

# Analytical and Numerical Treatment of a Singular Initial Value Problem in Avalanche Modeling<sup>\*</sup>

Othmar Koch, Ewa Weinmüller

*Institute for Applied Mathematics and Numerical Analysis, Vienna University of Technology, Austria*

---

## Abstract

We discuss a leading-edge model used in the computation of the run-out length of dry-flowing avalanches. The model has the form of a singular initial value problem for a scalar ordinary differential equation describing the avalanche dynamics. Existence, uniqueness and smoothness properties of the analytical solution are shown. We also prove the existence of a unique root of the solution. Moreover, we present a FORTRAN 90 code for the numerical computation of the run-out length. The code is based on a solver for singular initial value problems which is an implementation of the acceleration technique known as Iterated Defect Correction based on the implicit Euler method.

*Key words:* Singular initial value problem, existence of solution, numerical solution, implicit Euler method, Iterated Defect Correction, leading-edge model, avalanche run-out

*PACS:* 65L05, 34A12

---

## 1 The model

In this paper we discuss a leading-edge model due to McClung and Hungr for the description of the dynamics of dry-flowing avalanches (see [1] and [2]).

---

<sup>\*</sup> This project was supported in part by the Austrian Science Fund (FWF) grant P-15072-MAT.

*Email addresses:* [othmar@fsmat.at](mailto:othmar@fsmat.at) (Othmar Koch),  
[e.weinmueller@tuwien.ac.at](mailto:e.weinmueller@tuwien.ac.at) (Ewa Weinmüller).

*URLs:* <http://fsmat.at/~othmar> (Othmar Koch),  
<http://math.tuwien.ac.at/~ewa> (Ewa Weinmüller).

The model is a refinement of coarser, lumped-mass models, see for example [3], which often underestimate the run-up or run-out of the avalanche's leading front. Our aim is to provide a theoretical basis for the implementation of a FORTRAN code for the numerical computation of the run-out length of avalanches. The resulting test code is freely available.

In the model, five forces, driving and resistive, are combined to give the total force governing the avalanche's dynamics. The unit of the resulting differential equation is force per unit length, or  $\text{kg/s}^2$ . Subsequently, the five forces are briefly described. Parameters occurring in their definitions are given in Table 1, together with their values used in our numerical experiments<sup>1</sup>. A graphical illustration of the model and some of its parameters is given in Figure 1.

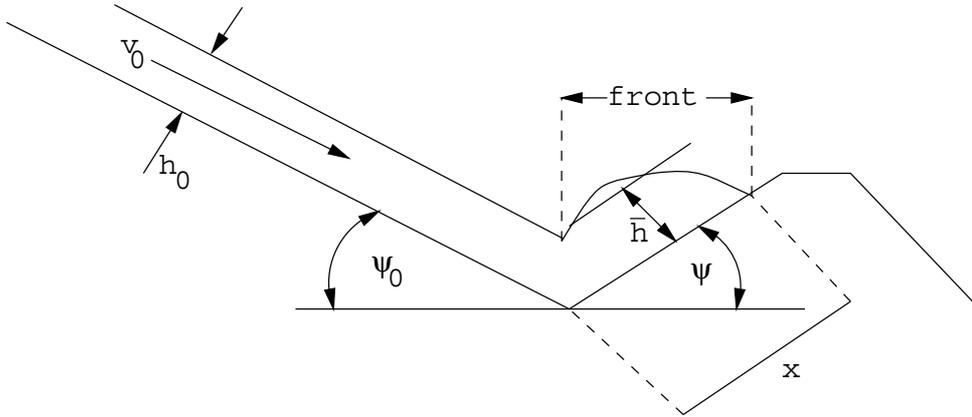


Fig. 1. Leading-edge model

**Driving force**  $T_1 = \bar{\rho} \bar{h} x g \sin \psi$ :

$\bar{\rho} \bar{h} x$  is the mass of the leading front.  $T_1$  stands for the gravity driving force or resisting force, where  $\psi$  is negative if run-up is considered. Note that this force is equal to 0 in the case of flat run-out ( $\psi = 0$ ).

**Momentum flux**  $T_2 = \bar{\rho} h_0 v_0^2 \cos(\psi_0 - \psi)$ :

This is the momentum flux between the main body of the avalanche and the front, calculated as the product of mass and velocity. This force vanishes if the incoming slope and the run-up slope are orthogonal.

