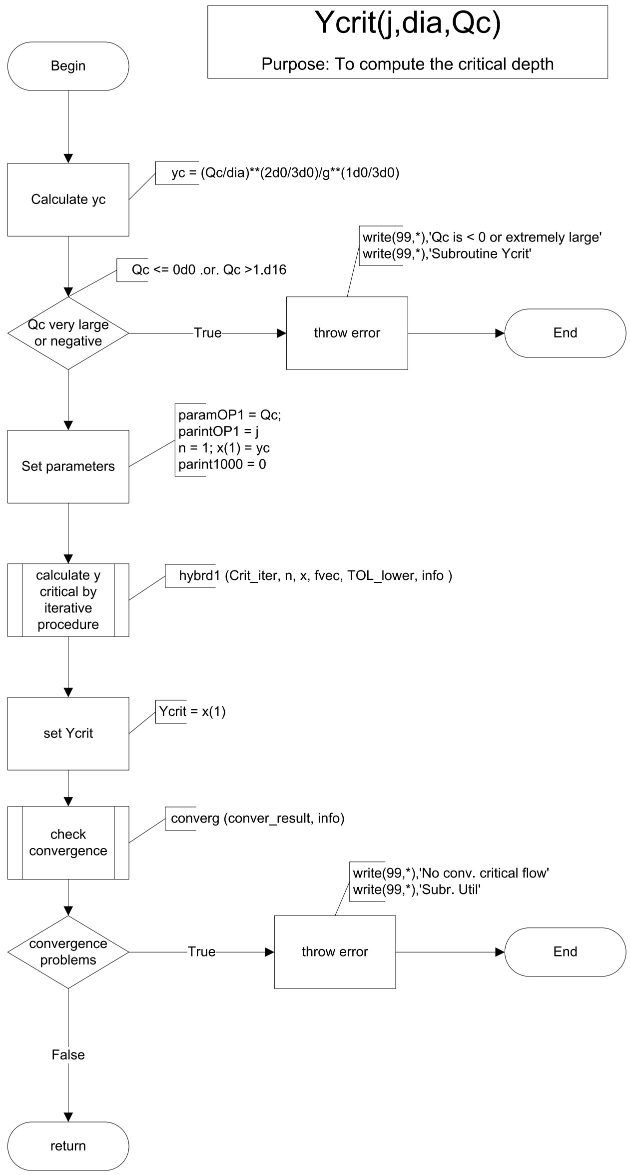


Figure 3.30: Flowchart for routine Ycrit

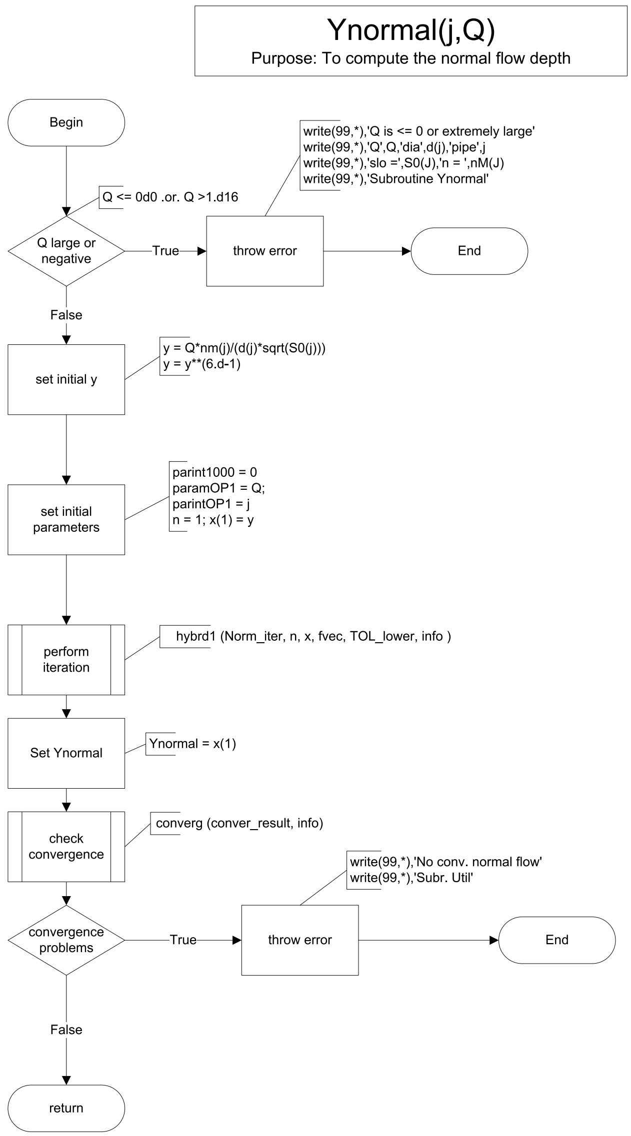


Figure 3.31: Flowchart for routine Ynormal

# Chapter 4

# Steps to Identify a Successful Simulation

The ITM model generates a file that has the same name as that of the input file plus the term *debug*. This file is located in the same folder as that of the input file. The *debug* file contains a history of the simulation that includes all errors generated at all time steps. To asses if the simulation results produced by the ITM model can be trusted few checks are suggested. The first check is to verify that the message “No convergence ...” is not shown continuously for a relatively long time at the same node or pipe location. The ITM model is an explicit approach that updates the flow variables in a cell using the fluxes at the cell interface. If convergence at a given cell interface is not achieved, the flux used at the new time step is the same as that of the previous time step. This approach is reasonable when the non-convergence (at same location) is produced continuously for few time steps. Otherwise the parameters used (e.g., tolerances) in the simulation need to be verified.

A second check that is related with the first one is due to mass conservation. If an incorrect flux was used for a relatively long time, it may produce a large mass conservation error. In this case, a warning showing the error volume appears in the debug file. If the volume error exceeds 10 % of the total volume that entered the system the program is stopped. A mass conservation error may be associated not only with flux errors at cell interfaces but also to the use of small pressure wave celerities in the simulation. Numerically, the elasticity of the pipe or tunnel (to store additional water) is a function of the pressure wave celerity ($a$). A pipe with $a$ equal to 50 m/s is more elastic than a pipe with $a$ equal to 1000 m/s. When using unrealistic pressure wave celerities (low values) it may produce large mass conservation errors.

Another source of errors may be because large tolerances are used in the simulation. In the latter case, even when there are not errors of convergence (because tolerances are large), the results might be unreasonable. For using and modifying the ITM model an advanced knowledge of theory on transient flows (gravity flow, mixed flow and waterhammer) is necessary.

# Acknowledgements

The ITM model would not have been possible without the financial support of the Metropolitan Water Reclamation District of Greater Chicago and of the Civil and Environmental Engineering Department of the University of Illinois at Urbana-Champaign.

# References

León, A.S., Oberg N., Schmidt, A.R., García, M.H. (2009a). User's manual for Illinois Transient Model-two equation model. A model for the analysis of transient free surface, pressurized and mixed flows in storm-sewer systems, University of Illinois at Urbana-Champaign.

León, A.S., Ghidaoui, M.S., Schmidt, A.R., García, M.H. (2009b). A robust two-equation model for transient mixed flows. *J. Hydraul. Research*. Tentatively accepted.