Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN

Geofisica

Ciclo 28

Settore Concorsuale di afferenza: 04/A4

Settore Scientifico disciplinare: GEO/10

TITOLO TESI

Numerical Modelling of Point-Masses Sliding on 2D Complex Surfaces as a Means to Investigate Rockslide Dynamics

Presentata da: Katharina Maria Elsen

Coordinatore Dottorato

Prof. Nadia Pinardi

Relatore

Prof. Stefano Tinti

Esame finale anno 2017

University of Bologna

Abstract

Department of Physics and Astronomy

Numerical Modelling of Point-Masses Sliding on 2D Complex Surfaces as a Means to Investigate Rockslide Dynamics

by Katharina Maria ELSEN

The present work introduces a new Lagrangian method to solve the equations of motion which model gravitational sliding of point masses on piecewise linear curves and piecewise planar surfaces. This method shall be incorporated in the already existing UBO-block model which can be used to model rockslide dynamics.

In the first case, the equations are solved for a curve, which is approximated by means of a continuous piecewise defined curve Σ formed by straight segments. In this special case, the solution can be computed analytically, dividing it into parts, one being the motion along the single segments and another being the effect of the transition between two segments.

This is possible as the equations consist of two types of terms: those containing gravity acceleration g but not containing derivatives of second order, and those that do contain second order derivatives, but not g. While the first terms determine the motion along the piecewise linear/planar segments, the second ones are sufficient to describe the transition between the elements.

In a second step, the method was generalized to the 2-dimensional case for which a semi-analytical solution to the equations was derived. While the transition between two planar elements can be computed exactly, the motion along planar segments generally must be computed numerically.

The method was implemented in a modular way in modern Fortran making use of typical HPC optimization methods. The goodness of the method is discussed by comparing the solutions to the ones computed for smooth curves and surfaces using the LSODE-solver for the time-integration. It is concluded that the new method significantly reduces the computational costs while reaching fully comparable results. The dependency of the solution on friction, grid resolution and initial position as well as its sensitivity to grid regularity were investigated using Monte-Carlo simulations.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

There are several problems in which gravitational sliding plays a role, one of which is the modelling of landslides. This is an issue that has received considerable attention in the last decades where several numerical models have been devised and implemented, since landslides are more and more considered one of the most diffuse and dangerous geohazards in the world. Several classifications of landslides have been given up to now, like those presented in Varnes, 1978, Cruden and Varnes, 1996 (modifications of the 1978 classification) or Hungr et al., 2001. These classifications contain for example the type of movement (the most important ones being slides, flows and falls), the involved material (e.g. earth, debris or rock), or the movement velocity, where the latter is of great importance for the hazard evaluation and depends on the type of landslide (debris avalanches can reach velocities of 100 m/s while some translational slides are as slow as 0.3 m in 5 years) (c.f. Highland and Bobrowsky, 2008). In figure 1.1 a classification of landslides, published by the U.S. Geological Survey (c.f. USGS, 2016), is given.

Depending on the type, different models are applied to model landslide. These contain models based on fluid-flow dynamics treating homogeneous fluids like those presented in Bouchut, 2004; Lucas et al., 2011 or two-phase, multiphase fluids such as in Abadie et al., 2010; Pudasaini, 2012; Bouchut et al., 2014 with equations solved through conventional (finite difference, finite volume, finite element) or unconventional (like smooth particle hydrodynamics, SPH) models (see e.g. Pastor et al., 2007; Pirulli and Sorbino, 2008; Hungr and McDougall, 2009) that are more suitable to treat the evolution of granular flows, debris flows and submarine turbid currents.

There are alternative numerical models based i) on the description of the landslide mass as an ensemble of very many interacting particles such as the distinct element method DEM, first applied to granular flows by Cundall and Struck Cundall and Strack, 1979 and later considerably enhanced and empowered (see Cheng et al., 2003; O'Sullivan, 2011) or ii) on the representation of the landslide body as a set of interacting blocks of irregular shape (UBO-block model, see Bortolucci, 2001; Tinti, Bortolucci, and Vannini, 1997; Tinti, Bortolucci, and Armigliato, 1999) that can be better used to handle more solid mass movements, such as rockslides and slumps.

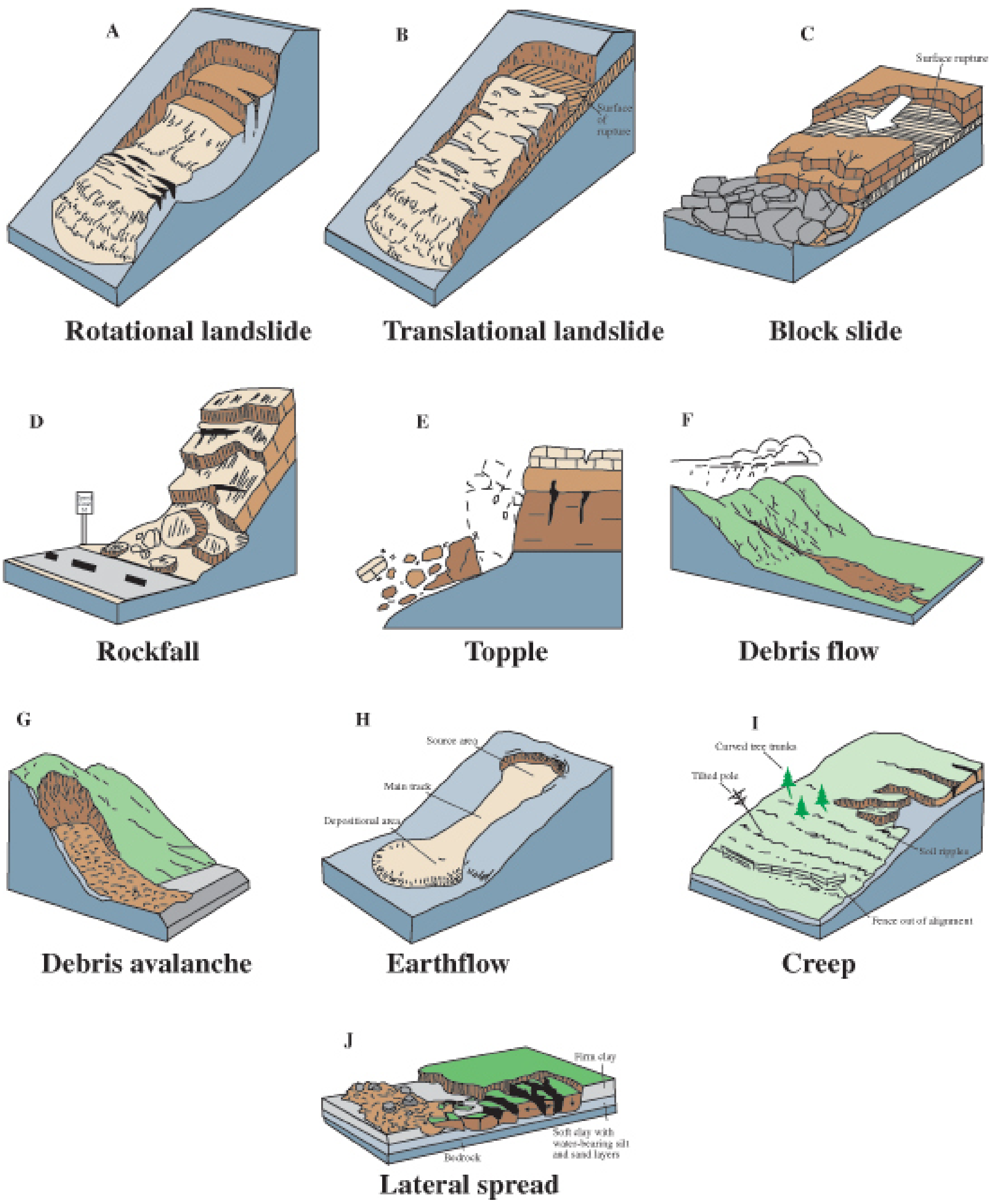


FIGURE 1.1: Classification of landslides published by the U.S. Geological Survey (c.f. USGS, 2016)

In this latter approach the blocks can become deformed (stretched, compressed) up to a given extent but they always keep their volume. Gravitational sliding strongly depends on the underlying topography, and therefore the way the sliding surface is treated in the models is an essential aspect of the landslide simulation codes. Generally speaking, the terrain is usually provided as a digital data set of points of which one knows the position and the altitude and can be represented basically in one of the following two ways: i) by using smooth functions to compute a higher order reconstruction of the real terrain from the given topography dataset or ii) by approximating the topography by a triangulation where the data-points are the vertices of the triangles. The main difference between the two options is that in the former method the resulting surface is smooth, usually admitting first and second derivatives, while triangulated surfaces are non-smooth at the passages between triangles and, if required by the model, impose specific techniques to calculate curvature-related quantities.

The foundation pillars of all landslide models are:

- the representation of the moving mass itself;
- the surface on which the mass is moving;
- the equations governing the motion of the mass;
- the numerical method used for solving the equations of motion.

The aim of the thesis is to derive and test a method for the modelling of point masses sliding on 2-dimensional complex surfaces surfaces that can be incorporated in the existing UBO-block model which was developed at the University of Bologna over the past two decades. The method must – in absence of friction – conserve the energy. This will be done for smooth surfaces as well as for piecewise planar surfaces. Moreover, an approximation to the curvature dependent friction term for non-smooth functions shall be found. In many landslide models this term is not implemented due to the lack of second order partial derivatives of the triangulated surface.

In the following chapter a short overview on selected landslide models and a more detailed description of the UBO-block model are given. Furthermore, the equations describing the gravitational sliding of a point mass on a surface under the influence of friction, are derived. The remaining thesis is organized as follows: in chapter 3, different methods for the reconstruction of smooth functions from given data sets are presented and tested. In addition, the Livermore solver for ODEs (LSODE), a solver that is suitable also for stiff ODEs is introduced and explained as a means to solve the equations of motion on smooth surfaces.

Chapter 4 concentrates on the solution of the equations in the 1-dimensional case. A new, fully analytical method for solving the equations of motion on piecewise linear curves is derived. The results obtained in different test cases are compared to the solutions computed on smooth functions using a specialized solver for stiff problems. Moreover, a detailed analysis of the new method is done, including different convergence tests.

In chapter 5 the 1-dimensional method is generalized to 2 dimensions. The results obtained by solving the equations of motion for a point moving on piecewise planar surfaces are compared to the solutions obtained on smooth surfaces. Again, the new method is analysed in terms of convergence and consistency.

Further tests are performed in chapter 6 using so called Monte Carlo simulations. In particular the sensitivity of the solution on grid regularity, but also its dependency on friction, grid resolution and initial position of the mass point will be investigated.

Chapter 7 describes how the newly developed method can be integrated in the UBO-block model. Finally a summary of the results and an outlook on possible future work are given in chapter 8.

The method was implemented in a modular way in modern Fortran, for the 1-dimensional as well as for the 2-dimensional case. The code was optimized using typical HPC (high performance computing) techniques and parallelized for the Monte-Carlo simulations.

Chapter 2

Framework

2.1 Overview on selected landslide models

In the following, two landslide models are presented more in detail to give an overview on already existing models. The first model is a not only suitable to model landslides but can be used for different gravity driven shallow water purposes. Depending on the kind of topography, two different versions of the model are available, one for small curvatures and another one for more general topographies. The second model is a very modern and complete, multi-dimensional two-phase model that was developed to suit in particular the simulation of different types of debris flows.

2.1.1 Gravity driven shallow water models for arbitrary topography

Throughout the work presented in Bouchut, 2004, two multi-dimensional models for gravity driven shallow water flows in several space dimensions over a general topography, are derived. One of the two models is valid for small slope variations (in the sense of small curvature) while a second model is valid for arbitrary topographies. The authors are particularly interested in the influence of the topography in the flow equations. For both models a Coulomb friction term can be added.

For modelling flows in one dimensional the classical Saint-Venant system is widely used. It is derived from the free surface incompressible Navier-Stokes equations in the regime of small slopes. A model that is suited to handle more general slopes, but is still 1-dimensional, is the Savage-Hutter model. It is valid in the in the regime of small slope variations and frequently used for the modelling of debris avalanches. However, significantly less models are developed for the multidimensional case.

The authors provide general equations that extend the Savage-Hutter theory to several space dimensions and, moreover, give a generalization for a model without small curvature assumption. Their models provide the following features from the incompressible Navier-Stokes equations with free surface Bouchut, 2004:

- The systems admit a conservative entropy equation, thus ensuring hyperbolicity (important to describe shock formation and for numerical stability),
- the models preserve the steady state of a lake at rest,
- the models are invariant under rotation,
- the models imply intrinsically the transportation of vorticity by the flow.

2.1.2 A general two-phase debris flow model

This model presented by Pudasaini, 2012 includes many physical aspects, among them enhanced viscous stress, virtual mass and generalized draw, making it one of the most generalized two-phase flow models. While the fluid phase of the model is characterized by its density ρ_f, the viscosity η_f, and isotropic stress distribution, the solid phase is characterized by its density ρ_s, internal and basal friction angles ϕ and δ, and anisotropic stress distribution, K (lateral earth pressure coefficient). In combination with relative motion between phases, these characteristics lead to two different mass and momentum balance equations for the solid and fluid phases.

In the following, $\mathbf{u}_f$ and $\mathbf{u}_s$ denote the velocities of the fluid and solid phase while α_s and $\alpha_f(=1-\alpha_a)$ are their volume fractions. Furthermore, the following assumptions are made: surface tension is negligible, interfacial solid and fluid pressures are identical to the fluid pressure, the solid and fluid components are incompressible, no phase change occurs.

The mass balance equations are:

$$\begin{aligned}\frac{\partial \alpha_s}{\partial t}+\nabla\cdot(\alpha_s\mathbf{u}_s)=0,\\ \frac{\partial \alpha_f}{\partial t}+\nabla\cdot(\alpha_f\mathbf{u}_f)=0,\end{aligned}$$

and the momentum equations (written in conservative form) are:

$$\begin{aligned}\frac{\partial}{\partial t}(\alpha_s\rho_s\mathbf{u}_s)+\nabla\cdot(\alpha_s\rho_s\mathbf{u}_s\otimes\mathbf{u}_s) &= \alpha_s\rho_s\mathbf{f}-\nabla\cdot\alpha_s\mathbf{T}_s+p\nabla\alpha_s+\mathbf{M}_s,\\ \frac{\partial}{\partial t}(\alpha_f\rho_s\mathbf{u}_f)+\nabla\cdot(\alpha_f\rho_f\mathbf{u}_f\otimes\mathbf{u}_f) &= \alpha_f\rho_f\mathbf{f}-\alpha_f\nabla p+\nabla\cdot\alpha_f\boldsymbol{\tau}_f+\mathbf{M}_f,\end{aligned}$$

where $\mathbf{f}$ is the body force density, $-\mathbf{T}_s$ is the negative Cauchy stress tensor, $\boldsymbol{\tau}_f$ is the extra stress for fluid, $\mathbf{M}$ is the internal force density (with $\mathbf{M}_s+\mathbf{M}_f=0$), $p\nabla\alpha_s$ accounts for the bouyant force and p the fluid pressure. The model contains constitutive equations for $\mathbf{M}_s$, $\mathbf{T}_s$ and τ_f. In addition to the 3-dimensional model, a 2-dimensional model using depth-averaging in z direction was derived in order to make the problem more tractable.

Numerical results indicated that the model is well suited to describe subaerial two-phase debris flows, particle-laden and dispersive flows, sediment transport or submarine debris flows.

2.2 The UBO-block landslide model: description of the general idea

In contrast to the models described above, the UBO-block landslide-model, presented for example in Bortolucci, 2001, Tinti, Bortolucci, and Vannini, 1997 or Tinti, Bortolucci, and Armigliato, 1999, is suitable for the modelling more solid mass movements, such as rockslides and slumps. It is based on a discretization of the moving mass using a number of 3-dimensional, contiguous blocks with quadrilateral base A and a given volume V that has to be conserved. The base of the blocks can become distorted up to a given level and as a consequence also the height of the block h is changing. Each block is represented by its barycentre $\mathbf{x}$ and four vertices $\mathbf{x}'$ where neighbouring blocks always share two vertices. The acceleration for each block is computed in its barycentre from which the barycentre-velocity $\mathbf{v}$ is obtained. The vertex-velocities $\mathbf{v}'$ in turn are interpolated from those barycentre-velocities. The motion of each block is obtained only by means of vertex-motion. These vertices move independently of each other such that throughout the motion of the mass, the blocks can – up to a given level of distortion – change their shape, but they cannot separate. Rolling and jumping of blocks is not supported either. As a consequence, during a single time-integration step, the motion of only a single vertex at a time has to be computed.

The sliding surface (on which the points motion is performed) is represented by planar triangles while for the time-integration an explicit Euler method is used. When moving from one triangle (of the surface) to another, the velocity of the point is corrected based on the inclination of the current and the subsequent surface element. The basic idea of the algorithm is described in Alg. 1. Let therefore N_k be the total number of blocks and N_j be the total number of vertices. The subscript k denotes a certain block and barycentre while the subscript j denotes a certain vertex. A certain time-step is denoted by an additional subscript i, respectively $i-1$ for the previous time-step. Accordingly, $\mathbf{v}_{i,k}$ denotes the velocity of the barycentre of block number k at time-step i and $\mathbf{v}'_{i,j}$ the velocity of vertex j at the same time-step. The barycentre and vertex positions are as well denoted by $\mathbf{x}_{i,k}$ and $\mathbf{x}'_{i,j}$, respectively. The acceleration of a certain block due to gravity (including friction) for a time-step i is denoted by $\mathbf{G}_{i,k}$ and acceleration due to block-block interaction by $\mathbf{F}_{i,k}$. The total acceleration is given by $\mathbf{a}_{i,k} = \mathbf{G}_{i,k} + \mathbf{F}_{i,k}$. The time-step itself is denoted by $\triangle t$.

Algorithm 1 UBO-block Algorithm

1: **for** $time = 1 : T$ **do**
2: **for** $block = 1 : N_k$ **do**
3: Compute accelerations $\mathbf{G}_{i-1,k}$
4: Compute pre-interaction barycentre velocity $\mathbf{u}_{i,k} = \mathbf{v}_{i-1,k} + \mathbf{G}_{i-1,k}\triangle t$
5: Compute post-interaction barycentre velocity $\mathbf{v}_{i,k} = \mathbf{u}_{i,k} + \mathbf{F}_{i-1,k}\triangle t$
6: From the barycentre velocities $\mathbf{v}_{i,k}$ interpolate vertex velocities $\mathbf{v}'_{i,j}$
7: Compute the new vertex positions $\mathbf{x}'_{i,j} = \mathbf{x}'_{i-1,j} + \frac{1}{2}(\mathbf{v}'_{i,j} + \mathbf{v}'_{i-1,j})\triangle t$ {For a better approximation of the transition point, the time-step $\triangle t$ is split into sub-steps; each time a vertex leaves a triangle, the direction of the velocity vector is adjusted to the slope of the new triangle}
8: **end for**
9: **end for**

2.3 The UBO-block landslide model: Equation of motion

The equation of motion of a material point P moving on a surface $\mathcal{S}$ can be obtained starting from the Newton law $m\mathbf{a} = \mathbf{F}_{tot}$, where m represents the mass of the point, $\mathbf{a}$ is its acceleration, and $\mathbf{F}_{tot}$ is the sum of all the forces acting on the point, which can be of two different kinds: active and reaction forces. The active forces, denoted in the following by $\mathbf{F}$, are known and they depend, in the most general case, on the position $\mathbf{x}$ of the point, on its velocity $\mathbf{v}$ and possibly of the time t. The reaction forces, denoted by $\boldsymbol{\Phi}$ are *a priori* unknown. Separating active and reaction forces, the Newton law reads as follows:

$$m\mathbf{a} = \mathbf{F} + \boldsymbol{\Phi}. \tag{2.1}$$

Under the assumption that the surface $\mathcal{S}$ is an *ideal* constraint, the reaction force $\boldsymbol{\Phi}$ is orthogonal to the surface. Denoting with $\mathbf{n}$ the unit vector normal to the surface and with $\mathbf{t}$ the unit vector associated to the projection of $\mathbf{F}$ on the plane Π tangent to the surface in the point P, it is easily seen that the three vectors appearing in Eq. (2.1) belong to the plane identified by the orthogonal unit vectors $\mathbf{t}$ and $\mathbf{n}$. Eq. (2.1) can therefore be projected along the directions $\mathbf{t}$ and $\mathbf{n}$ providing the following two scalar equations:

$$\begin{aligned} ma_t &= F_t, \\ ma_n &= F_n + \Phi_n, \end{aligned} \tag{2.2}$$

where

$$\begin{aligned} a_t &= \mathbf{a}\cdot\mathbf{t}, & a_n &= \mathbf{a}\cdot\mathbf{n}, & (\mathbf{a} &= a_n\mathbf{n}+a_t\mathbf{t}),\\ F_t &= \mathbf{F}\cdot\mathbf{t}, & F_n &= \mathbf{F}\cdot\mathbf{n}, & (\mathbf{F} &= F_t\mathbf{t}+F_n\mathbf{n}), \end{aligned}$$

and

$$\Phi_n = \boldsymbol{\Phi}\cdot\mathbf{n} \qquad (\boldsymbol{\Phi} = \Phi_n\mathbf{n})\,.$$

Being $\mathbf{F}\cdot\mathbf{t}$ the projection of $\mathbf{F}$ on the plane Π, and being $\mathbf{n}\perp\Pi$, the vector $\mathbf{F}-(\mathbf{F}\cdot\mathbf{t})\,\mathbf{t}$ is by construction parallel to $\mathbf{n}$. This leads to the following two vector relations

$$\mathbf{t}\cdot\mathbf{n} = 0, \qquad \mathbf{F}-(\mathbf{F}\cdot\mathbf{t})\,\mathbf{t} = \lambda\mathbf{n} \qquad (\lambda\in\mathbb{R})\,, \tag{2.3}$$

that uniquely identify[1] the unit vector $\mathbf{t}$.

Once the unit vector $\mathbf{t}$ is obtained, Eq. (2.2)$_1$ is an equation for the tangential component a_t of the acceleration $\mathbf{a}$. The normal component a_n can be obtained considering that the velocity $\mathbf{v}$ of the point is orthogonal to the unit vector $\mathbf{n}$ and then proceeding as follows:

$$\begin{aligned} \mathbf{v}\cdot\mathbf{n} = 0 \quad &\Rightarrow \quad \frac{d}{dt}(\mathbf{v}\cdot\mathbf{n}) = 0\\ &\Rightarrow \quad \mathbf{v}\cdot\frac{d\mathbf{n}}{dt}+\mathbf{n}\cdot\frac{d\mathbf{v}}{dt} = 0\\ &\Rightarrow \quad a_n = -\mathbf{v}\cdot\frac{d\mathbf{n}}{dt} = -\mathbf{v}\cdot(\nabla\mathbf{n}\,\mathbf{v})\,. \end{aligned}$$

The equation of motion Eq. (2.1) can now be cast in the following *pure* form (i.e. in which the unknown force $\boldsymbol{\Phi}$ does not appear):

$$\mathbf{a} = \frac{1}{m}(\mathbf{F}\cdot\mathbf{t})\,\mathbf{t}-\mathbf{v}\cdot(\nabla\mathbf{n}\,\mathbf{v})\,\mathbf{n}. \tag{2.4}$$

Incidentally, we note that once the motion has been determined, the reaction force can be obtained by means of Eq. (2.2)$_2$:

$$\Phi_n = ma_n - F_n = -m\mathbf{v}\cdot(\nabla\mathbf{n}\mathbf{v})-\mathbf{F}\cdot\mathbf{n}.$$

The explicit determination of the reaction force is useful when friction has to be included in the equation of motion. Friction, in fact, can be modelled by means of a force $\mathbf{F}_\mu$ parallel to the velocity vector $\mathbf{v}$, pointing in opposite direction, and

[1] When Eq. (2.3)$_{1,2}$ are projected along the axis of a reference frame, they provide three scalar equations for the two unknown components of $\mathbf{t}$ and the coefficient λ. More specifically, the first relation provides one scalar equation; the second relation, being all the involved vectors co-planar, provides two scalar equations.

with magnitude proportional to the magnitude of the reaction force, $|\mathbf{\Phi}|$, being the proportionality coefficient μ known as *friction coefficient*:

$$\mathbf{F}_\mu = -\mu\,|\mathbf{\Phi}|\,\frac{\mathbf{v}}{|\mathbf{v}|} = -\mu\,|-m\mathbf{v}\cdot(\nabla\mathbf{n}\mathbf{v}) - \mathbf{F}\cdot\mathbf{n}|\,\frac{\mathbf{v}}{|\mathbf{v}|}.$$

When friction is included in the list of forces acting on the material point, the equation of motion Eq. (2.1) becomes:

$$m\mathbf{a} = \mathbf{F} + |\mathbf{\Phi}| + \mathbf{F}_\mu,$$

and the pure equation Eq. (2.4) becomes:

$$\mathbf{a} = \frac{1}{m}(\mathbf{F}\cdot\mathbf{t})\,\mathbf{t} - \mathbf{v}\cdot(\nabla\mathbf{n}\,\mathbf{v})\,\mathbf{n} - \mu\left|\left(-\mathbf{v}\cdot(\nabla\mathbf{n}\mathbf{v}) - \frac{1}{m}\mathbf{F}\cdot\mathbf{n}\right)\right|\frac{\mathbf{v}}{|\mathbf{v}|}. \tag{2.5}$$

Note that the equations of motion derived so far, Eq. (2.4) and Eq. (2.5), have been written in a form independent from the reference frame, since no assumption has been made on the observer.

2.3.1 Motion of a material point moving on a surface subject to gravitational force

We shall now write Eq. (2.5) assuming that:

- the reference frame $Oxyz$ from which the motion is observed has the z-axis parallel to the vertical direction and pointing upwards (we denote with $\mathbf{i}_1$, $\mathbf{i}_2$ and $\mathbf{i}_3$ the unit vectors of the axis x, y and z, respectively);
- the only active force acting on the material point P (of mass m) is the gravitational force:

  $$\mathbf{F} = -mg\mathbf{i}_3,$$

  being g the gravity acceleration;
- the point is constrained to a surface described by the implicit equation $\varphi(x,y,z) = 0$, or by the explicit equation $z = f(x,y)$, being $\varphi(x,y,z) = z - f(x,y)$, where $f_{x/y}$, $f_{xx/xy/yy}$ and $\varphi_{x/y}$, $\varphi_{xx/xy/yy}$ are the first and second order partial derivatives of f and φ, respectively;
- friction cannot be neglected and the friction coefficient μ is constant.

With the above assumptions, the unit normal vector to the surface is given by[2]

$$\mathbf{n} = \frac{\nabla\varphi}{|\nabla\varphi|} = \frac{1}{\sqrt{\varphi_x^2+\varphi_y^2+\varphi_z^2}}\left(\varphi_x\mathbf{i}_1+\varphi_y\mathbf{i}_2+\varphi_z\mathbf{i}_3\right).$$

Letting α, β, γ be the components of the vector $\mathbf{t}$ in the chosen reference frame ($\mathbf{t} = \alpha\mathbf{i}_1+\beta\mathbf{i}_2+\gamma\mathbf{i}_3$), we find:

$$\mathbf{t}\cdot\mathbf{n} = 0 \qquad \Rightarrow \qquad \varphi_x\alpha+\varphi_y\beta+\varphi_z\gamma = 0,$$

$$\begin{gathered}(-mg\mathbf{i}_3\cdot\mathbf{t})\,\mathbf{t}+\lambda\mathbf{n} = -mg\mathbf{i}_3 \qquad \Rightarrow \\ -mg\gamma(\alpha\mathbf{i}_1+\beta\mathbf{i}_2+\gamma\mathbf{i}_3)+\frac{\lambda}{|\nabla\varphi|}\left(\varphi_x\mathbf{i}_1+\varphi_y\mathbf{i}_2+\varphi_z\mathbf{i}_3\right) = -mg\mathbf{i}_3.\end{gathered}$$

Solving the system, it is easily seen that:

$$\lambda = -mg\frac{\varphi_z}{|\nabla\varphi|},$$

$$\alpha\gamma = -\frac{\varphi_x\varphi_z}{|\nabla\varphi|^2}, \qquad \beta\gamma = -\frac{\varphi_y\varphi_z}{|\nabla\varphi|^2}, \qquad \gamma^2 = \frac{\varphi_x^2+\varphi_y^2}{|\nabla\varphi|^2},$$

and then:

$$\begin{aligned}\frac{1}{m}(\mathbf{F}\cdot\mathbf{t})\,\mathbf{t} &= -\frac{1}{m}\lambda\mathbf{n}-g\mathbf{i}_3 \\ &= -\alpha\gamma g\mathbf{i}_1-\beta\gamma g\mathbf{i}_2-\gamma^2 g\mathbf{i}_3 \\ &= \frac{\varphi_x\varphi_z}{|\nabla\varphi|^2}g\mathbf{i}_1+\frac{\varphi_y\varphi_z}{|\nabla\varphi|^2}g\mathbf{i}_2-\frac{\varphi_x^2+\varphi_y^2}{|\nabla\varphi|^2}g\mathbf{i}_3.\end{aligned} \tag{2.6}$$

Finally, the components of the term $\mathbf{v}\cdot(\nabla\mathbf{n}\,\mathbf{v})$ are calculated as follows:

$$\begin{aligned}\mathbf{v}\cdot(\nabla\mathbf{n}\,\mathbf{v}) &= \begin{pmatrix} v_x & v_y & v_z\end{pmatrix}\cdot\frac{1}{|\nabla\varphi|}\begin{pmatrix}\varphi_{xx} & \varphi_{xy} & \varphi_{xz}\\ \varphi_{xy} & \varphi_{yy} & \varphi_{yz}\\ \varphi_{xz} & \varphi_{yz} & \varphi_{zz}\end{pmatrix}\begin{pmatrix}v_x\\ v_y\\ v_z\end{pmatrix} \\ &= \frac{\varphi_{xx}v_x^2+\varphi_{yy}v_y^2+\varphi_{zz}v_z^2+2\varphi_{xy}v_xv_y+2\varphi_{xz}v_xv_z+2\varphi_{yz}v_yv_z}{|\nabla\varphi|},\end{aligned}$$

[2]Interpreting the surface $\mathcal{S}$ as the zero-level set of a (continuous) function $\varphi(x,y,z)$ defined on the whole space, i.e. the set of points $P(x,y,z)$ such that $\varphi(x,y,z) = 0$, the unit vector defined as $\mathbf{n} = \nabla\varphi/|\nabla\phi|$ points toward the semi-space where $\varphi > 0$. As a consequence, if $\varphi(x,y,z) = z-f(x,y)$, $\mathbf{n}$ points towards the semi-space *above* the surface.

and then

$$\begin{aligned}
&\mathbf{v}\cdot(\nabla\mathbf{n}\,\mathbf{v})\,\mathbf{n} = \\
&\quad \frac{\varphi_x\left(\varphi_{xx}v_x^2+\varphi_{yy}v_y^2+\varphi_{zz}v_z^2+2\varphi_{xy}v_xv_y+2\varphi_{xz}v_xv_z+2\varphi_{yz}v_yv_z\right)}{|\nabla\varphi|^2}\mathbf{i}_1+ \\
&\quad \frac{\varphi_y\left(\varphi_{xx}v_x^2+\varphi_{yy}v_y^2+\varphi_{zz}v_z^2+2\varphi_{xy}v_xv_y+2\varphi_{xz}v_xv_z+2\varphi_{yz}v_yv_z\right)}{|\nabla\varphi|^2}\mathbf{i}_2+ \\
&\quad \frac{\varphi_z\left(\varphi_{xx}v_x^2+\varphi_{yy}v_y^2+\varphi_{zz}v_z^2+2\varphi_{xy}v_xv_y+2\varphi_{xz}v_xv_z+2\varphi_{yz}v_yv_z\right)}{|\nabla\varphi|^2}\mathbf{i}_3.
\end{aligned} \tag{2.7}$$

Noting that

$$\mathbf{F}\cdot\mathbf{n} = -\frac{\varphi_z}{|\nabla\varphi|}mg \tag{2.8}$$

and combining Eq. (2.6), Eq. (2.7) and Eq. (2.8), the equation of motion (2.5) is written in terms of its Cartesian components as follows:

$$\begin{aligned}
\dot{v}_x =& \frac{\varphi_x\varphi_z}{|\nabla\varphi|^2}g - \frac{\varphi_x\left(\varphi_{xx}v_x^2+\varphi_{yy}v_y^2+\varphi_{zz}v_z^2+2\varphi_{xy}v_xv_y+2\varphi_{xz}v_xv_z+2\varphi_{yz}v_yv_z\right)}{|\nabla\varphi|^2} \\
&-\mu\left|\left(-\frac{\varphi_{xx}v_x^2+\varphi_{yy}v_y^2+\varphi_{zz}v_z^2+2\varphi_{xy}v_xv_y+2\varphi_{xz}v_xv_z+2\varphi_{yz}v_yv_z}{|\nabla\varphi|}+\frac{\varphi_z}{|\nabla\varphi|}g\right)\right|\frac{v_x}{|\mathbf{v}|}, \\
\dot{v}_y =& \frac{\varphi_y\varphi_z}{|\nabla\varphi|^2}g - \frac{\varphi_y\left(\varphi_{xx}v_x^2+\varphi_{yy}v_y^2+\varphi_{zz}v_z^2+2\varphi_{xy}v_xv_y+2\varphi_{xz}v_xv_z+2\varphi_{yz}v_yv_z\right)}{|\nabla\varphi|^2} \\
&-\mu\left|\left(-\frac{\varphi_{xx}v_x^2+\varphi_{yy}v_y^2+\varphi_{zz}v_z^2+2\varphi_{xy}v_xv_y+2\varphi_{xz}v_xv_z+2\varphi_{yz}v_yv_z}{|\nabla\varphi|}+\frac{\varphi_z}{|\nabla\varphi|}g\right)\right|\frac{v_y}{|\mathbf{v}|}, \\
\dot{v}_z =& -\frac{\varphi_x^2+\varphi_y^2}{|\nabla\varphi|^2}g - \frac{\varphi_z\left(\varphi_{xx}v_x^2+\varphi_{yy}v_y^2+\varphi_{zz}v_z^2+2\varphi_{xy}v_xv_y+2\varphi_{xz}v_xv_z+2\varphi_{yz}v_yv_z\right)}{|\nabla\varphi|^2} \\
&-\mu\left|\left(-\frac{\varphi_{xx}v_x^2+\varphi_{yy}v_y^2+\varphi_{zz}v_z^2+2\varphi_{xy}v_xv_y+2\varphi_{xz}v_xv_z+2\varphi_{yz}v_yv_z}{|\nabla\varphi|}+\frac{\varphi_z}{|\nabla\varphi|}g\right)\right|\frac{v_z}{|\mathbf{v}|},
\end{aligned} \tag{2.9}$$

where

$$\mathbf{v} = v_x\mathbf{i}_1+v_y\mathbf{i}_2+v_z\mathbf{i}_3 = v_x\mathbf{i}_1+v_y\mathbf{i}_2+(v_x\varphi_x+v_y\varphi_y)\,\mathbf{i}_3, \qquad |\mathbf{v}| = \sqrt{v_x^2+v_y^2+(v_x\varphi_x+v_y\varphi_y)^2}.$$

Recalling that $\varphi(x,y,z) = z - f(x,y)$, if the surface is given in explicit form $z = f(x,y)$, such that

$$\begin{gathered}
\varphi_x = -f_x, \quad \varphi_y = -f_y, \quad \varphi_z = 1, \\
\varphi_{xx} = -f_{xx}, \quad \varphi_{yy} = -f_{yy}, \quad \varphi_{xy} = -f_{xy}, \quad \varphi_{xz} = \varphi_{yz} = \varphi_{zz} = 0,
\end{gathered}$$

Eq. (2.9) can be rewritten as follows:

$$
\begin{aligned}
\dot{v}_x &= -\frac{f_x}{f_x^2+f_y^2+1}g - \frac{f_x\left(f_{xx}v_x^2+f_{yy}v_y^2+2f_{xy}v_xv_y\right)}{f_x^2+f_y^2+1} - \mu\,|\boldsymbol{\Phi}|\,\frac{v_x}{|\mathbf{v}|},\\
\dot{v}_y &= -\frac{f_y}{f_x^2+f_y^2+1}g - \frac{f_y\left(f_{xx}v_x^2+f_{yy}v_y^2+2f_{xy}v_xv_y\right)}{f_x^2+f_y^2+1} - \mu\,|\boldsymbol{\Phi}|\,\frac{v_y}{|\mathbf{v}|},\\
\dot{v}_z &= -\frac{f_x^2+f_y^2}{f_x^2+f_y^2+1}g + \frac{f_{xx}v_x^2+f_{yy}v_y^2+2f_{xy}v_xv_y}{f_x^2+f_y^2+1} - \mu\,|\boldsymbol{\Phi}|\,\frac{v_z}{|\mathbf{v}|}.
\end{aligned}
\tag{2.10}
$$

where

$$
\begin{aligned}
\boldsymbol{\Phi} &= \frac{f_{xx}v_x^2+f_{yy}v_y^2+2f_{xy}v_xv_y}{\sqrt{f_x^2+f_y^2+1}} + \frac{g}{\sqrt{f_x^2+f_y^2+1}},\\
|\mathbf{v}| &= \sqrt{v_x^2+v_y^2+(v_xf_x+v_yf_y)^2}.
\end{aligned}
$$

Note that, in case of a motion on a planar surface described by the equation

$$z = f(x,y) = ax+by+d,$$

Eq. (2.10) is further simplified as follows:

$$
\begin{aligned}
\dot{v}_x &= -\frac{ag}{a^2+b^2+1} - \mu\frac{v_xg}{\sqrt{a^2+b^2+1}\sqrt{v_x^2+v_y^2+(av_x+bv_y)^2}},\\
\dot{v}_y &= -\frac{bg}{a^2+b^2+1} - \mu\frac{v_yg}{\sqrt{a^2+b^2+1}\sqrt{v_x^2+v_y^2+(av_x+bv_y)^2}},\\
\dot{v}_z &= -\frac{\left(a^2+b^2\right)g}{a^2+b^2+1} - \mu\frac{(av_x+bv_y)\,g}{\sqrt{a^2+b^2+1}\sqrt{v_x^2+v_y^2+(av_x+bv_y)^2}}.
\end{aligned}
$$

2.3.2 Change of reference frame

As already pointed out, Eq. (2.5) is independent from the reference frame in which calculations are to be exploited. In the previous section, we have written the Cartesian components of this equation for a specific reference frame, having one axis parallel to the active force **F**, which was assumed to be constant, and with opposite direction. In the following, we shall denote such reference frame as *laboratory reference frame*, or *original reference frame*.

We propose now to write the component of the same equation expressed in a different reference frame, obtained from the original one by means of a rigid body rotation described by the rotation operator **R**. We shall denote this new reference frame as *rotated reference frame*.

We first recall that, if such new reference frame is defined by a basis of orthonormal vectors $\mathbf{e}_k$ $(k = 1,2,3)$, whose components, expressed in the original reference frame are the following

$$
\begin{aligned}
\mathbf{e}_1 &= R_{11}\mathbf{i}_1 + R_{21}\mathbf{i}_2 + R_{31}\mathbf{i}_3,\\
\mathbf{e}_2 &= R_{12}\mathbf{i}_1 + R_{22}\mathbf{i}_2 + R_{32}\mathbf{i}_3,\\
\mathbf{e}_3 &= R_{13}\mathbf{i}_1 + R_{23}\mathbf{i}_2 + R_{33}\mathbf{i}_3,
\end{aligned}
$$

then the following relation holds:

$$
\mathbf{e}_k = \mathbf{R}\mathbf{i}_k, \qquad \mathbf{R} = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33}. \end{pmatrix}
$$

and it is easily seen that R_{hk} represent the cosines of the angle Ψ_{hk}e between $\mathbf{i}_h$ and $\mathbf{e}_k$:

$$
R_{hk} = \mathbf{i}_h \cdot \mathbf{R}\mathbf{i}_k = \mathbf{i}_h \cdot \mathbf{e}_k = \cos\Psi_{hk}.
$$

As is well known from linear algebra, if a vector $\mathbf{u}$ has components u_1, u_2 and u_3 in the original reference frame, i.e. $\mathbf{u} = u_1\mathbf{i}_1 + u_2\mathbf{i}_2 + u_3\mathbf{i}_3$, then its components u_1', u_2' and u_3' in the rotated reference frame are given by:

$$
\begin{pmatrix} u_1' \\ u_2' \\ u_3' \end{pmatrix} = \mathbf{R}^T \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \qquad \Rightarrow \qquad
\begin{aligned}
u_1' &= R_{11}u_1 + R_{21}u_2 + R_{31}u_3,\\
u_2' &= R_{12}u_1 + R_{22}u_2 + R_{32}u_3,\\
u_3' &= R_{13}u_1 + R_{23}u_2 + R_{33}u_3,
\end{aligned}
$$

such that $\mathbf{u} = u_1'\mathbf{e}_1 + u_2'\mathbf{e}_2 + u_3'\mathbf{e}_3$.

2.4 Non-conservation of total energy in the UBO-block model

2.4.1 A frictionless test-case

A simple test with a single mass-point only was performed and compared to a reference solution. Therefore a smooth surface $\mathcal{S}$, given by an explicit function $f = f(x,y)$ serves as test surface. As for real terrains, the surface is not known exactly but only in terms of single data points (and therefore in particular is not smooth), the only way to evaluate the goodness of the model is to provide a smooth surface which is then triangulated. On the smooth surface the equations can be solved numerically, thus obtaining a reference solution while for the UBO-block model the triangulation of the same surface is used as test surface. On the basis of the reference solution then the quality of the approximated solution obtained using the UBO-block model can be judged.

In the following, the solution of the UBO-Block landslide model, which uses a discretization of the given smooth surface, will be called the *UBO-block solution*. As an exact solution to the problem is not known, the numerical solution of the underlying ODE-system solved on the smooth surface is considered the *reference solution*. It turned out that in the UBO-block solution – in absence of friction, and gravity being the only active force – total energy is not conserved while in contrast, it is conserved in the reference solution. This problem was the starting point of the present work and its causes are analysed in the following.

In the frictionless case Eq. (2.10) reduces to:

$$\begin{aligned}\dot{v}_x &= -g\frac{f_x}{1+f_x^2+f_y^2} + \frac{-f_x}{1+f_x^2+f_y^2}\left(v_x^2 f_{xx} + 2v_x v_y f_{xy} + v_y^2 f_{yy}\right),\\ \dot{v}_y &= -g\frac{f_y}{1+f_x^2+f_y^2} + \frac{-f_y}{1+f_x^2+f_y^2}\left(v_x^2 f_{xx} + 2v_x v_y f_{xy} + v_y^2 f_{yy}\right),\end{aligned} \tag{2.11}$$

where $z = f(x,y)$ is the sliding surface, along which a point mass is moving and $f_{x/y}$ and $f_{xx/xy/yy}$ are the first and second order partial derivatives of f. In the following, $\dot{v}_x$ and $\dot{v}_y$ will also be denoted by a_x and a_y, respectively. The test-surface is a paraboloid given by:

$$f(x,y) = \sum_{\substack{i,j=0\\ i+j\leq 2}}^{2} a_{ij}x^i y^j,$$

where the parameters a_{ij} are chosen as:

$$a_{20} = 0.00005, \quad a_{02} = 0.01, \quad a_{11} = 0, \quad a_{10} = -0.1, \quad a_{01} = 0, \quad a_{00} = 50.$$

The function f is convex $\forall x,y$.

Eq. (2.11) is a second order ODE-system with respect to $\mathbf{x} = (x,y)$. To solve this numerically it is convenient to rewrite it as a first order ODE-system:

$$\dot{\mathbf{y}}(t) = \frac{d\mathbf{y}}{dt} = \mathbf{F}(\mathbf{y},t), \quad \mathbf{y}(t_0) = \mathbf{y}_0,$$

with $\mathbf{y} = (x, v_x, y, v_y)$, and:

$$\mathbf{F} = \begin{pmatrix} v_x \\ -f_x \dfrac{g + \left(v_x^2 f_{xx} + 2 v_x v_y f_{xy} + v_y^2 f_{yy}\right)}{1 + f_x^2 + f_y^2} \\ v_y \\ -f_y \dfrac{g + \left(v_x^2 f_{xx} + 2 v_x v_y f_{xy} + v_y^2 f_{yy}\right)}{1 + f_x^2 + f_y^2} \end{pmatrix}. \tag{2.12a}$$

As the mass point is constrained to stay on the surface, the vertical components z and v_z are obtained by evaluating:

$$\begin{aligned} z &= f(x,y), \\ v_z &= v_x f_x + v_y f_y. \end{aligned}$$

The simulation starts at $(t_0 = 0)$ and is run until $(T = 20s)$, the step-size is chosen as $(\triangle t = 0.2s)$. The initial conditions are given with the initial position $(x_0, y_0) = (5, -85)$ and the initial velocity $(v_{x0}, v_{y0}) = (0, 0)$, which means:

$$\mathbf{y}(t_0) = (x_0, v_{x,0}, y_0, v_{y,0}) = (5, 0, -85, 0)$$

.

The sliding surface as well as the resulting trajectory of a point mass is shown in Fig.(2.1).

While in the reference solution the total energy remains constant, there is a gain of energy in the UBO-block solution. The problem is shown in Figs. 2.2-2.4. Fig. 2.2 and Fig. 2.3 depict the total energy plus its decomposition into kinetic and potential energy for the discretized (UBO-block) respectively reference solution. The increase of total energy until the end-time T of this simulation is about 0.67% of the total energy which is better seen in Fig. 2.4, also showing that the total energy is constant for the reference solution. Due to the law of *conservation of energy*, the total energy in an isolated system must remain constant over time. As in the present model there is no source or sink of energy the gain must originate from either numerical or model errors.

It sticks out that the total energy does not increase linearly but stepwise. While it is almost constant over wide intervals, there are two sudden rises at about $t = 5$ and $t = 16$. This behaviour is the key towards understanding the problem. Plotting acceleration, its variation and total energy of the UBO-block solution in one figure,

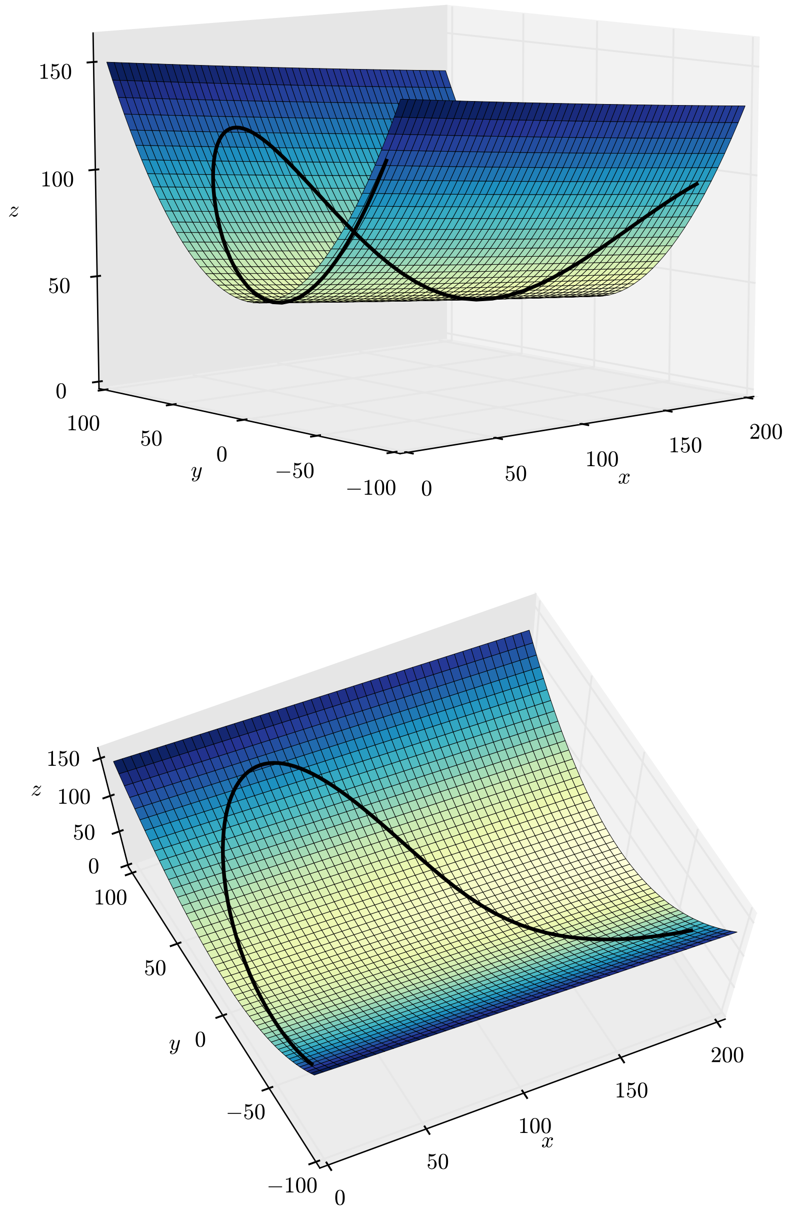


FIGURE 2.1: Sliding surface and reference trajectory for the test problem from two different angles.