**Dynamic Coulomb resistive force**  $T_3 = -\mu \bar{\rho} g \bar{h} x \cos \psi$ :

This is the frictional resisting force at the base of the leading front. The parameter  $\mu$  depends on the roughness of the ground. Naturally,  $T_3 = 0$  if the run-up slope is vertical.

**Turbulent resistive force**  $T_4 = -\frac{1}{2} \rho_t C_D v^2 x$ :

$T_4$  is the turbulent resisting drag force (at the top).

**Passive snow pressure force**  $T_5 = \frac{1}{2} \bar{\rho} g h_0^2 k_p \cos \psi_0 \cos(\psi_0 - \psi)$ :

$T_5$  models the fluid thrust between the main body and the front. Not surprisingly,  $T_5 = 0$  if the incoming slope is vertical and/or the incoming and outgoing slopes are orthogonal.

<sup>1</sup> We do not discuss the issue of parameter estimation here.

$\bar{\rho}$	mean density of the avalanche core	200 kg/m <sup>3</sup>
$g$	gravitation constant	9.81 m/s <sup>2</sup>
$\bar{h}$	mean flow depth along the run-up	3 m
$\psi$	slope angle (run-up)	$-\frac{\pi}{6}$
$h_0$	flow depth at $t = 0$	2 m
$v_0$	incoming speed	20 m/s
$\psi_0$	slope angle (incoming)	$\frac{\pi}{20}$
$\mu$	dynamic coefficient of friction (bottom)	0.155
$\rho_t$	mean density of the snow-dust-air mixture	10 kg/m <sup>3</sup>
$C_D$	drag coefficient	0.01
$k_p$	$k_p = \frac{\cos \psi_0 + \sqrt{\cos^2 \psi_0 - \cos^2 \phi}}{\cos \psi_0 - \sqrt{\cos^2 \psi_0 - \cos^2 \phi}}$ if $\phi \geq \psi_0$ $k_p = 1$ otherwise	$\approx 2.32$
$\phi$	internal friction angle	$\frac{5\pi}{36}$
$G_0$		$\approx 6.22$
$V$		$\approx 16.42$
$D_0$		$\approx 8.33 \cdot 10^{-5}$
$x = x(t)$	distance along the run-up slope	m
$v = v(t)$	avalanche speed	m/s

Table 1  
Parameters

The resulting differential equation is

$$\frac{d(\bar{\rho}\bar{h}vx)}{dt} = T_1 + T_2 + T_3 + T_4 + T_5.$$

Rewriting this relation and taking into account the continuity condition  $\bar{h}x = h_0v_0t$ , we obtain

$$v'(t) = -\frac{v(t)}{t} - D_0v^2(t) + \frac{V}{t} - G_0, \quad t > 0, \quad (1)$$

where

$$G_0 = g(\mu \cos \psi - \sin \psi),$$

$$V = v_0 \cos(\psi_0 - \psi) \left( 1 + \frac{k_p g h_0 \cos \psi_0}{2v_0^2} \right),$$

$$D_0 = \frac{\rho_t C_D}{2\bar{\rho}h}.$$

Our aim is to compute the velocity  $v(t)$  of the avalanche, determine the (unique) positive root of this function and integrate  $v(t)$  up to this point in order to obtain the run-out length of the avalanche.

## 2 Properties of the analytical solution

First, we study the analytical properties of the solution of the differential equation (1) subject to an initial condition posed at  $t = 0$ . A problem of this form is said to have a *singularity of the first kind*. Eqn. (1) can easily be analyzed using techniques developed in [4] for singular problems with a smooth inhomogeneous term in the right-hand side of the differential equation. The results can also be concluded from the more general discussion in [5].

It turns out that any bounded solution of (1) must satisfy the integral equation

$$v(t) = V - \frac{G_0}{2}t - tD_0 \int_0^1 sv^2(st) ds =: (\mathbf{KF}v)(t) + \varphi(t). \quad (2)$$

The operator  $\mathbf{KF}$  can be shown to be a contraction on a suitable interval  $[0, \delta]$ . Thus, a unique continuous solution exists in that interval, which can be extended for  $t > \delta$  due to classical theory. By substitution of (2) into (1) we conclude that this solution is arbitrarily smooth. Moreover, from (2) it follows that the only admissible initial condition for a wellposed problem is<sup>2</sup>  $v(0) = V$ .