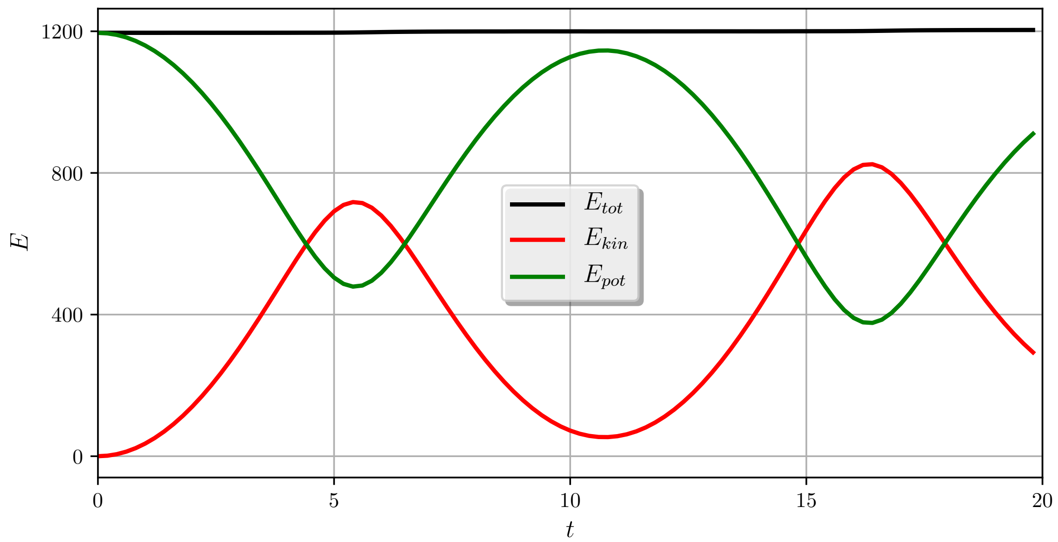


FIGURE 2.2: Total energy (black) and its decomposition into kinetic (red) and potential energy (green) (right) of the UBO-block solution.

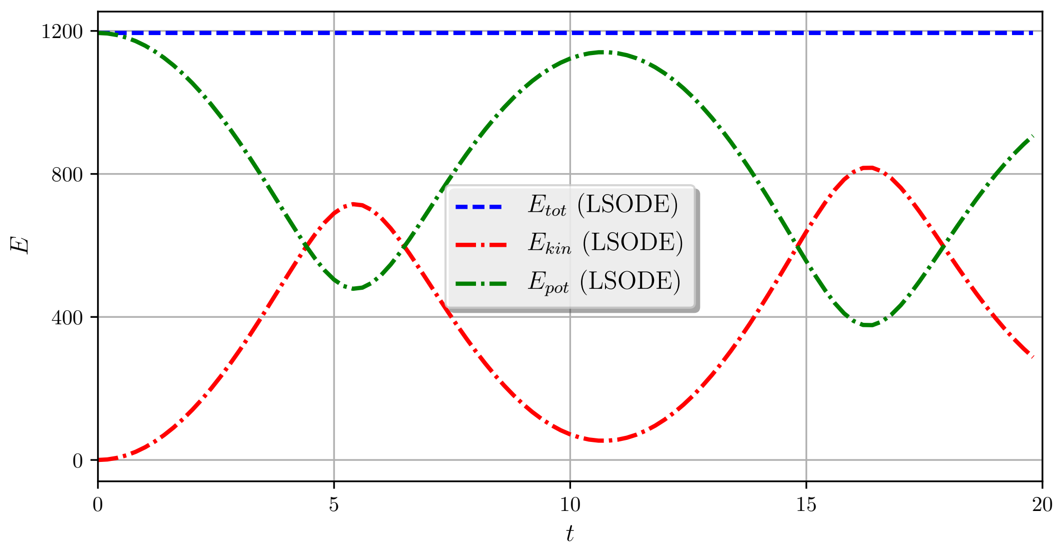


FIGURE 2.3: Total energy (blue) and its decomposition into kinetic (red) and potential energy (green) (right) of the reference solution.

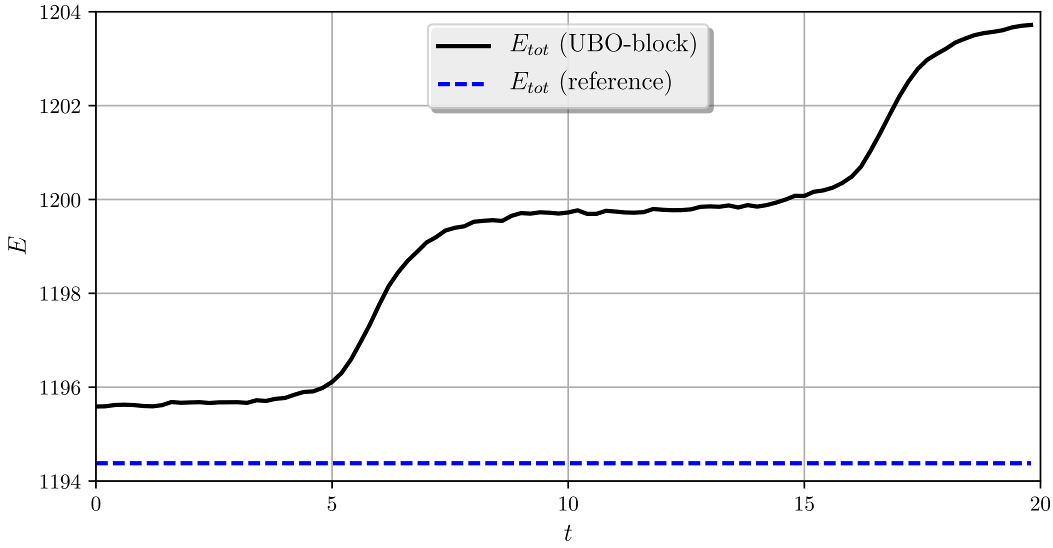


FIGURE 2.4: Comparison of the total energy for the UBO-block (black) and the reference solution (blue).

one finds a correlation between them. Corresponding to the x-component of the acceleration a_x we will hereby denote by a_x' the variation of a_x (cf. Fig.2.5).

In regions in which the acceleration shows little variation, the total energy is almost constant. Instead, in regions of high variation it is increasing significantly. Plotting a_x and a_x' for the sliding surface should clarify this, but, as in the 2-dimensional case the trajectory (in contrast to the 1-dimensional case) is a priori unknown and can generally be obtained only by solving the differential equation, a parabola given by $f(x) = 0.5x^2$ will be used instead. This curve is similar to the slice plane of the original surface (cut along the y-axis) such that the qualitative behaviour will be comparable. As in the UBO-block solution the second order derivatives are all *zero*, terms containing second derivatives will be neglected in the following:

$$a_x(x) = -g\frac{f_x}{1+f_x^2},$$

and therefore the first derivative of a_x with respect to x which can be considered the variation of the acceleration reads:

$$a_x'(x) = -g\frac{1-f_x^2}{(1+f_x^2)^2}.$$

The three functions, f, a_x and a_x' are plotted in Fig. 2.6 showing that the variation of a_x is increasing up the $x = 0$ and then decreasing again. This is additionally illustrated by the green lines, denoting the direction of the acceleration a_x in certain points of the curve. When solving the ODE on smooth surfaces, the first (slope) and second derivative of $f(x)$ and therefore the acceleration of the point cannot be

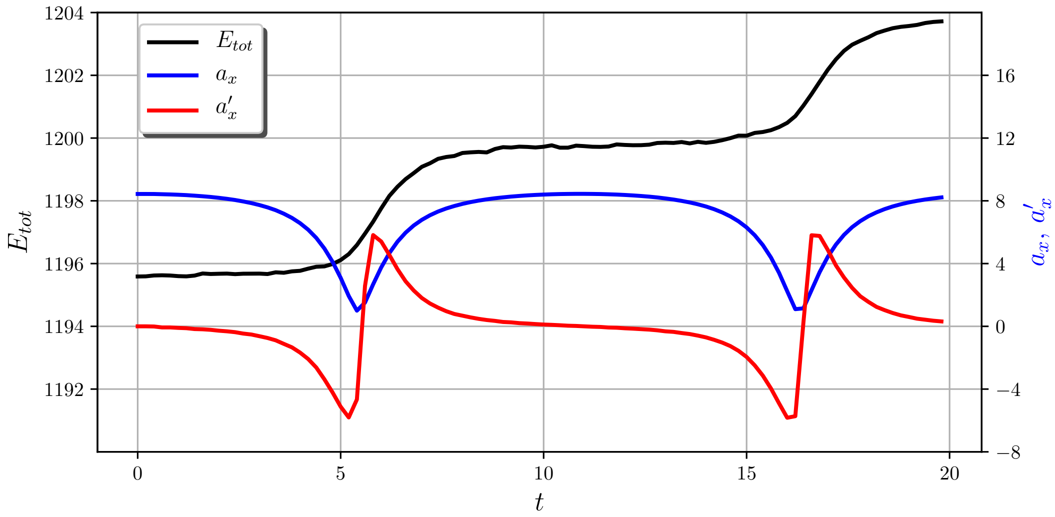


FIGURE 2.5: Total energy, acceleration and variation of acceleration (first derivative) for the UBO-block solution.

computed continuously as a time-discretization is unavoidable. As the shape of f instead changes continuously, this leads to mistakes that directly depend on a'_x. Even though the UBO-block model uses a discretized surface, the problem is the same, as the acceleration is updated only at the end of each time-step, even if meanwhile the point passes through several different triangles.

For a convex function $f(x)$ two cases have to be distinguished: the movement in direction of increasing x and the movement in direction of decreasing x. In the first case, the piecewise linear functions approximating the slope of the function overestimate the real slope as long as $f'(x) < 0$, whereas for $f'(x) \geq 0$ the slope is underestimated (as can easily be seen by the green lines in Fig.2.6). When moving in direction of decreasing x the real slope is overestimated for $f'(x) \geq 0$ and underestimated when $f'(x) < 0$.

This leads to an overestimation of the acceleration in case of increasing velocity, and to an underestimation when the velocity is decreasing resulting in an continuous overestimation of velocity which finally leads to the monotone increase of total energy as seen above.

In case of a concave, monotonically decreasing sliding surface like $f(x) = -x^2 + a,\ \ x \in \mathbb{R}_0^+,\ \ a \in \mathbb{R}$, one would find a continuous underestimation of the slope of the sliding surface and therefore a decrease in the total energy. Accordingly, the total energy will fluctuate for surfaces with varying convexity.

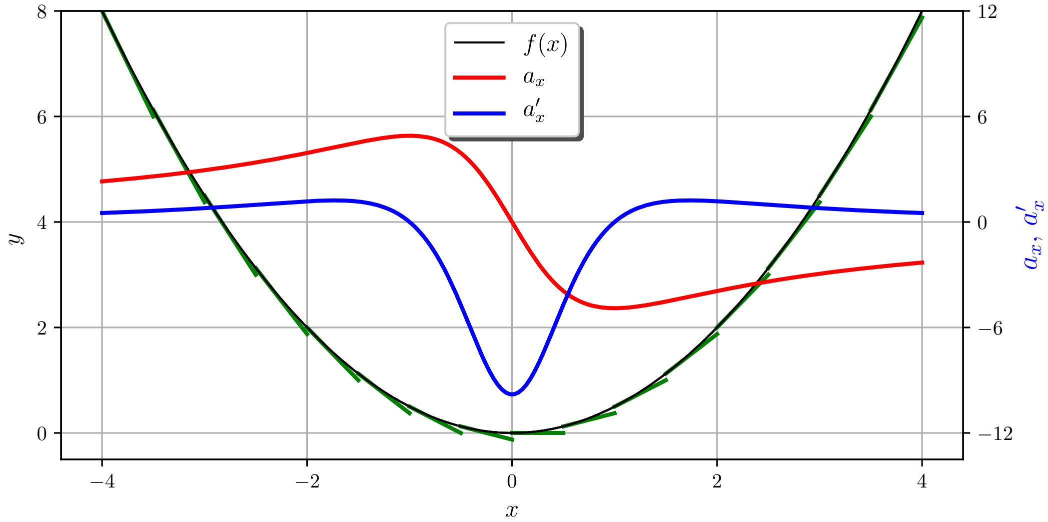


FIGURE 2.6: Sliding surface (black), its approximation by linear functions (green), tangential acceleration $a(x)$ (red) and its variation $a'(x)$ (blue). Note, that $\triangle x$ in the global coordinate system stays constant, which means, that in this case $\triangle x_s$, the length of the curve and therefore the area which is approximated by a linear function, becomes smaller in regions of higher variation of $a(x)$. Nonetheless, the error in the approximation is still much higher here.

2.4.2 Differences in the time-integration

This problem does not appear in the reference solution. While the integration scheme of the UBO-block model is basically an explicit Euler-scheme, the Livermore Solver for Ordinary Differential Equations (LSODE) Hindmarsh and Radhakrishnan, 1993 – a solver particularly suited for stiff ODEs – was used for the reference solution instead.

In order to evaluate the influence of the time-integration scheme on the solution of the equation, an Euler-scheme was applied also to the reference problem, leading to an increase of total energy as well, yet much higher than the one in the UBO-block solution (cf. Fig.2.7) namely about 32%. Furthermore different Runge-Kutta schemes of order two, three and four where implemented and used for the time integration of the reference problem. The solutions are displayed in Fig.(2.8) where the error in total energy decreases with the order of the method. Using a fourth order scheme finally leads to nearly constant total energy.

Compared to a fourth-order Runge-Kutta scheme, using LSODE the time-step can be chosen at least twenty times larger for obtaining comparable results thus reducing computational cost. The maximum time step under which the total energy remains approximately constant using the RK4-scheme is $\triangle t = 0.1$ whereas for LSODE $\triangle t = 2$ is sufficient (cf. Fig.2.9).

The above results suggest that Eq. (2.12) is a stiff ODE system. Moreover, the

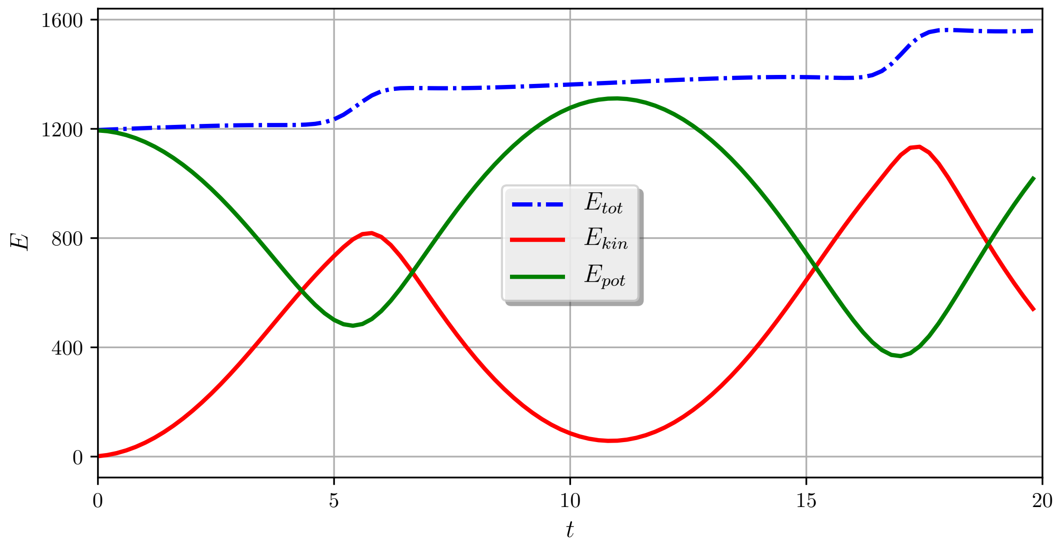


FIGURE 2.7: Total energy and its decomposition computed for the analytical solution using a Euler-forward scheme.

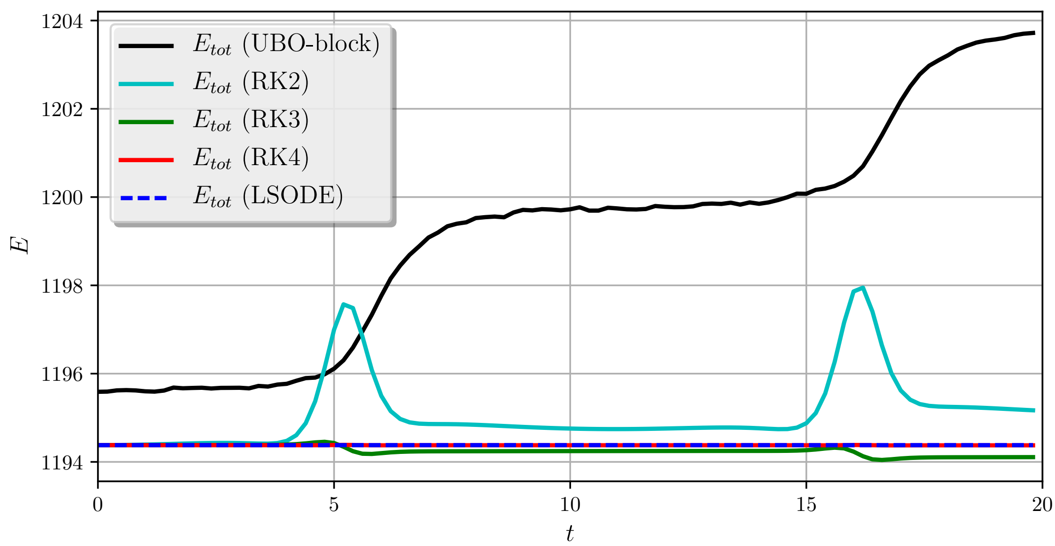


FIGURE 2.8: Total energy using different Runge-Kutta time integration schemes and the UBO-block scheme (black).

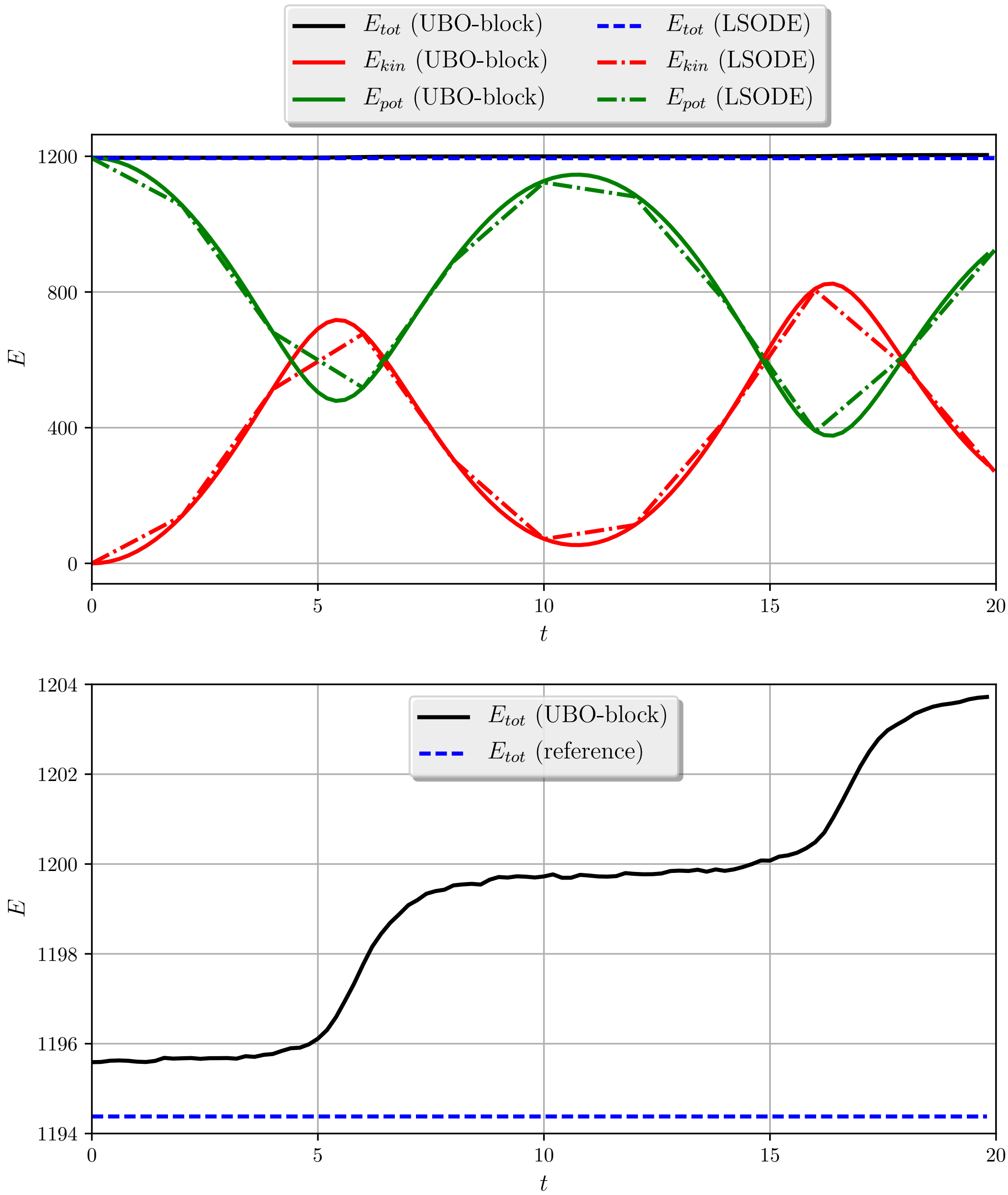


FIGURE 2.9: Total energy and its decomposition (black lines) computed for the analytical solution using the LSODE-solver with step size $\triangle t = 2$ (dashed lines) and the UBO-block scheme with time step $\triangle t = 0.2$ (solid lines).

observations show that the discretized problem is probably less stiff. This will be discussed in the next section.

2.5 Stiffness of the governing equations

Roughly spoken one can say, that the stiffness of an equation describes the dependency of the numerical solution on the step-size used for the discretization (here the time-discretization). This means that in case of a stiff equation extremely small time steps are required to obtain sufficiently good results, otherwise the numerical solution becomes unstable. From a more physical point of view stiffness can be interpreted as two or more phenomena taking place on very different time scales.

Mathematically, the stiffness of an equation can be determined by its eigenvalues. For a linear system of ODEs:

$$\mathbf{y}' = \mathbf{A}\mathbf{y}(x) + \mathbf{b}(x)$$

the stiffness can be determined by calculating the eigenvalues of $\mathbf{A}$ and then computing the stiffness ratio σ.

Definition 1. Let λ_t, $t = 1, 2, \dots, n$ be the eigenvalues of a matrix $\mathbf{A}$. Then $\overline{\lambda}$, $\underline{\lambda} \in \{\lambda_t\}_{t=1,\dots,n}$ are defined as:

$$|Re(\overline{\lambda})| \geq |Re(\lambda_t)| \geq |Re(\underline{\lambda})|$$

and the stiffness ratio σ as

$$\sigma := \frac{|Re(\overline{\lambda})|}{|Re(\underline{\lambda})|}.$$

As $\mathbf{A}$ does not change in time, the eigenvalues are constant and therefore the stiffness of the system does not change in time, too. For a non-linear system of ODEs the stiffness can only be calculated locally. Given:

$$\dot{\mathbf{y}}(t) = \mathbf{F}(\mathbf{y}, t),$$

the local stiffness of the system at $t = t_n$ is determined by the eigenvalues of its Jacobian $\mathbf{J}(t_n)$. As the Jacobian is changing in time, also the eigenvalues change and therefore the stiffness. Note, that in the case of a linear system of equations, the matrix $\mathbf{A}$ describing the system, coincides with its Jacobian.

Generally spoken, one can say that the larger σ, the stiffer is the system. More precisely, a system is considered to be stiff for $\sigma > 10^3$. This would also mean, that if $|Re(\underline{\lambda})| = 0$ σ is infinite. But: in the case that also $|Re(\overline{\lambda})|$ is sufficiently small, the system is not stiff at all.

In the following the Jacobians and their corresponding eigenvalues for the reference and discretized problem are computed. The Jacobian matrix $J_F(a)$ of a function $F:\mathbb{R}^n \to \mathbb{R}^m$ in a given point a is defined as:

$$J_F(a) := \frac{\partial \mathbf{F}}{\partial \mathbf{x}}(a) := \left(\frac{\partial F_i}{\partial x_j}(a)\right)_{i=1,\dots,m;j=1,\dots,n} = \begin{pmatrix} \frac{\partial F_1}{\partial x_1}(a) & \frac{\partial F_1}{\partial x_2}(a) & \cdots & \frac{\partial F_1}{\partial x_n}(a) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial x_1}(a) & \frac{\partial F_m}{\partial x_2}(a) & \cdots & \frac{\partial F_m}{\partial x_n}(a) \end{pmatrix}$$

Here we have $m=n=4$ and $a=(x,v_x,y,v_y)$. First the Jacobian for the reference solution, using the analytically given sliding surface f, is computed. Substituting:

$$\begin{aligned} A &:= 1+f_x^2+f_y^2, \\ B &:= v_x^2 f_{xx} + 2v_x v_y f_{xy} + v_y^2 f_{yy}, \end{aligned}$$

(2.12a) can be rewritten as:

$$\mathbf{F} = \begin{pmatrix} y_2 \\ -\frac{f_x}{A}(B+g) \\ y_4 \\ -\frac{f_y}{A}(B+g) \end{pmatrix},$$

and the Jacobian of this problem can be written as:

$$J_F(a) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\frac{d}{dx}\left(\frac{f_x}{A}\right)(B+g) & -\frac{f_x}{A}\left(\frac{d}{dv_x}(B)+g\right) & -\frac{d}{dy}\left(\frac{f_x}{A}\right)(B+g) & -\frac{f_x}{A}\left(\frac{d}{dv_y}(B)+g\right) \\ 0 & 0 & 0 & 1 \\ -\frac{d}{dx}\left(\frac{f_y}{A}\right)(B+g) & -\frac{f_y}{A}\left(\frac{d}{dv_x}(B)+g\right) & -\frac{d}{dy}\left(\frac{f_y}{A}\right)(B+g) & -\frac{f_y}{A}\left(\frac{d}{dv_y}(B)+g\right) \end{pmatrix},$$

with:

$$\begin{aligned} \frac{d}{dx}\left(\frac{f_x}{A}\right) &= \frac{2a_{20}A - 2f_x(2a_{20}f_x + a_{11}f_y)}{A^2}, \\ \frac{d}{dy}\left(\frac{f_x}{A}\right) &= \frac{a_{11}A - 2f_x(a_{11}f_x + 2a_{02}f_y)}{A^2}, \\ \frac{d}{dx}\left(\frac{f_y}{A}\right) &= \frac{a_{11}A - 2f_y(2a_{20}f_x + a_{11}f_y)}{A^2}, \\ \frac{d}{dy}\left(\frac{f_y}{A}\right) &= \frac{2a_{02}A - 2f_y(a_{11}f_x + 2a_{02}f_y)}{A^2}, \end{aligned}$$

and:

$$\frac{d}{dv_x}(B) = 2(v_x f_{xx} + 1),$$
$$\frac{d}{dv_y}(B) = 2(v_y f_{yy} + 1).$$

Evaluating this numerically using a fourth order Runge-Kutta scheme and a step-size $\triangle t = 0.2$ gives a maximum stiffness-ratio $\sigma_{max} = 6.79 \cdot 10^6$, a minimum of $\sigma_{min} = 274.6$ and an average of $\sigma_m = 4.37 \cdot 10^4$. Therefore the system can be considered to be stiff.

It has to be mentioned here that generally the stiffness strongly depends on the second order partial derivatives of the surface. In the present case, all second order partial derivatives are constant which has to be considered a particular case. For surfaces with non-constant second order partial derivatives, the equations will mostly be significantly stiffer than they are in the present case.

On the other hand, if f is described by a plane the matrix A becomes:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

The eigenvalues of this matrix are all *zero* and so the system is non-stiff, but indeed only for the motion on single planes. For the transition between two planes the stiffness-ratio cannot be computed as the derivatives are not defined in those points, but, in contrast to smooth surfaces, the number of transitions and therefore change in slope is finite. This explains why the two solutions, using the same time-integration scheme with the same step-size show a different behaviour but it also explains the stepwise increasing total energy. While in the reference problem the slope of the surface and thus also the points acceleration are changing continuously, this is not the case in the discretized problem where slope and acceleration only change at the transition point of two planes. This leads to a stronger increase in the total energy for the reference problem if an inappropriate solver is used for the time-integration. Generally, in regions with stronger variations of the surface the stiffness ratio is larger and the errors in the solution (visible in the gain of energy) are larger as well which is the reason for the stepwise changes in the total energy.

Generally, two different approaches are conceivable now.

1. Reconstructing a smooth surface from the given data points and solving the equations of motion numerically. This seems straightforward as in the example

problem the energy was conserved using the lsode-solver.

2. Improving the UBO-block model such that the energy is conserved. The big advantage of this method is that no surface needs to be reconstructed.

In the following sections, the two approaches will be compared and discussed more in detail. Therefore the next chapter concentrates on the reconstruction of smooth surfaces. In Chapters 4 and 5 an energy conserving method for the 1- and 2-dimensional case, using linearized curves and triangulated surfaces, respectively, will be derived.

Chapter 3

Smooth Reconstructed Surfaces as a Basis for the Solution of the Equations of Motion in 2D

In the first chapter it was shown that increasing the order of the time-integration method can significantly improve the results of the computation in terms of conservation of energy. Certainly, this holds only if the surface is a smooth function where also the first and second partial derivatives can be computed (either analytically or numerically) and are continuous. Moreover, eq. (2.11) is generally a stiff problem, where the stiffness depends on the sliding surface (more concretely, the degree of variation in the first and second derivatives). This makes it desirable to find a surface reconstruction that is not only smooth but also provides smooth first and second order partial derivatives.

Therefore this section concentrates on the reconstruction of sliding surfaces as a basis for the numerical solution of the equations. Hereby the focus is on the so called "*smooth surface reconstruction methods*" which are frequently used also in geophysical applications.

Note that however, even under these conditions, the problem remains stiff and therefore requires an implicit time-integration scheme that can handle those sort of problems. In case that the derivatives should not be sufficiently smooth, even a time integration method like the Livermore-Solver (known also as LSODE-solver) cannot longer guarantee a sufficiently good solution of the equations (in the frictionless case also ensuring energy-conservation) or too small time-steps are needed to obtain good results.

The reconstructed sliding surface will be denoted by $s(\mathbf{x}),\ \mathbf{x} = (x, y)$ and will – as already mentioned – be the result of a so called *smooth surface reconstruction*. Those methods provide a surface, as well as first and second order partial derivatives, that are given by globally defined (in contrast to piecewise-defined) analytical functions. Therefore, when evaluating the functions, no surface segments (in which a given point $\mathbf{x}$ is located) needs to be identified but the desired results can be computed directly.

Therefore, three different methods have been implemented and tested. They were –based on a given test problem – analysed in terms of their ability to provide a smooth reconstruction to given data points and in addition also the first and second derivatives were investigated.

3.1 Smooth fitting surfaces

One possibility for the reconstruction of smooth surfaces is the use of so called radial basis functions f. These depend only on the distance from the origin, which means that for a function f the following must hold: $f(\mathbf{x}) = f(\|\mathbf{x}\|)$ respectively for some given origin $\mathbf{x}_0$: $f(\mathbf{x},\mathbf{x_0}) = f(\|\mathbf{x}-\mathbf{x}_0\|)$. For the reconstruction $s(\mathbf{x})$ of a surface S a linear combination of such functions can be used. Given a number n of points $\mathbf{x}_k = (x_k, y_k)$ with $k = 1,\ldots,n$ and z_k the corresponding observed (function) values, this reads:

$$s(\mathbf{x}) = p(\mathbf{x}) + \sum_{k=1}^{n} \lambda_k f(\|\mathbf{x}-\mathbf{x}_k\|), \tag{3.1}$$

where the λ_k are the parameters to be determined and $p(\mathbf{x})$ is an optional polynomial of degree m, usually included to model global trends in the data. The weights λ_k, and the polynomial coefficients are then determined by the requirement

$$s(\mathbf{x}_k) = z_k,\ k = 1,\ldots,n.$$

The space of polynomials of degree m in d variables has dimension $M = \binom{m+d}{d}$. Hence, if the polynomial p is present, it introduces additional degrees of freedom that must be removed. Let $p_i : i = 1,...,M$ be a basis for all polynomials in d dimensions with degree at most m. To remove the additional degrees of freedom the following conditions have to be satisfied:

$$\sum_{k=1}^{n} \lambda_k p_i(\mathbf{x}_k) = 0,\ i = 1,\ldots,m.$$

In our case, the polynomial p is of degree *zero* and therefore just a constant parameter that will be denoted by λ_0. While the radial basis functions f are also called *ansatz*-functions, $s(\mathbf{x})$ is the so called *reconstructing* function. Let's in the following use the definition $r_k := (\|\mathbf{x}-\mathbf{x}_k\|)$, or $r_k = [(x-x_k)^2 + (y-y_k)^2]^{1/2}$ when using it in the Euclidean sense.

As it is described in (Billings et al. 2002, Interpolation of geophysical data using continuous global surfaces), many surface-fitting methods are defined naturally as the solution of certain variational problems. Therefore a penalty function $J(s)$ is defined and then the function $s(\mathbf{x})$ is chosen in the way to be the solution of the constrained optimization problem:

$$\min J(s) \quad \text{subject to} \quad s(\mathbf{x}_k) = f_k : k = 1, ..., n.$$

In the following, three different reconstructing functions and their derivatives are presented. As an example, the reconstructed surfaces are plotted for the first test-case. The number of grid points per coordinate-direction is 21, so overall, 441 grid points are used for the reconstruction. For the plotting, the double amount of grid points per direction was chosen to show the behaviour of the function in between the reconstructed points. In addition, the absolute errors between the reconstruction and the original surface are shown as well as the parameters of the reconstructing function. Then the motion of a point is simulated for different surfaces, reconstructed using the Inverse-Multi-Quadratic reconstruction. The test-surfaces will all be constant in y-direction.

3.2 The tension-spline

The first ansatz-function presented here is the so called tension-spline. In its original formulation this reconstruction leads to overshoots in regions with rapid changes of gradients. To eliminate this problem the so called tension parameter was introduced by Smith and Wessel (1990). Another setting was introduced by Mitasova and Mitas (1993) which is presented here.
Let's for a moment go back to equation (3.1). The polynomial $p(\mathbf{x})$ can be written as $p(\mathbf{x}) = \sum_{j=1}^{K} a_j q_j(\mathbf{x})$ where the q_j are a basis for the space of polynomials and a_j are the corresponding coefficients. The penalty function $J(s)$ was then chosen as:

$$J(s) = \sum_{k=0}^{\infty} \sum_{\mathbf{a}:|\mathbf{a}|=k} B_a \int_{\mathbb{R}^2} \left(\frac{\partial^{|\mathbf{a}|} s}{\partial x^{a_x} \partial y^{a_y}} \right)^2 \partial x \partial y$$

with $\mathbf{a} = (a_x, a_y)$, and $B_{\mathbf{a}} = |\mathbf{a}|! \varphi^{-2|\mathbf{a}|} / [a_x! a_y! (|\mathbf{a}| - 1)!]$ if $\mathbf{a} \neq 0$ and $B_a = 0$ otherwise. φ is the generalized tension parameter and controls the shape of the reconstructed surface. Mitasova and Mitas (1993) showed that the following ansatz-function minimizes $J(s)$ subject to interpolating the data:

$$f_k := -\left\{ \left[\ln \left(\frac{\varphi r_k}{2} \right)^2 \right] + \mathrm{E}_1 \left[\left(\frac{\varphi r_k}{2} \right)^2 \right] + C_e \right\},$$

where φ is the so called Tension-parameter whose choice is important for the quality of the reconstruction. For simplicity we will define $v_k := \left(\frac{\varphi r_k}{2} \right)^2$. Generally one can say that the greater the tension is, the more local will the behaviour be (Interpolation by Regularized Spline with Tension: I. Theory and Implementation, Mitasova and Mitas, 1993). For small values of r and v this can be approximated by

$$f_k \approx -\sum_{n=1}^{\infty} \frac{v_k^n}{n \cdot n!} \tag{3.2}$$

which is found using a Taylor-expansion for $\mathrm{E}_1(x)$:

$$\mathrm{E}_1(x) \approx C_e + \ln(|x|) + \sum_{n=1}^{\infty} \frac{v^n}{n \cdot n!}. \tag{3.3}$$

Equation (3.3) can be used to compute $\mathrm{E}_1(x)$ with floating point operations for real x between 0 and 2.5. For $x > 2.5$, the result is inaccurate due to cancellation.
Using this approximation, the terms C_e and $\ln(|x|)$ (in our case x=v and due to its definition it is always non-negative) cancel and we derive equation (3.2). For $v = 0$, f can be continued to $f(0) = 0$.
For simplicity we first compute the partial derivatives of v and with $r_k^2 = [(x-x_k)^2 + (y-y_k)^2]$ we get:

$$\begin{aligned}
v_{k,x} &= \frac{\varphi^2}{2}(x - x_k), \\
v_{k,xx} &= \frac{\varphi^2}{2}, \\
v_{k,xy} &= 0.
\end{aligned}$$

Therefore we find for the partial derivatives f_x^k, f_{xx}^k and f_{xy}^k:

$$\begin{aligned}
f_{k,x} &= \frac{v_{k,x}}{v_k}\left[e^{-v_k}\left(1 + \frac{1}{v_k}\right) - 1\right], \\
f_{k,xx} &= \frac{e^{-v_k}(v_{k,xx} - v_{k,x}^2) - v_{k,xx}}{v_k} + \frac{e^{-v_k}(v_{k,xx} - 2v_{k,x}^2) + v_{k,x}^2}{v_k^2} - \frac{2v_{k,x}^2 e^{-v_k}}{v_k^3}, \\
f_{k,xy} &= \frac{v_{k,x}v_{k,y}(v_k - e^{-v_k}((v_k+1)^2 + 1))}{v_k^3}.
\end{aligned}$$

The interior of the surface is reconstructed very well, with just a slight smoothing around the transition area ($x = 0$). Also f_x is smooth there, showing neither oscillation nor over-/undershoots, f_y is constant apart from the direct vicinity of the boundary (compare Figs. 3.1 and 3.2). As can be seen in Fig.(3.3) there are strong peaks around ($x = 0$) in f_{xx} and f_{yy} representing the sudden change in slope thus leading to a reconstruction that is very close to the original surface. (Note that the plots of the derivatives are rotated counter-clockwise with respect to the reconstructed surface

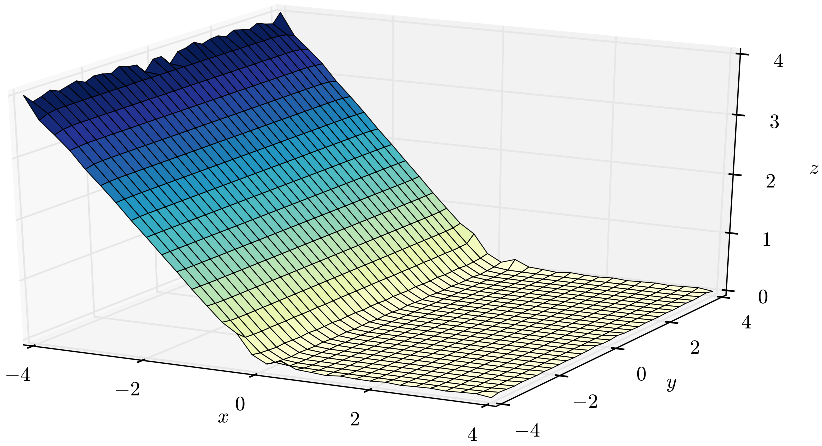


FIGURE 3.1: Reconstructed surface using the Tension Spline.

again to obtain a better visibility.) Fig. (3.4) shows the error ε between the reconstruction and the piecewise-linear surface $\varepsilon = z - z_r$. The biggest errors appear at the boundary of the surface where significant ripples are visible also in the reconstructed surface, moreover there are little errors also in the region of $x = 0$.

Altogether the Tension spline gives a very good reconstruction of the original surface (in the interior), conserving the fast change in slope while having very smooth first and second derivatives.

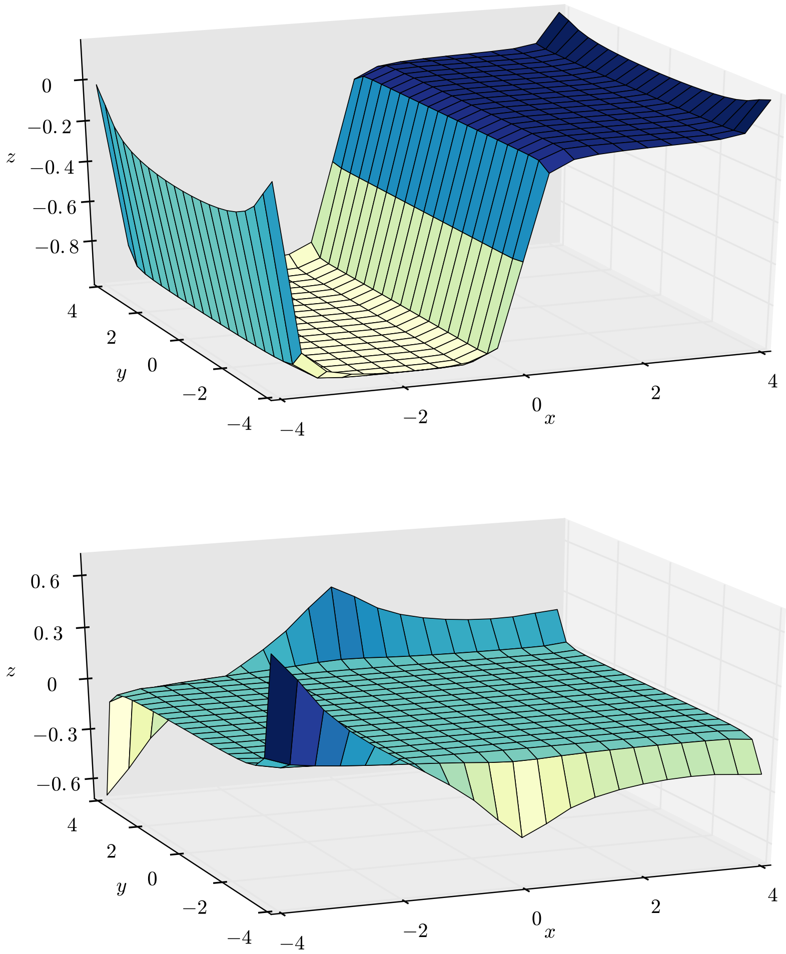


FIGURE 3.2: First order partial derivatives (top: f_x, bottom: f_y) of the reconstructed surface using the Tension Spline.

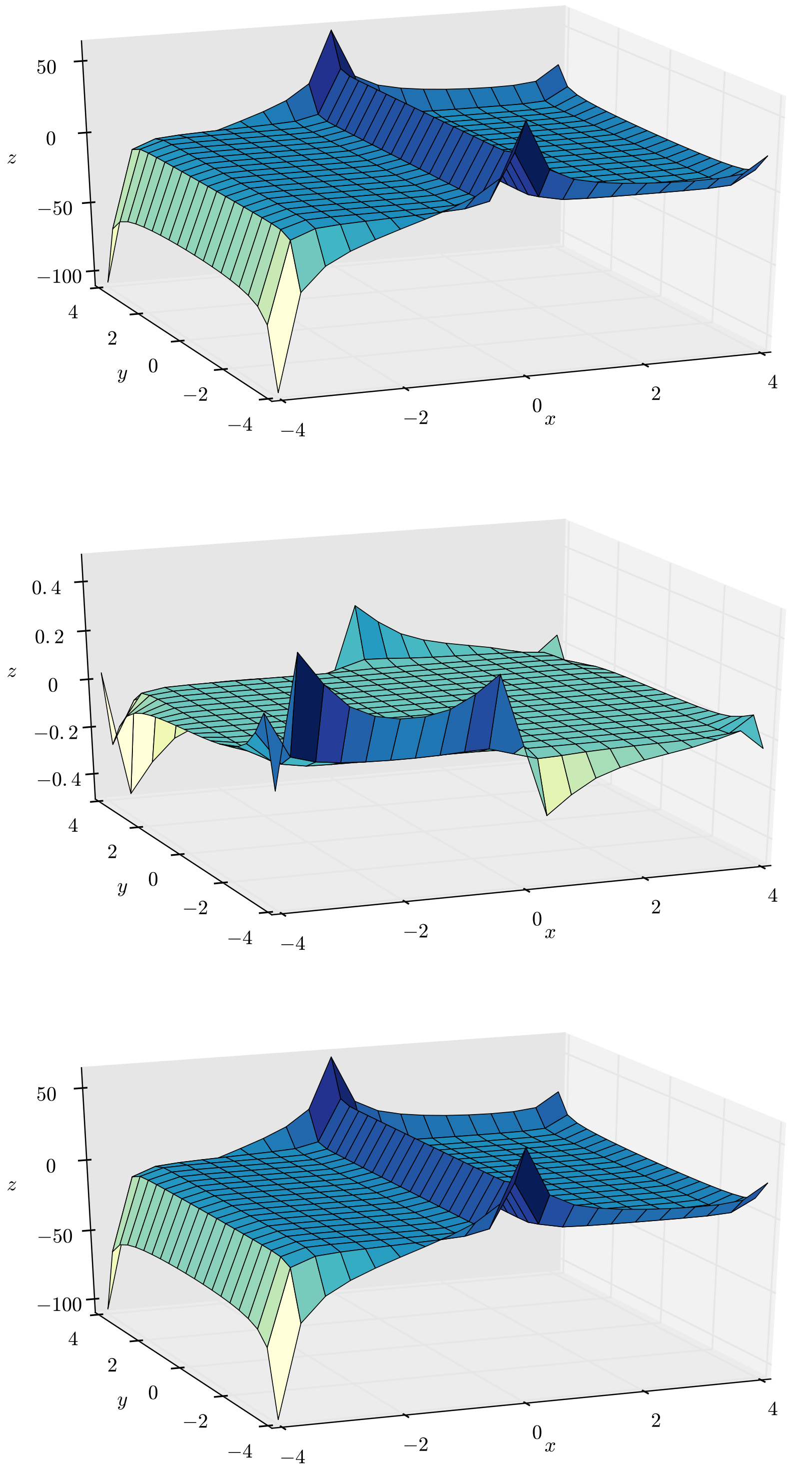


FIGURE 3.3: Second order partial derivatives (top: f_{xx}, middle: f_{xy} bottom: f_{yy}) of the reconstructed surface using the Tension Spline.

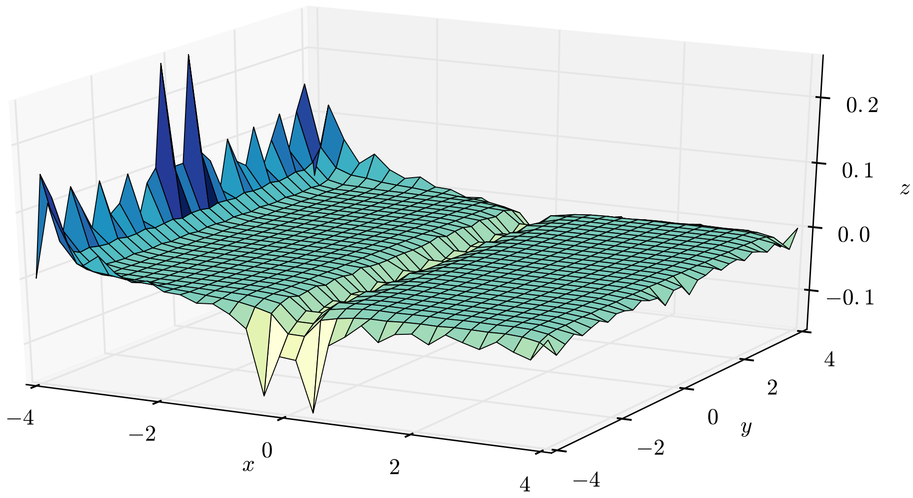


FIGURE 3.4: Tension spline: Difference to the original surface.

3.3 The thin-plate-spline

The second ansatz function is the thin-plate-spline which arises from the following penalty function:

$$J(s) = \int_{\mathbb{R}^2} \sum_{i=0}^{m+1} \binom{m+1}{i} \left(\frac{\partial^{m+1} s}{\partial x^i \partial y^{m+1-i}} \right)^2 \partial x \partial y.$$

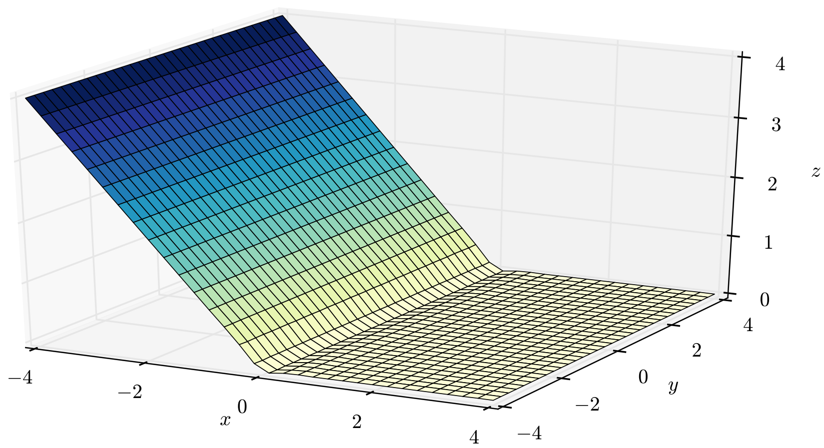


FIGURE 3.5: Reconstructed surface using the Thin-Plate Spline.

The reconstructing function satisfying this is:

$$f_k(r) := r_k^2 \ln(r_k),$$

with $\lim_{r\to\pm 0} f_{\text{tp},k}(r) \to 0$. For the partial derivatives we find:

$$
\begin{aligned}
f_x^k &= (x-x_k)(1+2\ln(r)),\\
f_{xx}^k &= (2\ln(r_k)+1)+\frac{2(x-x_k)^2}{r^2},\\
f_{xy}^k &= \frac{2(x-x_k)(y-y_k)}{r^2},
\end{aligned}
$$

where f_y^k and f_{yy}^k follow analogously.