As we are interested in computing the first positive root of  $v(t)$  in order to calculate the run-out distance  $X_R$  of the avalanche, we now prove that  $v(t)$  is monotonously decreasing and has a unique root  $t^* > 0$ . To this aim, we note that

$$v'(t) \leq -\frac{v(t)}{t} + \frac{V}{t} - G_0, \quad (3)$$

---

<sup>2</sup> Note that the numerical solution is very robust with respect to perturbations in the initial value. Even though no solution of the analytical problem exists, the numerical scheme has a unique solution which is very close to the solution of the initial value problem with the same parameters and correct initial value, see [6].

and

$$v_L'(t) = -\frac{v_L(t)}{t} + \frac{V}{t} - G_0 \quad (4)$$

is the linear version of (1). The solution of (4) can be computed explicitly,

$$v_L(t) = -\frac{1}{2}G_0t + V, \quad (5)$$

and accordingly the root  $t_L^*$  of  $v_L(t)$  satisfies

$$t_L^* = \frac{2V}{G_0}. \quad (6)$$

Thus, for the run-out distance we obtain

$$\int_0^{\frac{2V}{G_0}} \left(-\frac{1}{2}G_0t + V\right) dt = \frac{V^2}{G_0} =: \tilde{X}_R. \quad (7)$$

From the bound in (3) we conclude that the solution  $v$  of the nonlinear problem is monotonously decreasing and satisfies  $v'(t) \leq -G_0/2$ . Moreover, it has a unique positive root  $t^*$  which is bounded by  $t_L^*$ , and the run-out length computed for the avalanche is smaller than  $\tilde{X}_R$ .

### 3 The numerical method

For the numerical integration of the initial value problem (1) subject to  $v(0) = V$  we use an IVP integrator especially suited for singular problems. This solver is an implementation of *Iterated Defect Correction (IDeC)* based on the implicit Euler method and is described in detail in [7]. This acceleration technique was first proposed and analyzed for regular problems by Frank, [8]. We proceed with a short description of the method.

We assume to know the approximate solution  $v_h^{[0]} := v_h = (v_0, \dots, v_N)$  obtained for (1) by the implicit Euler method on an equidistant grid  $\Delta_h := (t_0, \dots, t_N)$ ,  $t_i = ih$ , and denote by  $p^{[0]}(t)$  the polynomial of degree  $N$  interpolating the values of  $v_h^{[0]}$ . Using this *Zadunaisky polynomial*, we construct a neighboring problem associated with (1) and solved exactly by  $p^{[0]}(t)$ ,

$$z'(t) = -\frac{z(t)}{t} - D_0z^2(t) + \frac{V}{t} - G_0 + d^{[0]}(t), \quad (8)$$

where

$$d^{[0]}(t) := p^{[0]'}(t) + \frac{p^{[0]}(t)}{t} + D_0(p^{[0]}(t))^2 - \frac{V}{t} + G_0.$$

Now we solve (8) by the implicit Euler method and obtain an approximate solution  $p_h^{[0]}$  for  $p^{[0]}(t)$ . This means that for the solution of the neighboring problem (8) we know the global error which we can use to estimate the unknown error of the original problem (1) and use this information to improve the solution<sup>3</sup>,

$$v_h^{[1]} := v_h^{[0]} + \left( R_h \left( p^{[0]} \right) - p_h^{[0]} \right).$$

We use these values to define a new interpolating polynomial  $p^{[1]}(t)$  by requiring  $p^{[1]}(t_j) = v_j^{[1]}$ ,  $j = 0, \dots, N$ . Now  $p^{[1]}(t)$  is used to construct a new neighboring problem analogous to (8), where again the exact solution is known, and the numerical solution of this neighboring problem serves to obtain the second improved solution  $v_h^{[2]} := v_h^{[0]} + \left( R_h \left( p^{[1]} \right) - p_h^{[1]} \right)$ . Clearly, this procedure can be iteratively continued.

For obvious reasons one does not use one interpolating polynomial for the whole interval in practice. Instead, a piecewise polynomial function composed of polynomials of (moderate) degree  $m$  is defined to specify the neighboring problem. There are two different ways in which we can extend the solution from a subinterval where one Zadunaisky polynomial is used to the next interval, see for example [9]. For the *global connection strategy*, we apply the implicit Euler method on the whole integration interval, and interpolate the values with a continuous, piecewise function  $p^{[0]}(t)$ . Continuing in this way, we successively obtain continuous piecewise polynomials  $p^{[j]}(t)$  defined on the whole interval. Alternatively, in the *local connection strategy*, we compute the IDeC iterates on one subinterval, and subsequently restart the iteration on the next interval using the most accurate approximation computed previously for the initial value.

An analysis of the numerical properties of the global connection strategy for singular problems with a smooth inhomogeneous term is given in [10]. It turns out that the analysis carries over to (1) if we can show that a bounded solution of the implicit Euler method exists. This follows from an estimate of the numerical solution on an interval  $[0, \delta]$ ,

$$|v_i| = \left| \prod_{l=1}^i \left( 1 + \frac{h}{t_l} \right)^{-1} v(0) + \sum_{l=1}^i \prod_{k=l}^i \left( 1 + \frac{h}{t_k} \right)^{-1} h t_l^{-1} \left( V - t_l D_0 v_l^2 - t_l G_0 \right) \right|$$

<sup>3</sup>  $\overline{R_h(g)} := (g(t_0), \dots, g(t_N))$  for a continuous function  $g$ .

$$\begin{aligned} &\leq \left| \prod_{l=1}^i \left(1 + \frac{h}{t_l}\right)^{-1} v(0) + \sum_{l=1}^i \prod_{k=l}^i \left(1 + \frac{h}{t_k}\right)^{-1} ht_l^{-1} (V - t_l G_0) \right| \\ &\leq \text{const}(V + \delta G_0), \quad i = 1, \dots, N, \end{aligned}$$

which uses a representation of the solution derived in [11]. We thus conclude that the following theorem holds:

**Theorem 1** *Consider the global connection strategy of the IDeC method based on the implicit Euler rule and on piecewise interpolation with polynomials of degree  $m$  for the numerical solution of problem (1). For the approximations obtained in the course of the iteration,*

$$\|v_h^{[j]} - R_h(v)\|_h := \max_{0 \leq l \leq N} |v_l^{[j]} - v(t_l)| = O(h^{j+1}) \quad (9)$$

*holds for  $j = 0, \dots, m - 1$ . Further iteration does not increase the asymptotic order of the approximation in general.*

Note that this result also means that the correction used in the last step of the iteration is an asymptotically correct estimate for the global error of the one-but-last iterate  $v_h^{[m-2]}$ . Accordingly, this quantity can also be used as a reasonable estimate for the global error of the final solution.

Unfortunately, for the local connection strategy we observe an order reduction down to  $O(h^2)$  in general. Apparently, this variant suffers from similar drawbacks as many common high-order one-step methods for singular problems (see for example [12]), because restarting the procedure on every subinterval amounts to the same as applying a one-step method of order  $m$ , see [9].

We choose a method with  $m = 4$  in the implementation of our FORTRAN 90 code for the solution of the problem of determining the run-out length of the avalanche. In Tables 2 and 3 we give numerical evidence of the asymptotic properties of IDeC discussed above. All test runs in this paper were performed on a Silicon Graphics Power Challenge XL R10000 with operating system IRIX V.6.5. The FORTRAN 90 programs were compiled with the MIPS PRO FORTRAN 90 COMPILER 7.2.1. The tests were run in IEEE double precision with  $\text{EPS} \approx 1.11 \cdot 10^{-16}$ .

To determine the empirical convergence orders of our numerical methods, we solve (1) on the interval  $[0, 6]$ , and compute the errors of the numerical approximations obtained for step-sizes  $h = 2^{-i}$ ,  $i = 1, 2, \dots$  with respect to a reference solution where  $h = 2^{-14} \approx 6.10 \cdot 10^{-5}$  was used. The values of  $V$  and  $G_0$  were chosen as in Table 1. We used  $D_0 = 0.065$ , however, because the order results are more meaningful when the nonlinear term in (1) is not negligible. Otherwise, the global error is almost entirely due to roundoff error and no

systematic results are observed. The following tables give the empirical convergence rates  $p$  and error constants  $c$  computed from the global errors “error” for every step-size  $h$ . Moreover, the error estimates “estimate” obtained from the IDeC iteration are shown. Note that these systematically overestimate the true errors, thus providing very reliable, though pessimistic information.