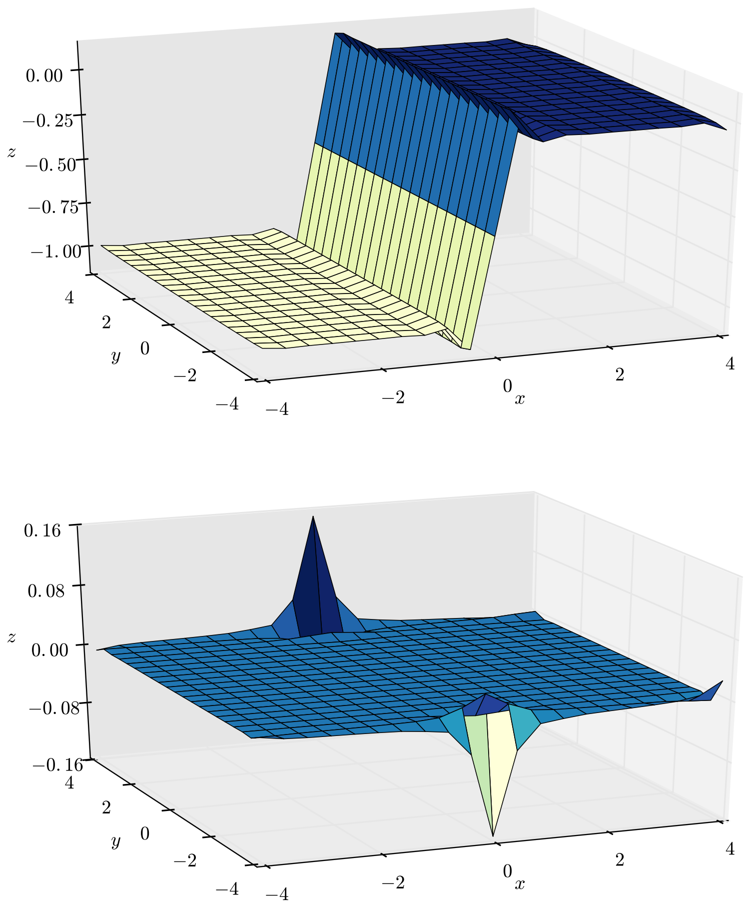


FIGURE 3.6: First order partial derivatives (top: f_x, bottom: f_y) of the reconstructed surface using the Tension Spline.

In contrast to the previous reconstruction one finds some sagging as the inclination changes $(x=0)$ (compare Fig.(3.5)). The first partial derivative f_x shows some over- and undershooting as can be seen in Fig.(3.6) while the peak in the second partial derivative f_{xx} is much lower than in the previous reconstruction (compare Fig.3.7) resulting in a less sudden change in slope. (Again the plots of the derivatives are rotated counter-clockwise with respect to the reconstructed surface.)

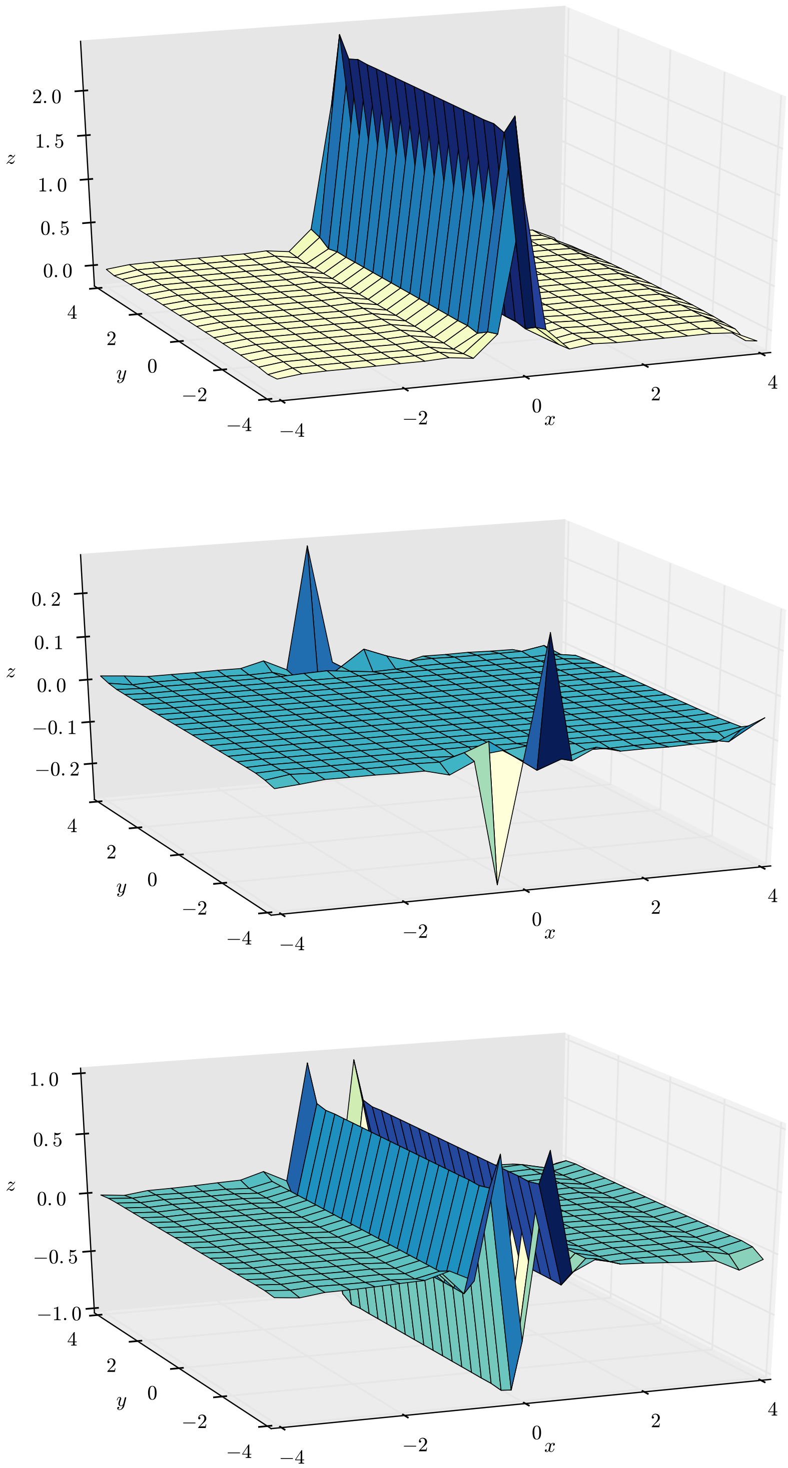


FIGURE 3.7: Second order partial derivatives (top: f_{xx}, middle: f_{xy} bottom: f_{yy}) of the reconstructed surface using the Tension Spline.

In figure (3.4) the absolute error between the reconstruction and piecewise-linear surface is shown. It has a maximum of about 0.03 where this error is concentrated around $x = 0$, that means in the region of changing curvature. For the rest of the surface, the error is approximately *zero*.

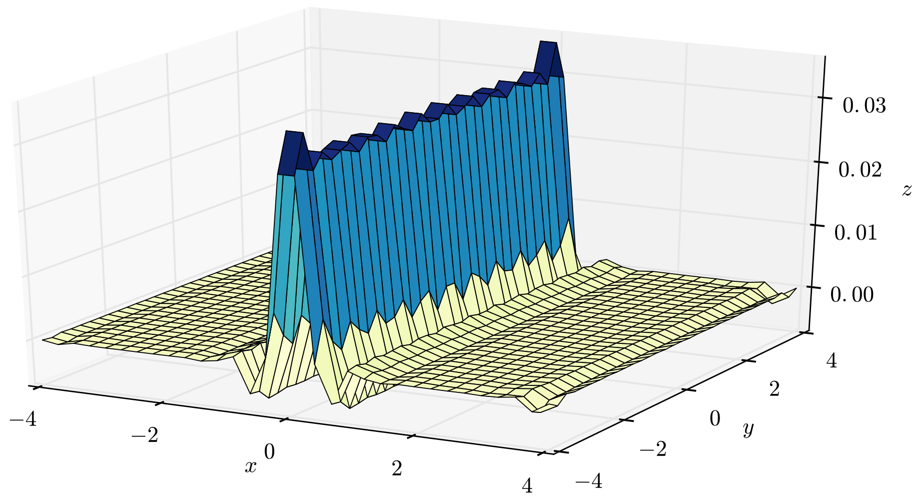


FIGURE 3.8: Difference to the original surface.

3.4 The inverse multi-quadratic reconstruction

The third reconstruction investigated here is not obtained from penalty function but uses a multi-quadratic approach for the ansatz-function. Therefore the reconstructing function is defined as follows:

$$f_k(r) := \frac{1}{\sqrt{1+r_k^2}}.$$

Due to the definition of $f_{\mathrm{imq},k}$, its function-values become smaller (going closer to *zero*) with increasing distance of $\mathbf{x}$ to the origin $\mathbf{x}_k$ and therefore the reconstruction can be described as being more local in some sense than others (that will follow later). We now find for the partial derivatives f_x^k, f_{xx}^k and f_{xy}^k:

$$\begin{aligned} f_x^k &= -\frac{(x-x_k)}{\sqrt{1+r_k^2}^3}, \\ f_{xx}^k &= \frac{3(x-x_k)^2-(1+r_k^2)}{\sqrt{(1+r_k^2)}^5}, \\ f_{xy}^k &= \frac{3(x-x_k)(y-y_k)}{\sqrt{1+r_k^2}^5}. \end{aligned}$$

f_y and f_{yy} follow analogously.

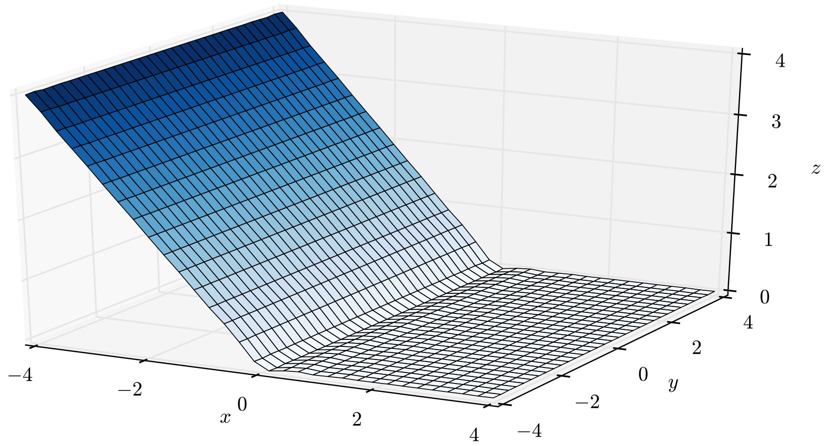


FIGURE 3.9: Reconstructed surface using the Inverse Multi-quadratic spline.

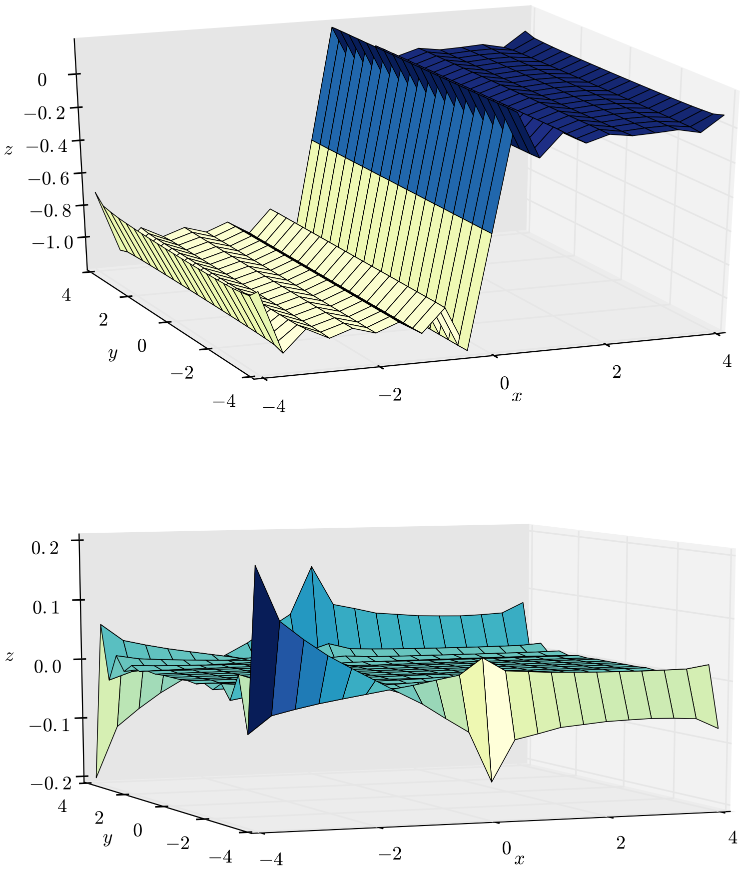


FIGURE 3.10: First order partial derivatives (top: f_x, bottom: f_y) of the reconstructed surface using the inverse multi-quadratic spline.

The reconstructed function is now shown in Fig.(3.9). It is basically able to take on the desired shape but however shows some sacking in vicinity $x = 0$. While, besides at the borders, f_y is quite constant, there are strong oscillations in f_x as can be seen in Fig.(3.10). In Fig.(3.11) the three second-order partial derivatives are depicted where f_{xx} shows a strong peak around $x = 0$ as well as significant oscillations the closer one gets to that point. (Note that for a better visibility the plots of the derivatives are rotated counter-clockwise with respect to the reconstructed surface.)

Figure (3.12) shows the error between the reconstruction and the original surface. The maximum absolute error is about 0.04 while the average error is about 0.01. It is the only reconstruction that shows errors to the original surface also in the originally planar areas.

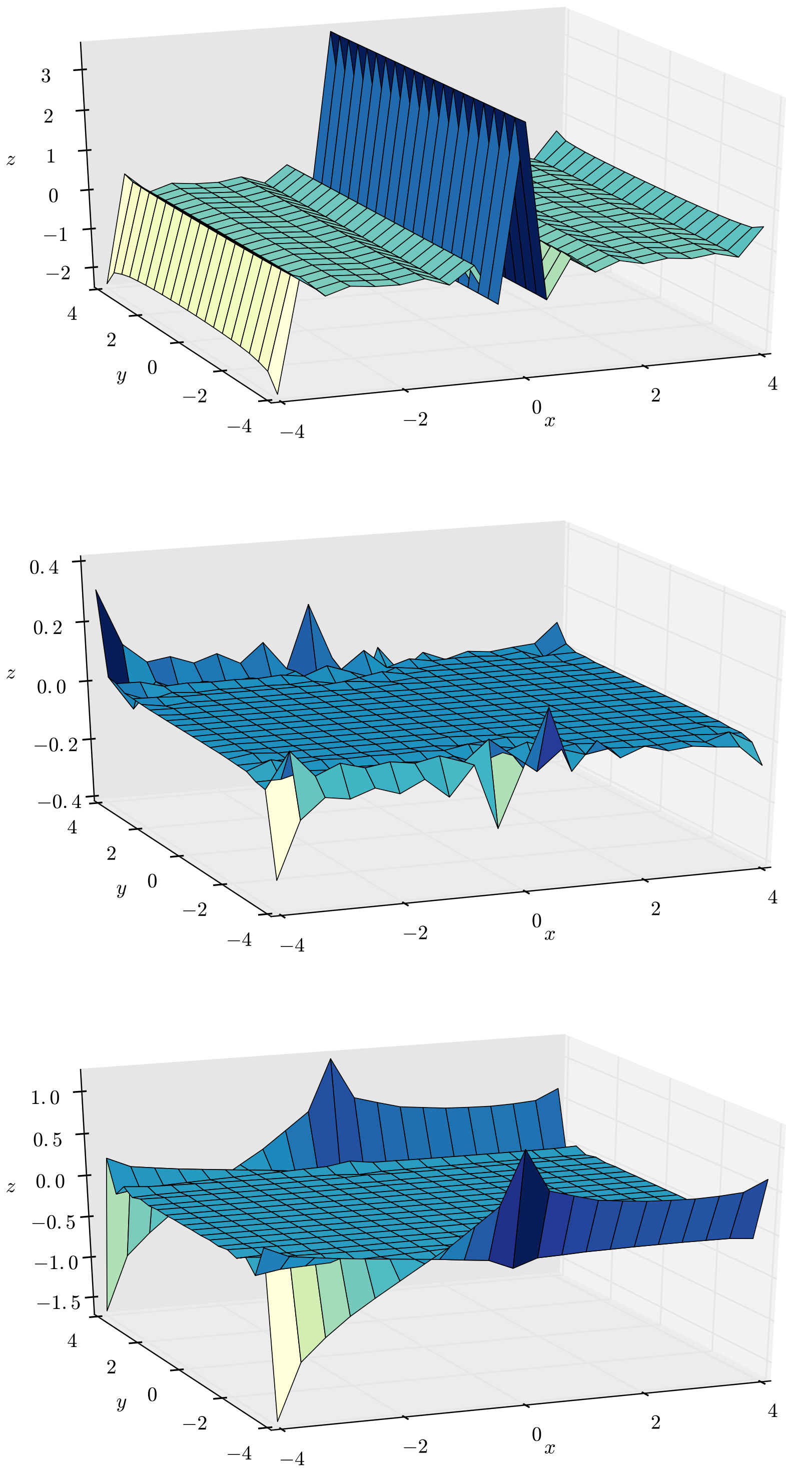


FIGURE 3.11: Second order partial derivatives (top: f_{xx}, middle: f_{xy} bottom: f_{yy}) of the reconstructed surface using the inverse multi-quadratic spline.

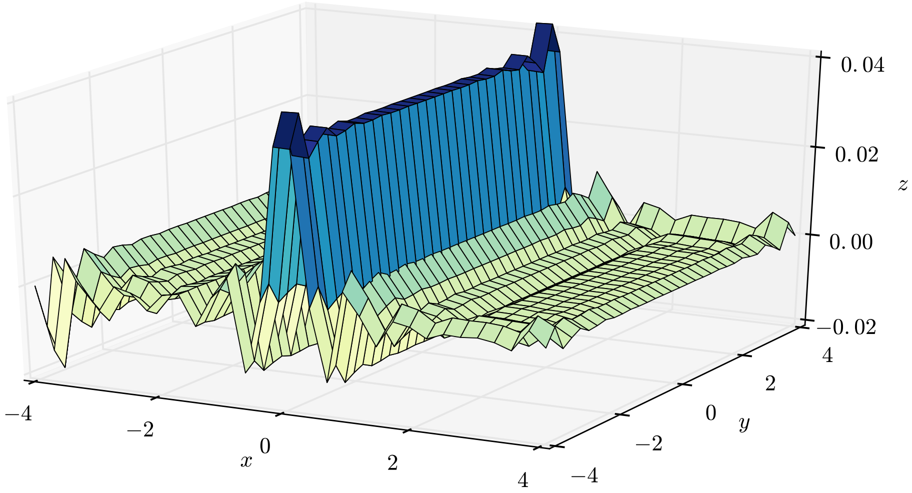


FIGURE 3.12: Difference to the original surface.

3.5 Time-integration using the Livermore solver

As pointed out in section 2.5, the equations of motion are generally a stiff problem and therefore a solver suited for such problems has to be used in order to obtain reasonable good results in the numerical solution. Among those solvers, the Livermore Solver for Ordinary Differential Equations (LSODE) Hindmarsh and Radhakrishnan, 1993 is a frequently used and highly optimized package of Fortran subroutines designed for the numerical solution of the initial value problem for a system of ordinary differential equations of the form

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}, t),$$

with initial conditions $\mathbf{x}(t_0) = \mathbf{x}_0$. While many of the classical solvers are for non-stiff ODEs only, LSODE is particularly well suited for stiff differential systems but can be used for non-stiff problems as well. Moreover, it contains different mechanisms that minimize computational costs.

Depending on the users needs, different options are available for the time-integration using LSODE where most methods require a Jacobian-matrix (i.e. the matrix of the partial derivatives), which can be either user-supplied or generated internally.

Adams-Moulton (AM) method: , best suited for non-stiff problems, no Jacobian needed;

BDF (backward differentiation formula) method: , best suited for stiff problems, Jacobian can be supplied as follows[1]:

- "Modified Newton iteration with user-supplied analytical Jacobian",
- "Modified Newton iteration with internally generated numerical Jacobian",
- "Modified Jacobi-Newton iteration with internally generated numerical Jacobian",
- "Modified Newton iteration with user-supplied banded Jacobian",
- "Modified Newton iteration with internally generated banded Jacobian".

Beyond that, the most important arguments of LSODE are the following[2]:

F: "The name of the user-supplied subroutine that computes the derivatives of the dependent variables with respect to the independent variable";

NEQ: "The number of first-order ordinary differential equations (ODE's) to be solved";

[1] compare Hindmarsh and Radhakrishnan, 1993
[2] compare Hindmarsh and Radhakrishnan, 1993

Y: "A vector of length NEQ (or more) containing the dependent variables";

T: "The independent variable. On the first call to LSODE, T must give the initial value of this variable";

RTOL: "The local relative error tolerance parameter for the solution";

ATOL: "The local absolute error tolerance parameter for the solution".

With the last two arguments, RTOL and ATOL, the following local error test is performed at each iteration step:

$$|(\text{local error in } x(i))| \leq \text{RTOL} \cdot |(y(i)) + \text{ATOL}(i)|$$

As for the reconstruction methods presented above, an analytically computed Jacobian usually becomes too complex, the stiff method with the internally generated full Jacobian was used for the numerical solution of the equations on the reconstructed surfaces. The results of those computations will – for the 1-dimensional case – be discussed in the next chapter where they are also compared to the solutions on piecewise linear curves. Accordingly, the results for the 2-dimensional case will be discussed and compared in the next chapter, dedicated to the solution of the equations on piecewise-planar surfaces.

Chapter 4

Analytical Solution of the 1D Equations of Motion on Piecewise-Linear Curves

4.1 Introduction and formulation of the problem

In this section a method for the analytical solution of the equations of motion on C_0-curves is derived and while the final goal is to simulate gravitational sliding on real terrains (topographies and/or bathymetries), this will first be done for the 1-dimensional problem i.e., a point mass sliding under the effect of gravity on curves made of piecewise linear segments. But, just as well as in the case of triangulated surfaces, the piecewise linear curves are non-smooth at the intersection point of the segments. The aim is to present a technique providing the analytical solution of the equations governing the point-mass motion on these curves. Once an analytical (here also exact) solution – which obviously guarantees energy conservation in case of *zero* friction – to this problem is found, the method is extended to the 2-dimensional case.

For a better illustration of our approach let us suppose that the terrain is given by a curve $\mathcal{C}$ that is at least of class C^2, and suppose further that we approximate $\mathcal{C}$ by a continuous curve Σ formed by straight segments, that may have discontinuous first derivative at the segment end points (or nodes). In order to overcome the non-smoothness problem of the piecewise curve Σ, we imagine (as indicated in Fig. 4.1) that in the neighbourhood of a node the curve is smoothed by a circular arc thus implying a constant curvature in this region. Note that in the figure, the proportions are strongly scaled in order to illustrate the idea. In the following, the sliding curve is denoted by Σ. Two consecutive linear segments of Σ are denoted by σ_i and σ_{i+1}, where each segment σ_i can be described by a function f_i of x:

$$f_i(x) = a_i x + b_i \quad \text{with} \quad x \in A_i \subset \mathbb{R}$$

where a_i and b_i are real coefficients, A_i is the range of x-values for the element σ_i

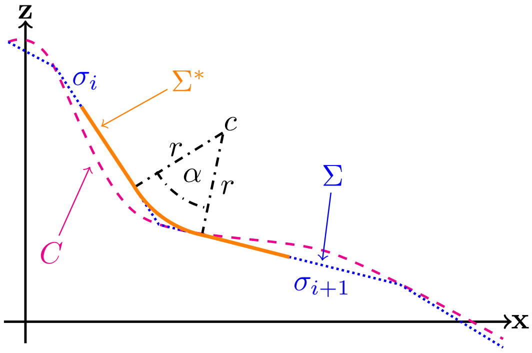


FIGURE 4.1: Curve approximation using straight segments and circular sectors: Smooth curve $\mathcal{C}$ (dashed, magenta), discretized curve Σ with linear segments σ_i and σ_{i+1} (densely dotted, blue) and smoothed curve Σ^* (orange, dashed). The circular sector is denoted by dash-dotted lines, r is the radius of the circular sector, α its angle and c the centre

and the index i is increasing with x. As mentioned above we want to smooth the discontinuities using a circular arc such that the derivatives at the beginning and end of the arc coincide with those of the linear segments. This is achieved if the radius r of the circular sector is normal to the linear segment itself in the point where they meet (compare also Fig. 4.1). It easily follows that the angle defining the circular sector is identical to the one enclosed by the two linear segments and so we found a simple and general criterion for choosing the angle of the circular arc. This angle will be denoted by α (see also Fig. 4.3). The smoothed curve will be denoted by Σ^* and is a C^1-function meaning that it is once continuously differentiable and therefore also has a second derivative. Note that as we did not construct a C^2-function, the second derivative is still not continuous itself. In general this would be a problem, however, in our case it is not as we will see later.

4.2 The solution for the piecewise linear curve Σ

4.2.1 The equations of motion

For the 1-dimensional case, i.e. the motion of a mass point on a curve $\mathcal{C}$, the following first order ODE system can be derived from Eq. (2.10):

$$\dot{\mathbf{y}}(t) = \frac{d\mathbf{y}}{dt} = \mathbf{F}(\mathbf{y},t), \quad \mathbf{y}(t_0) = \mathbf{y}_0, \tag{4.1}$$

where $\mathbf{y} = (x, v_x)$ and

$$\mathbf{F} = \begin{pmatrix} v_x \\ -f_x \dfrac{g + v_x^2 f_{xx}}{1+f_x^2} - \mu \,\mathrm{sgn}(v_x) \dfrac{g + v_x^2 f_{xx}}{1+f_x^2} \\ v_z \\ \dfrac{-f_x^2 g + v_x^2 f_{xx}}{1+f_x^2} - \mu \,\mathrm{sgn}(v_z) \dfrac{g + v_x^2 f_{xx}}{1+f_x^2} \end{pmatrix}. \tag{4.2}$$

Furthermore, $f = f(x)$ denotes the function describing the curve $\mathcal{C}$, while $f_x = f'(x)$ and $f_{xx} = f''(x)$ are the corresponding first and second derivatives, that supposedly exist everywhere along the curve $\mathcal{C}$.

As we assumed that the point mass is constrained to move on the curve, we can also obtain the height z and velocity v_z by evaluating

$$z = f(x) \qquad v_z = v_x f_x.$$

In the following the solution of (4.2) for the approximation Σ^* of the real curve $\mathcal{C}$ is described. While the solution we derive for $\Sigma*$ will be an analytical one, it will also be an approximation to the unknown solution on the corresponding C^2-curve $\mathcal{C}$. For a better understanding of the following steps, it is helpful to rewrite $\mathbf{F}$ as follows:

$$\mathbf{F} = \begin{pmatrix} v_x \\ -g \dfrac{f_x + \mu \,\mathrm{sgn}(v_x)}{1+f_x^2} - v_x^2 f_{xx} \dfrac{f_x + \mu \,\mathrm{sgn}(v_x)}{1+f_x^2} \\ v_z \\ -g \dfrac{f_x^2 + \mu \,\mathrm{sgn}(v_z)}{1+f_x^2} - v_x^2 f_{xx} \dfrac{1 + \mu \,\mathrm{sgn}(v_z)}{1+f_x^2} \end{pmatrix}. \tag{4.3}$$

While the left term of the rhs of equ. (4.3) contains g but does not depend on f_{xx} and therefore the curvature of f, the contrary is the case for the right term. Therefore, the left term will in the following also be denoted as "curvature-independent term", while the right term will be denoted as "curvature-dependent term". The main idea behind the subsequent steps is the following: if we let the radius r of the arc segments go to *zero*, also the length of the curve of those segments and therefore the time t needed to pass through the segments go to *zero*. Hence, the terms including gravity (those are at the same time the curvature-independent term) are become negligible. On the contrary, this is not the case for all curvature-dependent term which is shown in the following. As a consequence, in the transition area the problem reduces to the solution of the curvature-dependent terms while along the linear segments the curvature-dependent are *zero* (as the second derivatives are *zero*) and the problem reduces to the solution of the curvature-independent terms.

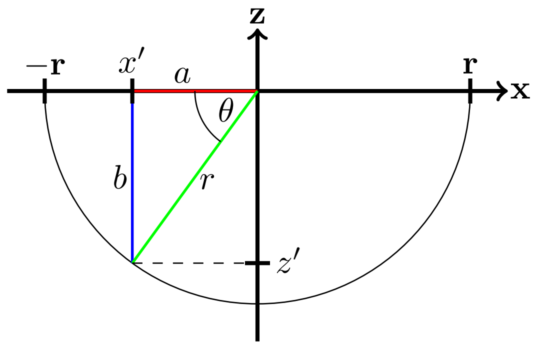


FIGURE 4.2: Half circle of radius r with argument x' and resulting value z' and angle θ.

Proof. The lower half of the circle is given by the following function:

$$f(x) = -\sqrt{r^2 - x^2}$$

restricted to the domain $x \in [-r, r]$, where r is the radius of the circle (cf. fig. 4.2). The first and second derivatives of f are:

$$f' = \frac{x}{\sqrt{r^2 - x^2}} \qquad \text{and} \qquad f'' = \frac{r^2}{(r^2 - x^2)^{3/2}}.$$

First it will be shown that the modulus of all curvature-independent terms of equ. (4.2) is always bounded. Notice therefore that the parameters g and mu as well as the velocities v_x and v_z are always bounded and will therefore not directly be considered in the following.

For the first term

$$\begin{pmatrix} -\frac{f_x g}{1+f_x^2} \\ -\frac{f_x^2 g}{1+f_x^2} \end{pmatrix} \tag{4.4}$$

we find:

$$\frac{f_x}{1+f_x^2} = \frac{\frac{x}{\sqrt{r^2-x^2}}}{\frac{r^2}{r^2-x^2}} = \frac{x}{r}\frac{\sqrt{r^2-x^2}}{r}, \tag{4.5}$$

where $\sqrt{r^2-x^2} = z$ and therefore $\dfrac{\sqrt{r^2-x^2}}{r} = \cos(\theta)$. Obviously, the values of $\cos(\theta)$ are in the range $[-1, 1]$. On the other hand, for the given domain $[-r, r]$ we have $-1 \leq \dfrac{x}{r} \leq 1$. It follows that for all values in the defined domain equ. (4.5) is bounded by $[-1,1]$. Moreover one finds that:

$$\frac{f_x^2}{1+f_x^2} = \frac{\dfrac{x^2}{r^2-x^2}}{\dfrac{r^2}{r^2-x^2}} = \frac{1}{r^2}x^2, \tag{4.6}$$

which is a parabola with parameter $\frac{1}{r^2}$. As the domain is restricted to $[-r,r]$ we find that (4.6) has minimum value *zero* and maximum value *one*. As therefore both components of (4.4) are bounded, also its modulus must be bounded. (More easily this is seen from a physical point of view as acceleration due to gravity can never exceed g (free fall).)

For the second term

$$\begin{pmatrix} \mu \operatorname{sgn}(v_x)\dfrac{g}{1+f_x^2} \\ \mu \operatorname{sgn}(v_z)\dfrac{g}{1+f_x^2} \end{pmatrix}. \tag{4.7}$$

We find that

$$\frac{1}{1+f_x^2} = \frac{1}{1+\dfrac{x^2}{r^2-x^2}} = 1-\left(\frac{x}{r}\right)^2.$$

As in the given domain $-1 \leq \dfrac{x}{r} \leq 1$ it follows that $0 \leq \left(\dfrac{x}{r}\right)^2 \leq 1$ and therefore $0 \leq 1-\left(\frac{x}{r}\right)^2 \leq 1$. Thus, both components of term (4.7) and therefore also its modulus are bounded. (Physically this is seen as this friction term is proportional to acceleration due to gravity which itself is bounded.)

It will now be shown that the curvature-dependent components are going to $\pm\infty$ for $r \to 0$.

For the third term the modulus is computed directly:

$$\left\| \begin{pmatrix} \dfrac{-f_x f_{xx}}{1+f_x^2} \\ \dfrac{f_{xx}}{1+f_x^2} \end{pmatrix} \right\| = \left(\frac{f_{xx}}{1+f_x^2}\right)^2 (f_x^2+1),$$

where for the left term of the rhs we have:

$$\frac{f_{xx}}{1+f_x^2} = \frac{\dfrac{r^2}{(r^2-x^2)^{3/2}}}{1+\dfrac{x^2}{r^2-x^2}} = \frac{r^2-x^2}{(r^2-x^2)^{3/2}} = \frac{1}{\sqrt{r^2-x^2}}, \tag{4.8}$$

where for the domain $[-r,r]$ the values of r^2-x^2 are in the range $[0,r]$ and therefore

$$\lim_{r\to 0}\frac{1}{\sqrt{r^2-x^2}}=\infty \qquad \text{and thus also} \qquad \lim_{r\to 0}\left(\frac{f_{xx}}{1+f_x^2}\right)^2=\infty.$$

On the other hand

$$f_x^2+1=\frac{x^2}{\sqrt{r^2-x^2}}+1=\frac{r^2}{r^2-x^2} \tag{4.9}$$

is a family of curves that, in the given range, has a single minimum for $x=0$ where $f(0)=1$, and two poles, one for $x=-r$ and one for $x=r$ with $\lim_{x\to\pm r}=\infty$. Accordingly, the values of (4.9) are in the range $[1,\infty]$ such that finally we find:

$$\lim_{r\to 0}\left(\frac{f_{xx}}{1+f_x^2}\right)^2(f_x^2+1)=\infty.$$

For the last term

$$\begin{pmatrix}\frac{f_{xx}}{1+f_x^2}\\ \frac{f_{xx}}{1+f_x^2}\end{pmatrix},$$

we already know from (4.8) that the limit of the components is ∞.

As the curvature-dependent components are going to ∞ as $r\to 0$ while the curvature-independent terms are bounded, the latter are negligible in that case. □

For simplicity the curvature-independent and -dependent terms are written separately and will be denoted by $\mathbf{F}_g$ and $\mathbf{F}_c$, respectively, such that the solution of (4.1) reduces to

$$\dot{\mathbf{y}}(t)=\frac{d\mathbf{y}}{dt}=\mathbf{F}_g(\mathbf{y},t), \quad \mathbf{y}(t_0)=\mathbf{y}_0,$$

for the linear segments σ_i, where

$$\mathbf{F}_g=\begin{pmatrix} v_x\\ -g\frac{f_x}{1+f_x^2}-\mu\,\mathrm{sgn}(v_x)\frac{g}{1+f_x^2}\end{pmatrix} \tag{4.10}$$

and the subscript g, indicates gravity dependency. For the circular segments instead the problem reduces to

$$\dot{\mathbf{y}}(t)=\frac{d\mathbf{y}}{dt}=\mathbf{F}_c(\mathbf{y},t), \quad \mathbf{y}(t_0)=\mathbf{y}_0,$$

where

$$\mathbf{F}_c = \begin{pmatrix} v_x \\ \frac{-f_x}{1+f_x^2} v_x^2 f_{xx} - \mu\,\mathrm{sgn}(v_x) \frac{1}{1+f_x^2} v_x^2 f_{xx} \end{pmatrix}. \tag{4.11}$$

and the subscript c denotes the curvature-dependency. In Eq. (4.11) one can distinguish a term that acts normal to the velocity vector and changes its direction but not its modulus, namely $\frac{-f_x}{1+f_x^2} v_x^2 f_{xx}$, and a friction term that acts tangential to the direction of the velocity and changes the speed but does not influence the direction of motion, that is $-\mu\,\mathrm{sgn}(v_x)\frac{1}{1+f_x^2} v_x^2 f_{xx}$. This property is easily seen using also the vertical acceleration a_z. Considering – for the 1-dimensional case – only curvature-dependent components, Eq. (2.10) becomes:

$$\begin{aligned} \dot{v}_x &= -\frac{f_x}{1+f_x^2} v_x^2 f_{xx} - \mu\,\mathrm{sgn}(v_x) \frac{f_{xx} v_x^2}{1+f_x^2}, \\ \dot{v}_z &= \frac{1}{1+f_x^2} v_x^2 f_{xx} - \mu\,\mathrm{sgn}(v_x) \frac{f_x f_{xx} v_x^2}{1+f_x^2}. \end{aligned} \tag{4.12}$$

Denoting by $\mathbf{a}_n$ and $\mathbf{a}_t$ the left and right part of Eq. (4.12) we find:

$$\mathbf{a}_n = \begin{pmatrix} -\frac{f_x}{1+f_x^2} v_x^2 f_{xx} \\ \frac{1}{1+f_x^2} v_x^2 f_{xx} \end{pmatrix} = \frac{v_x^2 f_{xx}}{1+f_x^2} \begin{pmatrix} -f_x \\ 1 \end{pmatrix},$$

where the vector $(-f_x, 1)^\mathsf{T}$ and therefore $\mathbf{a}_n$ are normal to the curve and thus to the direction of motion. For $\mathbf{a}_t$ instead we find:

$$\mathbf{a}_t = \begin{pmatrix} -\mu\,\mathrm{sgn}(v_x) \frac{f_{xx} v_x^2}{1+f_x^2} \\ -\mu\,\mathrm{sgn}(v_x) \frac{f_x f_{xx} v_x^2}{1+f_x^2} \end{pmatrix} = \frac{-\mu\,\mathrm{sgn}(v_x) f_{xx} v_x^2}{1+f_x^2} \begin{pmatrix} 1 \\ f_x \end{pmatrix},$$

where $(1, f_x)^\mathsf{T}$ and therefore also $\mathbf{a}_t$ are tangential to the surface and thus also to the direction of motion.

Observe that Eq. 4.11 does not depend on gravity, and therefore would be sufficient to describe the motion of the point mass on a curve lying on a horizontal plane.

Starting with Eq. 4.10 we will study the two types of motion (i.e gravity-dependent and curvature-dependent) separately. Later on we will see how they can be combined together. We anticipate that our strategy will be to compute the motion on the piecewise curve Σ and to recognize that the motion along the straight segments of Σ is fully governed by the gravity-dependent component of the equations, while in the transition from one segment to the next, the governing contribution comes from the curvature-dependent component.

4.2.2 The motion on the straight segments of Σ

Along a straight segment with constant slope the second derivative of the profile function f_{xx} is identically zero and therefore the motion is fully determined by the curvature-independent components.The acceleration $a_{x,0}$ is given by:

$$a_{x,0} = -g\frac{f_x}{1+f_x^2} - \mu\,\mathrm{sgn}(v_{x,0})\frac{g}{1+f_x^2}, \tag{4.13}$$

which we get from 4.10. From the definition of Σ we find that for the certain linear segment σ_i we have $f_x = a_i$ and therefore:

$$a_{x,0} = -g\frac{a_i}{1+a_i^2} - \mu\,\mathrm{sgn}(v_{x,0})\frac{g}{1+a_i^2} = const., \tag{4.14}$$

There is no loss of generality if we assume that the mass of the point is unitary, i.e. that $m = 1$. Since the driving acceleration $a_{x,0}$ is constant (c.f. (4.14)), the solution corresponding to the initial conditions $(x(t_0), v_x(t_0)) = (x_0, v_{x,0})$ is:

$$x(t) = x_0 + v_{x,0}(t-t_0) + \frac{1}{2}a_{x,0}(t-t_0)^2,$$

and

$$v_x(t) = v_{x,0} + a_{x,0}(t-t_0), \tag{4.15}$$

allowing us to compute the exact position and velocity of the point for any time instant of interest. Assuming we know the final position $x_t = x(t)$ we can also solve for t:

$$t_{1,2} = t_0 + \frac{-v_{x,0} \pm \sqrt{v_{x,0}^2 - 2a_{x,0}(x_0 - x_t)}}{a_{x,0}}, \tag{4.16}$$

Three cases are possible:

- Both values $t_{1/2} - t_0$ are positive, i.e. the mass-point will cross x_t twice. This is the case if the direction of motion changes, but outside the current triangle. Starting with velocity $v_z > 0$, the point moves upwards, crosses x_t for the first time at $t = t_1$ and stops (i.e. $v_z = 0$) somewhere outside the current segment σ_i at time $t_s \in [t_1, t_2]$. Then it continues in a downwards motion (i.e. $v_z < 0$), crossing x_t for the second time at $t = t_2$, re-entering element σ_i.

- One values of $t_{1/2} - t_0$ is positive, the other negative. Either the mass is moving with negative v_z from the beginning, or, the motion changes direction within the current segment σ_i. In both cases, the negative value of $t_{1/2} - t_0$ (from a theoretical point of view) accounts for going backwards in time which from a physical point of view is impossible. The only valid value of t is the positive one.

- Both values $t_{1/2} - t_0$ are negative. The mass-point is moving with velocity $v_z < 0$ (downwards) and already passed by the point x_t. It could return to that point only moving backwards in time which is excluded.

Letting x_l and x_r be the left and right endpoint of a linear segment σ_i, Eq. (4.16) has to be evaluated twice, once for each of the two endpoints. For $x_t = x_l$ and $x_t = x_r$ altogether four values t_i are found where, without loss of generality, it can be assumed that $t_{1/2}$ are the solutions for $x_t = x_l$ and $t_{3/4}$ the solutions for $x_t = x_r$, respectively. The final time of intersection t_{int} is given by:

$$t_{int} = \min(t_i - t_0 | t_i - t_0 > 0),\ i = 1, \ldots, 4,$$

i.e. the smallest positive solution of Eq. (4.16). Since it is trivial to derive the vertical components of the motion vector (i.e. acceleration, velocity and position) from the horizontal ones and to see that both acceleration and velocity are vectors parallel to the segments, the above simple formula enable us to fully compute the motion of the point mass along each linear segment, just adapting its initial velocity and acceleration.

Special case: the motion stops In this model we consider only a constant friction coefficient μ. From Eq.s (4.15) and (4.13) one finds that the motion of a point can stop if either $\operatorname{sgn}(a_x) \neq \operatorname{sgn}(v_x)$ or $\mu \geq |f_x|$, or also both are the case. In this case t_{stop}, the time at which a point stops, is computed as follows:

$$\begin{aligned} & v_x(t_{stop}) = 0 = v_{x,0} + a_{x,0} t_{stop} \\ \Rightarrow \quad & t_{stop} = -\frac{v_{x,0}}{a_{x,0}} \end{aligned}$$

If the point stops but $\mu < |f_x|$ then the point reached a reversal point (point of maximal height). As the slope is steep enough to overcome friction, the point will continue moving, but velocity will change its sign (the point then moves downwards). If in contrast $\mu \geq |f_x|$ the slope is too slight and the point finally stops.

4.2.3 The motion in the transition region between the linear segments of Σ

Strictly speaking, in any node connecting two linear segments of the approximating curve Σ the second derivative f_{xx} does not exist and Eq. 4.2 cannot be applied. In order to overcome this, we consider instead a smoothed version of the curve Σ that we call Σ^* that is obtained by using circular sectors. Accordingly, when the point approaches the point of intersection of two linear segments it undergoes a circular

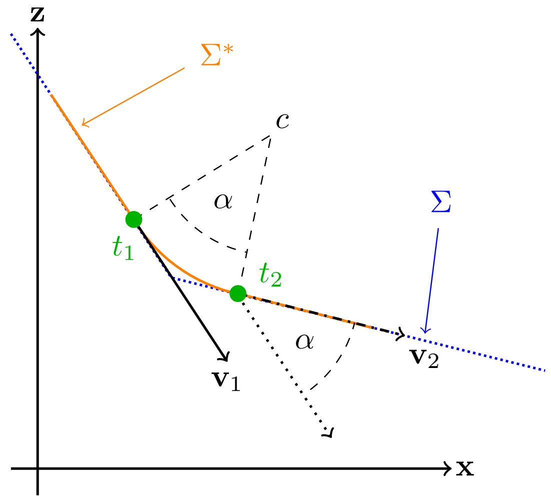


FIGURE 4.3: Rotation of the velocity vector: angle of rotation of the point equals the change of direction of the velocity vector

motion which enables us to incorporate centripetal acceleration and friction. We will show that in the transition region the motion of the mass is dominated only by the Eq. 4.11 that is notably g-independent. Let's suppose that this is true, and let's consider separately the contribution of the term without and with friction in this equation. In Fig. 4.3 we show (strongly scaled up) how the direction of motion is changing during the passage of the point through a circular sector with radius r, centre c and angle α. Displayed are the discretized surface Σ and smoothed surface Σ^*, the circular sector of angle α (dashed lines), the position of a point when entering the arc at time t_1 and on exit at time t_2 (marked by green dots) and the corresponding velocities $\mathbf{v}_1$ and $\mathbf{v}_2$ (marked respectively by a solid and dash-dotted arrows). The translation of $\mathbf{v}_1$ into the point of exit (dotted arrow) is given to indicate the rotation (α) of the velocity vector. It can easily be shown that the angle of the circular arc equals the change of direction of the velocity vector.

If only the frictionless terms are taken into account, there is no loss of kinetic energy, and the magnitude of the velocity is expectedly conserved in the transition, and, as the point is forced to stay on the curve Σ^*, it follows that $\mathbf{v}_1$ and $\mathbf{v}_2$ must both be tangent to the curve. Therefore the angle enclosed by the two linear segments σ_{i-1} and σ_i must be equal to the one enclosed by $\mathbf{v}_1$ and $\mathbf{v}_2$. It follows that the vector $\mathbf{v}_2$ can be obtained by a simple rotation of $\mathbf{v}_1$, i.e.:

$$\mathbf{v}_2 = \mathbf{R}_\alpha \mathbf{v}_1, \tag{4.17}$$

where $\mathbf{R}_\alpha$ is the well known rotation matrix and α denotes the angle of rotation. Note that this does not depend on the radius and therefore the length of the arc but only on α itself. This property is crucial and will be exploited later.

4.2.4 Introducing friction

Let $\mathbf{F}$ be the friction force and $\mathbf{\Phi}$ the reaction of the surface exerted on the body. From the motion equation:

$$m\mathbf{a} = \mathbf{F}_\mu + \mathbf{\Phi},$$

in case of gravity-independent circular motion of a point with mass m and velocity $\mathbf{v}$ along an arc of radius r, we have:

$$\mathbf{F}_\mu + |\Phi_n|\,\mathbf{n} = m\dot{v}\mathbf{t} + \frac{mv^2}{r}\mathbf{n},$$

and:

$$\mathbf{F}_\mu = -\operatorname{sgn}(v)\mu\,|\Phi_n|\,\mathbf{t}, \tag{4.18}$$

where $\mathbf{t}$ and $\mathbf{n}$ are unit vectors respectively tangential and normal to the curve. It follows that:

$$-\operatorname{sgn}(v)\mu\,|\Phi_n|\,\mathbf{t} + |\Phi_n|\,\mathbf{n} = m\dot{v}\mathbf{t} + \frac{mv^2}{r}\mathbf{n},$$

and therefore we find:

$$-\operatorname{sgn}(v)\mu\,|\Phi_n| = m\dot{v}, \qquad \text{and} \qquad |\Phi_n| = \frac{mv^2}{r},$$

$$\Rightarrow -\operatorname{sgn}(v)\mu\frac{mv^2}{r} = m\dot{v}.$$

Finally we end up with:

$$\dot{v} = -\operatorname{sgn}(v)\mu\frac{v^2}{r}, \tag{4.19}$$

Hence, if we want to understand how the speed of a point mass moving along a (horizontal) circular path evolves in time, we need to solve equation (4.19). We imagine for the moment that the point is moving inside the circle which hence is a convex curve. As radius and friction coefficient are constant we substitute for convenience $\frac{\mu}{r}$ with ξ. With $v(0) = v_0$ and $\operatorname{sgn}(v) = \operatorname{sgn}(v_0)$ (as now the speed can only decrease) we find:

$$v(t) = \frac{v_0}{1 + \operatorname{sgn}(v_0)v_0\xi t}. \tag{4.20}$$

Note that $v(t)$ has no singularity as $|v_0|$, ξ and t are always positive and for $t \to \infty$ we find that $v \to 0$. Obviously, for $v_0 = 0$ also $v(t) = 0$ for all values of t. This can be treated as a special case. We assume therefore in the following that $v_0 \neq 0$ in order to ensure that $\operatorname{sgn}(v_0)$ is always different from *zero*.

The goal of the following computations is to find the final velocity v_f of a point P after travelling through a particular circular arc with angle α. Therefore we first need to find the function describing the angle through which a point passed through for a certain time t. This function will be denoted by $\beta(t)$.

In a first step we convert the velocity $v(t)$ into the angular velocity $\omega(t)$ by multiplying $v(t)$ by the inverse of the radius. In the general case the following relation holds $\omega = d\beta/dt$ and thus we find β by integrating $\omega(t)$ in time:

$$\beta(t) = \int_0^t \omega(t')dt' = \int_0^t \frac{v_0}{r(1+\mathrm{sgn}(v_0)v_0\xi t')}dt',$$

As $(1+\mathrm{sgn}(v_0)v_0\xi t') = (1+|v_0|\xi t') > 0$, the integral can be solved using the rule of logarithmic integration:

$$\begin{aligned}\beta(t) &= \int_0^t \frac{\mathrm{sgn}(v_0)}{\xi r}\frac{\mathrm{sgn}(v_0)v_0\xi}{1+\mathrm{sgn}(v_0)v_0\xi t'}dt' = \\ &= \frac{\mathrm{sgn}(v_0)}{\xi r}\ln(1+\mathrm{sgn}(v_0)v_0\xi t) = \\ &= \frac{\mathrm{sgn}(v_0)}{\xi r}\ln(1+|v_0|\,\xi t)\end{aligned}$$

If t is a particular time that we are interested in, then obviously $\beta(t)$ is the corresponding angle. As $(|(|v_0)\xi t) > 0$ it follows that also $\ln(1+|(|v_0)\xi t) > 0$ and as also $\xi r > 0$ one finds that $\beta(t) > 0$ if $v_0 > 0$ and on the other hand $\beta(t) < 0$ if $v_0 < 0$. Inverting the function we find the time t needed by a point with initial velocity v_0 to move along a circular arc of angle β, defined by the radius r:

$$t(\beta) = \frac{e^{(\mathrm{sgn}(v_0)\beta\xi r)}-1}{\mathrm{sgn}(v_0)v_0\xi}. \tag{4.21}$$

Inserting (4.21) into (4.20), and back-substituting ξr with μ we find the final velocity in dependency of the angle β:

$$v(\beta) = \frac{v_0}{e^{\mathrm{sgn}(v_0)\mu\beta}} = \frac{v_0}{e^{\mu|\beta|}}, \tag{4.22}$$

as $\beta > 0$ for positive initial velocities and $\beta < 0$ in the opposite case. This final velocity will also be denoted by v_f.