Table 2 confirms the results of Theorem 1, the final IDeC iterate shows an empirical convergence order of 4. In Table 3, an order reduction down to order 2 is observed for the local connection strategy.

$h$	$p$	$c$	error	estimate
0.50000E+00	4.56	0.21E-01	0.89047E-03	0.46120E-02
0.25000E+00	4.29	0.17E-01	0.45300E-04	0.44726E-03
0.12500E+00	4.14	0.14E-01	0.25521E-05	0.49353E-04
0.62500E-01	4.07	0.12E-01	0.15150E-06	0.58005E-05
0.31250E-01	4.03	0.10E-01	0.92304E-08	0.70321E-06
0.15625E-01	4.01	0.10E-01	0.56989E-09	0.86570E-07
0.78125E-02	3.99	0.94E-02	0.35692E-10	0.10739E-07

Table 2

Convergence of the global connection strategy ( $D_0 = 0.065$ )

$h$	$p$	$c$	error	estimate
0.50000E+00	3.32	0.64E-02	0.64389E-03	0.39904E-02
0.25000E+00	1.82	0.22E-02	0.18225E-03	0.14202E-02
0.12500E+00	1.89	0.25E-02	0.48978E-04	0.37032E-03
0.62500E-01	1.95	0.28E-02	0.12621E-04	0.86696E-04
0.31250E-01	1.96	0.29E-02	0.32405E-05	0.20653E-04
0.15625E-01	1.97	0.30E-02	0.82336E-06	0.50424E-05
0.78125E-02	1.98	0.31E-02	0.20802E-06	0.12464E-05

Table 3

Convergence of the local connection strategy ( $D_0 = 0.065$ )

In spite of the drawbacks of the local connection strategy, we have to rely on this version of IDeC to obtain a first approximation for the location of the root  $t^*$  of  $v(t)$ . As the integration interval is not known a priori, we cannot use the global connection strategy for this task. After determining this interval, we can restart the algorithm with the global connection strategy to obtain a better numerical approximation for  $t^*$ . Our procedure allows to omit this step, however, if computation time is critical and reduced accuracy is tolerated. The rerun yields a more accurate solution and a reliable error estimate.

After the computation of numerical approximations for  $v(t)$  and  $t^*$ , we determine the avalanche’s run-out length by numerical quadrature. In this process, we use Newton-Cotes quadrature rules of orders 4 and 6 to determine an approximation for  $X_R$  and an estimate for the error introduced by numerical quadrature. Finally, the estimates for the errors of the IDeC method and the numerical quadrature are combined to yield an error estimate for the numerical approximation of  $X_R$ .

More details on the above numerical procedure can be found in [6].

#### 4 Test results

Finally, we give the results of a test run of our code for the computation of the run-out of avalanches according to the model (1). We used the choice of parameters given in Table 1. We do not attempt to study the issues of parameter estimation and model validation here. Rather, we want to demonstrate that our discussion of the analytical and numerical properties of (1) yields a firm basis for our code, which can efficiently be used to investigate the issues relevant in applications.

result	program results	
	global connection strategy	local connection strategy
$t^*$	5.27379405258795	5.27379405258823
error of $t^*$	6.684849889 E-14	2.163802616E-10
$X_R$	43.25747367208126	43.25747367210393
error of $X_R$	2.646977960 E-22	0
global error	1.095998293 E-12	6.727368662 E-10

Table 4  
Output from test run

Table 4 gives the numerical approximations for  $t^*$  and  $X_R$ , computed using either the local connection strategy only or additionally the global connection strategy of IDeC. Moreover, estimates for the errors introduced in the computation of  $t^*$ , the numerical quadrature and the combined effects for the error in  $X_R$  are given. Note that the error estimates for  $t^*$  and the quadrature are usually not returned by our routine if only the local connection strategy is used, because these may be unreliable due to the order reduction observed in that case. For the computation of the numerical solution, we use an equidistant step-size  $h = 2^{-7}$ . It may be interesting to compare the results with the theoretical bounds from §2,  $t^* \leq 5.27696133$ ,  $X_R \leq 43.3138030$ . Obviously,

the influence of the nonlinear term is only weak because  $D_0$  is chosen very small.