It follows that in case of circular motion with constant radius r and in absence of additional tangential acceleration (e.g. due to gravity) the change of speed due to friction is determined only by the initial velocity, the friction coefficient and the angle of the circular arc, but – as well as the change of direction – it is independent of the radius itself and also of time. This radius- and time-independence enables us to reduce the influence of the curvature-dependent components to the point of change of slope as no time-integration is needed for its computation. We can therefore also assume that the change of direction and modulus of velocity in this point is an instantaneous one.

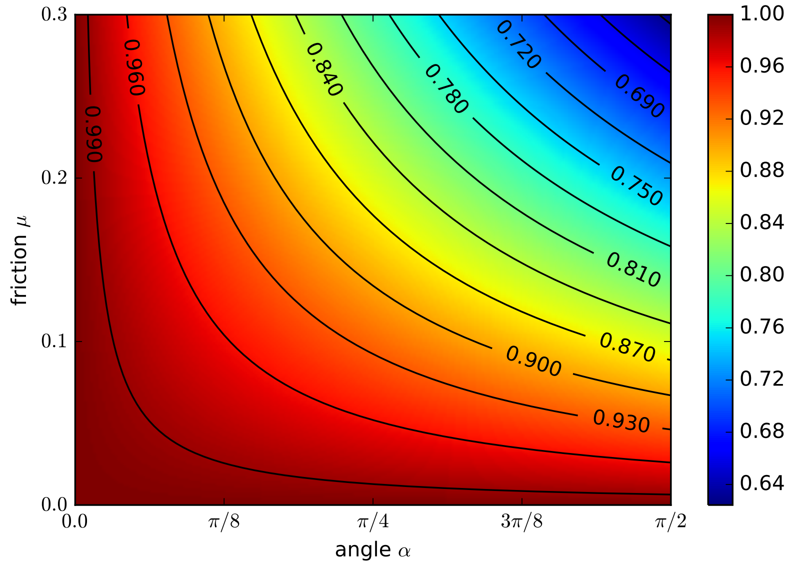


FIGURE 4.4: Velocity modification factor Γ for convex curves, plotted as a function of the friction coefficient μ and the angle of the circular arc α

We assume now a given angle α and friction coefficient μ. As the final velocity only depends on the angle and friction coefficient we can also write:

$$\Gamma(\alpha,\mu) = e^{\mu|\alpha|}, \tag{4.23}$$

and call this the velocity modification factor. To get the final velocity we just multiply v_0 by Γ. From (4.22) and (4.23) we can directly deduce that if either α or μ equals *zero*, there is no change in the speed while for $(\alpha\mu) \gg 0$ the final velocity is vanishingly *zero*. This can also be seen in Fig. (4.4) where the contour plot shows the velocity modification factor as a function of the friction coefficient and the angle of the circular arc. The change of velocity is denoted by $\triangle v$ where:

$$\triangle v = v_0 - v_f = v_0(1 - e^{\mu|\alpha|}).$$

Looking at the friction-containing terms of (4.2):

$$-\mu\,\mathrm{sgn}(v_x)\left(\frac{g+v_x^2 f_{xx}}{1+f_x^2}\right),$$

and

$$-\mu\,\mathrm{sgn}(v_z)\left(\frac{g+v_x^2 f_{xx}}{1+f_x^2}\right),$$

Let for the following be

$$\tilde{\Phi} = g + v_x^2 f_{xx}$$

. We will now compare the loss of velocity due to friction for a point moving on a (monotonous) C^2-curve and the piecewise linear segment approximating this curve, respectively. A subscript c to $\tilde{\Phi}$ will denote the value for $\mathcal{C}$ while a subscript l stands for the linear segment. Altogether three cases are possible:

Case 1: The curve $\mathcal{C}$ is convex (comp. figure 4.5) $\Rightarrow f_{xx} > 0$. It immediately follows

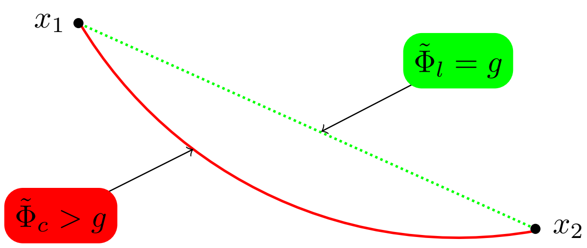


FIGURE 4.5: Influence of friction for a convex curve

that also $\tilde{\Phi}_c > 0$, and moreover $\tilde{\Phi}_c > g$. On the linear segment instead, as $f_{xx} = 0$ we have $\tilde{\Phi}_l = g$ and therefore $\tilde{\Phi}_c > \tilde{\Phi}_l$. Hence, in comparison with the motion on the curve $\mathcal{C}$, the point experiences a less strong loss of velocity when moving along the linear segment. This is compensated by the velocity modification factor that additionally decreases the velocity of the point in the transition point between two segments.

Case 2a: The curve $\mathcal{C}$ is concave (comp. figure 4.6) $\Rightarrow f_{xx} > 0$ but still $g > -v_x^2 f_{xx}$ and therefore $\tilde{\Phi}_c > 0$. On the linear segment instead, as $f_{xx} = 0$ we still have

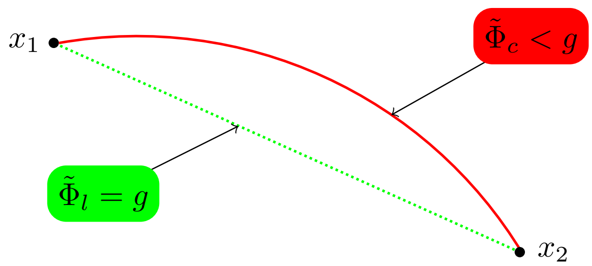


FIGURE 4.6: Influence of friction for a concave curve

$\tilde{\Phi}_l = g$ and therefore $\tilde{\Phi}_c < \tilde{\Phi}_l$. In comparison with the motion on the curve $\mathcal{C}$, the point now experiences a stronger loss of velocity when moving along the linear segment which will be compensated by a gain of velocity using again the velocity modification factor which thus has to be adapted for concave curves.

Case 2b: The curve $\mathcal{C}$ is concave $\Rightarrow f_{xx} > 0$ and $g < -v_x^2 f_{xx}$ and therefore $\tilde{\Phi}_c < 0$. It follows that $\tilde{\Phi}_c$ increases the velocity which obviously is an unphysical behaviour Nonetheless, we will consider this case for the moment and discuss possible solutions to the problem later on.

On the linear segment, as $f_{xx} = 0$ we again have $\tilde{\Phi}_l = g$ and therefore $\tilde{\Phi}_c < \tilde{\Phi}_l$. In comparison with the motion on the curve $\mathcal{C}$, where the velocity increases, the point experiences a loss of velocity when moving along the linear segment which will be compensated by a gain of velocity using again the velocity modification factor.

We will now adapt the velocity modification factor such that it holds also for concave curves and therefore look again at equation (4.18):

$$\mathbf{F}_\mu = -\operatorname{sgn}(v)\mu\Phi_n\mathbf{t},$$

In order to obtain a gain of velocity instead of a loss the sign in (4.18) needs to be changes such that we finally get:

$$\dot{v} = \operatorname{sgn}(v)\mu\frac{v^2}{r}. \tag{4.24}$$

Solving (4.24) and then following the procedure above we find that the following velocity modification factor Γ' for a point with initial velocity v_0 moving on a circular arc with given angle α and friction coefficient μ:

$$\Gamma'(\alpha,\mu) = e^{-\mu|\alpha|},$$

and thus we can define a more general Γ as follows:

$$\Gamma(\alpha,\mu) = e^{\pm\mu|\alpha|},$$

where the positive sign accounts for convex and the negative sign for concave curves. According to figure (4.4) the velocity modification factor for concave curves is displayed in figure (4.7) for different values of α and μ.

4.2.5 Friction: alternative strategies for concave curves

We just found that under certain circumstances, equation (4.2) allows a gain of velocity due to the friction containing terms which is an unphysical behaviour This could be avoided making one of the two following assumptions:

- detachment from the surface is allowed. The point detaches from the surface in the moment in which $g = -v_x^2 f_{xx}$. Further motion is described by additional equations that model the free motion through the air until the point finally ”lands” on the curve $\mathcal{C}$;

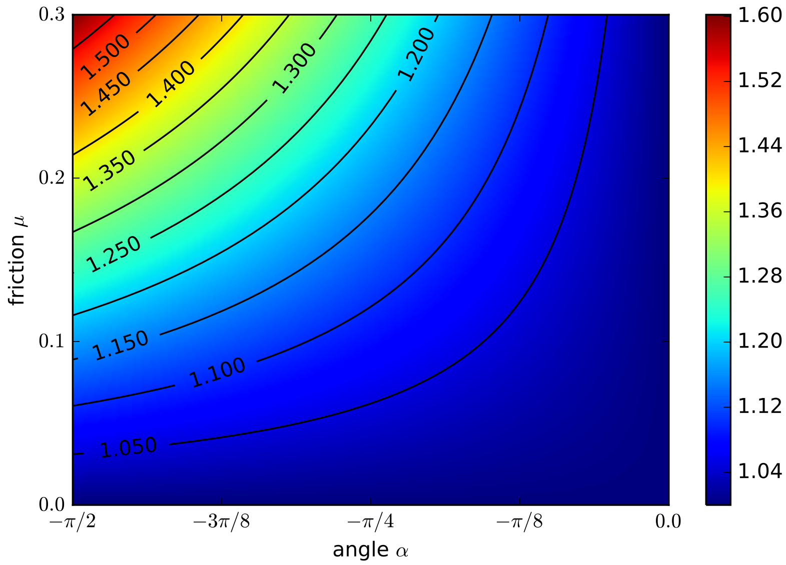


FIGURE 4.7: Velocity modification factor Γ for concave curves, plotted as a function of the friction coefficient μ and the angle of the circular arc α

- point is fully constrained to the surface, i.e. the motion can be seen as bounded from bottom and top. In this case $\mathbf{\Phi}$ must be replaced by $|\mathbf{\Phi}|$ in equ. (4.2).

In real world landslide simulations however, one will probably never find the case of a mass detaching from the surface wherefore the two solutions proposed above are not considered in the implementation of the problem. For testing the method developed in the following, a possibly unphysical behaviour of the equations in some sections of the test-surfaces is not problematic as the presented method works well also in this case. Nonetheless, the two alternative methods shall briefly be described in the following.

Detachment from the surface A very simple model for the motion of a point that detaches from the surface is assumed here, where air resistance and friction are neglected. In other words:

$$a_x = 0, \qquad a_y = -g;$$

and therefore:

$$v_x = v_{x,0}, \qquad v_y = v_{y,0} - gt.$$

It follows that the position of the point is given by:

$$x = x_0 + v_{x,0}t, \tag{4.25}$$

$$y = y_0 + v_{y,0}t - \frac{1}{2}gt^2. \tag{4.26}$$

The first step is to compute whether or not the point detaches from the surface. For a smooth (C^2) function this is the case iff throughout the motion the condition

$$g + v_x^2 f_{xx} = 0 \tag{4.27}$$

becomes true.

This condition certainly fails for piecewise linear functions as the point can only detach from the surface in the points of transition between two elements where in turn f_{xx} does not exist. From a physical point of view, the point should detach from the surface at each transition point, provided that the curve is concave in this point. If, in contrast, the solution on the piecewise linear curve shall be an approximation to the solution on a corresponding smooth curve, a different condition for the detachment from the surface has to be found that resembles condition (4.27). Therefore we can use the net-velocity per segment after the velocity modification factor has been applied.

Let v_0 and v_1 be the points velocity at the beginning and end of the segment σ_i with endpoints x_i and x_{i+1} and let v_{fr} be the loss of velocity due to friction that the point experiences when moving from x_i to x_{i+1}. v_{fr} can be computed independently as it does not depend on the points velocity. Recall therefore that

$$a_x = -g\frac{f_x}{1+f_x^2} - \mu\,\mathrm{sgn}(v_x)\frac{g}{1+f_x^2}.$$

v_{fr} can be computed after knowing the time of intersection t_{int} with the respective endpoint of σ_i:

$$v_{fr} = \mu\,\mathrm{sgn}(v_x)\frac{g}{1+f_x^2}t_{int}.$$

We then compute the velocity modification factor Γ and the resulting change of velocity

$$\triangle v = v_1(1-\Gamma)$$

which represents the loss or gain of velocity at the point of intersection. Using this the following approximation could be done: detachment from the surface is assumed if the change of velocity resulting from the friction containing terms results in a loss of speed, i.e.:

$$\mathrm{sgn}(v_x)(v_{fr} + \triangle v) < 0. \tag{4.28}$$

In a second step we need to find the intersection between the curve and the trajectory of the point. If the curve is defined piecewise, those intersections have to be computed element-wise until the correct element and point of intersection are found. One possible way to the solution of the problem is shown in the following. It is convenient here to express the elements of $\mathcal{C}$ in parameter-form, i.e. an element σ_i is defined by three parameters a_i, b_i and c_i such that:

$$ax + by = c, \tag{4.29}$$

where, knowing the two endpoints (x_i, y_i) and (x_{i+1}, y_{i+1}) of element σ_i, the parameters a_i, b_i and c_i can be computed using the following relations:

$$\begin{aligned}
a &= y_i - y_{i+1}, \\
b &= x_{i+1} - x_i, \\
c &= x_{i+1} y_i - x_i y_{i+1}.
\end{aligned}$$

Inserting the equations for x and y, (4.25) and (4.26) respectively, in equation (4.29) we get:

$$\begin{aligned}
& a(x_0 + v_{x,0} t) + b(y_0 + v_{y,0} t - \frac{1}{2} g t^2) = c, \\
\Rightarrow \quad & (a x_0 + b y_0 - c) + (a v_{x,0} + b v_{y,0}) t - \frac{1}{2} g t^2 = 0,
\end{aligned}$$

which we have to solve for t to get the two intersections with the element σ_i:

$$t_{1/2} = \frac{-(a v_{x,0} + b v_{y,0}) \pm \sqrt{(a v_{x,0} + b v_{y,0})^2 - 2(a x_0 + b y_0 - c) g}}{2(a x_0 + b y_0 - c)}.$$

Plugging the two possible solutions for t into equation (4.25) we find two potential points of intersection x_{int}. If x_{int} is located within the element σ_i, i.e. $[x_i \leq x_{int} \leq x_{i+1}]$, the point of intersection has been found, assuming that x_{int} is not also the point of detachment. Otherwise we have to proceed with the next element. Obviously, starting from the point of detachment we only have to check the elements σ_i that lie in the direction of the points "flight".

Friction: point fully constrained to the surface If the point is fully constrained to the surface and thus $\mathbf{\Phi}$ is replaced by $|\mathbf{\Phi}|$, the friction containing terms of equ. (4.2) become

$$-\mu \operatorname{sgn}(v_x) \frac{g + v_x^2 f_{xx}}{1 + f_x^2} \quad \Rightarrow \quad -\mu \operatorname{sgn}(v_x) \left| \frac{g + v_x^2 f_{xx}}{1 + f_x^2} \right| = -\mu \operatorname{sgn}(v_x) \frac{|g + v_x^2 f_{xx}|}{1 + f_x^2}$$

and

$$-\mu\,\mathrm{sgn}(v_z)\frac{g+v_x^2 f_{xx}}{1+f_x^2} \quad\Rightarrow\quad -\mu\,\mathrm{sgn}(v_z)\left|\frac{g+v_x^2 f_{xx}}{1+f_x^2}\right| = -\mu\,\mathrm{sgn}(v_z)\frac{\left|g+v_x^2 f_{xx}\right|}{1+f_x^2},$$

where

$$\left|g+v_x^2 f_{xx}\right| = \begin{cases} +g+v_x^2 f_{xx}, & \text{for } g \geq -v_x^2 f_{xx} \\ -g-v_x^2 f_{xx}, & \text{for } g < -v_x^2 f_{xx}. \end{cases}$$

For a C^2-curve where the second derivative is defined everywhere, this approach is easily applied by using the following rhs $\tilde{\mathbf{F}}$ instead of (4.2):

$$\tilde{\mathbf{F}} = \begin{pmatrix} v_x \\ -f_x \dfrac{g+v_x^2 f_{xx}}{1+f_x^2} - \mu\,\mathrm{sgn}(v_x)\dfrac{\left|g+v_x^2 f_{xx}\right|}{1+f_x^2} \\ v_z \\ \dfrac{-f_x^2 g+v_x^2 f_{xx}}{1+f_x^2} - \mu\,\mathrm{sgn}(v_z)\dfrac{\left|g+v_x^2 f_{xx}\right|}{1+f_x^2} \end{pmatrix}.$$

For piecewise linear curves instead, a case distinction has to be done. This can be done using the same approximation as in the previous paragraph. In a first step the equations of motion are solved using the rhs $\mathbf{F}_+ = \mathbf{F}$ (comp. (4.2)). If condition (4.28) is not fulfilled, i.e. the change in speed resulting from the friction term leads to a gain of speed the equations of motion have to be recomputed for the respective segment using the rhs $\mathbf{F}_-$:

$$\mathbf{F}_- = \begin{pmatrix} v_x \\ -f_x \dfrac{g+v_x^2 f_{xx}}{1+f_x^2} - \mu\,\mathrm{sgn}(v_x)\dfrac{-(g+v_x^2 f_{xx})}{1+f_x^2} \\ v_z \\ \dfrac{-f_x^2 g+v_x^2 f_{xx}}{1+f_x^2} - \mu\,\mathrm{sgn}(v_z)\dfrac{-(g+v_x^2 f_{xx})}{1+f_x^2}. \end{pmatrix}.$$

4.2.6 Friction: further considerations

In this subsection we want to present two important features of the results derived above. First we compare the theoretical change of velocity on a circular path to that on a corresponding piecewise-linear discretization. We will show that the overall change and therefore the final velocity after passing through a certain angle α is equal in both cases, if the discretization conserves the angle.

Consider therefore a circle segment with angle α and a decomposition of α with

$$\alpha = \alpha_1 + \alpha_2 + \cdots + \alpha_n = \sum_{i=0}^{n} \alpha_i.$$

TABLE 4.1: Relative arrival times t_r for different values of μ and α

α	$\mu = 0.05$	$\mu = 0.1$	$\mu = 0.15$	$\mu = 0.2$
$\pi/2 \equiv 90.0^\circ$	0.785	0.784	0.782	0.779
$\pi/4 \equiv 45.0^\circ$	0.947	0.947	0.947	0.946
$\pi/10 \equiv 18.0^\circ$	0.992	0.992	0.992	0.991
$\pi/20 \equiv 9.0^\circ$	0.998	0.998	0.998	0.998
$\pi/40 \equiv 4.5^\circ$	0.999	0.999	0.999	0.999

Let v_0 denote the initial velocity before entering the circle segment and v_f the final velocity on exit. Using equation (4.22) it is easy to find for the final velocity v_f:

$$v_f = \frac{v_0}{\exp\left(\mu \sum_{i=1}^{n} \alpha_i\right)} = \frac{v_0}{\prod_{i=1}^{n} \exp\left(\mu \alpha_i\right)} = v_0 \prod_{i=1}^{n} \Gamma_i,$$

where $\Gamma_i = \Gamma(\mu\alpha_i)$. It follows that the final velocity is also independent of a possible decomposition of the angle α into sub-angles. So, in case of horizontal circular motion the final velocity on a smooth circle is the same as on a discretized one if the total angle α is conserved. Only the arrival times differ depending on the roughness of the discretization as the discretized curve is slightly longer. The differences for different rough to fine discretizations in dependency of the friction coefficient can be seen in Table 4.1. They were computed using the following relation:

$$t_r = \frac{t_C}{t_L} = \frac{\tanh(\frac{\alpha\mu}{2})}{\mu \tan(\frac{\alpha}{2})}, \tag{4.30}$$

where t_r is the ratio between the time t_C needed on the smooth circle and t_L which is the time needed on the discretized curve. Computing the limit of t_r for $\alpha \to 0$, one finds that $t_r \to 1$ meaning that $t_L \to t_C$. Note that also the ratio t_r is radius-independent again. In Table 4.1 the relative arrival times t_r are shown for some values of α and μ.

In the following, Eq. (4.30) is explained more in detail. Assume a curve that is once approximated by two straight segments and once by a circular segment, where the angle of the latter equals the one enclosed by the two linear segments (comp. Fig. 4.3). What are the arrival times of two points each moving along the two curves, starting at time t_1 with initial velocity v_0, arriving in the final position at t_2? To compute this, we first have to know the length of curve L for the piecewise linear curve which can be computed from the angle α and the radius r. The situation is shown in Fig. 4.8.

We need to know the length L of the linear segment. For simplicity let's define:

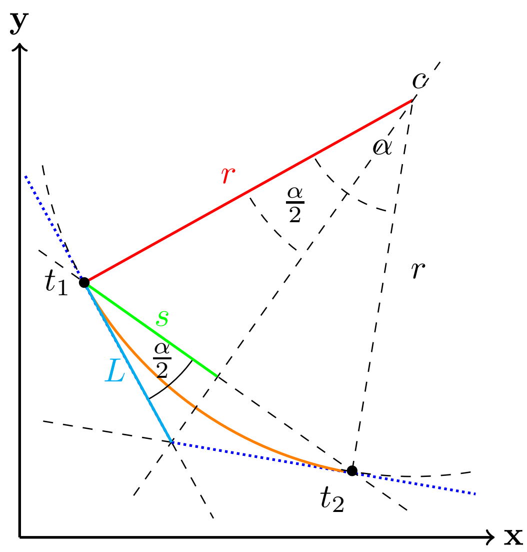


FIGURE 4.8: Computation of the relative arrival time

$\theta = \alpha/2$. Then we find:

$$\begin{aligned}
\sin(\theta) &= \frac{s}{r} \quad \Rightarrow \; s = r\sin(\theta), \\
\cos(\theta) &= \frac{s}{L} \quad \Rightarrow \; L = \frac{r\sin(\theta)}{\cos(\theta)} = r\tan(\theta).
\end{aligned}$$

We know that the velocity of the point moving down the piecewise linear curve is constant, except in the change of slope where the velocity decreases instantaneously following Eq. 4.22. We denote the time needed to reach the point of changing slope with t_{L_1} and the one to reach the final position from there by t_{L_2}. Together: $t_L = t_{L_1} + t_{L_2}$. The final time for the point moving along the circle segment is denoted by t_C. Using the above equations we find:

$$\begin{aligned}
t_{L_1} &= \frac{r\tan(\frac{\alpha}{2})}{v_0}, \\
t_{L_2} &= r\tan(\tfrac{\alpha}{2})\tfrac{e^{\mu\alpha}}{v_0}, \\
t_L &= \frac{r}{v_0}\tan(\tfrac{\alpha}{2})\left(1 + e^{\mu\alpha}\right).
\end{aligned}$$

For the circular arc the time is:

$$t_C = \frac{r}{v_0}\left(\frac{e^{\mu\alpha} - 1}{\mu}\right),$$

and hence the ratio between t_L and t_C is given by:

$$t_r = \frac{t_C}{t_L} = \frac{e^{\mu\alpha} - 1}{\mu \tan(\frac{\alpha}{2})\,(1 + e^{\mu\alpha})} = \frac{\tanh(\frac{\alpha\mu}{2})}{\mu \tan(\frac{\alpha}{2})},$$

For the limit $\alpha \to 0$ we get:

$$\lim_{\alpha\to 0} t_r = \lim_{\alpha\to 0} \frac{\mu \tanh(\frac{\alpha}{2})}{\tan(\frac{\alpha}{2})\mu} = \frac{0}{0},$$

meaning that we have to apply de l'Hopital's rule:

$$\lim_{\alpha\to 0} t_r = \lim_{\alpha\to 0} \frac{\left(\tanh(\frac{\alpha\mu}{2})\right)'}{\left(\mu \tan(\frac{\alpha}{2})\right)'} = \lim_{\alpha\to 0} \frac{\left(1 - \tanh^2\left(\frac{\alpha\mu}{2}\right)\right)\frac{\mu}{2}}{\mu\left(1 + \tan^2\left(\frac{\alpha}{2}\right)\right)\frac{1}{2}} \to 1.$$

As in real applications we are generally dealing with relatively small angles, we can certainly assume that the arrival times are approximately equal. As an example: assume that we want to approximate a half circle by 10 linear segments (which is a pretty rough discretization), the angle α is already $\pi/10$ and therefore the relative time is about 0.992 (meaning that the time needed on the circle segment is 0.992 times the time needed on the piecewise linear approximation).

4.2.7 The solution strategy

In the previous subsections we have found that the solution of Eq. (4.2) can be computed in parts, one for the single linear segments σ_i of Σ and one for its transition points that are smoothed using an imaginary circular arc. In the latter case the problem can further be split into a normal component, thus changing the direction of motion but not the speed, and a tangential component, which is changing the speed but not the direction of motion. In the transition areas, the change of direction is computed by a simple rotation of the velocity vector of the point mass that is independent from the arc radius. Just as well, the effect of friction is independent of the arc radius, and so the speed of the sliding mass is changed through the factor Γ. This allows us to build the curve Σ^* in such a way that smoothing arcs connecting adjacent linear segments have an infinitesimally small radius, which means that the curves Σ^* and Σ practically coincide one with another. This enables us to use different equations in different part of the curve. Along the straight segments the mass motion is governed only by the curvature-independent contribution given by Eq. (4.10), while at the transition nodes one simply rotates the velocity vector by means of Eq. (4.17) and changes the speed magnitude by means of the modification factor Γ deduced above. We observe that the computations along the geometrical elements of the curve are quite simple and involve only elementary expressions. A missing element for the validation of this strategy regards the time taken by the mass to move along the curve Σ and the curve Σ^*. To this purpose, one can compare the time taken by the mass to go from the first end of the segment σ_i to the second end of the adjacent segment σ_{i+1} along the two curves. The difference is only due to the time taken in the transition region around the common end of the two segments, where in one case the mass slides on the segments, whereas in the other it runs along the arc. Since the transition region becomes smaller and smaller as the arc radius goes to zero, it is easy to see that also the difference in time goes to zero, and therefore the splitting strategy we have devised is correct.

The method is also briefly sketched in algorithm 2.

Algorithm 2 Algorithm for 1D analytical solution

logical :: inner {true if point does not leave current segment}
while ($time < T_{end}$) **do** {external loop, time-step: $\triangle t$}
  inner=$.false.$, $\tilde{\triangle} t = \triangle t$ {$\tilde{\triangle} t$=remaining time}
  while (inner==.false.) **do**
    Compute acceleration a_x
    if (Stop conditions fulfilled) **then**
      Stop simulation
    else
      Compute intersection time t_{int} and point x_{int}
      if ($\tilde{\triangle} t \leq t_{int}$) **then**
        inner=$.true.$
        $x_i = x_{i-1} + v_{x,i-1}\tilde{\triangle} t + 0.5 a_{x,i-1}\tilde{\triangle} t^2$
        $v_x, i = v_{x,i-1} + a_{x,i-1}\tilde{\triangle} t$
      else
        $x_i = x_{int}$
        $v_x, i = v_{x,i-1} + a_{x,i-1} t_{int}$
        Correct direction and modulus of velocity
        $\tilde{\triangle} t = \tilde{\triangle} t - t_{int}$
      end if
    end if
  end while
end while

4.3 Results 1: Evaluation of the two different methods

This section is dedicated to the comparison of the solution of the equations on piecewise linear curves with the numerical solution obtained using smooth surface reconstruction methods in combination with the lsode-solver. The two methods are evaluated using a set of four different test-cases, accounting for possibly critical characteristics of real topographies such as sudden change in slope.

4.3.1 Test-Cases

All test-cases are defined on the domain $[-4,4]$. They are constructed such as to represent different critical characteristics of real topographies such as a sudden change in slope. While the first test-case is a piecewise-linear function and therefore contains an instantaneous change in slope, the other test-functions are smooth. The tests will be applied in the following to compare the results obtained on piecewise-linear curves to those obtained on smooth reconstructed curves (using the methods presented in chapter 2) using LSODE for the time-integration.

Test-Case 1:

$$g_1(x) = \begin{cases} -x & \text{if} \quad (x,y) \in [-4,0), \\ 0 & \text{if} \quad (x,y) \in [0,4]. \end{cases}$$

This is the only non-smooth function, representing an instantaneous change in slope. During the reconstruction the corner in $x = 0$ must be smoothed while the piecewise-linear parts should remain as linear as possible.

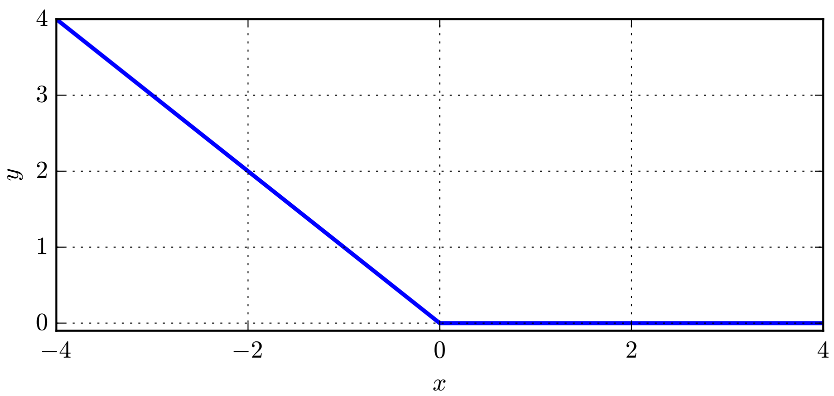


FIGURE 4.9: Test-curve for case 1.

Test-Case 2: A kind of normalized arctan-function. Due to the definition of $g_{2,x}$ the inclination of the function g_2 before the change of slope is given by the parameter φ, where the inclination afterwards is *zero* by definition. We will present a case with $g_{2,x} \approx -1$ for $x > 0$ and $g_{2,x} \approx 0$ for $x > 0$. This test is in some sense a smoothed version of test case 1. The first derivative is given by:

$$g_{2,x} = \frac{1}{\pi} \arctan(\varphi x) - 0.5,$$

which gives the following reconstructing function:

$$g_2(x) = \frac{1}{\pi} x \arctan(\varphi x) - \frac{1}{(2\pi\varphi)} \log(\varphi^2 x^2 + 1) - 0.5x,$$

and second derivative:

$$g_{2,xx} = \frac{\varphi}{\pi} \cdot \frac{1}{1+\varphi^2 x^2}.$$

Similar to the first test case, the slope is first negative and then becomes almost *zero* after passing through $x = 0$ but this is done in a smooth way.

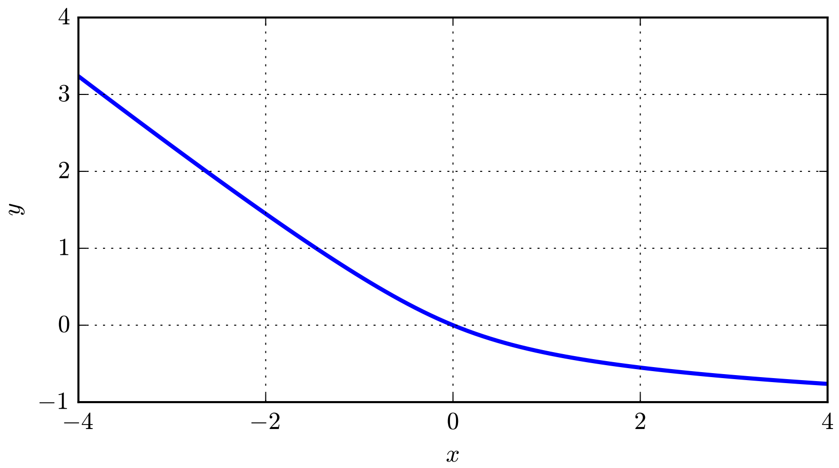


FIGURE 4.10: Test-curve for case 2.

Test-Case 3: A parabola

$$g_3(x) = \varphi x^2,$$

where φ is the scale-factor of g_3. In contrast to the other test functions, g_3 has constant second derivative and should be approximated better by the radial basis functions than by the piecewise linear curve. It stands for a moderate change in slope, having constant second derivative.

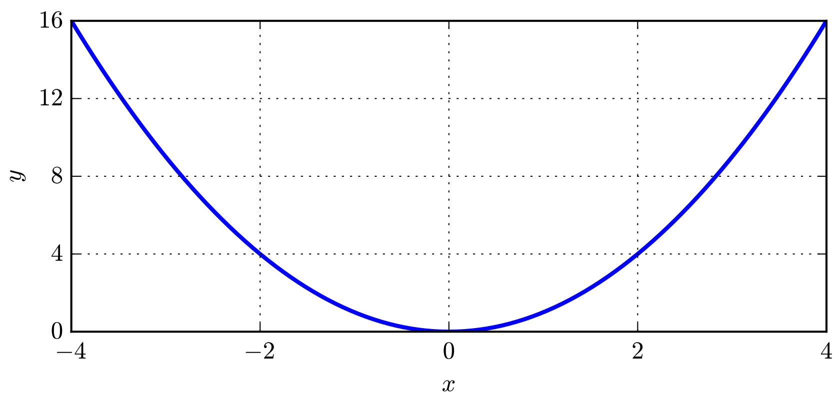


FIGURE 4.11: Test-curve for case 3.

Test-Case 4: A basic arctan-function. The definition of this function is very similar to test-case 2 but in contrast the slope for $x > 0$ is not close to *zero* but has the same absolute value as for $x < 0$, just with opposite sign. Again, it is determined by the parameter φ. Compared to test-case 3 the change in slope is more sudden.

$$g_{4,x} = \arctan(\varphi x).$$

Therefore the following reconstructing function is found:

$$g_4(x) = x\arctan(\varphi x) - \log(\varphi^2 x^2 + 1)/(2\varphi) + \eta y,$$

and second derivative:

$$g_{4,xx} = \varphi \frac{1}{1+\varphi^2 x^2}.$$

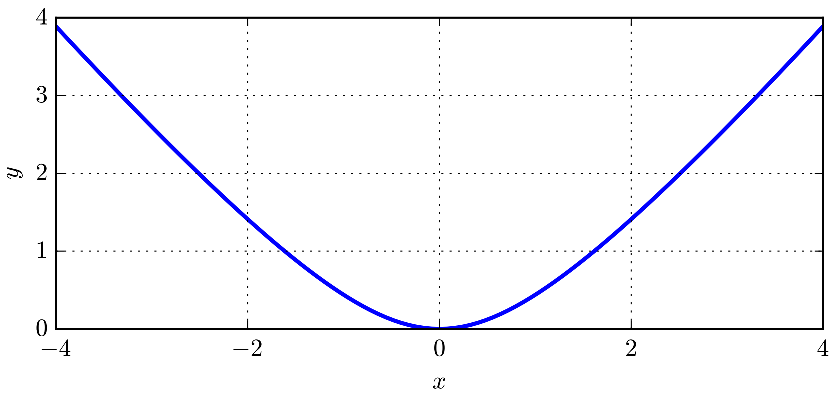


FIGURE 4.12: Test-curve for case 4.

4.3.2 Comparison with the numerical results obtained on smooth reconstructed curves

As also for smooth reconstructed curves the equations are still stiff, the results on those curves were obtained using LSODE with a time-step of $\Delta t = 0.1$ seconds for the time-integration. The solution computed for the corresponding piecewise-linear surface were stored at the same time-intervals. The initial conditions for all tests were chosen as follows: $x_0 = -4.0$, $v_{x,0} = 0.0$.

To also get information on the influence of the discretization, the simulations were run on different coarse/fine grids, where for the coarse grid 6 points and 21 for the fine grid were taken. For all test-cases, first of all the point's movement along the x-axis in time and then the trajectory are shown. Afterwards the components of the energy are plotted. Finally the velocity in x-direction and the modulus of the velocity along the trajectory are shown. The plots are done each for the coarse and fine grid. The piecewise linear reconstructions are in the following denoted by

”lin”, while ”rbf” (radial basis function, the basis of the used smooth reconstructed curves/surface) denotes the smooth reconstructed curves.

Test-case 1

$$f(x) = \begin{cases} -x & \text{if} \quad (x) \in [-4,0), \\ 0 & \text{if} \quad (x) \in [0,4]. \end{cases}$$

This is the only test-case in which the control-surface is not a smooth function but it is given by a piecewise-analytical function. Therefore the Inverse-Multi-Quadratic reconstruction is compared to the piecewise-linear surface. Obviously, the two simulations differ quite much on a coarse grid, also as there is no data point for $x = 0$ and in addition, a smooth surface can – by definition – never fully interpolate the instantaneous change in inclination. The differences in the trajectory for the coarse grid are explained by the fact that in this case the reconstructed surface shows a slight sagging in between of two points of the disretization. As the initial positions are exactly in between of two of these points, also the trajectory is different as in the piecewise-linear case. Even the height of the surface is not equal due to that, leading overall to a lower velocity especially at the beginning of the movement. Altogether, this leads to a lower total energy. On the contrary, for the finer grid, the differences almost disappear.

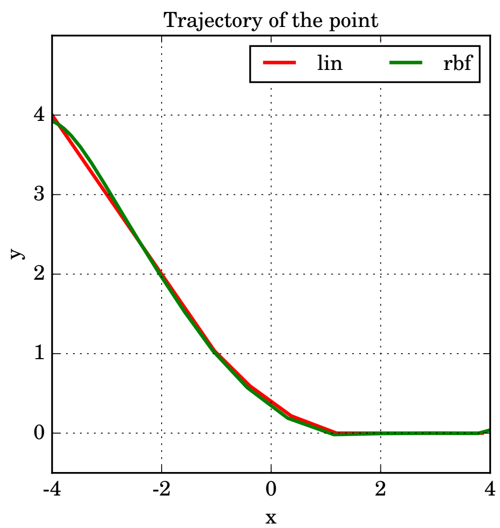


FIGURE 4.13: Test-Case 1: Trajectories for the piecewise linear sliding curves for a discretization with 6 grid points.

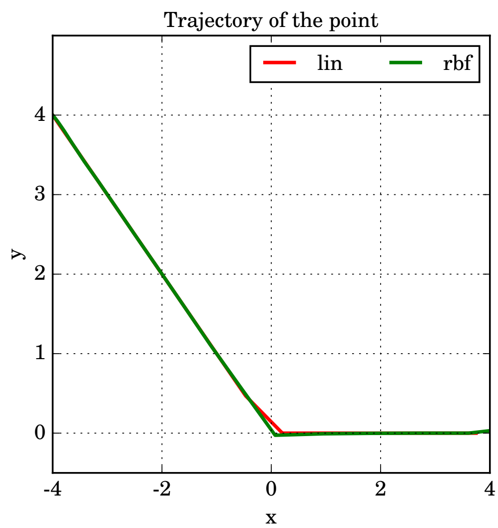


FIGURE 4.14: Test-Case 1: Trajectories for the piecewise linear sliding curves for a discretization with 21 grid points.

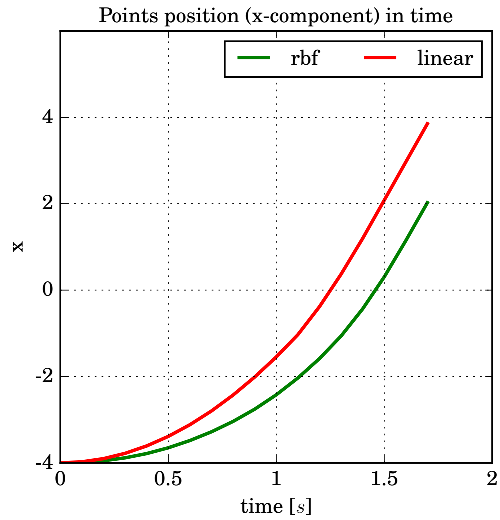


FIGURE 4.15: Test-Case 1: x-component of the point's position in time during motion on a piecewise-linear surface for a discretization with 6 grid points.

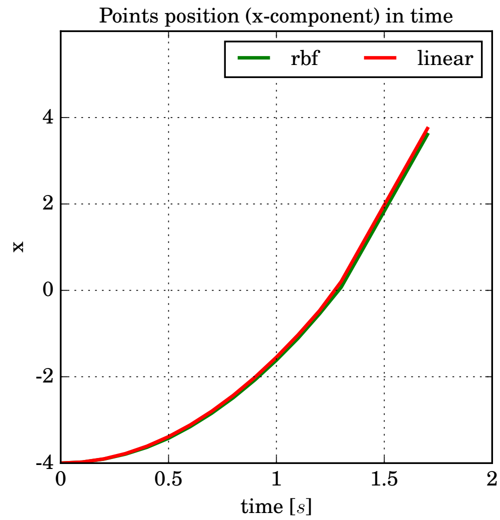


FIGURE 4.16: Test-Case 1: x-component of the point's position in time during motion on a piecewise-linear surface for a discretization with 21 grid points.

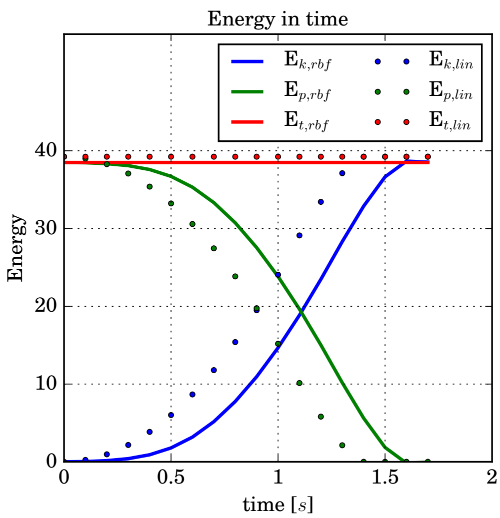


FIGURE 4.17: Test-Case 1: Kinetic, potential and total energy for the simulation on a piecewise-linear surface for a discretization with 6 grid points.

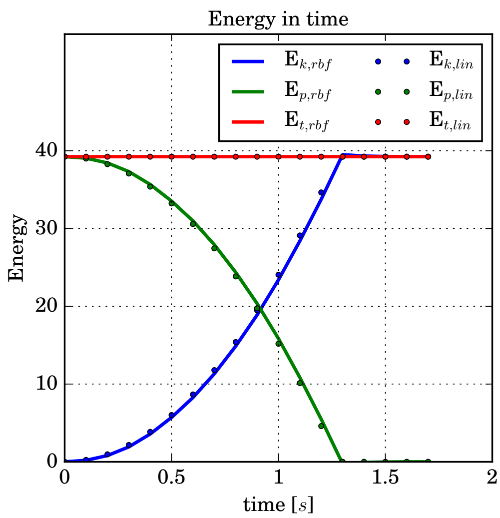


FIGURE 4.18: Test-Case 1: Kinetic, potential and total energy for the simulation on a piecewise-linear surface for a discretization with 21 grid points.

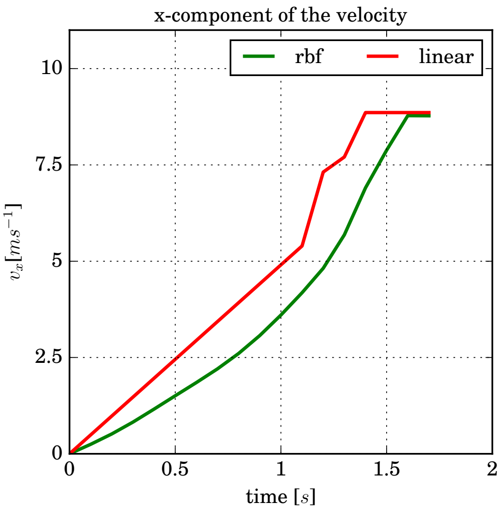


FIGURE 4.19: Test-Case 1: x-component of the point's velocity in time during motion on a piecewise-linear surface for a discretization with 6 grid points.

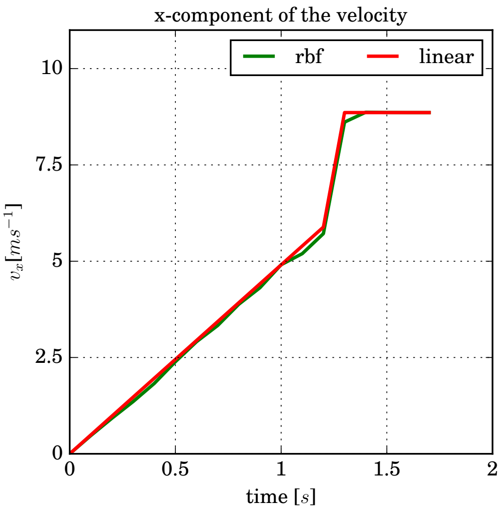


FIGURE 4.20: Test-Case 1: x-component of the point's velocity in time during motion on a piecewise-linear surface for a discretization with 21 grid points.

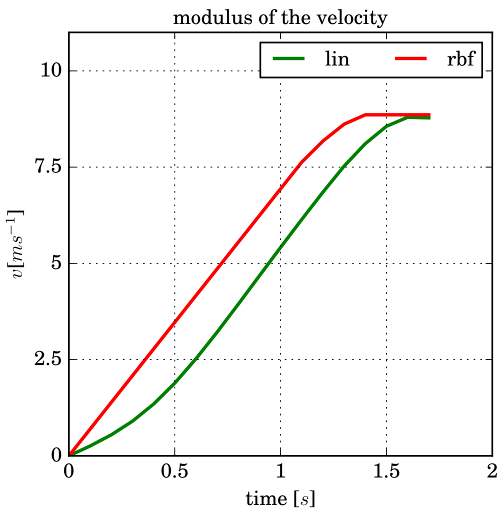


FIGURE 4.21: Test-Case 1: The modulus of the point's velocity in time during motion on a piecewise-linear surface for a discretization with 6 grid points.

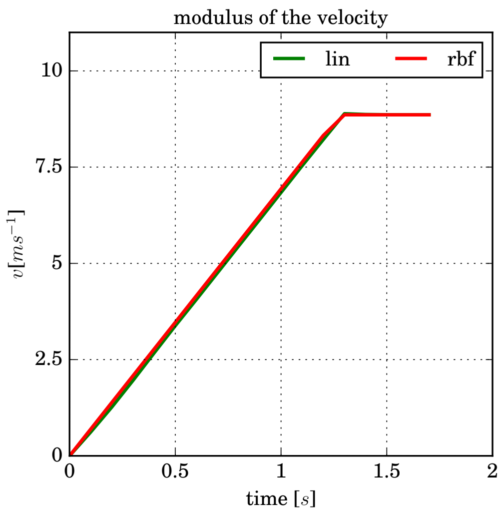


FIGURE 4.22: The modulus of the point's velocity in time during motion on a piecewise-linear surface for a discretization with 21 grid points.

Test-case 2

$$f(x,y) = \frac{1}{\pi} x \arctan(\varphi x) - \frac{1}{(2\pi\varphi)} \log(\varphi^2 x^2 + 1) - 0.5x.$$

The final time for this test was chosen to be $1.7s$ as this is the time at which the point leaves the reconstructed surface.
Over all, this test case is very similar to the above presented, just it is smooth along all the surface. Also the results are very similar to the ones above.

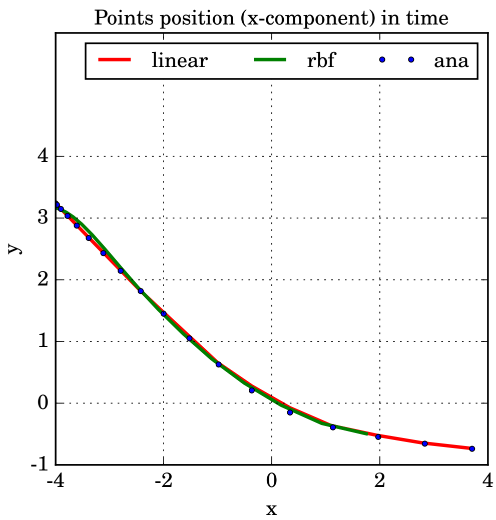


FIGURE 4.23: Test-Case 2: Trajectories for the sliding curve defined using a ”normalized” arctan for a discretization with 6 grid points.

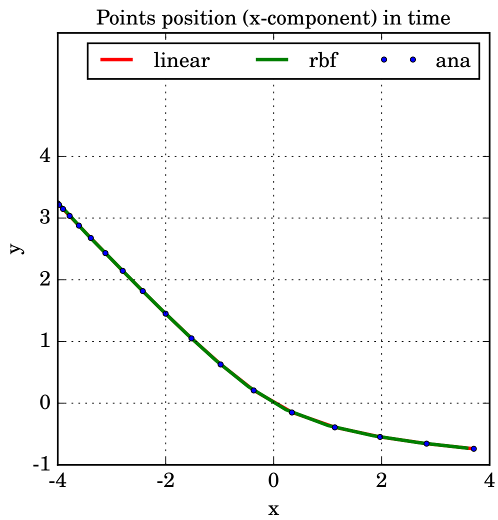


FIGURE 4.24: Test-Case 2: Trajectories for the sliding curve defined using a ”normalized” arctan for a discretization with 21 grid points.

Test-case 3

$$f(x,y) = \varphi x^2.$$

This example is different from the previous two in a sense that the surface is generally smoother, showing variations in the inclination along all the surface. Here the RBF-reconstruction performs better then the piecewise-linear one. The reason for this can be seen in Figs. 4.37 and 4.38 , where a zoom on the trajectory is shown. One can see that the piecewise-linear segments are not on the surface and also they cannot capture the minimum of the ”real” function which can be done a better by the RBF. However, the differences in this example are not too extreme and both reconstructions can perform well in the simulation.

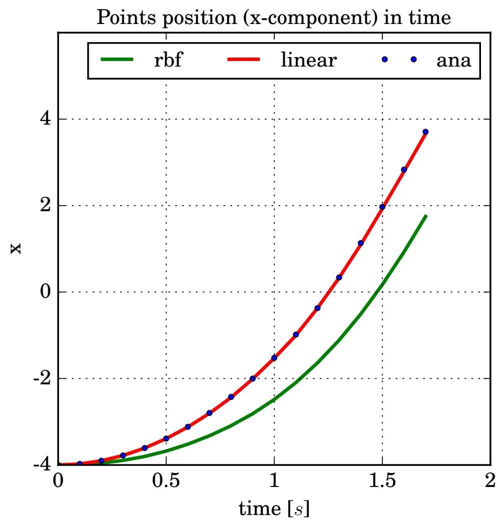


FIGURE 4.25: Test-Case 2: x-component of the point's position in time during motion on a surface defined using a "normalized" arctan for a discretization with 6 grid points.

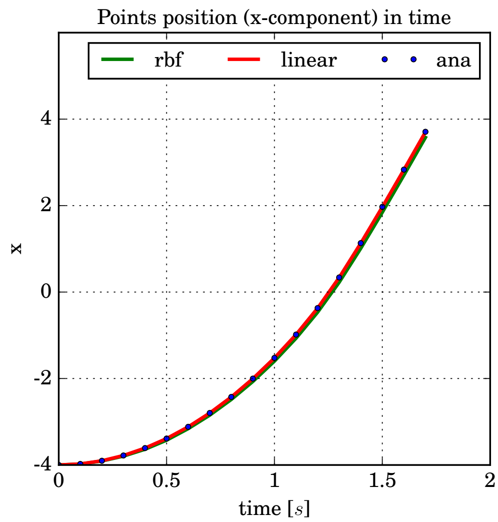


FIGURE 4.26: Test-Case 2: x-component of the point's position in time during motion on a surface defined using a "normalized" arctan for a discretization with 21 grid points.

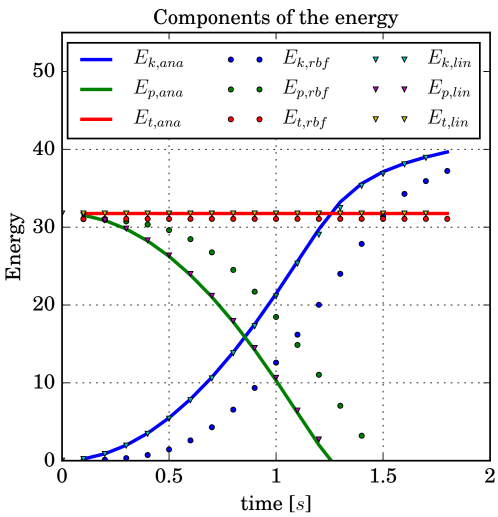


FIGURE 4.27: Test-Case 2: Kinetic, potential and total energy for the simulation on a surface defined using a "normalized" arctan for a discretization with 6 grid points.

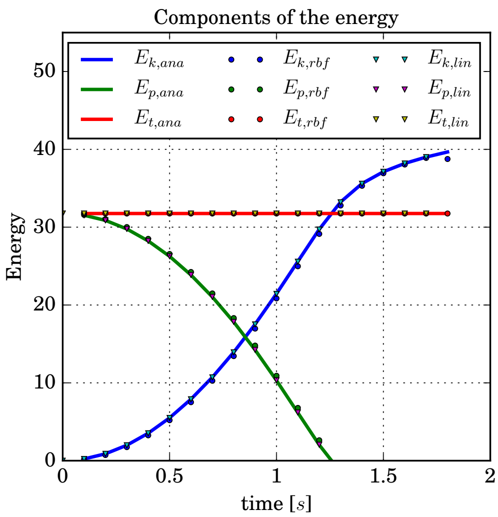


FIGURE 4.28: Test-Case 2: Kinetic, potential and total energy for the simulation on a surface defined using a "normalized" arctan for a discretization with 21 grid points.

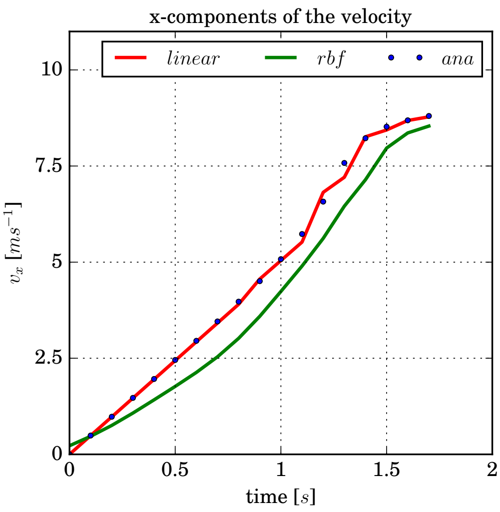


FIGURE 4.29: Test-Case 2: x-component of the point's velocity in time during motion on a surface defined using a "normalized" arctan for a discretization with 6 grid points.

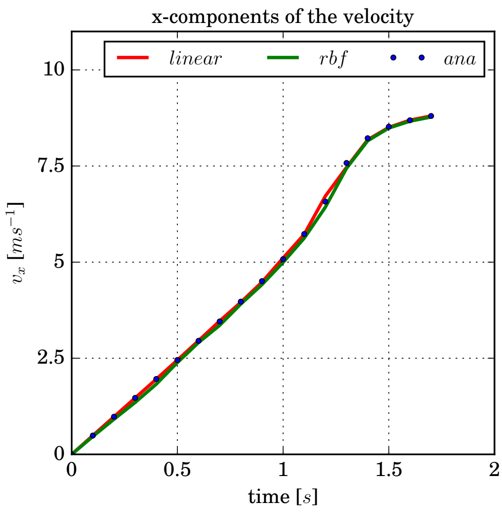


FIGURE 4.30: Test-Case 2: x-component of the point's velocity in time during motion on a surface defined using a "normalized" arctan for a discretization with 21 grid points.

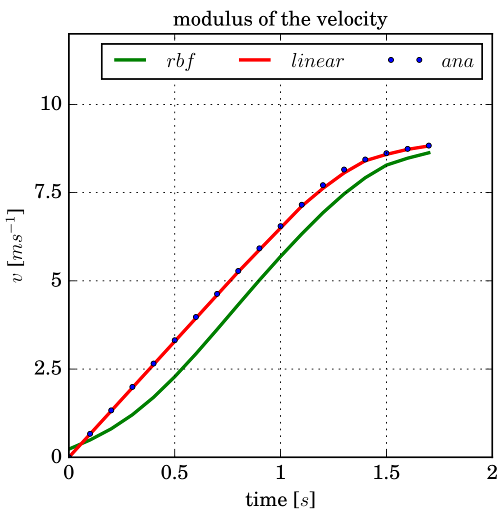


FIGURE 4.31: Test-Case 2: The modulus of the point's velocity in time during motion on a surface defined using a "normalized" arctan for a discretization with 6 grid points.

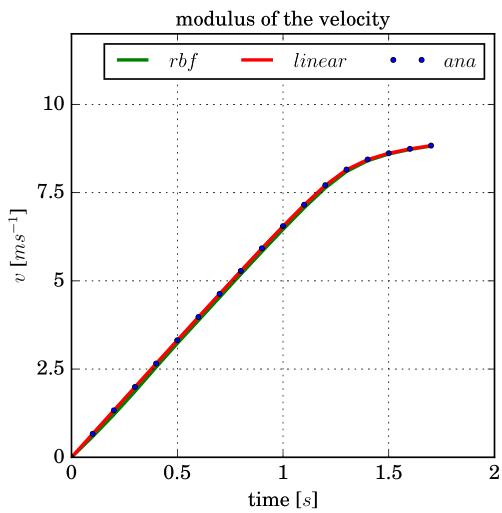


FIGURE 4.32: Test-Case 2: The modulus of the point's velocity in time during motion on a surface defined using a "normalized" arctan for a discretization with 21 grid points.

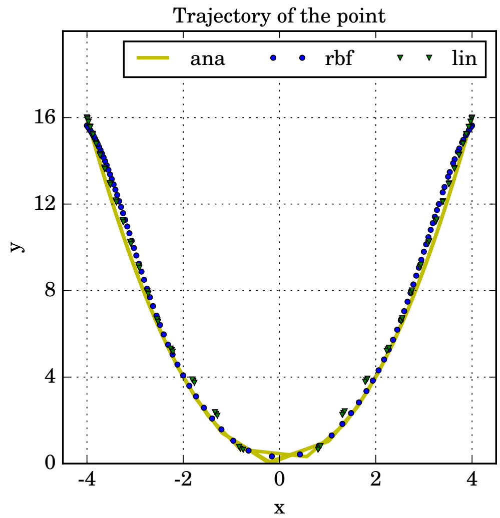


FIGURE 4.33: Test-Case 3: Trajectories for the parabola for a discretization with 6 grid points.

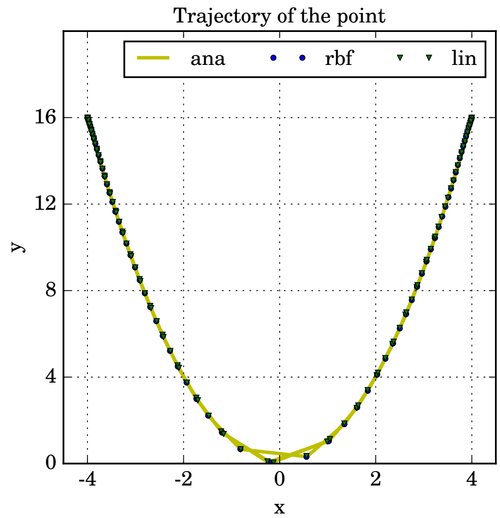


FIGURE 4.34: Test-Case 3: Trajectories for the parabola for a discretization with 21 grid points.

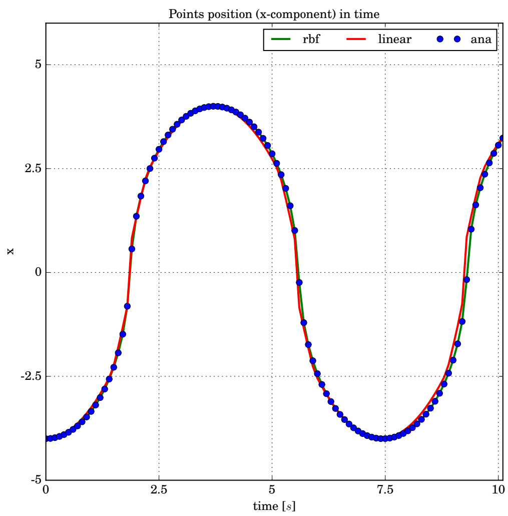


FIGURE 4.35: Test-Case 3: x-component of the point in time during motion on a parabolic surface for a discretization with 6 grid points.

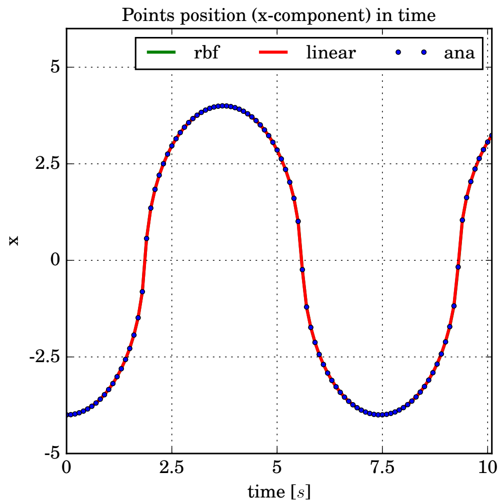


FIGURE 4.36: Test-Case 3: x-component of the point in time during motion on a parabolic surface for a discretization with 21 grid points.

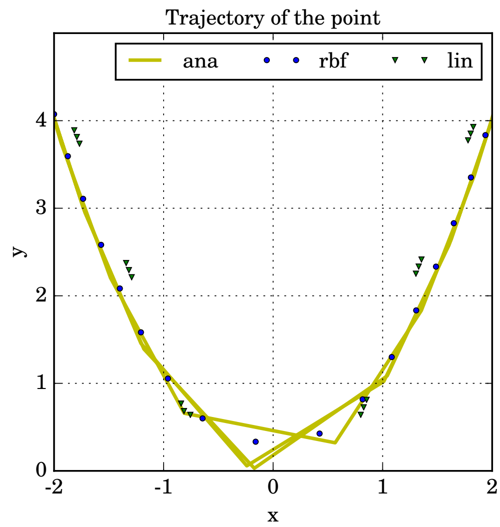


FIGURE 4.37: Test-Case 3: Zoom on the trajectory of the point in time during motion on a parabolic surface for a discretization with 6 grid points. Zoomed area is $[-2,2] \times [0,4]$.

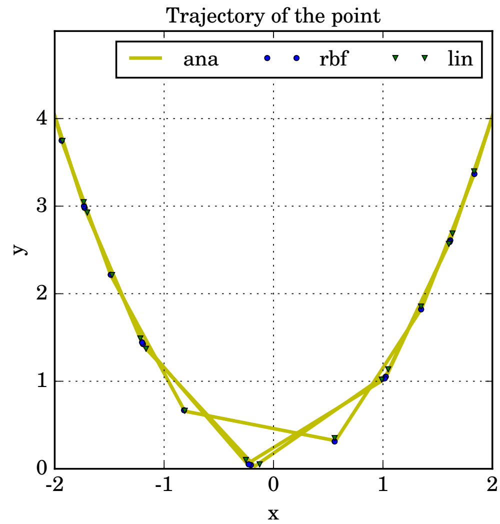


FIGURE 4.38: Test-Case 3: Zoom on the trajectory of the point in time during motion on a parabolic surface for a discretization with 21 grid points. Zoomed area is $[-2,2] \times [0,4]$.

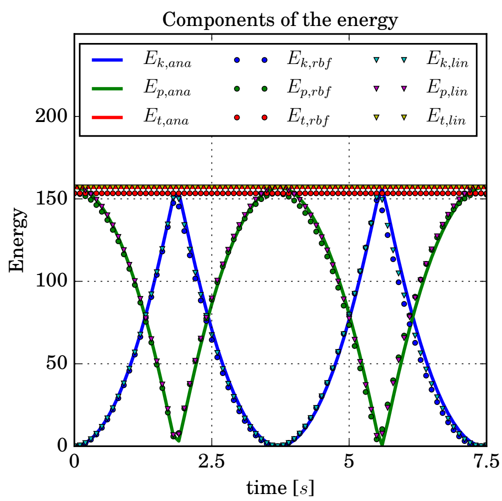


FIGURE 4.39: Test-Case 3: Kinetic, potential and total energy for the simulation on a parabolic surface for a discretization with 6 grid points.

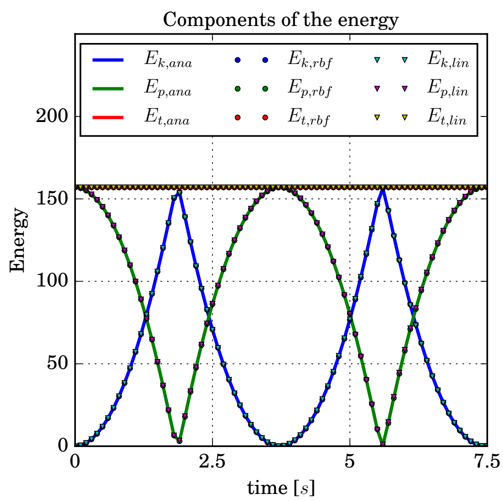


FIGURE 4.40: Test-Case 3: Kinetic, potential and total energy for the simulation on a parabolic surface for a discretization with 21 grid points.

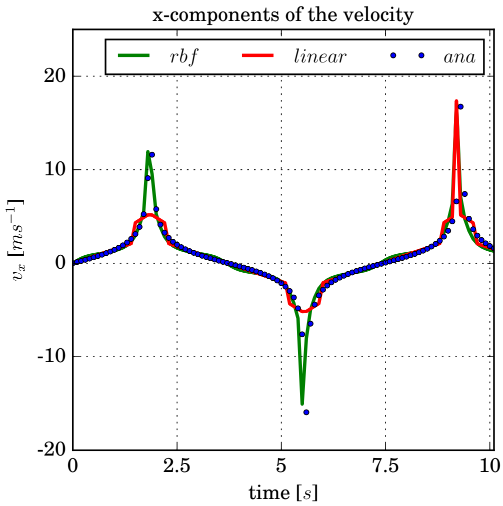


FIGURE 4.41: Test-Case 3: x-component of the point's velocity in time during motion on a parabolic surface for a discretization with 6 grid points.

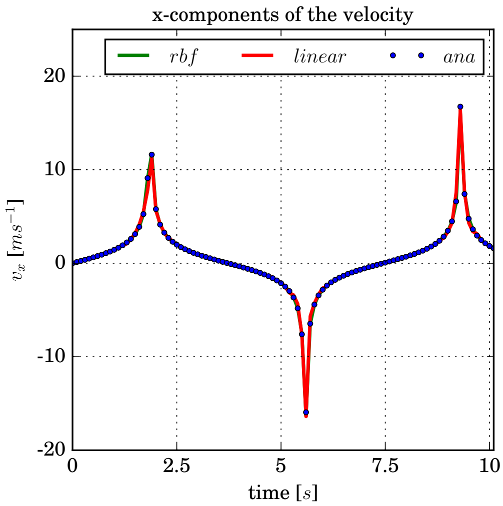


FIGURE 4.42: Test-Case 3: x-component of the point's velocity in time during motion on a parabolic surface for a discretization with 21 grid points.

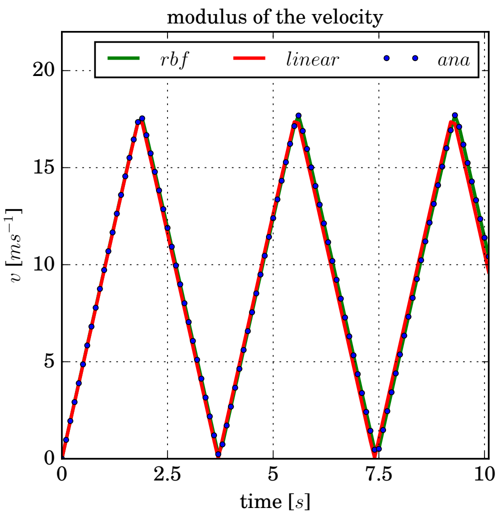


FIGURE 4.43: Test-Case 3: The modulus of the point's velocity in time during motion on a parabolic surface for a discretization with 6 grid points.

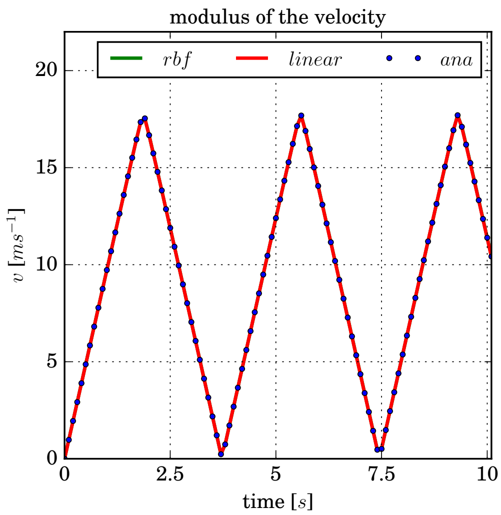


FIGURE 4.44: Test-Case 3: The modulus of the point's velocity in time during motion on a parabolic surface for a discretization with 21 grid points.

Test-case 4

$$f(x,y) = x\arctan(\varphi x) - \log(\varphi^2 x^2 + 1)/(2\varphi),$$

The sliding surface here is very similar to the one in test-case two, especially as also here the first derivative changes very quickly in $x = 0$, whilst it is almost constant elsewhere. This can also be seen in the results of the reconstruction, where the RBF-reconstruction can capture the real behaviour less good than the piecewise-linear can, at least on the coarse grid. For the finer grid the simulations show, as already before, almost no difference.

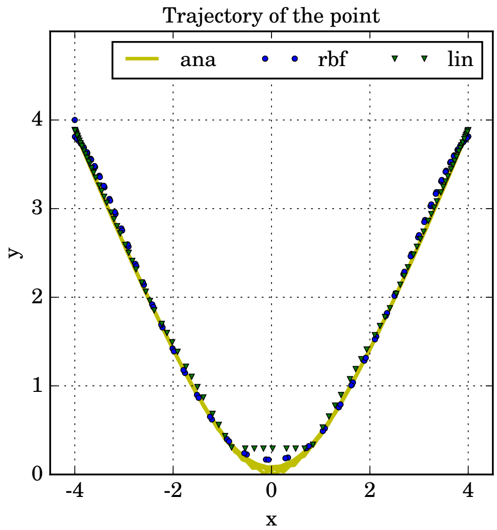


FIGURE 4.45: Test-Case 4: Trajectory on the sliding curve defined using an arctan for a discretization with 6 grid points.

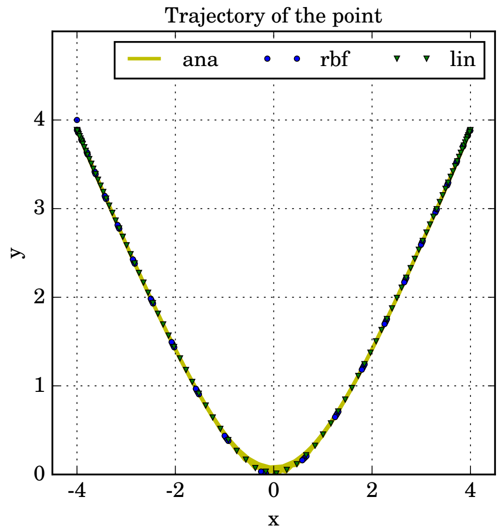


FIGURE 4.46: Test-Case 4: Trajectory on the sliding curve defined using an arctan for a discretization with 21 grid points.

4.3.3 Comparison between the solution on piecewise linear curves and the numerical solution on smooth reconstructed surfaces

In cases where the inclination of the function is almost constant over a long interval (which also means that the function itself is quite linear) and then changes rapidly (but within a short interval), the piecewise-linear reconstruction is performing better on coarse grids as they naturally imitate this behaviour. It lies in the nature of the RBF-reconstructions, as they are designed to smooth a surface, that they are performing less good for those cases. These differences however vanish with increasing refinement of the grid.

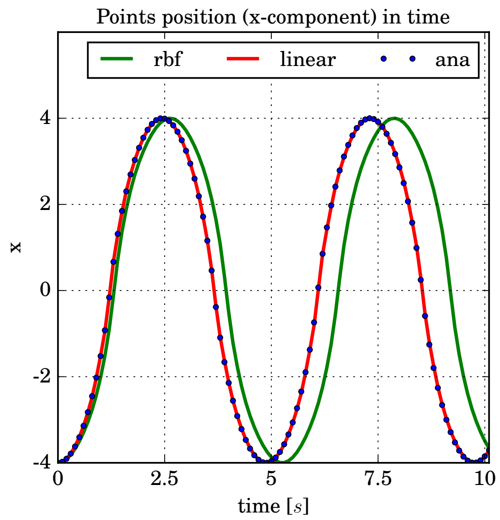


FIGURE 4.47: Test-Case 4: x-component of the point in time during motion on a surface defined using an arctan for a discretization with 6 grid points.

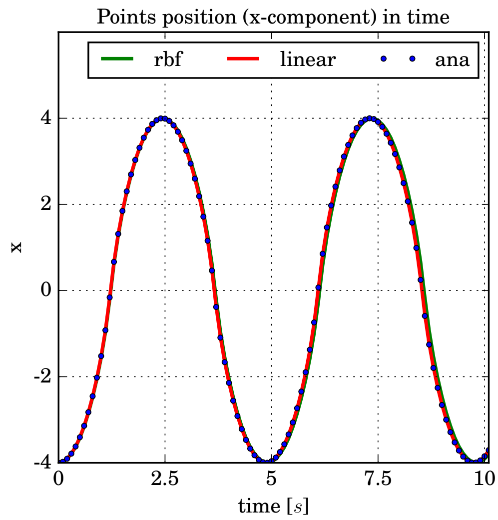


FIGURE 4.48: Test-Case 4: x-component of the point in time during motion on a surface defined using an arctan for a discretization with 21 grid points.

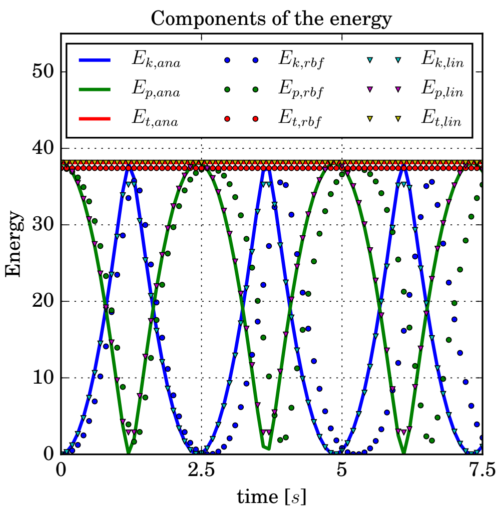


FIGURE 4.49: Test-Case 4: Kinetic, potential and total energy for the simulation on a surface defined using an arctan for a discretization with 6 grid points.

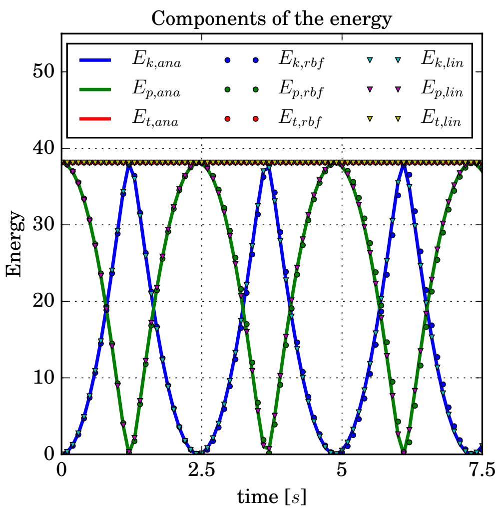


FIGURE 4.50: Test-Case 4: Kinetic, potential and total energy for the simulation on a surface defined using an arctan for a discretization with 21 grid points.

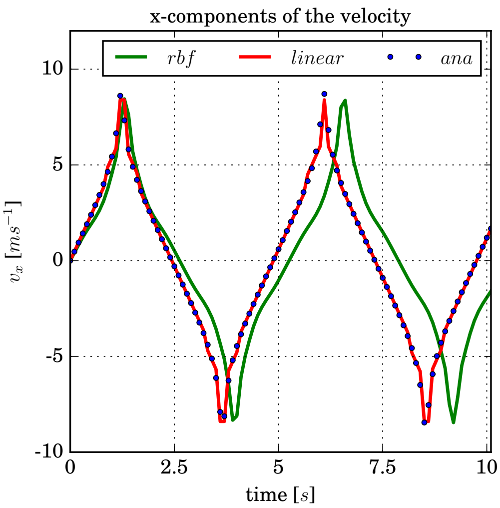


FIGURE 4.51: Test-Case 4: x-component of the point's velocity in time during motion on a surface defined using an arctan for a discretization with 6 grid points.

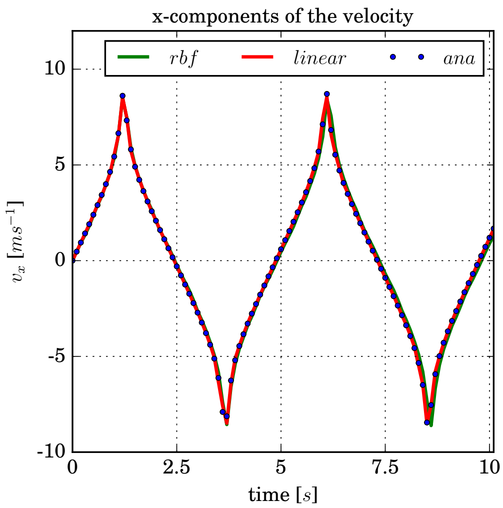


FIGURE 4.52: Test-Case 4: x-component of the point's velocity in time during motion on a surface defined using an arctan for a discretization with 21 grid points.

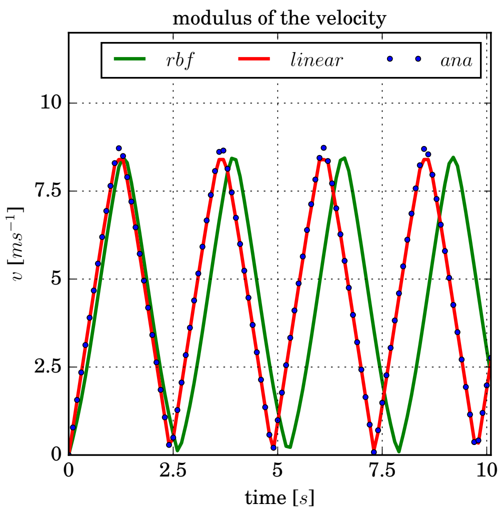


FIGURE 4.53: Test-Case 4: The modulus of the point's velocity in time during motion on a surface defined using an arctan for a discretization with 6 grid points.

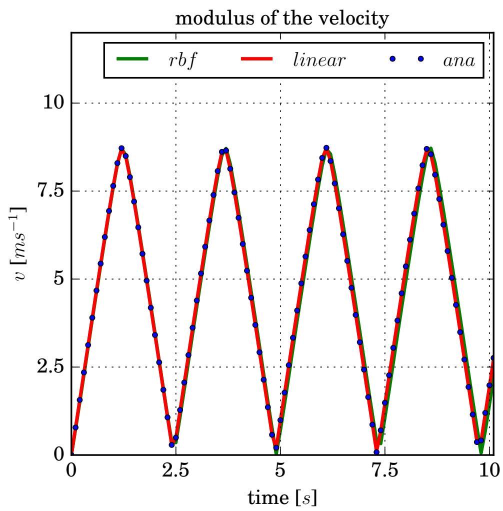


FIGURE 4.54: Test-Case 4: The modulus of the point's velocity in time during motion on a surface defined using an arctan for a discretization with 21 grid points

Piecewise linear curve + exact time-integration:	**Smooth curve + numerical time-integration:**
• Careful choice of data-points for the piecewise-linear approximation necessary; • identification of intersection-point & -time and new triangle will be more difficult in 2F; • might need grid refinement to reconstruct local behaviour; • simple time-integration; • no restriction to step size.	• Smooth surface reconstruction (min C_2) needed;, • requires solver for stiff systems (lsode, computationally expensive); • small time-step (depending of stiffness) needed; • might better take on the behaviour of the real surface; • no segment-identification necessary (in 2D triangle).

On the contrary, the RBF-reconstructions perform better on smooth data as they can capture the behaviour of the underlying functions. In the third test case (parabola), the function is not longer linear and the inclination is changing linearly. This is not as good imitated by a piecewise-linear reconstruction. However, the parabola chosen here is not a very extreme example such that also the piecewise-linear reconstruction can still perform well.

The results also show, how important a careful choice of the discretization is. Too rough grids in regions of strong variation will unavoidably lead to significant errors in the solution. Instead, if the discretization of the curve is chosen with care, both methods perform comparably good in terms of accuracy. The advantages and disadvantages of the two methods are displayed in the table below.

As altogether the solution on piecewise linear curves is computationally much less expensive, this is the method of choice in the following.

4.4 Results 2: Convergence tests and more complex surfaces

This section is dedicated to convergence tests on one hand and the evaluation of the goodness of the model on more complex curves on the other hand. In the following different analytically given C^2 curves (denoted as curves $\mathcal{C}$ before) are approximated by discretized piecewise curves of type Σ. The solutions obtained using the presented method (named *"UBO-block solution"*) were compared with the numerical solution obtained on the analytical surfaces (called *"reference solution"*), using the Livermore-Solver for stiff ODEs Hindmarsh and Radhakrishnan, 1993 for the time-integration. The stiffness of the problem is also the reason for which at least continuity in the second derivative of C is required. The smoother the second derivative, the better the reference solution will be.

Note that the reference solution is not necessarily the exact solution of the system, as in general this is not known. We assume however that the reference solution is sufficiently close to the unknown exact solution of the system on the curve $\mathcal{C}$. The difference between our solution and the reference solution will – for convenience – be called the *error*, even though it is not an error in the usual sense. Instead it shall be an expression for the differences between the solutions on the piecewise linear curves and the reference solutions, measured as the Euclidean distances of the point-masses on the two curves at each time-step.

A single time-step is denoted by a subscript i, and M is the total number of time-steps, while the Euclidean distance is denoted by $\delta\mathbf{x}_i$ (see below). The errors are measured in the L_2- and L_∞-norm, where the first gives an average and the second a maximal error, from which the corresponding estimated convergence rates can be computed. The L_2- and L_∞-errors will be denoted by e_2 and e_∞, where for example $e_2(T_N)$ denotes the error in the L_2-norm for a discretization of N linear segments, the corresponding numerical convergence rates are denoted by $\mu_2(T_N)$ and $\mu_\infty(T_N)$. Note that the computation of the convergence rates depends on if the exact solution is known or not and on the refinement steps. Usually, and this is what is also done here, the refinement is done using a factor *two*. The following equations for μ_2 and μ_∞ only hold if an exact/reference solution is known.

$$e_2 = \frac{1}{M}\sqrt{\sum_{i=1}^{M}\delta\mathbf{x}_i^2}, \qquad \mu_2(T_N) = \log_2\left(\frac{e_2(T_{N-1})}{e_2(T_N)}\right),$$

$$e_\infty = \max_{i=1,M}\delta\mathbf{x}_i, \qquad \mu_\infty(T_N) = \log_2\left(\frac{e_\infty(T_{N-1})}{e_\infty(T_N)}\right),$$

where:

$$\delta\mathbf{x}_i = ||\mathbf{x}_i^d - \mathbf{x}_i^r||,$$

and the superscript d denotes the discretized and r the reference solution.

So, the L_∞-error is the maximum distance between the reference solution and the solution on the piecewise linear curves, evaluated over all time steps, whereas the L_2-error gives the average distance between the two solutions, measured also over all the simulated time. Using these two error-norms we will compute the convergence rates for different test cases. Simulations were run with an increasing number of line segments, defining the discretization Σ of the analytic curve $\mathcal{C}$, where N (the number of segments) was doubled in the consecutive simulations.

The two test cases presented in the following seemed the most significant for us. First, in order to analyse the convergence of the method, a half circle curve was chosen. This curve is symmetric by definition and can easily be discretized in a way that all linear segments are of equal length (equidistant along the curve). Moreover, as it has constant curvature, this discretization also ensures that the angles between two consecutive segments are always equal. Finally, in case of frictionless motion the exact solution is known (mathematical pendulum). For the second test case we chose a more complex one, showing convex and concave segments as well as strong changes in the first and second derivatives. In addition, we find a number of local maxima and minima, allowing us to study also oscillations along the curve. All distances are given in $meters$, times in $seconds$, velocities in m/s and accelerations in m/s^2.

4.4.1 Circular motion (convergence test)

In the following three examples the equations are solved on a half circle of unit radius where the curve $\mathcal{C}$ is given by:

$$f(x) = \sqrt{1-x^2}+1, \quad x \in \left[\cos\left(\pi + \frac{\pi}{32}\right), \cos\left(2\pi - \frac{\pi}{32}\right)\right].$$

The above restriction of the domain had to be chosen in order to avoid undefined first derivatives in the smooth surface, as would be the case for $x = \cos(\pi)$ and $x = \cos(2\pi)$. In these two points the tangents to the curve would be vertical, thus requiring a free fall of the point. While this is not a problem in our method, it is not supported in the numerical solution of the ODE-system which requires analytically given derivatives. Using the domain given above, we can avoid these singularities in the reference solution. The total length of the curve is therefore given by $\ell_C = \frac{15}{16}\pi$. By ℓ_t the total length of the trajectory will be denoted. The initial conditions are given by

$$(x_0, z_0) = \left(\cos\left(\pi + \frac{\pi}{32}\right), \sin\left(\pi + \frac{\pi}{32}\right)\right)$$
$$(v_{x,0}, v_{z,0}) = (0,0).$$

TABLE 4.2: Error and order of convergence for the solution to the full equations on a half circle

N	L_2-error	L_2-order	L_∞-error	L_∞-order
16	3.28e-03	-	9.80e-03	-
32	8.87e-04	1.89	2.69e-03	1.86
64	2.11e-04	2.07	6.00e-04	2.16
128	5.17e-05	2.03	1.52e-04	1.98
256	1.54e-05	1.74	4.14e-05	1.88

TABLE 4.3: Errors (in percent %) relative to the total length ℓ_C of the curve for the solution of the full equations on the half circle.

N	L_2-error (%)	L_∞-error (%)
16	0.1114	0.3327
32	0.0301	0.0910
64	0.0072	0.0203
128	0.0017	0.0052
256	0.0005	0.0014

Even though this is not exactly a *half circle* we will – for convenience – call it this way throughout the following. In order to conserve the angle of the circular segment on which the point mass is moving, we chose an ”outer discretization” of the arc. This means that the linear segments of Σ are tangential to the analytical curve in the given data-points and implies that Σ is longer than $\mathcal{C}$, but has the advantage of conserving the total angle. We use different discretizations T_N by approximating the half-circle by N equal tangential segments. We now go into detail with the different test-setups.

a) Full equations In the first test we solve the full Eq. 4.2 for $\mu = 0.1$. Fig. 4.55 (top) shows the position on the x-axis of the point-mass vs. time (left scale) as well as the differences between the reference and the solutions on the piecewise linear curve, measured as the L_2-distances (right scale). The point oscillates along the curve, steadily decreasing in speed, and stops after *two and a half* full periods, corresponding to an approximate total length of $\ell_t \approx 1.4\pi$ of the trajectory.

As can be seen also from the errors (dotted curves), the solutions on the piecewise linear curve differ very little from the reference solution such that they are overlapping in the figure. The errors and convergence-rates of the discretized towards the reference solution are given in Table 4.2. Here it can be seen that the solutions on the piecewise linear curve converge towards the reference solution with an order of convergence of about *two*. The errors between the two solutions in relation to the total length of the smooth curve in *percent* are given in Table 4.3.

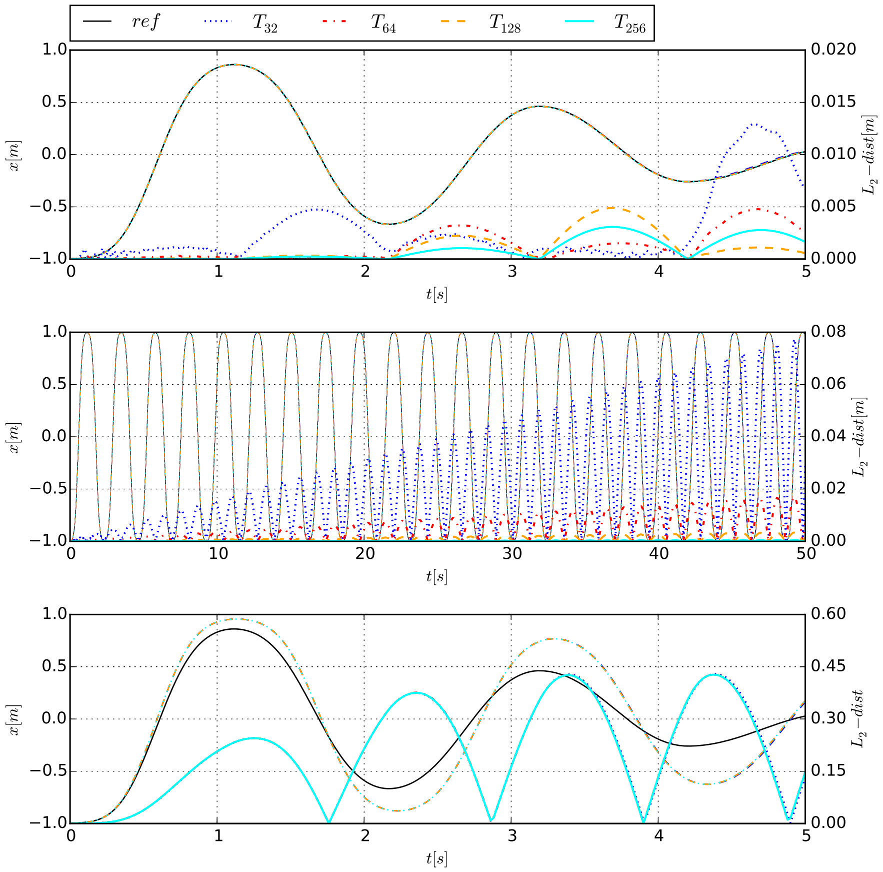


FIGURE 4.55: First Test Case: Motion along a half circle. Top: Full equations; Middle: Frictionless motion; Bottom: Omitting curvature dependent friction; *x*-components of the point trajectories on different discretizations in time (left scale) and difference to the reference solution (right scale, dotted lines) are shown. Notice that in all cases the trajectories are overlapping and therefore indistinguishable

TABLE 4.4: Convergence rates for frictionless motion on a half circle in the first 20 full periods

N	L_2-error	L_2-order	L_∞-error	L_∞-order
16	2.61e-03	–	2.27e-01	–
32	6.09e-04	2.10	7.65e-02	2.10
64	1.30e-04	2.22	1.62e-02	2.24
128	2.20e-05	2.60	2.58e-03	2.65
256	1.41e-06	3.96	2.69e-04	3.26

TABLE 4.5: Errors (in percent %) relative to the total length ℓ_C of the curve for the solution of the frictionless equations on the half circle

N	L_2-error (%)	L_∞-error (%)
16	8.86e-02	7.70e-00
32	2.06e-02	2.60e-00
64	4.41e-03	5.50e-01
128	7.50e-04	8.76e-02
256	5.00e-05	9.10e-03

b) Frictionless motion In this example we solve the equations for $\mu = 0$, meaning frictionless motion, in order to show how the two solutions evolve in time in case that the motion is not decaying. It follows that theoretically the point should steadily oscillate along the curve, always reaching the extrema $\cos(\pi + \pi/32)$ and $\cos(2\pi - \pi/32)$ with velocity $|\mathbf{v}| = 0$. The distance between the points on the two curves is measured in the L_2-norm at each time step. The simulation was run for $50s$, which corresponds to little more than 20 full periods. The errors in time are displayed in Fig. 4.55 (middle)). One can see that the errors are linearly increasing, though they are not amplifying in time. This behaviour is explained through the different lengths of the discretized curves which also lead to differing arrival times at the reversal points $\cos(\pi + \pi/32)$ and $\cos(2\pi - \pi/32)$ of the curve. The result is a shift in time of the solution on the piecewise linear curve compared to the reference solution which for each single period is equal to all the others. With time this simply adds up to the final error but cannot amplify. For $\lim_{N\to\infty} \ell_{T_N} \to \ell_C$ and therefore the errors are going to *zero*. The convergences rates measured in the L_2- and L_∞-norm are about 2 as can be seen from Table 4.4, the corresponding relative errors are given in 4.5. In this case also the exact solution (mathematical pendulum) is known and therefore the errors and convergence rates of the solution on the piecewise linear curve towards the exact solution can be computed as well as the error of the reference solution. The results are shown in Tables 4.6 and 4.7. The error between the reference and exact solution is comparable to the one for solution on the piecewise linear curve with $N = 256$ where the latter is even a little better. It follows that we cannot find better results for

TABLE 4.6: Convergence rates for frictionless motion on a half circle towards exact solution for first 50s

N	L_2-error	L_2-order	L_∞-error	L_∞-order
ref	4.48e-06	-	4.21e-04	-
16	2.61e-03	-	3.28e-01	-
32	6.13e-04	2.09	7.73e-02	2.09
64	1.35e-04	2.18	1.70e-02	2.18
128	2.66e-05	2.34	3.34e-03	2.34
256	3.99e-06	2.74	5.02e-04	2.74

TABLE 4.7: Errors (in percent %) relative to the total length ℓ_C of the curve for the solution of the frictionless equations on the half circle towards exact solution for first 50s

N	L_2-error (%)	L_∞-error (%)
ref	1.63e-06	2.61e-04
16	8.86e-04	1.11 e-01
32	1.81e-04	2.62e-02
64	4.58e-05	5.77e-03
128	9.03e-06	1.13e-03
256	1.35e-06	1.70e-04

even finer discretizations when comparing it to the reference solution as they would start to diverge again.

c) Influence of the different friction components In the last example the importance of curvature-dependent friction component on the solution of the equations of motion will be pointed out. While the curvature-independent component of the friction term is acting along the whole surface and hence it seems obvious to incorporate it in the computations, the curvature-dependent component in case of a piecewise linear surface – due to the missing second derivative – is not present, and can play a role only in the moment of changing from one linear segment to another. Moreover, in the point of change of slope, the second derivative, and hence curvature-dependent components, are not defined. This difficulty might be a reason for curvature dependent friction being neglected in some numerical landslide models, however, we will show here that the results can become highly inaccurate if doing so.

In the following the reference solution of the full equations is compared to the solution on the piecewise linear curve, not taking into account the curvature-dependent component of the friction, that is by omitting to change the point velocity at each node through the modification factor Γ. The result can be seen in Fig. 4.55

(bottom), showing how strong the errors can increase in time if only gravity dependent friction is considered. The reference solution is depicted by a solid, black line, the solutions for different grid resolutions of the piecewise-linear curve are given by dashed lines and the differences between the reference solution and the solutions on piecewise-linear curves are given by dotted lines. It is seen immediately that the reference solution decays much faster than the solution on the piecewise linear curve. While in the reference solution the motion stops after $5s$, it takes $7.35s$ if curvature dependent friction is not considered, meaning that the motion needs about 47% more time to stop.

4.4.2 Parabola with additional roughness

This test is based on the test surface used in chapter 4.1 to illustrate the non-conservation of total energy. The aim is to show that now the total energy can be conserved even for a comparably rough surface. The 1-dimensional test-curve is depicted in Fig. 4.56. The underlying function is a parabola with the same spatial dimensions as the

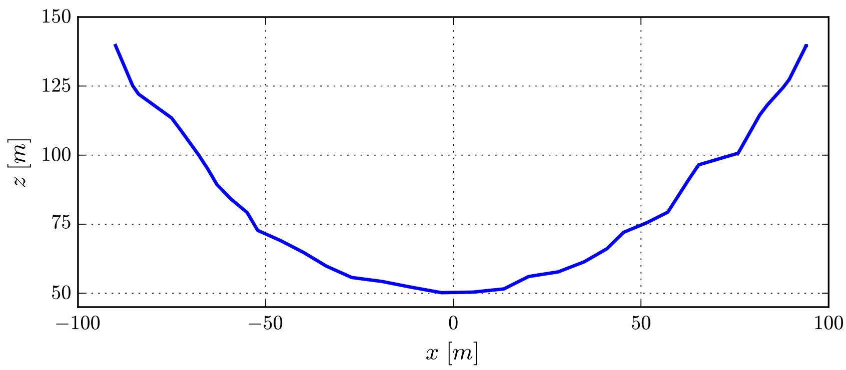


FIGURE 4.56: The sliding surface used for the problem.

test-surface from chapter 4.1 but roughness was added in order to theoretically make the curve more "complicated" for the model to treat, meaning: due to the additional roughness, the function contains several significant changes in slope throughout the whole curve. Not correcting the acceleration at each transition point would lead to noticeable errors in the total energy but also small inaccuracies in the method or the implementation are more likely to become visible than for a less rough curve. Fig. 4.57 shows the trajectory (i.e. x- and z-component) in time. Fig. 4.58 depicts the velocity components v_x and v_z in time that show noticeable fluctuation caused by the roughness of the surface. Fig. 4.59 finally shows the total energy as well as its decomposition into kinetic and potential energy plotted in time. It can be seen that despite the roughness of the curve and the resulting fluctuations in velocity, the total energy of the system is conserved (the error relative to the total energy is about $1.083 \cdot 10^{-5}\%$).

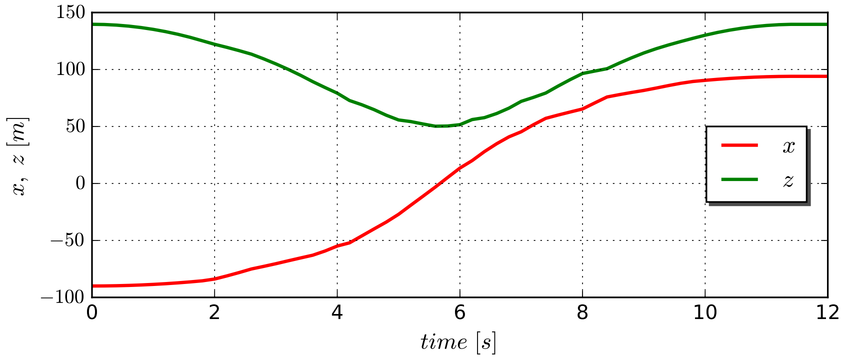


FIGURE 4.57: The x- and z-components of the points position in time.

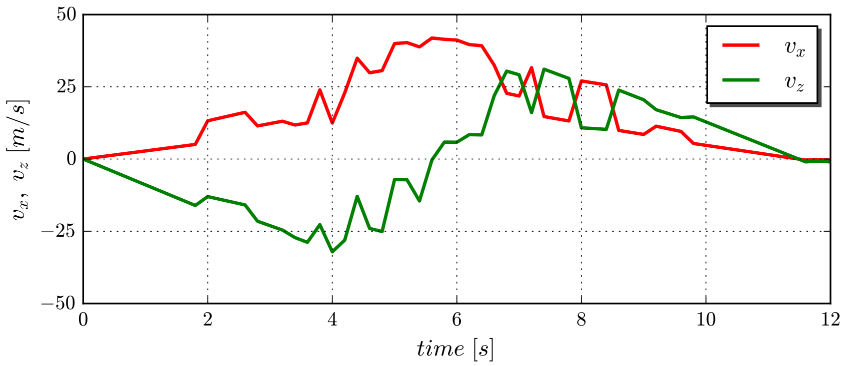


FIGURE 4.58: The x-and z-components of the velocity for the points position in time.

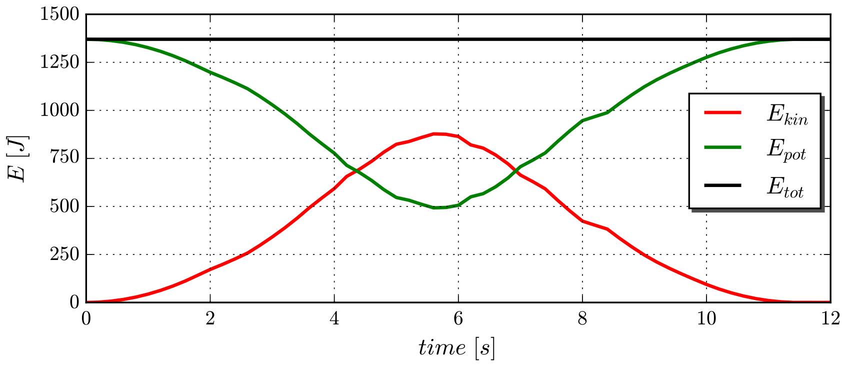


FIGURE 4.59: The kinetic, potential and total energy in time. Total energy is conserved.