## 5 The FORTRAN code

The FORTRAN 90 files of our code for the computation of the run-out length of avalanches are freely available from either

<http://fsmat.at/~othmar/software.html>

or

<http://math.tuwien.ac.at/~ewa>

These also include the IVP integrator. A technical description of our code is given in [6]. The IDeC implementation is discussed in [7]. Note that we use a slightly adjusted version of this solver. Technical details about this procedure can be found in the documentation included in the source code. Since the code consists of several modules, we add a short description of the file organization. The main program is `avalanche.f`. The second file is called `parameters.f` and contains the model parameters. This is the only file that should be modified by the user. The third part is an interface between the main part and the subroutines performing the different parts of the calculation. This file is called `work.f`. The IDeC routine is called `ieulidec.f90`. This module requires the additional modules `types.f90`, `dynvect.f90` and `linalg.f90`. Make sure to use the right sequence for compilation, for example

```
f90 types.f90 dynvect.f90 linalg.f90 ieulidec.f90
  parameters.f work.f avalanche.f
```

The files made available at the moment are a test version which may still undergo some revision. We would be grateful for any suggestions about possible improvements in the usability and reliability of the code.

## Acknowledgements

We thank DI Sigrid Wieshofer and Dr. Karl Kleemayr from the Vienna University of Agriculture (BoKu) for bringing this problem to our attention. Moreover, we owe the implementation of the FORTRAN 90 code for the avalanche problem to DI Alexander Paul and DI Andreas Traxler. The IVP integrator that was used is due to Dr. Peter Kofler.

## References

- [1] D. M. McClung, O. Hungr, An equation for calculating snow avalanche run-up against barriers, in: *Avalanche Formation, Movement and Effects*, IAHS Publ. no. 162, 1987, pp. 605–612.
- [2] D. M. McClung, A. I. Mears, Dry-flowing avalanche run-up and run-out, *J. Glaciol.* 41 (138) (1995) 359–369.
- [3] A. Voellmy, Über die Zerstörungskraft von Lawinen, *Schweiz. Bauztg.* 73 (1955) 159–165, 212–217, 246–249, 280–283.
- [4] O. Koch, P. Kofler, E. Weinmüller, Initial value problems for systems of ordinary first and second order differential equations with a singularity of the first kind, *Analysis* 21 (2001) 373–389.
- [5] F. d. Hoog, R. Weiss, On the boundary value problem for systems of ordinary differential equations with a singularity of the second kind, *SIAM J. Math. Anal.* 11 (1980) 41–60.
- [6] O. Koch, A. Paul, A. Traxler, E. Weinmüller, Efficient numerical solution of a singular initial value problem in avalanche modeling, *Techn. Rep. ANUM Preprint Nr. 11/02*, Inst. for Appl. Math. and Numer. Anal., Vienna Univ. of Technology, Austria, available at <http://www.math.tuwien.ac.at/~inst115/preprints.htm> (2002).
- [7] W. Auzinger, O. Koch, P. Kofler, E. Weinmüller, The application of shooting to singular boundary value problems, *Techn. Rep. Nr. 126/99*, Inst. for Appl. Math. and Numer. Anal., Vienna Univ. of Technology, Austria, available at <http://fsmat.at/~othmar/research.html> (1999).
- [8] R. Frank, The method of Iterated Defect Correction and its application to two-point boundary value problems, Part I, *Numer. Math.* 25 (1976) 409–419.
- [9] R. Frank, C. Überhuber, Iterated Defect Correction for the efficient solution of stiff systems of ordinary differential equations, *BIT* 17 (1977) 146–159.
- [10] O. Koch, E. Weinmüller, Iterated Defect Correction for the solution of singular initial value problems, *SIAM J. Numer. Anal.* 38 (6) (2001) 1784–1799.
- [11] O. Koch, P. Kofler, E. Weinmüller, The implicit Euler method for the numerical solution of singular initial value problems, *Appl. Num. Math.* 34 (2000) 231–252.
- [12] F. d. Hoog, R. Weiss, The application of Runge-Kutta schemes to singular initial value problems, *Math. Comp.* 44 (1985) 93–103.