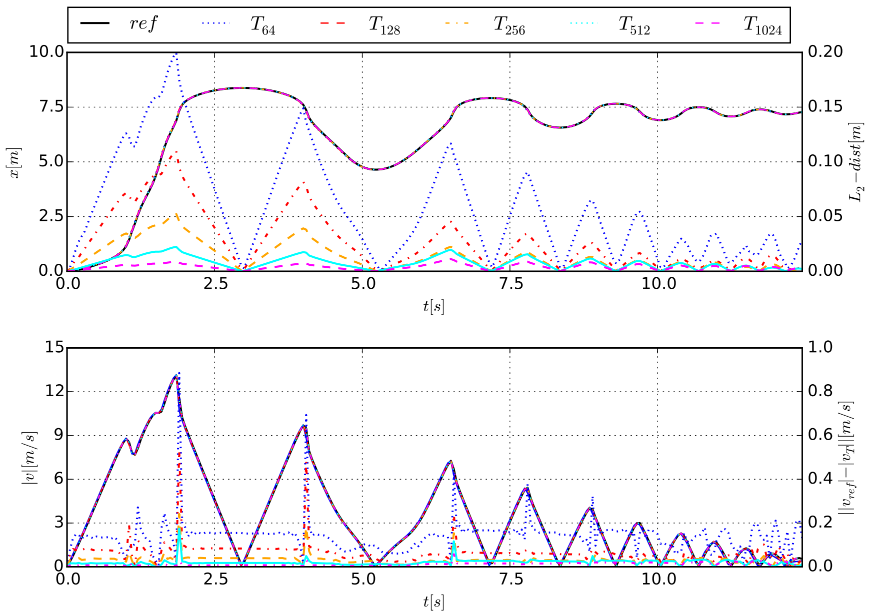


FIGURE 4.60: Test case: High-order polynomial, full equations over 12.5s

4.4.3 High-order polynomial

One more test was done for a more complex surface given by a polynomial of degree eight, where:

$$f(x) = \sum_{i=0}^{8} c_i x^i,$$

with coefficients c_i and $f : [0,9] \to [0,10]$. The polynomial was constructed using a second polynomial of degree *eight*, denoted by $\bar{f}(x)$ with:

$$\bar{f}(x) = \sum_{i=0}^{8} \bar{c}_i x^i,$$

where the coefficients c_i written in vector-form are:

$$\bar{c} = [-123.0, 353.53, -369.96913, 200.962114, -63.19811,$$
$$11.926022, -1.330871, 0.080853156, -0.002058209].$$

Now the function is scaled (approximately) to the range $[0,9] \times [0,10]$ using $dx = 1.3$ and $s = (9.0 - dx)/9.24$. The new coefficients c are obtained as follows:

$$c = \frac{10\bar{c}}{\bar{f}(dx)}.$$

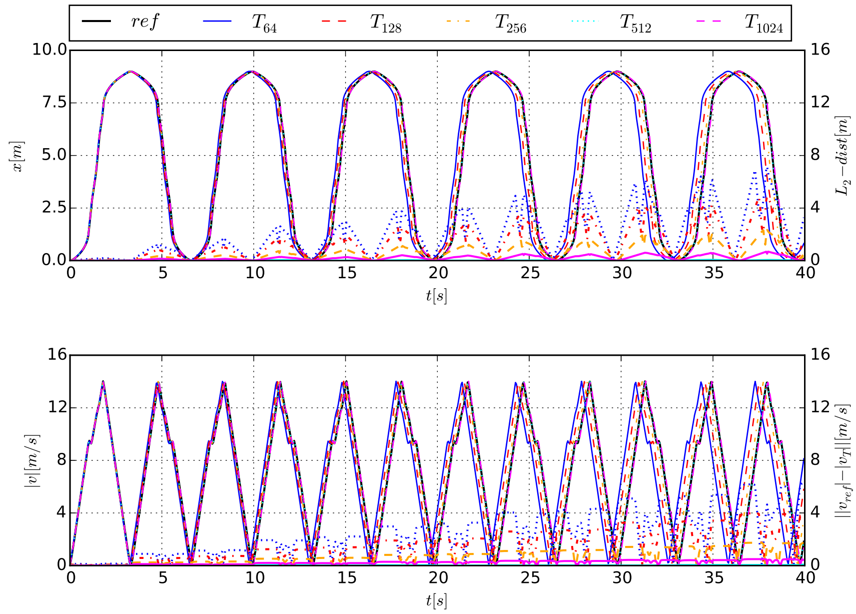


FIGURE 4.61: Test case: High-order polynomial, frictionless motion in the first 40s

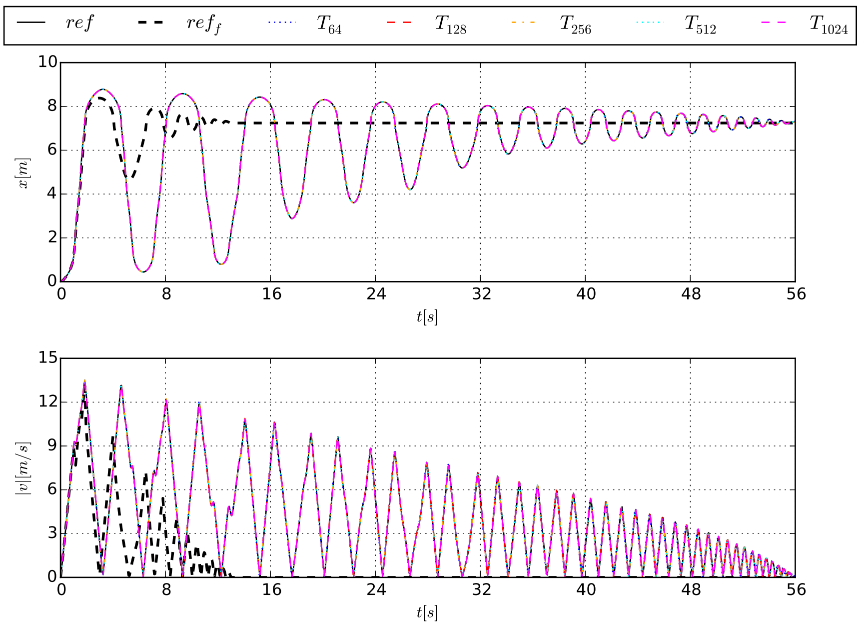


FIGURE 4.62: Test case: High-order polynomial, full equations (reference solution) and omitting curvature dependent friction (solution on the piecewise linear curve). Differences in speed not plotted here

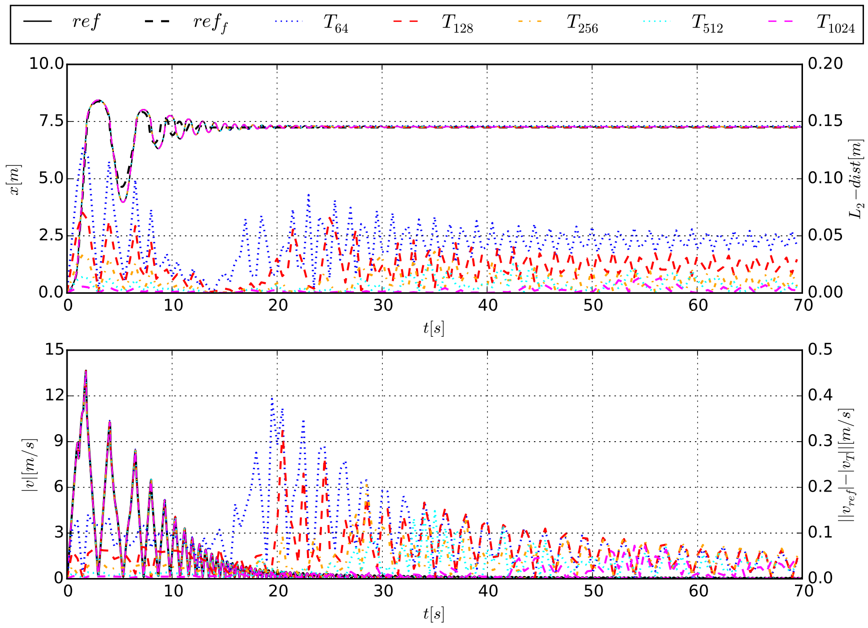


FIGURE 4.63: Test case: High-order polynomial, full equations (reference solution) and omitting gravity dependent friction (solution on the piecewise linear curve)

The final polynomial – scaled to the desired domain – then reads:

$$f(x) = \sum_{i=0}^{8} c_i (sx + dx)^i.$$

This polynomial is displayed in Fig. 4.64. The total length of the curve is $\ell_C \approx 23.83562$. The curve changes between concave and convex six times and has one local minimum and maximum. Restricting to the domain in which f is defined, the global minimum is located at $x \approx 7.3$ and global maxima are located at $x = 0$ and $x = 9$ and have the same value, such that the motion of a point mass in the frictionless case covers the whole domain. Seen from left to right (increasing x) the curve is characterized by a long bumpy slope, followed by a very steep rise, connected through a valley of small radius, thus leading to strong centripetal acceleration.

As in the previous example, we will show the results for the full equations, for frictionless motion and in case the influence of the curvature dependent friction is neglected. The initial conditions are given by

$$(x_0, z_0) = (0.0, f(0.0))$$
$$(v_{x,0}, v_{z,0}) = (0, 0).$$

The data-points are chosen equidistant in x-direction. While this is not the best choice in terms of performance it is very general and simplifies the grid refinement. In order

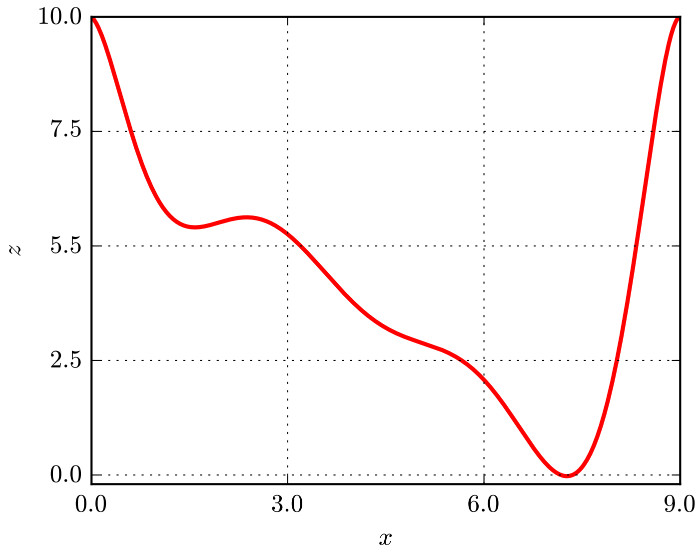


FIGURE 4.64: Sliding surface: polynomial of degree *eight*

to keep the discretization simple, we just connected the given data-points by linear segments. The number of segments ranges from 64 to 1024 and was doubled in each step. The corresponding discretizations are denoted by T_{64} with $\ell_{T_{64}} \approx 23.80921$ to T_{1024} with $\ell_{T_{1024}} \approx 23.83545$ where generally ℓ_T denotes the length of a certain discretized curve. It follows that compared to the length of the reference curve the relative error in total length ranges from 0.1108% to 0.007%. The results of the computations are displayed in Figs. 4.60-4.63. All figures show on the top the x-component of the point trajectories on different discretizations in time (left scale) and the differences to the reference solution (right scale, dotted lines). On the bottom the modulus of the velocity in time (left scale) and the differences to the reference solutions are displayed (right scale, dotted lines).

a) Full equations As in the previous example, the first test for the polynomial surface is the solution of the full equations with $\mu = 0.1$. Results are shown in Fig. 4.60. The motion is followed until the point finally stops after about $12.5s$. It can be seen that the solutions on the piecewise linear curve are very close to the reference solution and even the drop of velocity in the local minimum is very well captured. The L_2-error relative to the length of the curve ℓ_C varies between 0.0195% (T_{64}) and 0.0012% (T_{1024}), whereas the relative L_∞ varies between 0.837% and 0.047%.

b) Frictionless motion Also in the frictionless case, displayed in Fig. 4.61 we find that the solutions are converging if the discretized curve converges to the analytical one. Only for very fine discretizations the error is increasing again, but, as also the reference solution is not exact we cannot finally tell if this is more due to an error in the discretized or the reference solution.

Nonetheless, neither the larger errors for coarser discretizations nor the ones for very fine discretizations are too important in the real world applications as the they are only due to a time shift of the motion, meaning that the point is arriving slightly later than it would on the reference surface. Much more important is the general behaviour of the motion, especially the velocities in certain points and these do coincide very well when not considering the time-shift. As the errors are not really amplifying, but just summing up (and showing this was the main aim of this test case), we can take a look at only the first full period of motion where it can be seen best that all the characteristic points, like the maximum height of the point or its maximal velocity are really well captured by all the solutions on the piecewise linear curves. Especially the errors in the velocity are very small and this is one of the most crucial points for us.

The errors relative to the length of the curve ℓ_C vary between 0.3945% (T_{64}) and 0.0029% (T_{1024}) for the L_2-error, and between 29.919% (T_{64}) and 0.241% (T_{1024}) for the L_∞-error.

c) Influence of the curvature and gravity dependent friction In this example we want to point out the strong influence of curvature dependent friction illustrated in Fig. 4.62, but also the influence of gravity dependent friction was investigated and is shown in Fig. 4.63. This was done to visualize better the different effects of the two friction components. Therefore we once computed the motion while omitting curvature dependent friction (c) and once leaving out gravity dependent friction (d). Again, in both cases the reference solution for the full equations is plotted next to the solutions on the piecewise linear curves. The reference solutions for the "incomplete" equations are plotted too, but are totally overlapped by the curves of the solutions on the piecewise linear curves. In this example we can clearly see, that the two components account for totally different behaviours within the decaying of the motion. While in both cases the run-times of the simulations are significantly longer, leaving out curvature dependent friction leads to a quite slow decaying solution: it stops after $56s$ which is about 4.2 times the time needed using the full equations. Leaving out the tangential component instead does not, on a first glance, change the behaviour of the motion too much. After about $15s$, only 1.12 times the time of the full equations, there is no more significant motion, but we still find tiny oscillations in the motion that persist even after $70s$. So, in order to get the correct behaviour of the motion, we really need both components of the friction. Omitting one will always lead to errors, even though these errors heavily depend on the kind of surface. While on plane surfaces with few changes in curvature, curvature dependent friction

TABLE 4.8: CPU-times for the discretized (left) and numerical solution (right) for different discretizations resp. accuracies

N	L_2-error	L_∞-error	RT	tol_a	tol_r	$\triangle t$	L_2-error	L_∞-error	RT
16	2.61e-03	3.28e-01	0.301	10^{-5}	10^{-4}	0.02	3.54e-03	5.49e-01	10.41
				10^{-5}	10^{-4}	0.01	1.11e-03	2.11e-01	11.17
32	6.13e-04	7.73e-02	0.618	10^{-6}	10^{-5}	0.02	7.98e-04	8.45e-02	11.41
				10^{-6}	10^{-5}	0.01	4.65e-04	6.69e-02	12.38
64	1.35e-04	1.70e-02	1.237	10^{-7}	10^{-6}	0.005	2.36e-04	2.10e-02	18.81
				10^{-7}	10^{-6}	0.0025	1.01e-04	1.05e-02	23.79
128	2.66e-05	3.34e-03	2.471	10^{-8}	10^{-7}	0.0005	2.21e-05	2.11e-03	66.46
256	3.99e-06	5.02e-04	4.947	10^{-9}	10^{-8}	0.0001	4.48e-06	4.21e-04	258.72

obviously plays a minor role compared to gravity dependent friction, its importance is increasing with the complexity of the surface.

All together, the above results show how important it is to capture the changes in slope and curvature and to really solve the full equations. It also shown how well the decoupling of the equations into curvature-and gravity-dependent components works.

4.4.4 Runtime-comparison

Although a comparison of an exact method with a numerical one is not really fair to do, we want in the following to give a comparison of the runtimes of the two methods presented above. Both programs were mainly written in Fortran 90/95, but using also some Fortran 2003-standard. The two programs are both structured in the same way, sharing the same main-program and functions to compute for example the accelerations. They only differ in their time-integration method. For the numerical solution the highly optimized Fortran-routines written by A. Hindmarsh et al. Hindmarsh and Radhakrishnan, 1993 where used unmodified.

As test-case we chose the half circle with no friction and a final time of $50.0s$. As the run-times for this example are too short to directly compare them, the simulation was run 10,000 times in a row in order to achieve more reliable values. Within this cycle only the time-integration is executed, where before each single cycle the initial values were reset. All preliminary computations as well as allocations were done outside the measured cycle.

The comparison was done as follows: using the errors for the solution on the piecewise linear curve given in Table 4.6 we chose – for the numerical (reference) solution – a combination of tolerances and step-size that minimize the run-time resulting in errors as close as possible to the ones of the solution on the piecewise linear curve (for the first three discretizations two values for the numerical solution are given). The run times were measured five times and averaged. Table 4.8 gives the run-times (RT) in seconds (measured as CPU-time) for both solutions (for different accuracies) as well as the chosen tolerances (absolute tol_a and relative tol_r, for more information see the LSODE documentation Hindmarsh and Radhakrishnan, 1993) and time-steps (in seconds) for the numerical solution.

Even though these numbers should not be overrated, they highly suggest that the new method is far more efficient than the numerical solution, given totally equal results in terms of accuracy.

4.5 Conclusions

We presented a method with which the equations of motion on a discretized curve formed by straight segments can be solved analytically using the approximation of infinitely small circular sectors to smooth discontinuities. In particular, it provides an efficient way for the computation of centripetal acceleration (most notably the corresponding friction term) without needing any information on second derivatives.

The method allows the splitting of the equations into a gravity-dependent and curvature-dependent component and to solve these parts independently of each other. This was possible since we exploited the property that for horizontal (gravity-free) circular motion the change of velocity depends neither on the radius of the circular sector nor on the time needed to pass trough it, but only on the friction coefficient and the angle of the circular segment. It also does not depend on a possible discretization and/or splitting of the angle into sub-angles.

In the examples it was shown that the method works well, reaching about second order of convergence. The results obtained on discretized curves by splitting the equations of motion into its gravity-dependent and curvature-dependent components are fully comparable to those obtained by solving the equations of motions numerically on corresponding smooth curves. Even though – mainly due to the differences in the lengths of the curves – there are small phase shifts between the solutions, they do not have a strong impact on the final results. This is explained by the fact that there are only very small differences in the velocities of the two solutions, especially if one would compare these not in time, but in space, meaning the velocities of the two solutions measured in approximately the same position. It follows that the phase shift only influences slightly the arrival time, but hardly the velocity and also final position, that is the position where the mass stops in a damped motion.

Omitting curvature dependent friction instead can lead to completely different results and significantly changes the behaviour of the sliding body. It could be shown in the examples that curvature dependent friction influences the motion more than grid refinement. In the second example we saw that the maximum error when omitting curvature dependent friction was about 100 times larger than the maximum error in the solution of the full equations.

A big advantage of the presented method is its low computational cost as no approximations have to be made. All necessary computations can be done exactly, which also leads to a high accuracy. A final remark is that, even though not mentioned in the results, the total energy is conserved in all cases in absence of friction. This is a direct result of the exact solution and high accuracy in the computations, since it is not directly imposed in the equations themselves.

Chapter 5

Semi-Analytical Solution for the 2-Dimensional Problem

5.1 Introduction and Formulation of the Problem

In the previous section a method for the exact solution of the equations in the more simple 1-dimensional case using discretized curves was presented. In the 2-dimensional case, surfaces represented by planar triangles will be treated and thus the method must be generalized. While the technique presented in the following is not restricted to triangles but works with arbitrary planar polygons, in terrain models, triangles are the most common structures however, which is why in the following we will mainly talk about triangles and triangulations. The data describing these terrains are usually given in the form of isolines from which one can extract/reconstruct a set of data points (x,y,z). There are several methods that can be used to reduce the noise from the raw data and to create data points that are regularly distributed in the x/y-plane; three of those methods were already presented in chapter 3 , a larger collection is found in Billings, Beatson, and Newsam, 2002a; Billings, Beatson, and Newsam, 2002b for example. From these data points in turn we create a triangulation which then will be the basis for our computations.

The following notations will be used throughout the subsequent sections: Let $\mathcal{S}$ be the original sliding surface, and let $D \subset \mathbb{R}^2$ be the domain over which $\mathcal{S}$ is defined. Then $\mathcal{S}$ is described by a function $f : D \to \mathbb{R}$. The discretized surface is denoted by Π; single elements of Π are denoted by σ_i and can be described as a function of x and y:

$$\pi_i = f_i(x,y) = a_i x + b_i y + d_i \quad \text{with} \quad (x,y) \in A_i \subset D$$

Unlike it is possible in 1D, single elements in 2D are not longer ordered in a certain (unique) way, as the method is not restricted to structured discretizations. Instead their order depends on the particular implementation of the discretization.

5.2 The solution on arbitrarily triangulated surfaces

The key point of the method presented for the 1-dimensional case in the previous chapter, is the separation of the motion into its curvature-independent and curvature-dependent components. While the first are fully determining the motion of a mass-point on a linear segment (or planar surface in the 2-dimensional case), the latter are active only in the moment of transition from one curve- ore surface-element to another. We already found that during this (instantaneous) transition, gravity-dependent components do not play any role such that the separation of the components is feasible. This technique shall now be applied to the 2-dimensional case. However, while in the 1D case acceleration along a linear segment is constant, in 2D this is generally not the case and therefore the motion can only be computed analytically in the frictionless case. In general, the solution on planar elements has to be computed numerically. A possible sliding surface and trajectory of a point, illustrating this problem, are given in Figs. 5.1a and 5.1b.

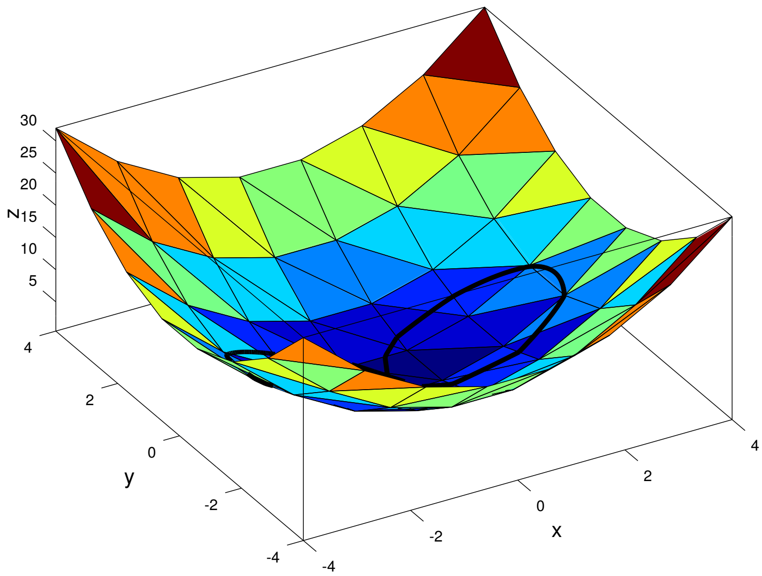
The 2-dimensional problem – obtained from Eq. (2.10) – is given by the following first order ODE system:

$$\mathbf{F} = \begin{pmatrix} v_x \\ -f_x \dfrac{g + \left(v_x^2 f_{xx} + 2v_x v_y f_{xy} + v_y^2 f_{yy}\right)}{1 + f_x^2 + f_y^2} \;-\; \mu v_x \dfrac{g + \left(v_x^2 f_{xx} + 2v_x v_y f_{xy} + v_y^2 f_{yy}\right)}{v\sqrt{1 + f_x^2 + f_y^2}} \\ v_y \\ -f_y \dfrac{g + \left(v_x^2 f_{xx} + 2v_x v_y f_{xy} + v_y^2 f_{yy}\right)}{1 + f_x^2 + f_y^2} \;-\; \mu v_y \dfrac{g + \left(v_x^2 f_{xx} + 2v_x v_y f_{xy} + v_y^2 f_{yy}\right)}{v\sqrt{1 + f_x^2 + f_y^2}} \end{pmatrix}, \tag{5.1}$$

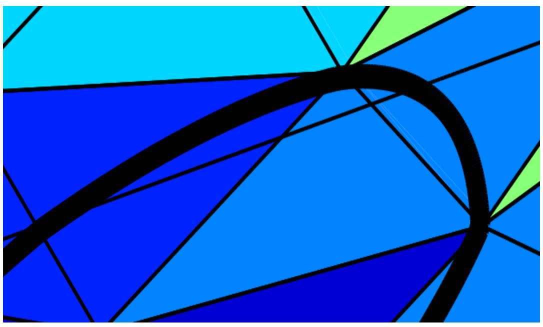
where $v = \sqrt{v_x^2 + v_y^2 + (v_x f_x + v_y f_y)^2}$, $f = f(x,y)$ is the given surface (in explicit form) and $f_{x/y}$ and $f_{xx/xy/yy}$ are the corresponding first and second order partial derivatives. v_x and v_y are the x- and y-components of the velocity vector.

As already discussed in the previous chapter, equation (5.1) becomes un-physical as $\left(g + v_x^2 f_{xx} + 2v_x v_y f_{xy} + v_y^2 f_{yy}\right)$ becomes negative, but as in the 1-dimensional case, detachment from the surface is extremely unlikely for real world landslides such that this problem will be neglected in the following. Again, the un-physical behaviour described by the equations for certain combinations of velocity and concavity is not a problem for the solution of the equations itself.

Analogous to the 1-dimensional case, $\mathbf{F}$, the solution of the 2-dimensional equations of motion is carried out in parts as again the gravity-independent terms are negligible for the motion in the transition area between two planar segments, and all curvature-dependent terms are zero if the surface is planar. Accordingly, for these two independent problems, two rhs are defined that will be denoted by $\mathbf{F}_g$ and $\mathbf{F}_c$,



(A) Sliding surface (discretized paraboloid) with trajectory of a point.



(B) Close up on a section of the trajectory of a point moving on the discretized paraboloid.

FIGURE 5.1: Sliding surface (discretized paraboloid) with close up on trajectory.

respectively.

$$
\mathbf{F}_g = \begin{pmatrix} v_x \\ -\dfrac{f_x g}{1+f_x^2+f_y^2} - \dfrac{\mu v_x g}{\sqrt{1+f_x^2+f_y^2}\sqrt{v_x^2+v_y^2+v_z^2}} \\ v_y \\ -\dfrac{f_y g}{1+f_x^2+f_y^2} - \dfrac{\mu v_y g}{\sqrt{1+f_x^2+f_y^2}\sqrt{v_x^2+v_y^2+v_z^2}} \end{pmatrix},
$$

$$
\mathbf{F}_c = \begin{pmatrix} v_x \\ -\dfrac{f_x\left(v_x^2 f_{xx}+2v_xv_yf_{xy}+v_y^2f_{yy}\right)}{1+f_x^2+f_y^2} - \dfrac{\mu v_x\left(v_x^2 f_{xx}+2v_xv_yf_{xy}+v_y^2f_{yy}\right)}{\sqrt{1+f_x^2+f_y^2}\sqrt{v_x^2+v_y^2+v_z^2}} \\ v_y \\ -\dfrac{f_y\left(v_x^2 f_{xx}+2v_xv_yf_{xy}+v_y^2f_{yy}\right)}{1+f_x^2+f_y^2} - \dfrac{\mu v_y\left(v_x^2 f_{xx}+2v_xv_yf_{xy}+v_y^2f_{yy}\right)}{\sqrt{1+f_x^2+f_y^2}\sqrt{v_x^2+v_y^2+v_z^2}} \end{pmatrix}, \tag{5.2}
$$

where $v_z = (v_x f_x + v_y f_y)$.

5.2.1 The motion on the planar segments of Π

In the following, the solution for the motion on the planar segments of the discretized surface Π will be derived, meaning that only $\mathbf{F}_g$ will be considered as all second order partial derivatives are *zero*. At first only the solution for the frictionless case is derived which, as well as in the 1-dimensional case – can be computed analytically.

The frictionless case: $\mu = 0$ In the frictionless case $\mathbf{F}$ further reduces to:

$$
\hat{\mathbf{F}}_g = \begin{pmatrix} v_x \\ -f_x \dfrac{g}{1+f_x^2+f_y^2} \\ v_y \\ -f_y \dfrac{g}{1+f_x^2+f_y^2} \end{pmatrix}.
$$

It follows that $\mathbf{a} = (a_x, a_y)$ for a certain planar element π_i, defined by the three parameters a_i, b_i and d_i, is given by:

$$
\begin{aligned}
a_x &= -g\frac{f_x}{1+f_x^2+f_y^2} = -g\frac{a_i}{1+a_i^2+b_i^2}, \\
a_y &= -g\frac{f_y}{1+f_x^2+f_y^2} = -g\frac{b_i}{1+a_i^2+b_i^2}.
\end{aligned}
$$

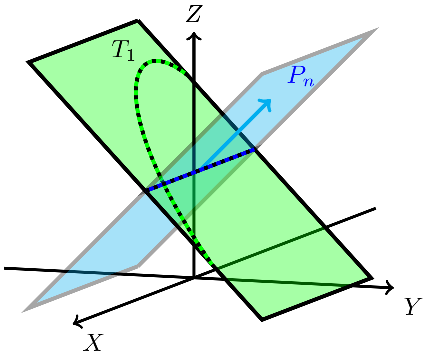


FIGURE 5.2: Intersection between parabolic trajectory and edge. The auxiliary plane P_n through the edge of the triangle is shown in light blue.

Obviously, a_x and a_y, the accelerations in x and y, are constant and so the motion can be described according to the 1D-case using the following two well known vector equations:

$$\mathbf{x}(t) = \mathbf{x}_0 + \mathbf{v}_0(t - t_0) + \frac{1}{2}\mathbf{a}_0(t - t_0)^2,$$

and

$$\mathbf{v}(t) = \mathbf{v}_0 + \mathbf{a}_0(t - t_0), \tag{5.3}$$

where $\mathbf{x} = (x, y)$ and $\mathbf{v} = (v_x, v_y)$. The vertical components z and v_z directly follow as the point is forced to stay on the surface:

$$z = f(x, y)$$
$$v_z = v_x f_x + v_y f_y$$

As $\mathbf{x}_0$, $\mathbf{v}_0$ and $\mathbf{a}_0$ are constant, the resulting trajectory is described by a parabola in t. This means that for every edge of a triangle two possible intersection points have to be considered, while altogether four intersections are possible. The trajectory is given by a parametric curve $\mathbf{p}(t)$ and the boundary is given by three nodes $\mathbf{n}_i = (x_i, y_i, z_i) \in \mathbb{R}^3$. As in $\mathbb{R}^3$ edges cannot be given in coordinate-form, they are replaced by planes perpendicular to the triangle, passing through those edges. This is illustrated in figure 5.2. These auxiliary planes P_n can generally be described by:

$$Ax + By + Cz + D = 0, \tag{5.4}$$

with real parameters A, B, C and D, while the trajectory is given by:

$$\mathbf{p}(t) = \mathbf{x}_0 + \mathbf{v}_0 t + \frac{1}{2}\mathbf{a}_0 t^2. \tag{5.5}$$

To find the time t at which a particular trajectory and plane intersect, the components of Eq. (5.5) are plugged into Eq. (5.4):

$$\begin{aligned}
&\frac{1}{2}(Aa_{x_i} + Ba_{y_i} + Ca_{z_i})t^2 + \\
&(Av_{x_i} + Vv_{y_i} + Cv_{z_i})t \;+ \\
&(Ax_i \;+ By_i \;+ Cz_i) \quad + D = 0,
\end{aligned} \tag{5.6}$$

which is easily solved for t.

The intersection time is denoted by t_{int}; evaluating $\mathbf{p}(t)$ in $t = t_{int}$ also gives the intersection point x_{int}. Just as well one finds the velocity in this point x_{int} using Eq. (5.3). As however there are three edges per triangle and solving Eq. (5.6) gives up to two real solutions per edge, we first need to identify the correct intersection time before being able to compute x_{int}. Just as well as in the 1-dimensional case, we are moving forward in time and are searching for the first possible intersection, hence, t_{int} must be the smallest positive real solution of Eq. (5.6).

The general case: $\mu > 0$ In the general case acceleration is not longer constant as the friction term depends on the velocity (not just on its direction as in the 1-dimensional case). So, unless an analytical solution for the system is known, the trajectory of the mass-point has to be computed numerically, that means the following system of equations has to be solved:

$$\dot{\mathbf{y}} = \frac{d\mathbf{y}}{dt} = \mathbf{F}_g(\mathbf{y}),$$

where:

$$\mathbf{F}_g = \begin{pmatrix} v_x \\ -\dfrac{gf_x}{1+f_x^2+f_y^2} - \dfrac{\mu g v_x}{v\sqrt{1+f_x^2+f_y^2}} \\ v_y \\ -\dfrac{gf_y}{1+f_x^2+f_y^2} - \dfrac{\mu g v_y}{v\sqrt{1+f_x^2+f_y^2}} \end{pmatrix}. \tag{5.7}$$

and $v = \sqrt{v_x^2 + v_y^2 + (f_x v_x + f_y v_y)^2}$.

The above system can be integrated numerically using an explicit Runge-Kutta scheme. As the second order partial derivatives and therefore the acceleration do not change within single planer segments, no solver for stiff equations is required in this case.

Certainly, as the equations cannot be solved exactly, also the exact point and time of transition between triangles cannot be computed analytically but only approximated. To obtain accurate results though, the following should be kept in mind:

1. Choose step-size according to the resolution of the grid and the (approximated maximal) velocity of the point;

   ⇒ avoid a too big number of time-steps which would make computations unnecessarily time-consuming as well as too few steps thus leading to incorrect numerical results;

2. After each time step check if the point is located in- or outside the current surface element;

   ⇒ this is easily done using barycentric coordinates (compare A.1), a technique that is not only computationally cheap but also provides information whether a point is located in the interior of the triangle or on the boundary and in the latter case also if it is lying on an edge or coincides with one of the three nodes (this method can be extended to general convex sets (compare Warren et al., 2004) and thus is not restricted to triangles);

3. Use bisection technique in order to optimize the computation of intersection time and point.

The listing below explains the bisection technique, as it was used in the presented method, more in detail:

1. Bisection is started when during time-integration the point is located outside the triangle for the first time;

2. As long as any of the following three conditions is $false$, the time-integration is continued from $\mathbf{x}_i$ on (i.e. the last computed point $\mathbf{x}_{i+1}$ is ignored) using half the step-size;

   (a) the distance between the last two points of the time-integration is less than a given ε_x, i.e. $||\mathbf{x}_{i+1} - \mathbf{x}_i|| < \varepsilon_x$,

   (b) the relative change in velocity is less than a given ε_v,
   i.e. $\dfrac{||\mathbf{v}_{i+1} - \mathbf{v}_i||}{||\mathbf{v}_{i+1}||} < \varepsilon_v$,

   (c) $\mathbf{x}_{i+1}$ is located outside the triangle

3. If all three conditions are fulfilled, the bisection is stopped and intersection-time and -point, t_{int} and $\mathbf{p}(t_{int})$ respectively, are computed.

The interpolated point and time of transition can be found by intersecting the position vector representing the trajectory with those of the three edge of the current element. Let therefore $\mathbf{p}_1$ and $\mathbf{p}_2$ denote the last two positions of the point before the bisection was stopped, $\mathbf{p}_1$ being the point inside the triangle and $\mathbf{p}_2$ the point outside. Let further $\mathbf{v}_1$ and $\mathbf{v}_2$ be the two vertices of a certain edge. Then the point of intersection can be found by solving the following vector equation for λ_1 and λ_2:

$$\mathbf{p}_1 + \lambda_1(\mathbf{p}_2 - \mathbf{p}_1) = \mathbf{v}_1 + \lambda_2(\mathbf{v}_2 - \mathbf{v}_1). \tag{5.8}$$

There is always a solution to this equation unless the two lines are parallel. The point of intersection is found by either inserting λ_1 in the lhs or λ_2 in the rhs of the equation.

As the point of intersection (in x and y) is the same when projecting the triangle on the (x,y)-plane, it is sufficient for the solution of Eq. (5.8) to consider only the x- and y-components. Altogether three possible intersection of the trajectory with the triangle can be found, but only one of them is also a valid intersection in the sense that

- the point of intersection lies on the boundary of the triangle and
- the point of intersection lies between $\mathbf{p}_1$ and $\mathbf{p}_2$.

This is the case if and only if both factors, λ_1 and λ_2, are in the range $[0,1]$ and is illustrated in figure 5.3 which shows one valid (S_1) and two invalid intersections $(S_2,\ S_3)$. All three intersections are explained in the following where $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ are the three vertices of the triangle:

$\mathbf{S_1}$: intersection between trajectory and edge e_3 (given by $\mathbf{v}_1 + \lambda_2(\mathbf{v}_2 - \mathbf{v}_1)$);
$\lambda_1, \lambda_2 \in [0,1]$

$\Rightarrow$ valid intersection

$\mathbf{S_2}$: intersection between trajectory and edge e_1 (given by $\mathbf{v}_2 + \lambda_2(\mathbf{v}_3 - \mathbf{v}_2)$);
$\lambda_1 < 0,\ \lambda_2 \in [0,1]$

$\Rightarrow$ no valid intersection, point would have to travel backwards in time

$\mathbf{S_3}$: intersection between trajectory and edge e_2 (given by $\mathbf{v}_3 + \lambda_2(\mathbf{v}_1 - \mathbf{v}_3)$);
$\lambda_1 \in [0,1],\ \lambda_2 > 1$

$\Rightarrow$ no valid intersection, point of intersection lies outside the triangle

The point of intersection is then found by evaluating one of the two following equations:

$$\mathbf{p}(t_{int}) = \mathbf{p}_1 + \lambda_1(\mathbf{p}_2 + \mathbf{p}_1),$$
$$\mathbf{p}(t_{int}) = \mathbf{v}_1 + \lambda_2(\mathbf{v}_2 + \mathbf{v}_1).$$

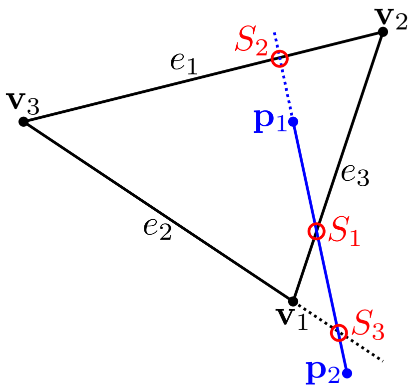


FIGURE 5.3: Possible intersections (S_1, S_2 and S_3, marked by red circles) of a linear trajectory (blue line) with a planar triangle (given by black lines).

While theoretically both equations must give the same result, using the second one is recommended in the final implementation as it directly ensures an intersection-point on the edge of the triangle. This is of great importance within the code.

Knowing the point of intersection $\mathbf{p}(t_{int})$, also the intersection time is easily computed by solving Eq. (5.9) for t, using linear interpolation (recall that from condition (b) it follows that the velocity is approximately linear).

$$\mathbf{p}(t_{int}) = \mathbf{x}_0 + \mathbf{v}_0 t_{int} \qquad \Rightarrow \qquad t_{int} = \frac{\mathbf{p}(t_{int}) - \mathbf{x}_0}{\mathbf{v}_0}. \tag{5.9}$$

Note that due to the linear approximation of the motion in vicinity of the intersection-point, only one intersection-time t_{int} is computed (in contrast to the 1-dimensional case) for a particular couple of points $\mathbf{p}_1$ and $\mathbf{p}_2$. t_{int} is a valid intersection-time if it is non-negative.

5.2.2 The motion in the transition region between two elements of Π

For computing the transition between two neighbouring triangles it is again assumed that gravity does not play a role. Moreover, it will be shown that the solution of the curvature-dependent components only requires knowledge about the current and the future element (i.e. before and after transition) of the surface. Recall that the

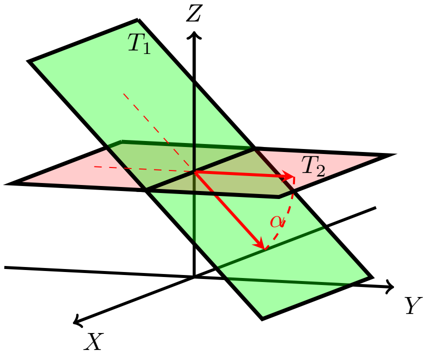


FIGURE 5.4: The planes containing the two triangles T_1 and T_2 and the angle α enclosed by them. The red lines indicate the vectors lying in the planes, normal to the axis of rotation.

curvature-dependent equations $\mathbf{F}_c$ (c.f. Eq. (5.2))read:

$$\mathbf{F}_c = \begin{pmatrix} v_x \\ -f_x \dfrac{\left(v_x^2 f_{xx} + 2 v_x v_y f_{xy} + v_y^2 f_{yy}\right)}{1+f_x^2+f_y^2} \quad - \quad \mu v_x \left[\dfrac{\left(v_x^2 f_{xx} + 2 v_x v_y f_{xy} + v_y^2 f_{yy}\right)}{v\sqrt{1+f_x^2+f_y^2}} \right] \\ v_y \\ -f_y \dfrac{\left(v_x^2 f_{xx} + 2 v_x v_y f_{xy} + v_y^2 f_{yy}\right)}{1+f_x^2+f_y^2} \quad - \quad \mu v_y \left[\dfrac{\left(v_x^2 f_{xx} + 2 v_x v_y f_{xy} + v_y^2 f_{yy}\right)}{v\sqrt{1+f_x^2+f_y^2}} \right] \end{pmatrix},$$

It can easily be verified that the left side of $\mathbf{F}_c$ act normal to the surface and therefore only changes the direction of motion but does not influence the speed while the right side acts tangential to the direction of motion and thus influences the speed but not the direction of motion. From Eq. (2.9) we get:

$$\begin{aligned} \dot{v}_{c,x} &= -f_x \frac{v_x^2 f_{xx} + v_x v_y f_{xy} + v_y^2 f_{yy}}{1+f_x^2+f_y^2} \quad - \quad \mu v_x \frac{v_x^2 f_{xx} + v_x v_y f_{xy} + v_y^2 f_{yy}}{1+f_x^2+f_y^2}, \\ \dot{v}_{c,y} &= -f_y \frac{v_x^2 f_{xx} + v_x v_y f_{xy} + v_y^2 f_{yy}}{1+f_x^2+f_y^2} \quad - \quad \mu v_y \frac{v_x^2 f_{xx} + v_x v_y f_{xy} + v_y^2 f_{yy}}{1+f_x^2+f_y^2}, \\ \dot{v}_{c,z} &= \frac{v_x^2 f_{xx} + v_x v_y f_{xy} + v_y^2 f_{yy}}{1+f_x^2+f_y^2} \quad - \quad \mu (f_x v_x + f_y v_y) \frac{v_x^2 f_{xx} + v_x v_y f_{xy} + v_y^2 f_{yy}}{1+f_x^2+f_y^2}. \end{aligned} \tag{5.10}$$

Denoting by $\mathbf{a}_n$ and $\mathbf{a}_t$ the left and right part of Eq. (5.10) we find:

$$\mathbf{a}_n = \begin{pmatrix} \dot{v}_{c,x} \\ \dot{v}_{c,y} \\ \dot{v}_{c,z} \end{pmatrix} = \frac{v_x^2 f_{xx} + v_x v_y f_{xy} + v_y^2 f_{yy}}{1+f_x^2+f_y^2} \begin{pmatrix} -f_x \\ -f_y \\ 1 \end{pmatrix},$$

where the vector $(-f_x, -f_y, 1)^\mathsf{T}$ and therefore $\mathbf{a}_n$ is normal to the curve and thus to the direction of motion. For $\mathbf{a}_t$ instead we find:

$$\mathbf{a}_t = \begin{pmatrix} -\mu v_x \dfrac{v_x^2 f_{xx} + v_x v_y f_{xy} + v_y^2 f_{yy}}{1 + f_x^2 + f_y^2} \\ -\mu v_y \dfrac{v_x^2 f_{xx} + v_x v_y f_{xy} + v_y^2 f_{yy}}{1 + f_x^2 + f_y^2} \\ -\mu (f_x v_x + f_y v_y) \dfrac{v_x^2 f_{xx} + v_x v_y f_{xy} + v_y^2 f_{yy}}{1 + f_x^2 + f_y^2} \end{pmatrix}$$
$$= \frac{v_x^2 f_{xx} + v_x v_y f_{xy} + v_y^2 f_{yy}}{1 + f_x^2 + f_y^2} \begin{pmatrix} v_x \\ v_y \\ f_x v_x + f_y v_y \end{pmatrix},$$

where $(v_x, v_y, f_x v_x + f_y v_y)^\mathsf{T} = (v_x, v_y, v_z)^\mathsf{T} = \mathbf{v}$ and therefore also $\mathbf{a}_t$ is tangential to the direction of motion.

Therefore Eq. (5.2) can again be split up into two components denoted by $\mathbf{a}_{c,n}$ for the normal and $\mathbf{a}_{c,t}$ for the tangential component that on piecewise planar surfaces can be solved independently of each other. Even though the trajectory in 2D is non-linear, in a sufficiently small neighbourhood around a particular point (e.g. the intersection between two planes), the motion is approximately linear which in turn means that the same principles as in the 1-dimensional case can be applied.

To compute the direction of the velocity $\mathbf{v}_1$ after the transition, a 3D rotation of the velocity vector $\mathbf{v}_0$ has to be performed, thus projecting it from the old to the new surface. Let T_1 and T_2 be two planar triangles in 3D and let P_1 and P_2 be the two planes in which the triangles are lying. Hence, the axis of rotation is the line in which those two planes intersect and the angle of rotation is the angle enclosed by them. This is indicated in figure 5.4.

Let $\mathbf{n}_{T_1}$ and $\mathbf{n}_{T_2}$ be the unit normal vectors to the triangles T_1 and T_2. Then the angle of rotation α, and the (normalized) direction of the axis of rotation $\boldsymbol{\varepsilon} = (\varepsilon_x, \varepsilon_y, \varepsilon_z)$, are:

$$\alpha = \arcsin\left(|\mathbf{n}_{T_1} \times \mathbf{n}_{T_2}|\right) \quad \text{and} \quad \boldsymbol{\varepsilon} = \frac{\mathbf{n}_{T_1} \times \mathbf{n}_{T_2}}{|\mathbf{n}_{T_1} \times \mathbf{n}_{T_2}|}.$$

In $\mathbb{R}^3$ a rotation can uniquely be described by a quaternion q (for more information see appendix B), where $\mathbf{q} = \cos\frac{\alpha}{2} + \boldsymbol{\varepsilon}\sin\frac{\alpha}{2}$, or written in its components:

$$q_0 = \cos\frac{\alpha}{2},$$
$$q_1 = \varepsilon_x \sin\frac{\alpha}{2}, \quad q_2 = \varepsilon_y \sin\frac{\alpha}{2}, \quad q_3 = \varepsilon_z \sin\frac{\alpha}{2},$$

leading to the following rotation matrix:

$$\mathbf{R}_\alpha = \begin{pmatrix} q_0^2+q_1^2-q_2^3-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2-q_1^2+q_2^3-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2-q_1^2-q_2^3+q_3^2 \end{pmatrix},$$

and the velocity after the transition $\mathbf{v}_2$ is given by:

$$\mathbf{v}_2 = \mathbf{R}_\alpha \mathbf{v}_1.$$

That followed, the change in speed can be computed. In chapter 4 we showed that in case of (gravity-independent) horizontal circular motion, where a point of given initial velocity passes through a circular segment with angle γ, the change of speed due to friction – given by the friction coefficient μ, is described by the so called velocity modification factor Γ:

$$\Gamma = \exp(-\mu\gamma),$$

where (independently of the direction of motion) $\gamma > 0$ and therefore $\Gamma < 1$.

Multiplying v (the modulus of $\mathbf{v}$) by Γ gives the speed of the point after passing through the circular segment where the angle γ is given in radians.

Recall that in the 1-dimensional case, the curve in vicinity of the transition point is smoothed using a circular arc whose angle coincides with the angle enclosed by the two linear segments of the curve(and therefore the trajectory) which in turn coincides with the angle between the two velocity vectors. This is not longer the case in 2D.

Instead of smoothing the whole surface, only the trajectory that in an ε-neighbourhood of the transition point is approximately linear is smoothed using a circular arc. The two linear segments (before and after the transition from one triangle to another) of the trajectory in this neighbourhood are represented by the two corresponding velocity vectors where $\mathbf{v}_0$ is the direction of velocity before the transition and $\mathbf{v}_1$ the one after. Then the angle γ needed to compute Γ is exactly the one enclosed by $\mathbf{v}_0$ and $\mathbf{v}_1$ and can easily be computed via their scalar product:

$$\cos(\gamma) = \frac{\mathbf{v}_0 \cdot \mathbf{v}_1}{\|\mathbf{v}_0\|\|\mathbf{v}_1\|}.$$

Note that in general the angles α (the angle built by the two triangles) and γ (angle enclosed by the velocity vectors) are not equal.

As the factor Γ is independent of the radius r of the circular segment, we can choose r to be infinitely small such that the length of the circular segment goes to *zero* and thus the change of speed is performed instantaneously.

This retrospectively also justifies the splitting of equ. 5.1 into its gravity-and curvature-dependent terms as the gravity-dependent terms do not play a role within an infinitely small period of time (in which the point passes through the circular segment). On the other hand, as the surface is planar almost everywhere, and thus curvature-dependent terms do not play a role most of the time but their effect is restricted to the point of transition.

We find that the velocity modification factor together with and a simple rotation of the velocity vector are sufficient to describe the transition of a point from one planar element to another. In particular we do not need information on the curvature of the surface even though it is part of the original equations. This not only offers a very simple and fast way to compute the transition; in the results section we will also see that the method provides a high accuracy. In the following section different test cases are performed.

5.2.3 The solution strategy

For a better understanding we will briefly sketch an algorithm for solving the equations for the motion of one point on arbitrarily discretized surfaces. We assume that a 2-dimensional discretization of the sliding surface is already given. A sketch of the 2D code is given in algorithm 3 where the interior of a triangle T is denoted by T° and its boundary by ∂T.

The time-step used for the numerical integration is denoted by $\triangle t_n$ while the actual time used for a particular Runge-Kutta time-integration step is denoted by $\triangle t_{RK}$, the intersection time of the trajectory with the boundary of T is given by t_{int} while $\tilde{\triangle} t$ denotes the remaining time of a single time-step $\triangle t_n$ in case the time-integration is split into two or more sub-steps. The simulation is running until either the point stops moving or would leave the discretized area. At each time, the point P is associated with one of the triangles T of the discretization D of $\mathcal{S}$.

For each triangle T, Eq. (5.7) is solved either numerically or analytically (in case of constant acceleration); at the transition between two elements the direction and modulus of the point's velocity must be corrected using the rotation matrix R and the modification factor Γ. As for each triangle we need to know the intersection-time and -point, a splitting of the time-step into several, individual intermediate time-steps (dt_{int}) usually is required. The intersection time is defined as the time that is needed by P to reach ∂T starting from its current position.

Algorithm 3 Algorithm for 1D analytical solution

logical :: inner {true if point does not leave current triangle}
while <$time < T_{end}$> **do** {external loop, time-step: $\triangle t$}
  $\tilde{\triangle} t = \triangle t$; inner=.$true$.
  while ($\tilde{\triangle} t > 0$) **do**
    $\triangle t_{RK} = \min(\tilde{\triangle} t, \triangle t_n)$
    $\mathbf{Y} = [x_{i-1}, v_{x,i-1}, y_{i-1}, v_{y,i-1}]$
    call Runge-Kutta($\mathbf{F}$,$\mathbf{Y}$,$\triangle t_{RK}$)
    $\mathbf{X}_i = [Y_1, Y_3]$; $\mathbf{V}_i = [Y_2, Y_4]$
    if ($\mathbf{X}_i \subset \partial T$) **then**
      inner=.$false$.
      $\tilde{\triangle} t = \tilde{\triangle} t - \triangle t_{RK}$
      exit
    else if ($\mathbf{x}_i \not\subseteq T$) **then**
      call Bisection($\mathbf{X}_{i-1}, \mathbf{X}_i, \mathbf{V}_{i-1}, \mathbf{V}_i$)
      if ($\mathbf{X}_i \subset \partial T$) **then**
        inner=.$false$.
        exit
      else
        if ($\mathbf{x}_i \not\subseteq T$) **then**
          inner=.$false$.
          Compute point, velocity and time of intersection ($\mathbf{X}_{int}$, $\mathbf{V}_{int}$, t_{int})
          $\mathbf{X}_i = \mathbf{X}_{int}$; $\mathbf{V}_i = \mathbf{V}_{int}$; $t_i = t_{int}$
          exit
        else if ($\mathbf{x}_i \subset T^{\mathrm{o}}$) **then**
          inner=.$false$.
          $\tilde{\triangle} t = \tilde{\triangle} t - \triangle t_{RK}$
          exit
        end if
      end if
    else
      $\tilde{\triangle} t = \tilde{\triangle} t - \triangle t_{RK}$
    end if
  end while
  if (inner==.$true$.) **then** $\tilde{\triangle} t = 0$
  end if
end while

5.3 Numerical results 1: Comparison to the numerical results obtained on smooth reconstructed surfaces

5.3.1 Test-cases:

All test-cases are constructed such as to represent different critical characteristics of real topographies such as a sudden change in slope. While the first test-case is a piecewise-planar function and therefore contains an instantaneous change in slope, the other cases are smooth functions. Test case 1 was also used in chapter 2 to test the different reconstruction methods concerning their ability to imitate a triangulated surface.

Test-Case 1:

$$g_1(x,y) = \begin{cases} -x+0.5y & \text{if} \quad (x,y) \in ([-4,0) \times [-14,4]), \\ 0.5y & \text{if} \quad (x,y) \in ([0,14] \times [-14,4]). \end{cases}$$

This is the only non-smooth function, representing an instantaneous change in slope. During the reconstruction the corner in $x = 0$ must be smoothed while the piecewise-planar parts should remain as planar as possible.

Test-Case 2: A kind of normalized arctan-function in x that is constant in y. Due to the definition of $g_{2,x/y}$ we can choose the inclination of the function g_2 before the change of slope, where the inclination afterwards is *zero* by definition. The whole surface can in addition be inclined in y-direction. However, here we will present a case with $g_{2,x} \approx -1$ for $x > 0$ and $g_{2,x} \approx 0$ for $x > 0$ and no additional inclination in y-direction. This test is in some sense a smoothed version of test case 1. The two derivatives are given by:

$$g_{2,x} = \frac{1}{\pi} \arctan(\varphi x) - 0.5, \qquad g_{2,y} = \eta,$$

which gives the following reconstructing function:

$$g_2(x,y) = \frac{1}{\pi} x \arctan(\varphi x) - \frac{1}{(2\pi\varphi)} \log(\varphi^2 x^2 + 1) - 0.5x + \eta y,$$

and second order partial derivatives:

$$g_{2,xx} = \frac{\varphi}{\pi} \cdot \frac{1}{1+\varphi^2 x^2}, \qquad g_{2,xy} = g_{2,yy} = 0.$$

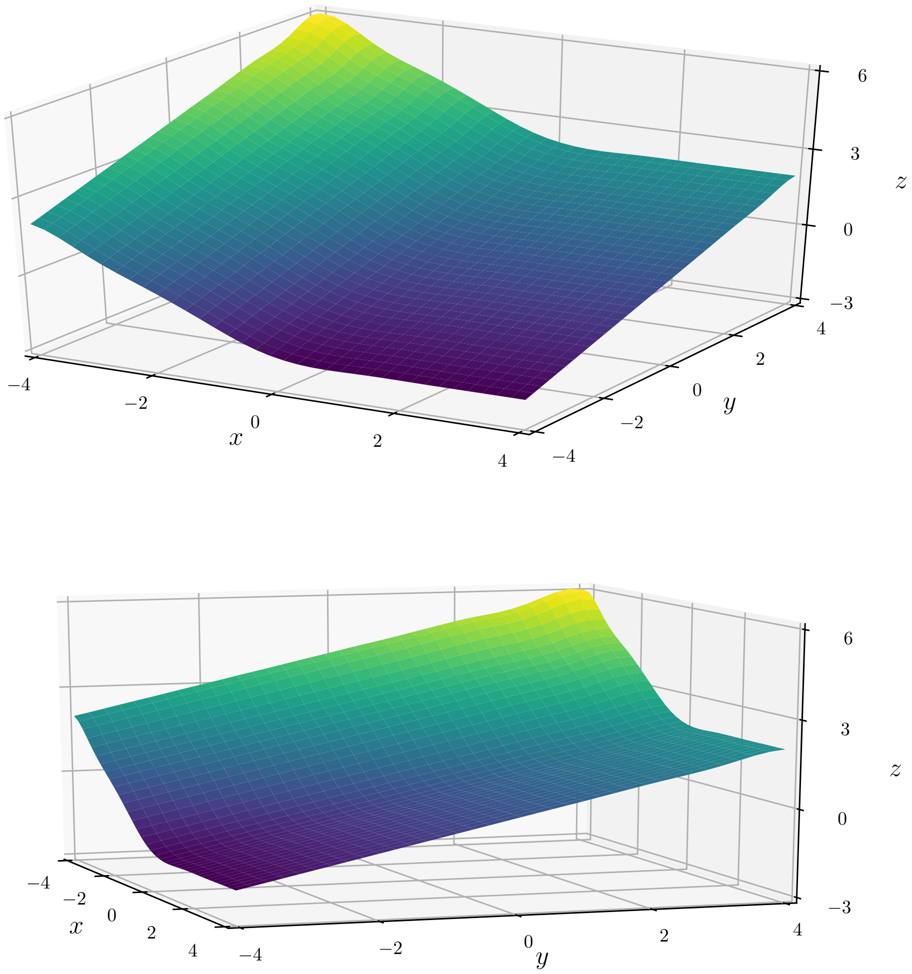


FIGURE 5.5: Surface for test case 1 from two different angles.

Similar to the first test case, the slope in x-direction is first negative and then becomes almost *zero* after passing through $x = 0$ but this is done in a smooth way. Setting η different from *zero* one adds an additional slope in y-direction. In the present case the parameters were chosen as follows: $\varphi = 1$, $\eta = 0.5$, and the domain is given by the range $[-4,14] \times [-14,4]$

Test-Case 3: A paraboloid

$$g_3(x,y) = \varphi x^2 + \eta_1 y^2 + \eta_2 y.$$

In contrast to the other test functions g_3 has constant second derivatives and should be approximated better by the radial basis functions. It stands for a moderate change in slope, having constant second order partial derivatives. The parameters were chosen $\varphi = 1$, $\eta_1 = 0.1$ and $\eta_2 = 0.5$. The domain is $[-4,4] \times [-8,4]$.

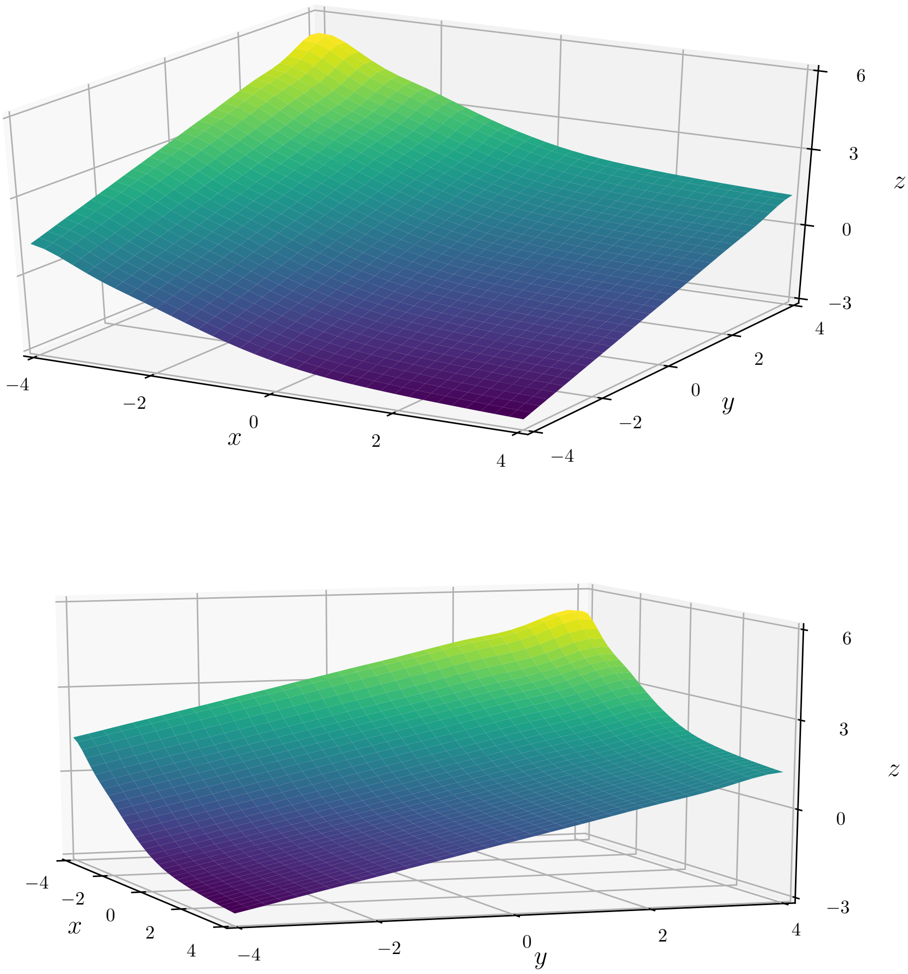


FIGURE 5.6: Surface for test case 2.

Test-Case 4: Based on an arctan-function in x-direction and a parabola in y-direction. The definition of this function is a combination of the test-cases 2 and 3, but in contrast the slope for $x > 0$ is not close to *zero* but has the same absolute value as for $x < 0$, just with opposite sign. The slope becomes steeper for larger absolute values of φ_1 and the change of slope more sudden for larger values of φ_2. With the following parameters: $\varphi_1 = \varphi_2 = 2$ the change of slope is more sudden then in case 3. The remaining parameters are set to: $\eta_1 = 0.1$, $\eta_2 = 0.5$. The domain is again $[-4,4] \times [-8,4]$

$$g_{4,x} = \varphi_1 \arctan(\varphi_2 x), \qquad g_{4,y} = 2\eta_1 y + \eta_2.$$

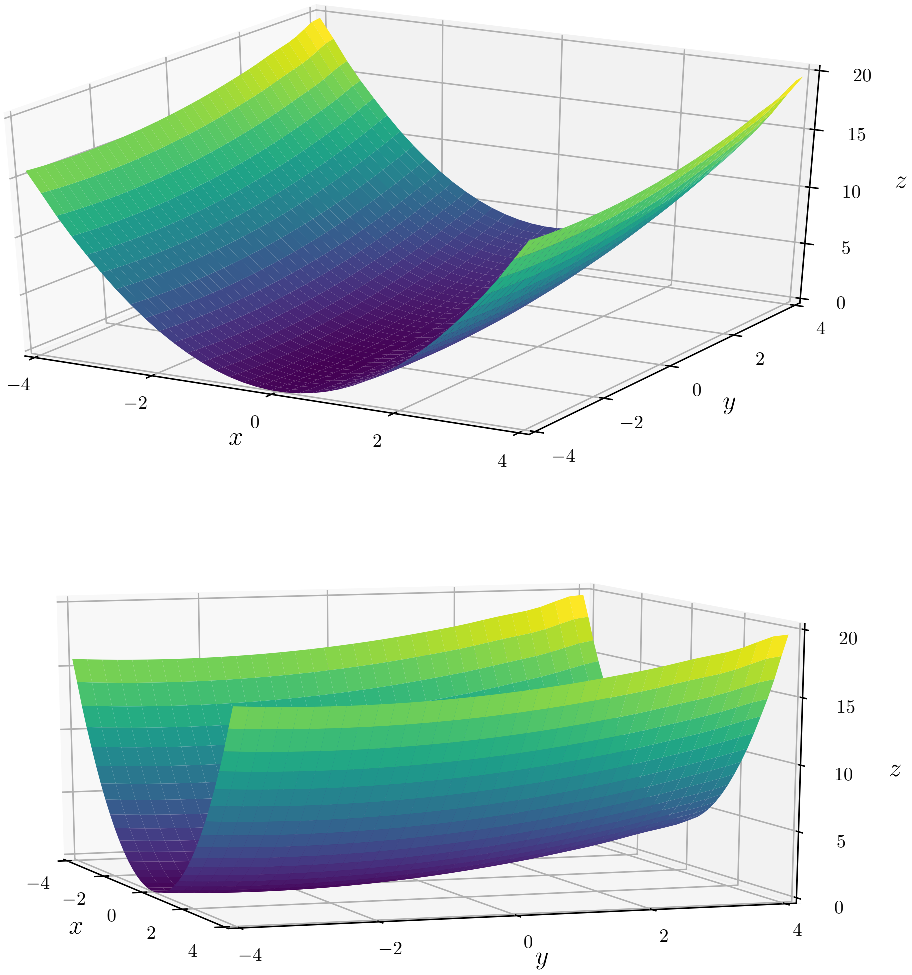


FIGURE 5.7: Surface for test case 3.

Therefore the following reconstructing function is found:

$$g_4(x,y) = \varphi_1(x \arctan(\varphi_2 x) - \log(\varphi_2^2 x^2 + 1)/(2\varphi_2)) + \eta_1 y^2 + \eta_2 y,$$

and second order partial derivatives:

$$g_{4,xx} = \varphi_1 \left(\varphi_2 \frac{1}{1+\varphi_2^2 x^2} \right), \qquad g_{4,xy} = 0, \qquad g_{4,yy} = 2\eta_1.$$

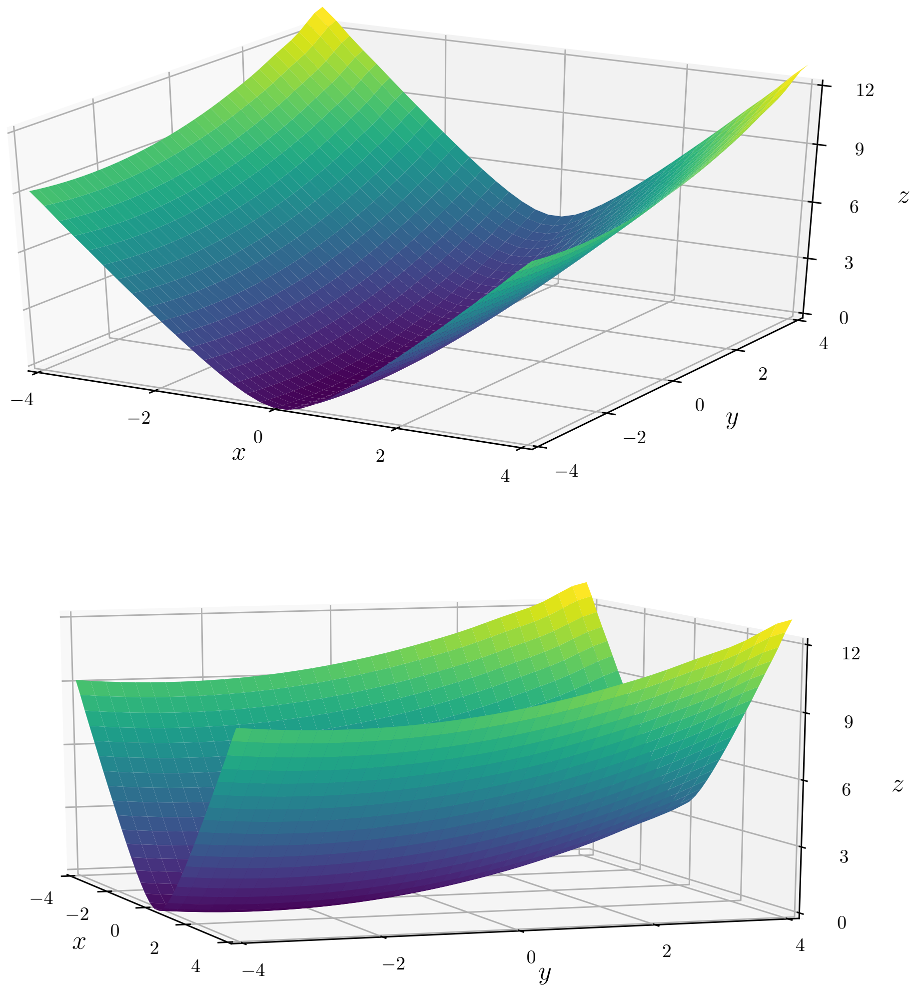


FIGURE 5.8: Surface for test case 4.

5.3.2 Comparison of solutions computed on triangulated versus smooth surfaces for the frictionless case

The solutions computed on smooth surfaces – obtained using smooth surface reconstruction by means of radial basis functions – are in the following compared to the solutions computed on piecewise planar surfaces in a qualitative way. Therefore, the four test surfaces presented above will be used in the following.

In the previous sections two methods were presented for the solution of the equations on piecewise planar surfaces. A fully analytical one and a semi-analytical one. While the first methods can be used to compute the solution of the equations of motion exactly in absence of friction, the second one approximates the solution along the planar segments and only computes the change of speed in the transition point between two elements analytically. Hence it can be used also in the case of non-zero friction. In the case of zero friction, obviously both approaches can be used even though the first one is slightly more accurate and in particular provides an even better conservation of energy as the point of transition and the respective velocity do not have to be approximated. However, in order to judge the more general method (i.e. including friction), the semi-analytical approach will be used also in the frictionless case in the following as in the friction containing case the total energy cannot be judged as no exact solution is known.

For every test case, surfaces with different levels of refinement are used, more concrete, the number of grid points per coordinate direction are $N = 11, 21, 41$, so the total number of grid points is $N_t = 121, 441, 1681$. Finer grids turned out not to give better results for the solution on smooth reconstructed surfaces, which was prone to too strong roughness in the second order partial derivatives. Tests with a finer grid were also run for the piecewise planar surfaces, which – as can be expected – turned out to give better results with an increasing number of grid points, also beyond $N = 41$.

As reference, the solutions obtained on the original surfaces using the LSODE solver, are used. In order to evaluate also the conservation of total energy, the tests are run without friction, i.e. $\mu = 0$. For the comparison of the total energy the potential energy is assumed to be *zero* for the initial time step t_0, as, given the x- and y-coordinate of the point's initial position the corresponding z-coordinate varies with different reconstructions of the surface, thus leading to different potential energy if the *zero*-level for the potential energy would be equal for all reconstructions. As – due to the initial conditions – the kinetic energy at time t_0 is *zero*, too, also the total energy is *zero* at the initial time.

In the following test cases, the solutions on piecewise planar surfaces together with the solutions on smooth reconstructed surfaces will also be denoted as "approximated solutions" as they are assumed to approximate the exact solution on the original smooth surface.

Results of test case 1: In contrast to the remaining three test cases, the solution on the original surface is computed via the newly developed method instead of using the LSODE solver, due to the discontinuity in the first and second order partial derivatives for $x = 0$. The so computed solution is exact.

The test-surface for the full domain and the trajectories of the point for the surfaces with $N = 11$ and $N = 41$ points are shown in figure 5.9. The two trajectories for the smooth surfaces, obtained using the radial basis functions, are denoted by RBF uning a yellow dash-dotted line for $N = 11$ and a magenta dashed line for $N = 41$. The respective trajectories of the solutions on the piecewise planar surfaces (PP) are denoted by a turquoise dash-dotted ($N = 11$) and a white dotted ($N = 41$) line.

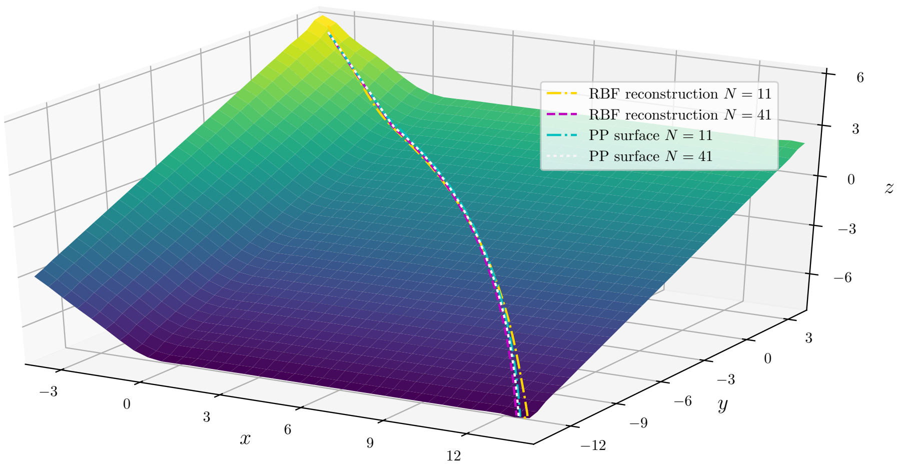


FIGURE 5.9: Test case 1: Surface and trajectories for friction coefficient $\mu = 0$. Only the solutions for $N = 11$ and $N = 21$ as well as the reference solution are plotted.

In the given domain, no big difference can be seen in the two trajectories on the piecewise planar surface and the finer smooth surface. Only the solution on the lower-resolution smooth surface differs slightly. The same is seen in figure 5.10 were additionally the solutions for $N = 21$ are plotted. While the three solutions for the piecewise planar surfaces are nearly identical, coinciding with the reference solution, the solutions on the RBF surfaces seam to approximate the reference solution for an increasing number of grid points (for a definite answer a stronger refinement must have been computed). The overlapping solutions for the piecewise planer reconstructions are not surprising as this type of reconstruction obviously fits a piecewise planar surface best. (Note however, that due to the choice of N, the reconstructed surfaces do not coincide with the original one.)

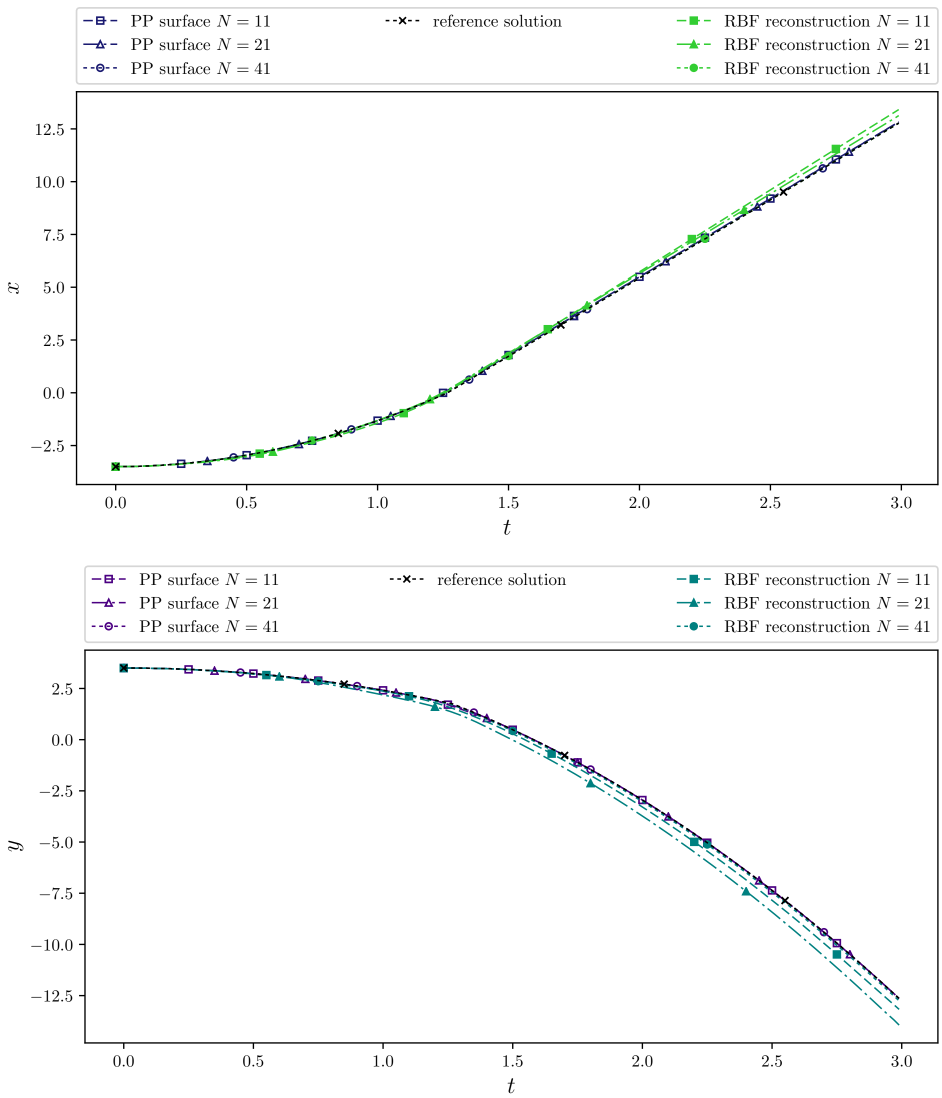


FIGURE 5.10: Test case 1: x- and y-coordinates of the trajectories for friction coefficient $\mu = 0$. The solutions for $N = 11, 21, 41$ as well as the reference solution are shown.

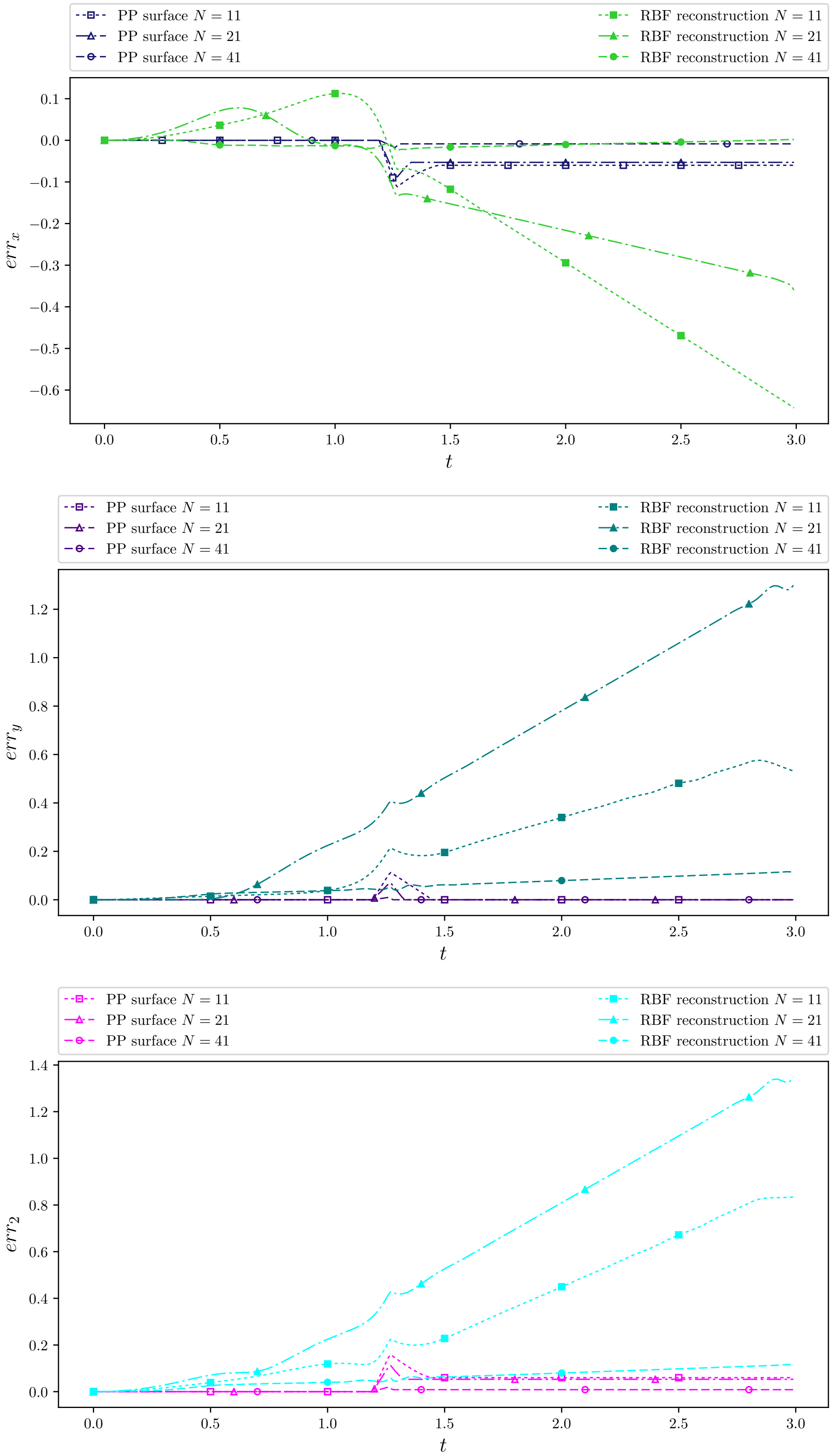


FIGURE 5.11: Test case 1: Absolute errors between the trajectories and the reference solution for $\mu = 0$, split in x- and y-coordinates (top and middle). Euclidean distances between reference and approximated solutions (bottom).

The corresponding absolute errors between the approximated and the reference solutions for the x- and y- coordinates (top and middle) as well as the L_2-distances (distance between a point on the reference solution and a point on the approximated solution at a particular time t) are plotted in figure 5.11. Here it is seen more in detail how well the solutions on the PP surfaces approximate the reference solution while the errors for $N = 11$ and $N = 21$ for the solutions on the RBF are still significant. Note in particular the error-peaks that appear in the point of change of slope.

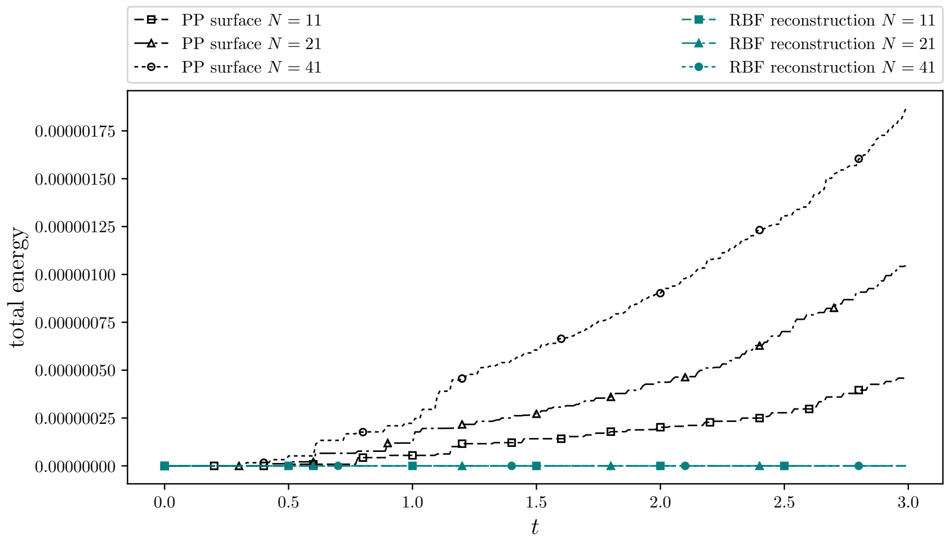


FIGURE 5.12: Test case 1: Total energy plotted for all solutions for $\mu = 0$.

The total energy is practically conserved for all six trajectories as can be seen in figure 5.12.

Altogether, the semi-analytical time-integration method in combination with the piecewise planer surfaces performs again significantly better than the LSODE solver in combination with smooth reconstructed surfaces.

Results of test case 2: As already mentioned, the second test case is similar to the first on, but is smooth everywhere. The results are presented in the following.

The trajectories for $N = 11$ and $N = 41$ as well as the reference solution on the original surface are shown in figure 5.13. Altogether, the solutions are slightly closer to each other than in the previous case which can also be seen in the following figures.

Figure 5.14 shows the x- and y-coordinates of the trajectories in time. The corresponding absolute errors are shown in figure 5.15 (top and middle). On the bottom of the same figure the distance (measured in the L_2-norm) between the trajectories is depicted. Compared to the first test case we find a very similar picture, but without showing the error-peaks during change of slope.

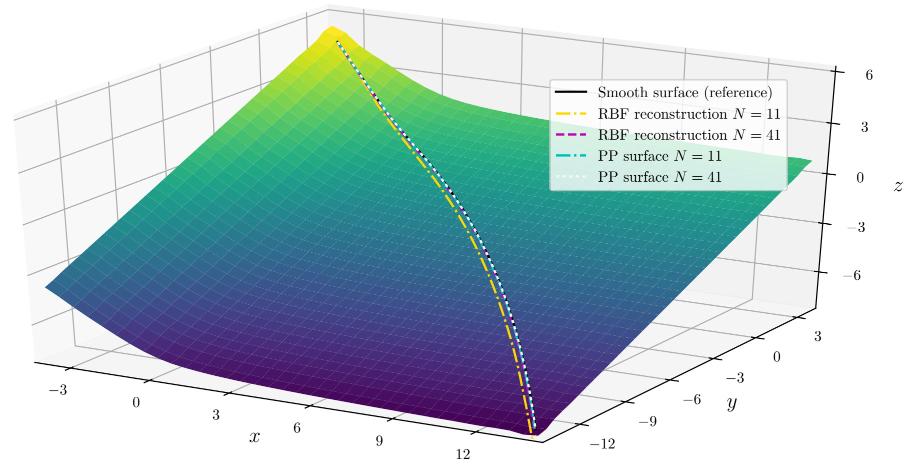


FIGURE 5.13: Test case 2: Surface and trajectories for friction coefficient $\mu = 0$. Only the solutions for $N = 11$ and $N = 21$ as well as the reference solution are plotted.

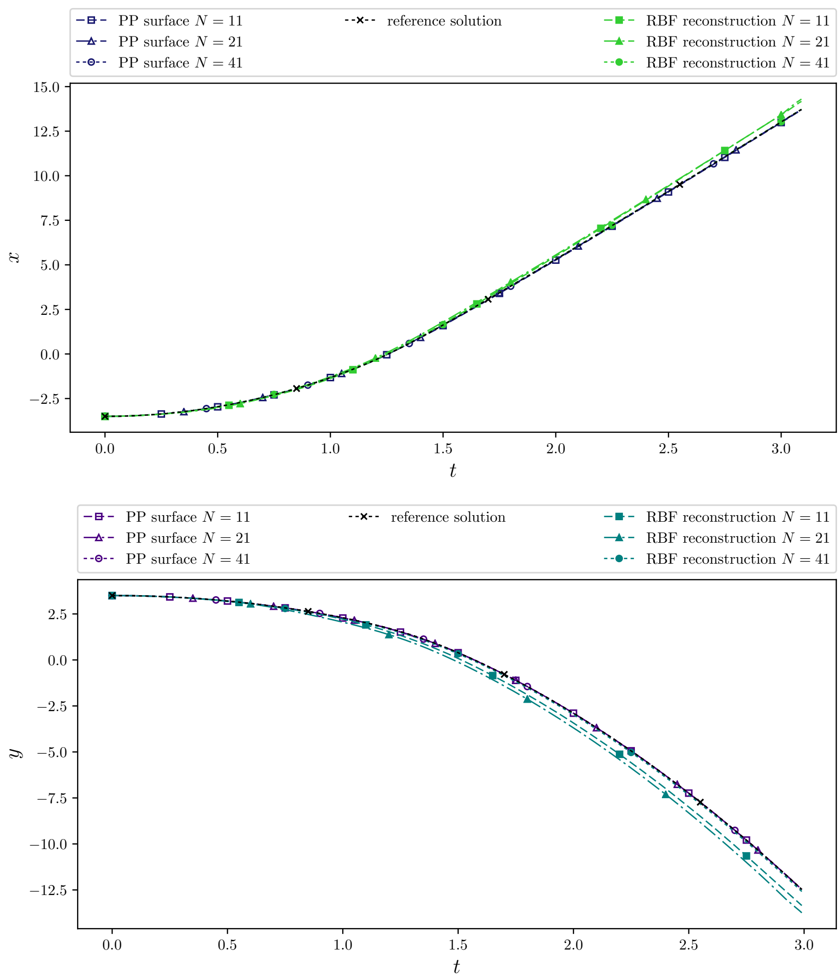


FIGURE 5.14: Test case 2: x- and y-coordinates of the trajectories for friction coefficient $\mu = 0$. The solutions for $N = 11, 21, 41$ as well as the reference solution are shown.

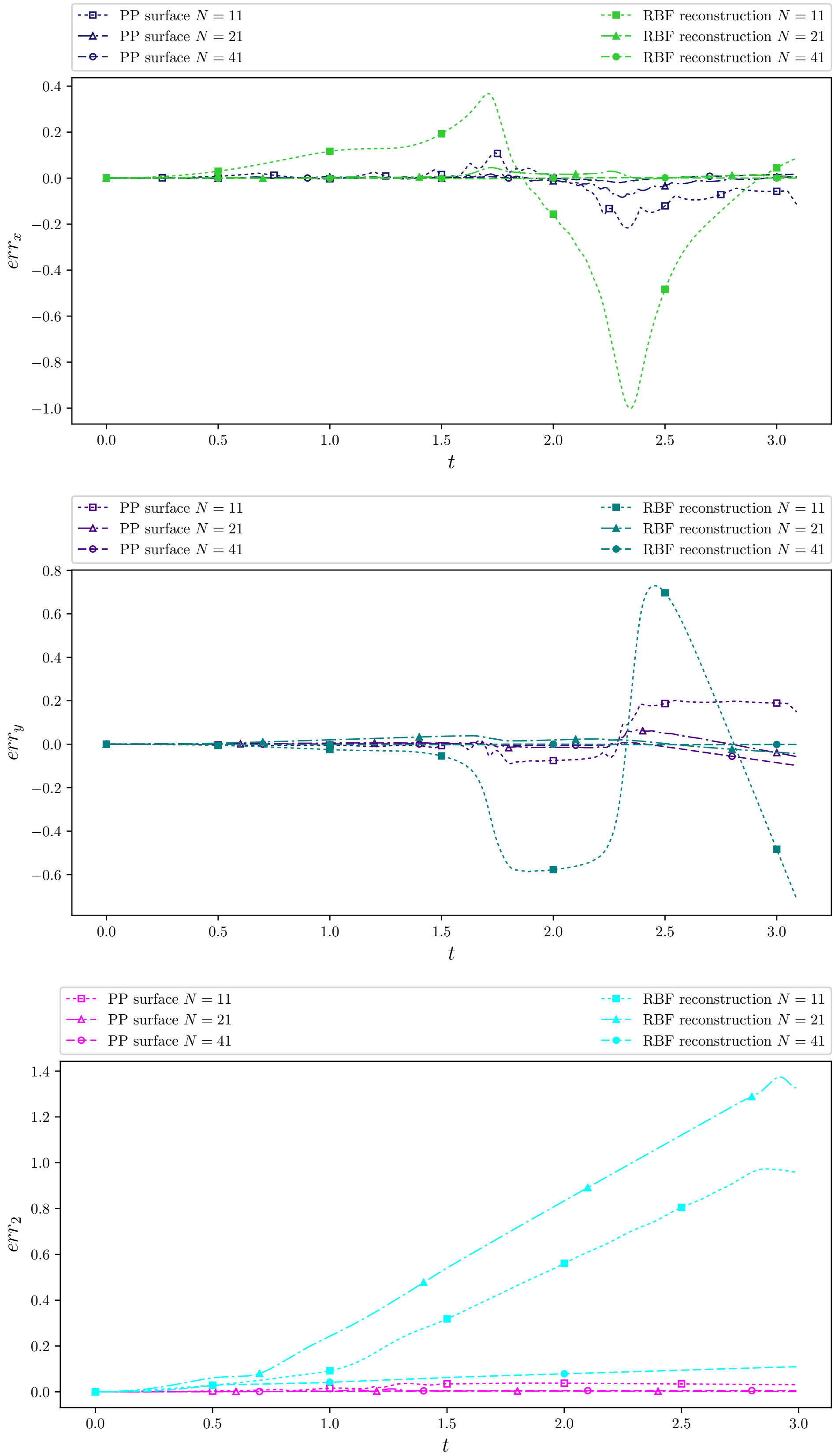


FIGURE 5.15: Test case 2: Absolute errors between the trajectories and the reference solution for $\mu = 0$, split in x- and y-coordinates (top and middle). Euclidean distances between reference and approximated solutions (bottom).

Still, the errors in the solutions on the RBF surface with $N = 11$ and $N = 21$ increasing significantly when the point moves through the region of changing slope.

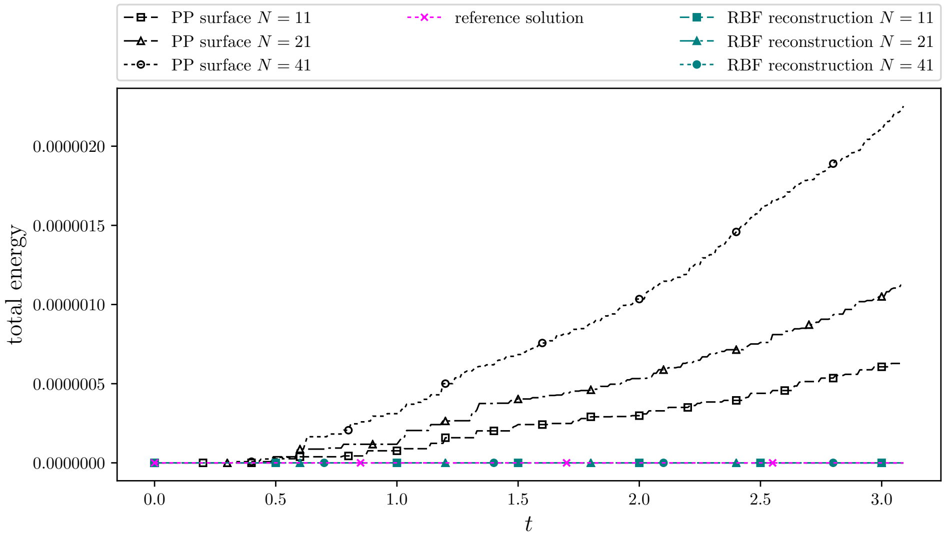


FIGURE 5.16: Test case 2: Total energy plotted for all solutions for $\mu = 0$.

The total energy is conserved again, with a slight gain in energy for the solutions on the PP surfaces. For the solution on the PP surfaces as well as for the reference solution, no gain of energy can be identified from the figure, even though there are marginal changes in the total energy as well, as can be seen from the data (in the range of $10e^{-8}$).

Altogether, the semi-analytical time-integration method in combination with the piecewise planer surfaces performs again significantly better than the LSODE solver in combination with smooth reconstructed surfaces.

Results of test case 3: In the following the results of the paraboloid test case are presented. It is the test case in which we find the strongest difference between an approximated and the reference solution. Part of the trajectories and the surface are shown in figure 5.17. It is seen immediately that the yellow, dash-dotted curve that represents the trajectory for the RBF surface with $N = 11$ grid points, differs significantly from the remaining ones that leave the domain at approximately $\mathbf{x} = (-3, -8)$ and therefore are not further plotted. The yellow trajectory instead changes direction with respect to y twice and and leaves the domain at approximately $\mathbf{x} = (3, -8)$.

The following figures are plotted for a final time $t_{End} = 3.5s$, the time the reference point needs to leave the domain. Hence, the strongest differences between the reference solution and the RBF solution for $N = 11$ is not depicted in the following. Figure 5.18 shows the x- and y-coordinates of the trajectories in time. Here, the

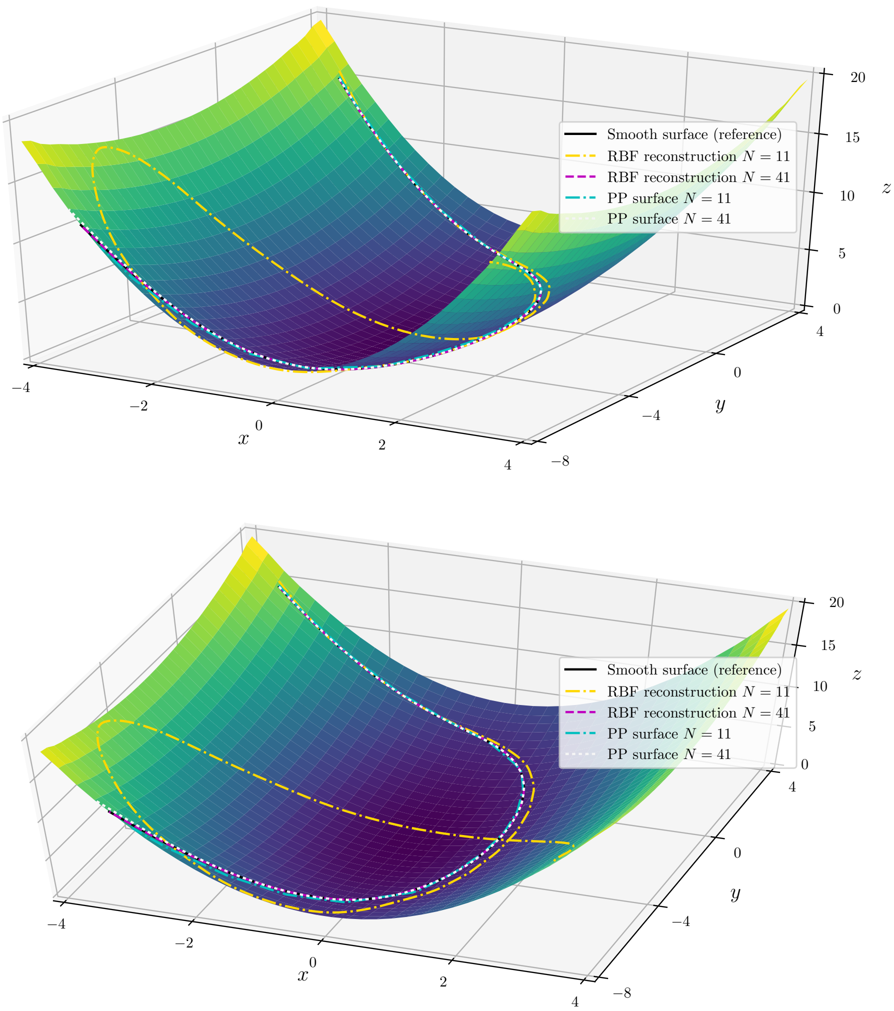


FIGURE 5.17: Test case 3: Surface and trajectories for friction coefficient $\mu = 0$ from two different angles. Only the solutions for $N = 11$ and $N = 21$ as well as the reference solution are plotted.

green dashed curve represents the RBF solution for $N = 11$. It is easily seen that this solution reaches is maximum in x later and in addition the maximum value is larger than in the remaining curves. Moreover it is seen in the upper figure (x) that the PP solution for $N = 11$ leaves the domain earlier than the reference solution (the curve ends in a straight line after $t \approx 3$ which means that after this time the solution does not change any more as the point left the domain). In the lower figure of 5.18 one can see the change in direction with respect to the y-coordinate for the blue dashed curve which represents the RBF solution for $N = 11$. Together 5.18 shows how a small difference between the reference and the approximated solution at the beginning of the simulation can turn into a important one later.

The errors can be seen even better in figure 5.19 where the errors for the RBF solution with $N = 11$ stand with respect to the remaining ones, but also for the PP

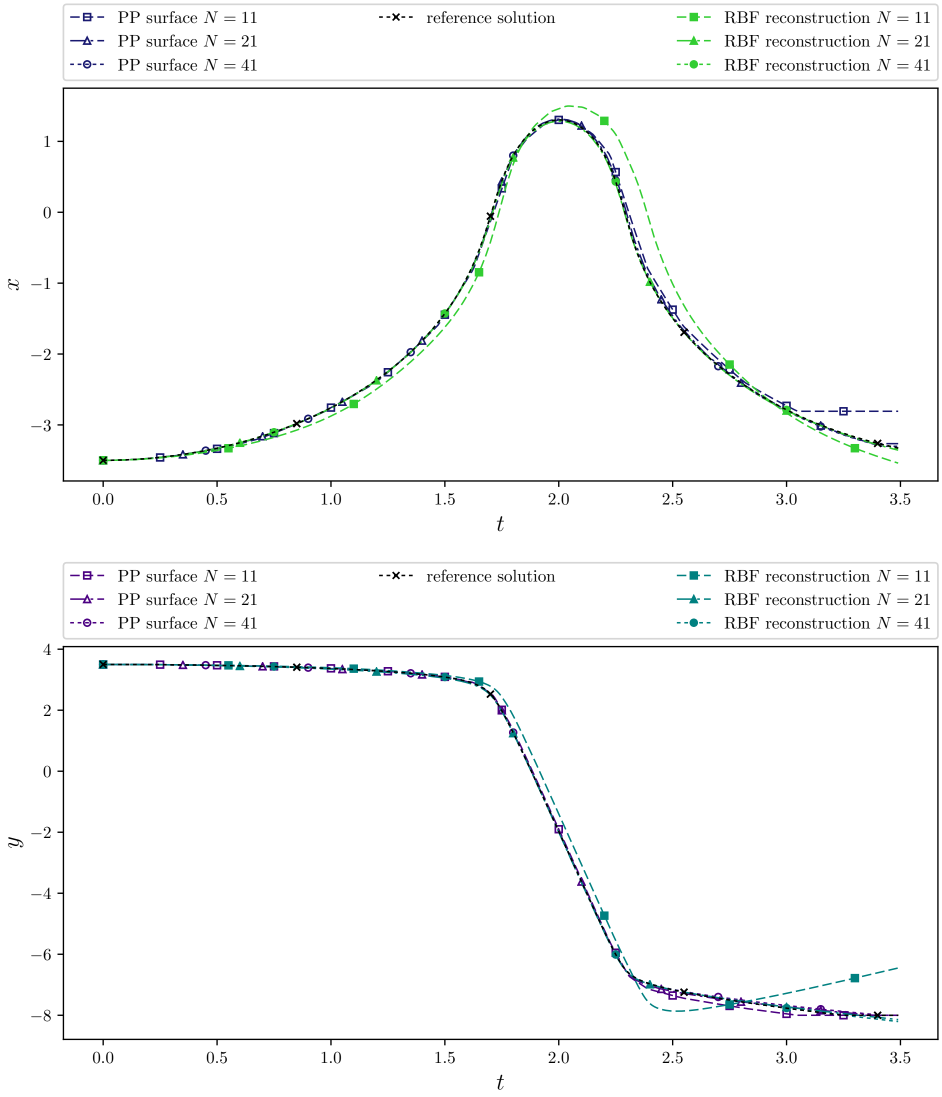


FIGURE 5.18: Test case 3: x- and y-coordinates of the trajectories for friction coefficient $\mu = 0$. The solutions for $N = 11, 21, 41$ as well as the reference solution are shown.

solution with $N = 11$ it can be seen again that a small error at the beginning can turn into a significantly larger one later on. In general however, the errors are much smaller than in the previous two test cases and the solution lie closer to each other.

Finally, the total energy for the approximated solutions and the reference solution are depicted in figure 5.20. Again the energy is practically conserved for all solutions.

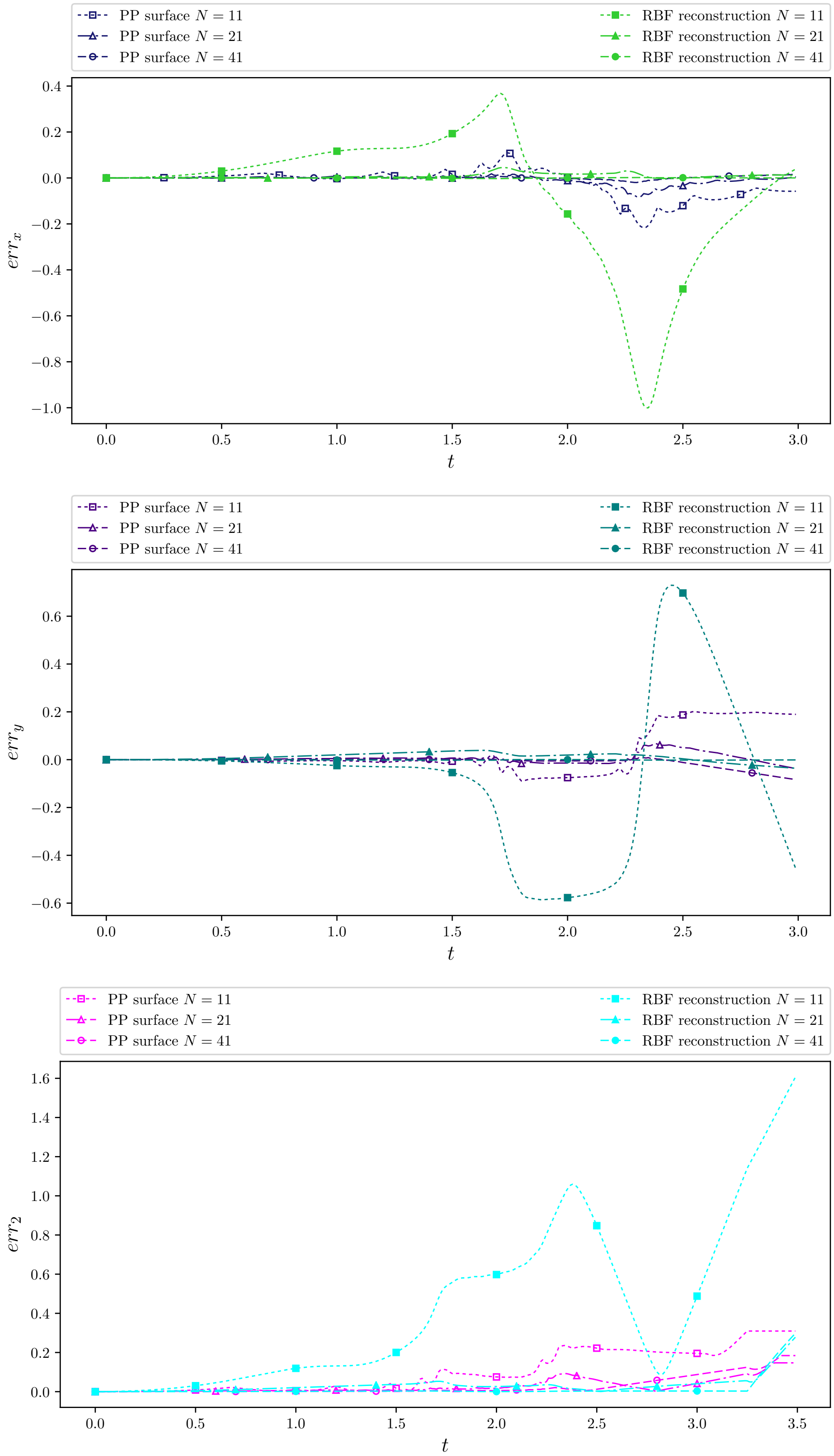


FIGURE 5.19: Test case 3: Absolute errors between the trajectories and the reference solution for $\mu = 0$, split in x- and y-coordinates (top and middle). Euclidean distances between reference and approximated solutions (bottom).

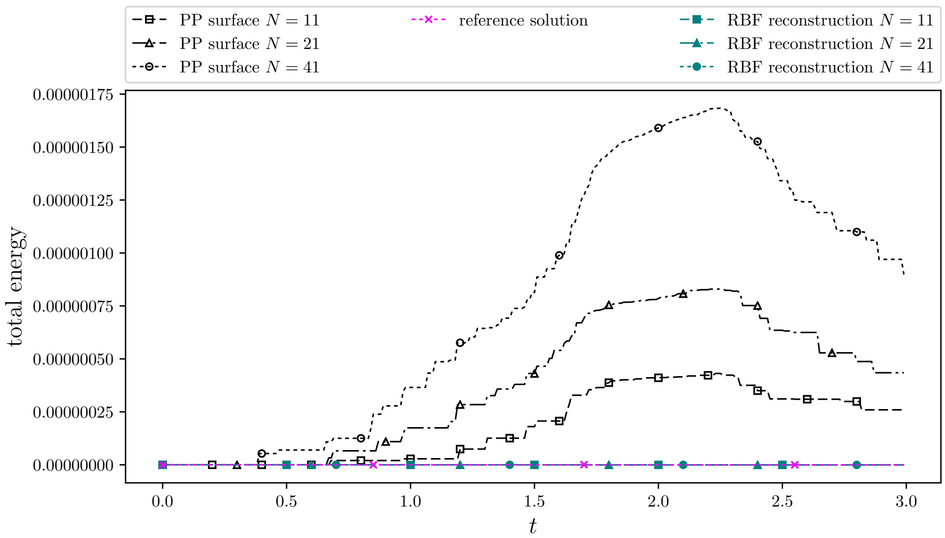


FIGURE 5.20: Test case 3: Total energy plotted for all solutions for $\mu = 0$.

Results of test case 4: The last test case is a combination of the previous two and is characterized by a quick change of slope in x-direction and a soft change in slope in y-direction.

Figure 5.21 shows the surface and the approximated solutions for $N = 11$ and $N = 41$, respectively, as well as the reference solution from two different angles. One can see immediately that the solution obtained on the piecewise planar reconstruction with $N = 11$ grid points differs significantly from all the other solutions.

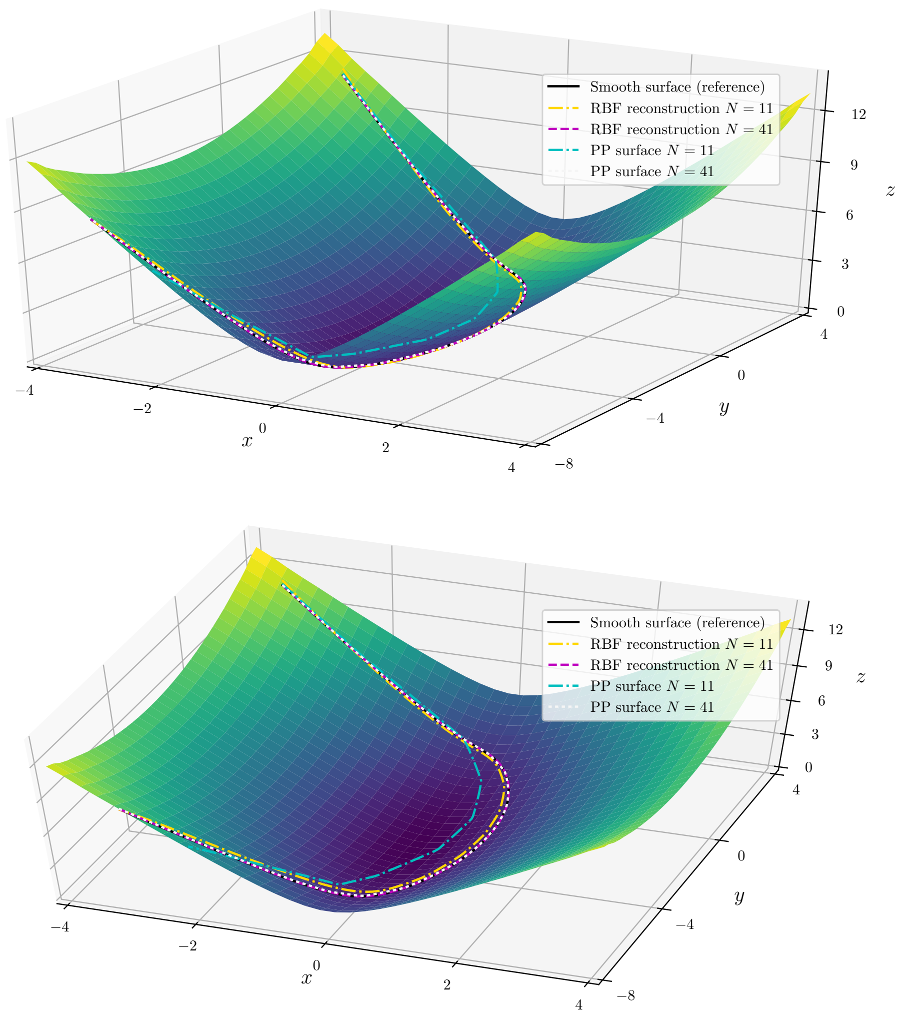


FIGURE 5.21: Test case 4: Surface and trajectories for friction coefficient $\mu = 0$ from two different angles. Only the solutions for $N = 11$ and $N = 21$ as well as the reference solution are plotted.

This becomes even more evident in figure 5.22 were the x- and y-coordinates of the trajectories are plotted, the approximated ones for all refinement levels as well as the reference. The difference between the PP solution for $N = 11$ and the remaining solutions is most evident in the upper figure (x-coordinate) but can be seen also in the lower one (y-coordinate).

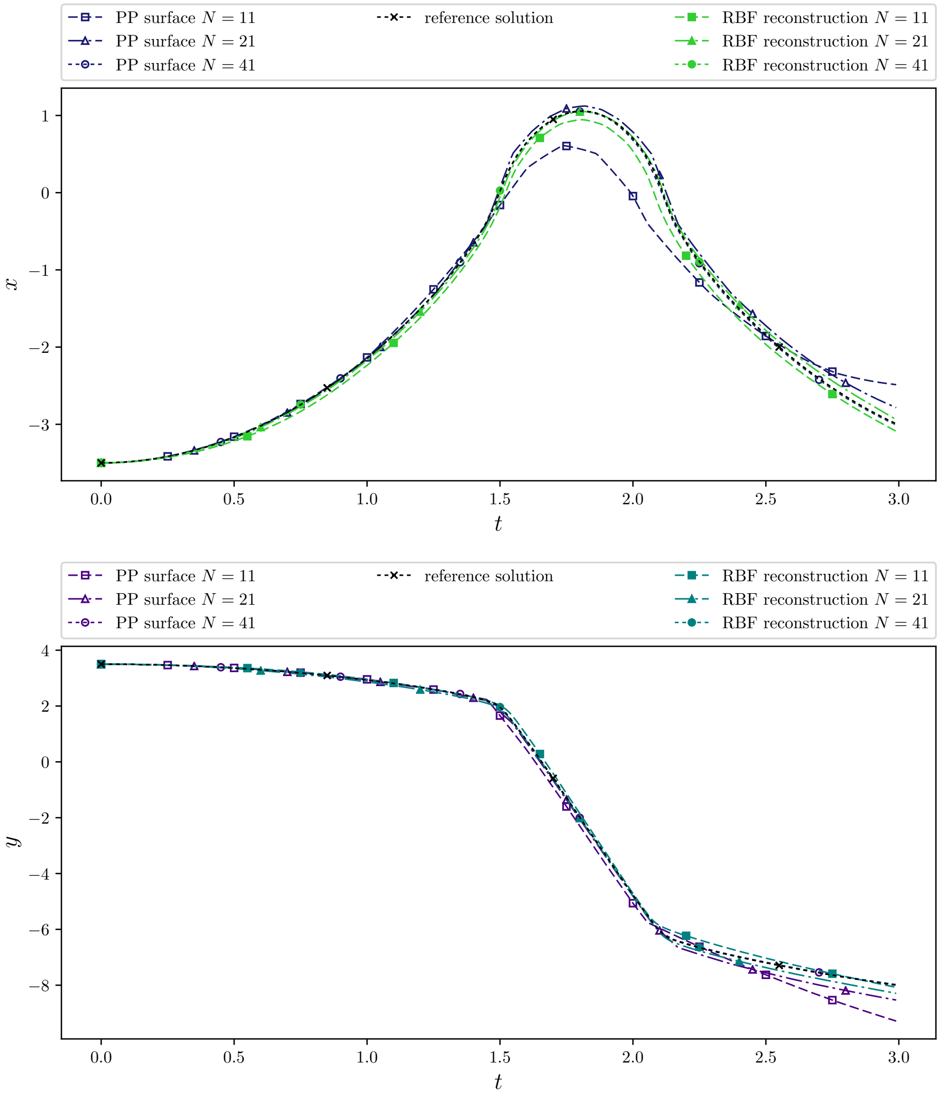


FIGURE 5.22: Test case 4: x- and y-coordinates of the trajectories for friction coefficient $\mu = 0$. The solutions for $N = 11, 21, 41$ as well as the reference solution are shown.

Figure 5.23, which shows the absolute errors in terms of x and y as well as the Euclidean distances between the trajectories illustrates this even better, showing a strong peak in the x- as well as in the y-error (top and middle figure). Moreover there are significant errors for the PP surface and $N = 21$, and for the RBF surface for $N = 11$, in both the x- and the y-error. The same picture is found looking at the bottom figure (L_2-distance). For the PP surface with $N = 11$ the strongest difference to the reference solution is found while errors are still significant for the PP surface with $N = 21$ and the RBF surface with $N = 11$.

The total energy is again conserved for all solutions even though there are marginal changes in the PP based solutions.

Altogether the smooth reconstructed surfaces in combination with the here presented semi-analytical time-integration method perform better in this test case. For the final refinement step $N = 41$, the differences are, like in the previous two test cases, negligible.

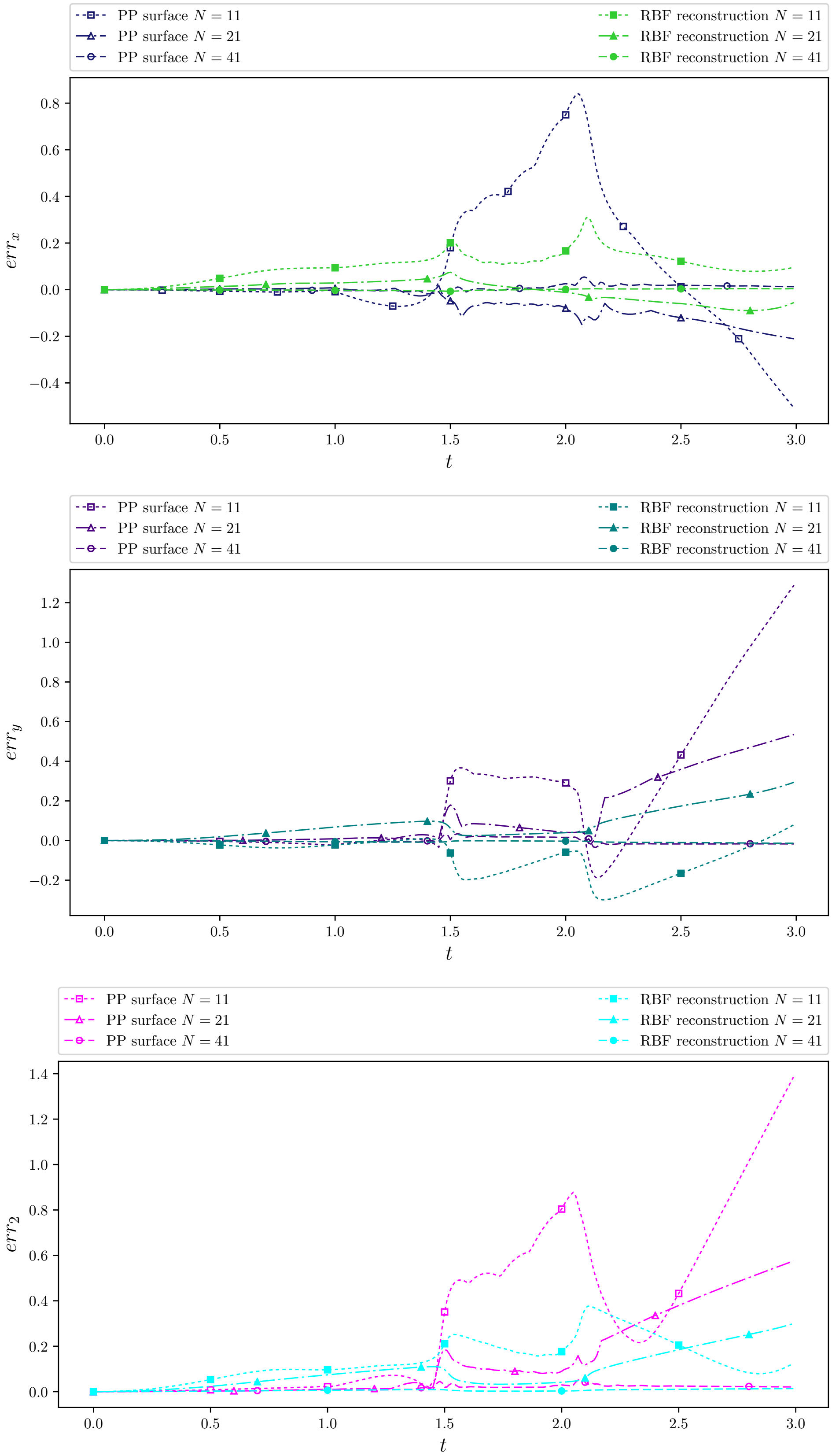


FIGURE 5.23: Test case 4: Absolute errors between the trajectories and the reference solution for $\mu = 0$, split in x- and y-coordinates (top and middle). Euclidean distances between reference and approximated solutions (bottom).

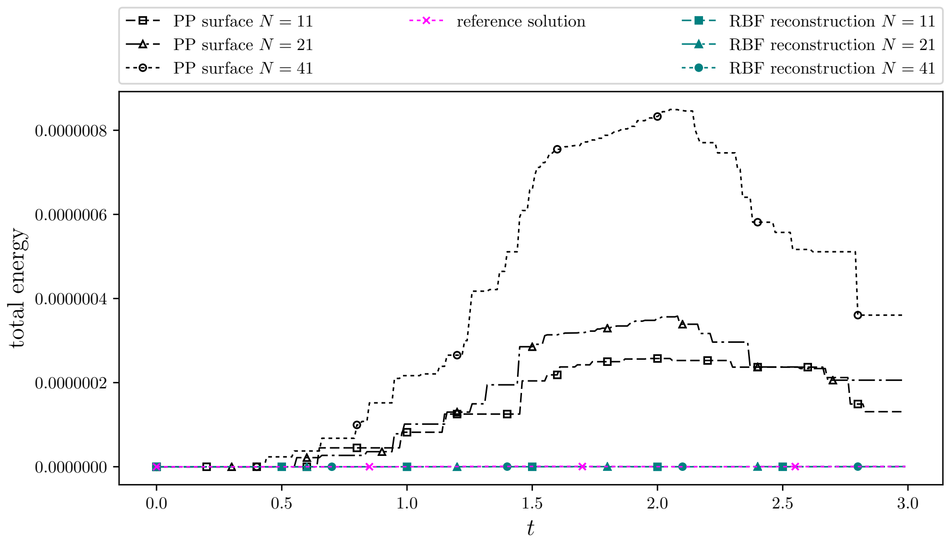


FIGURE 5.24: Test case 4: Total energy plotted for all solutions for $\mu = 0$.

5.3.3 Comparison of solutions computed on triangulated versus smooth surfaces under the influence of friction

The four test cases are repeated setting the friction coefficient $\mu = 0.15$. The results are presented in the following. The solutions computed on smooth surfaces – obtained using smooth surface reconstruction by means of radial basis functions – are again compared to the solutions computed on piecewise planar surfaces in a qualitative way.

As in the frictionless case, surfaces with three different levels of refinement are used where the number of grid points is still $N = 11, 21, 41$.

Results of test case 1: The test-surface for the full domain and the trajectories of the point for the surfaces with $N = 11$ and $N = 41$ points are shown in figure 5.25. The two trajectories for the smooth surfaces, obtained using the radial basis functions, are denoted by RBF uning a yellow dash-dotted line for $N = 11$ and a magenta dashed line for $N = 41$. The respective trajectories of the solutions on the piecewise planar surfaces (PP) are denoted by a turquoise dash-dotted ($N = 11$) and a white dotted ($N = 41$) line.

In comparison with the frictionless case, the trajectories leave the domain at smaller values of x (trajectories are more bended towards $x = 0$) with a slightly larger variation in between the trajectories themselves.

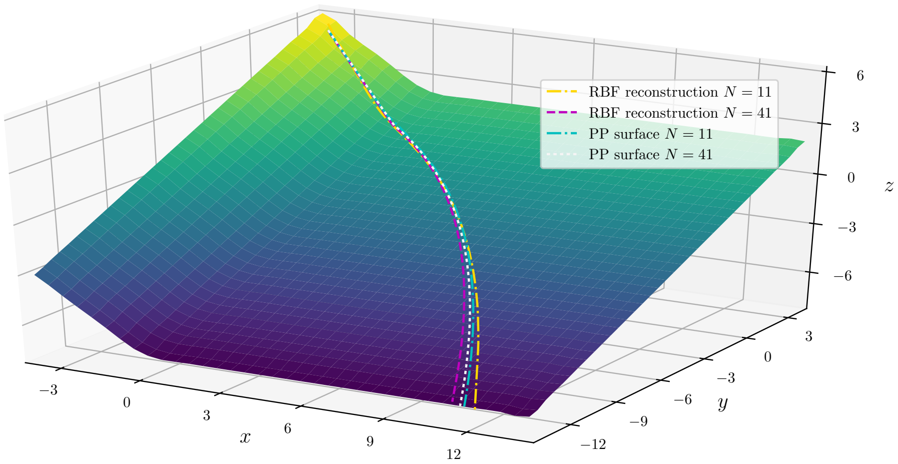


FIGURE 5.25: Test case 1: x- and y-coordinates of the trajectories for friction coefficient $\mu = 0.15$. The solutions for $N = 11, 21, 41$ as well as the reference solution are shown.

Overall, the results look very similar to the frictionless case. The solutions on the piecewise planar surfaces perform significantly better than the ones one the RBF surfaces if compared to the reference solution as can already be seen in figure 5.26.

While the PP surface solutions appear to lie on one curve together with the reference solution, there is a considerably larger variation in the curves belonging to the RBF surfaces. This is even more evident in figure 5.27 where on the top and in the middle the errors in x- and y-coordinate are depicted. The bottom picture shows – as in the frictionless case – the distance between the approximated and the reference trajectories, measured in the L_2-norm. Altogether, the solutions on the piecewise planar surfaces are significantly closer to the reference solution, even though the errors are larger than for $\mu = 0$.

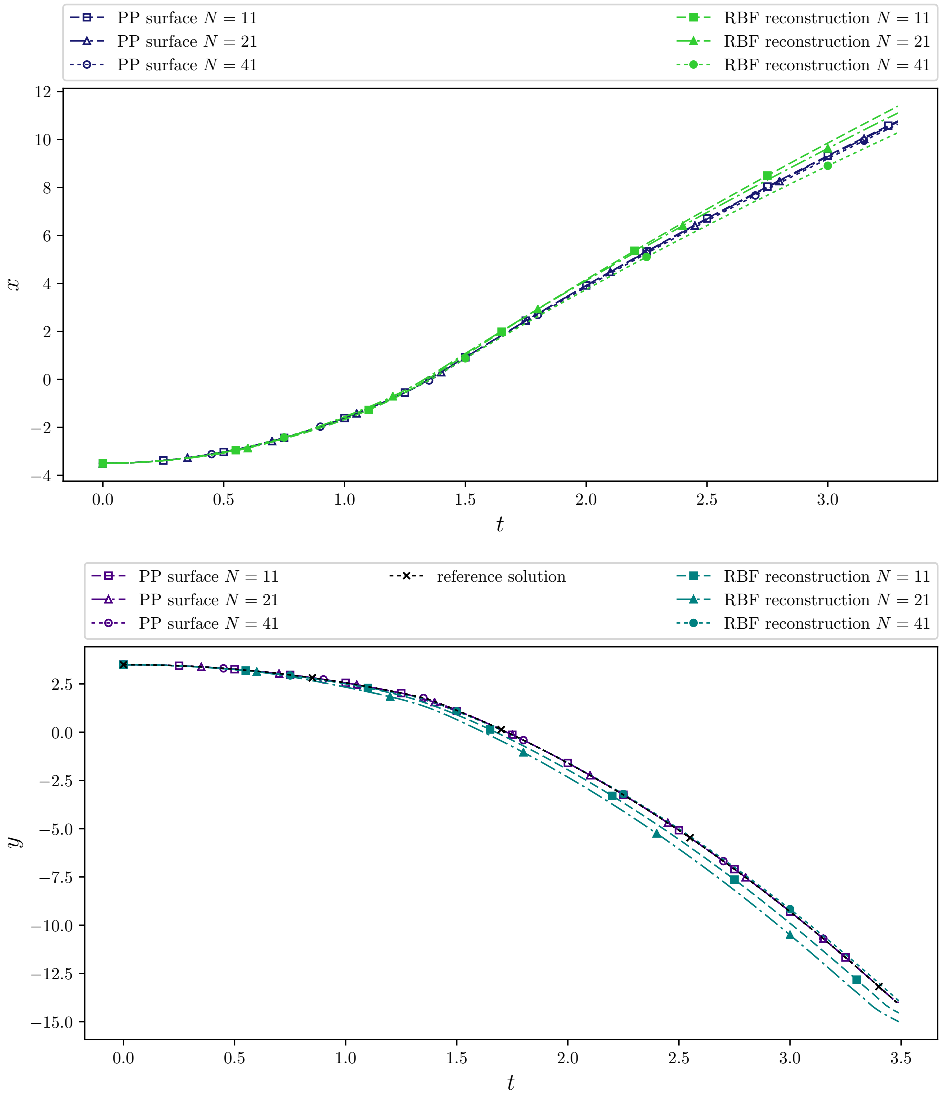


FIGURE 5.26: Test case 1: x- and y-coordinates of the trajectories for friction coefficient $\mu = 0.15$. The solutions for $N = 11, 21, 41$ as well as the reference solution are shown.

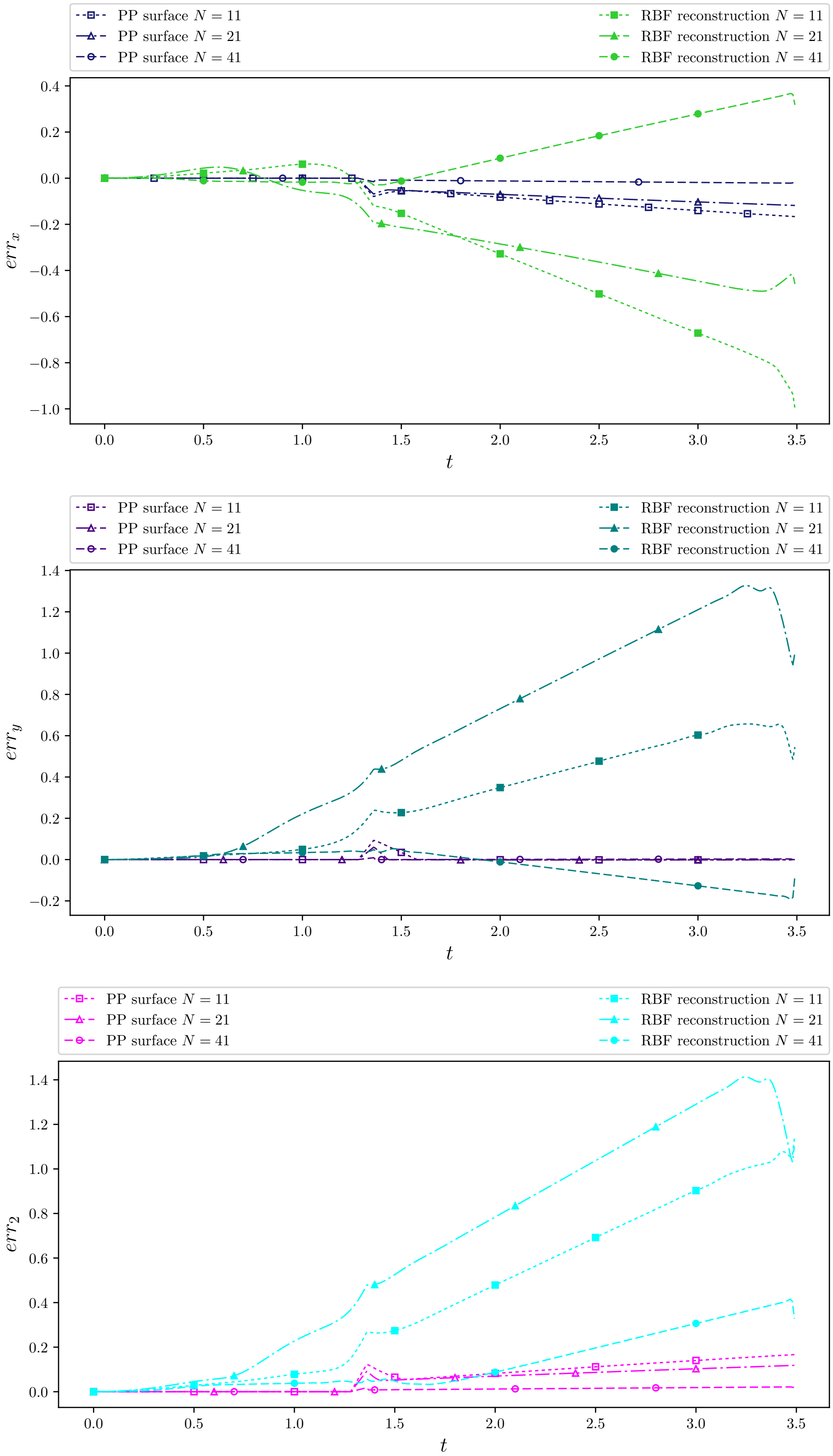


FIGURE 5.27: Test case 1: Absolute errors between the trajectories and the reference solution for $\mu = 0.15$, split in x- and y-coordinates (top and middle). Euclidean distances between reference and approximated solutions (bottom).

Results of test case 2: Like already in the frictionless case, the results of this simulation are very similar to test case 1 but there is a less strong variation between the different trajectories.

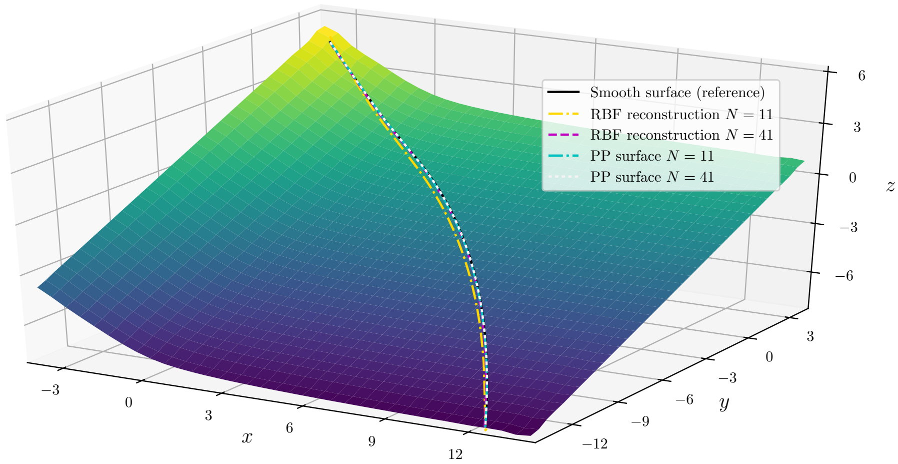


FIGURE 5.28: Test case 2: Surface and trajectories for friction coefficient $\mu = 0.15$. Only the solutions for $N = 11$ and $N = 21$ as well as the reference solution are plotted.

As can be seen in figure 5.29, the trajectories computed on the piecewise planar surfaces seemingly overlap while there is a stronger variation in the solutions on the smooth reconstructed surfaces.

Again, this is seen better in figure 5.30, in particular in the bottom picture that depicts the Euclidean distance between the approximated and the reference solution. The errors of the solutions computed on the piecewise planar surfaces are small and the solution on the finest RBF surface is in a comparable range. The solutions computed on the RBF surfaces with $N = 11$ and $N = 21$ instead are several times larger such that overall also in this test, the solutions computed on the piecewise planar surfaces are clearly better than those computed on the smooth reconstructed surfaces.

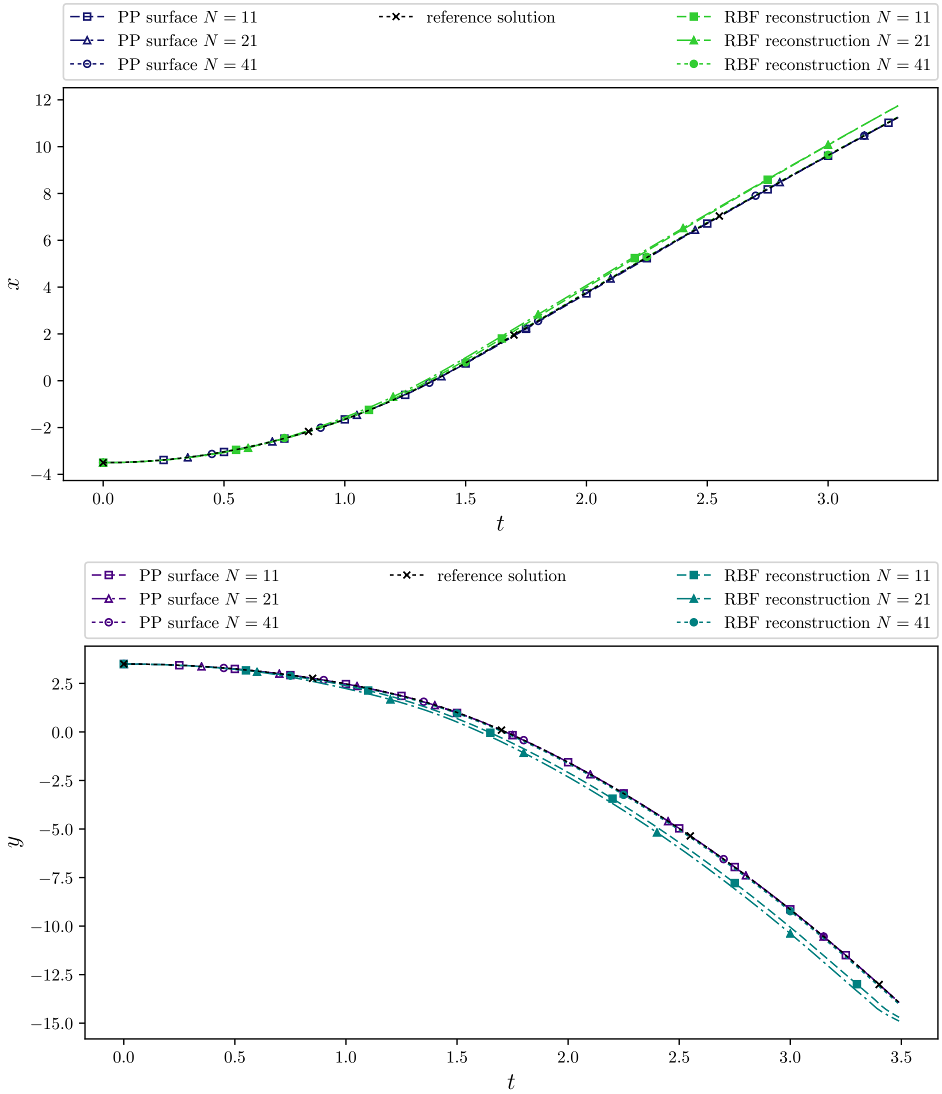


FIGURE 5.29: Test case 2: x- and y-coordinates of the trajectories for friction coefficient $\mu = 0.15$. The solutions for $N = 11, 21, 41$ as well as the reference solution are shown.

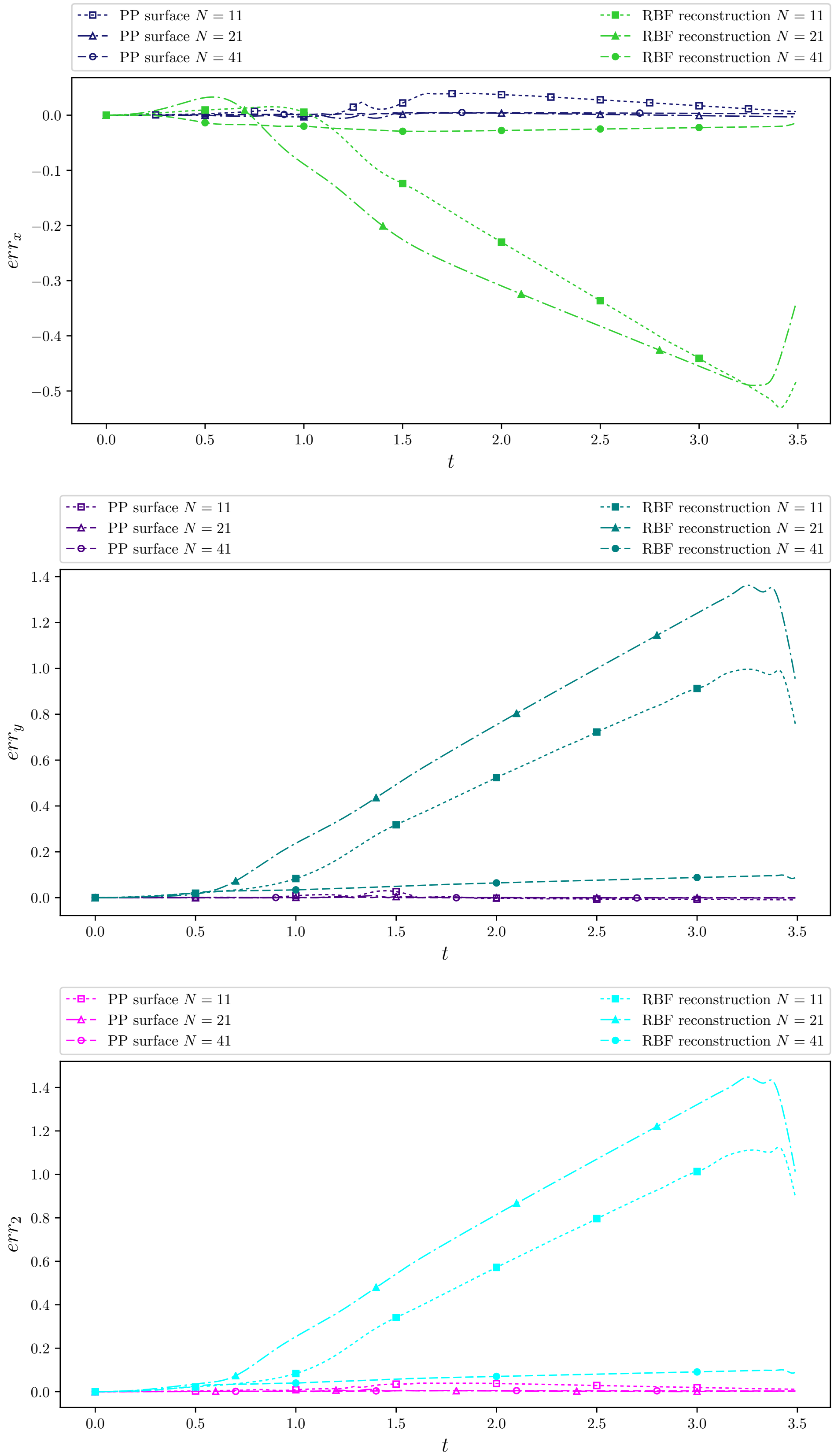


FIGURE 5.30: Test case 2: Absolute errors between the trajectories and the reference solution for $\mu = 0.15$, split in x- and y-coordinates (top and middle). Euclidean distances between reference and approximated solutions (bottom).

Results of test case 3: In contrast to the frictionless cases and the first two cases including friction, the motion in this test will stop and none of the the trajectories leaves the domain. The motion can therefore be followed until the end (i.e. until the motion finally stops). Figure 5.31 shows this from two different perspectives.

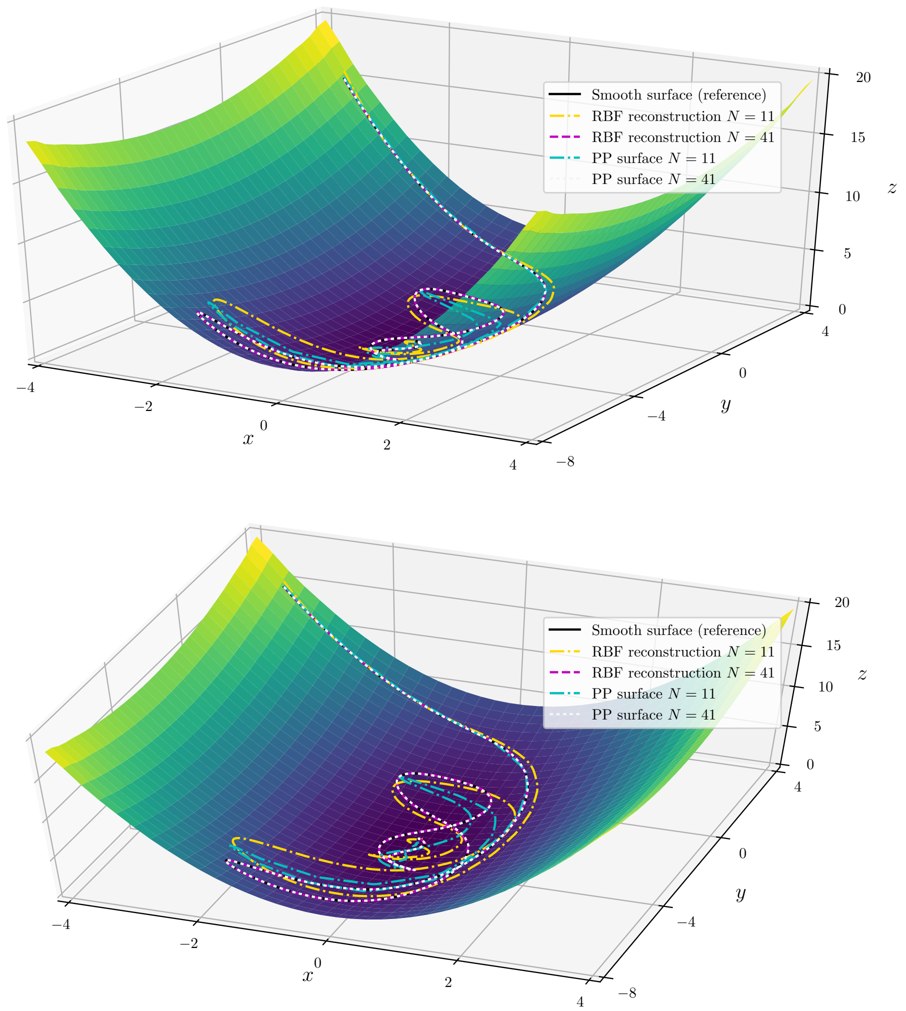


FIGURE 5.31: Test case 3: Surface and trajectories for friction coefficient $\mu = 0.15$ from two different angles. Only the solutions for $N = 11$ and $N = 21$ as well as the reference solution are plotted.

The x- and y-coordinates of the different trajectories are seen more in detail in figure 5.32. They suggest that both methods (the semi-analytical solution on the PP surfaces and the solutions computed on the smooth reconstructed surfaces using LSODE) perform comparably well in terms of accuracy.

This is confirmed in figure 5.33 which shows the different errors between the approximated and the reference solutions. While the maximum error is in each case stronger for the solution on the RBF surface with $N = 11$, there are considerable errors as well for solutions on the PP surfaces for $N = 11$ and $N = 21$. For $N = 41$

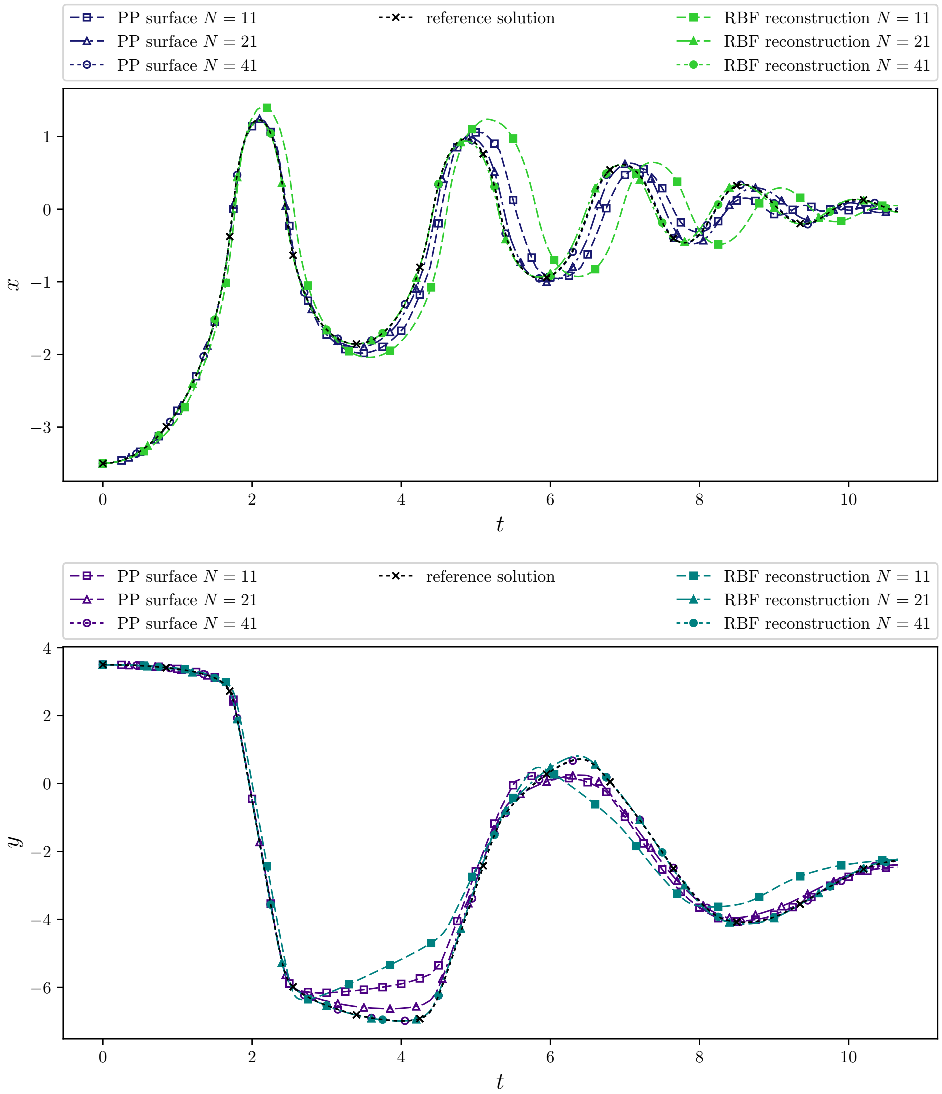


FIGURE 5.32: Test case 3: x- and y-coordinates of the trajectories for friction coefficient $\mu = 0.15$. The solutions for $N = 11, 21, 41$ as well as the reference solution are shown.

the errors become comparable small for both methods even if the RBF based methods performs slightly better.

Altogether, the results do not suggest that any of the two methods should be preferred over the other here, as long as the surface reconstruction is sufficiently accurate.

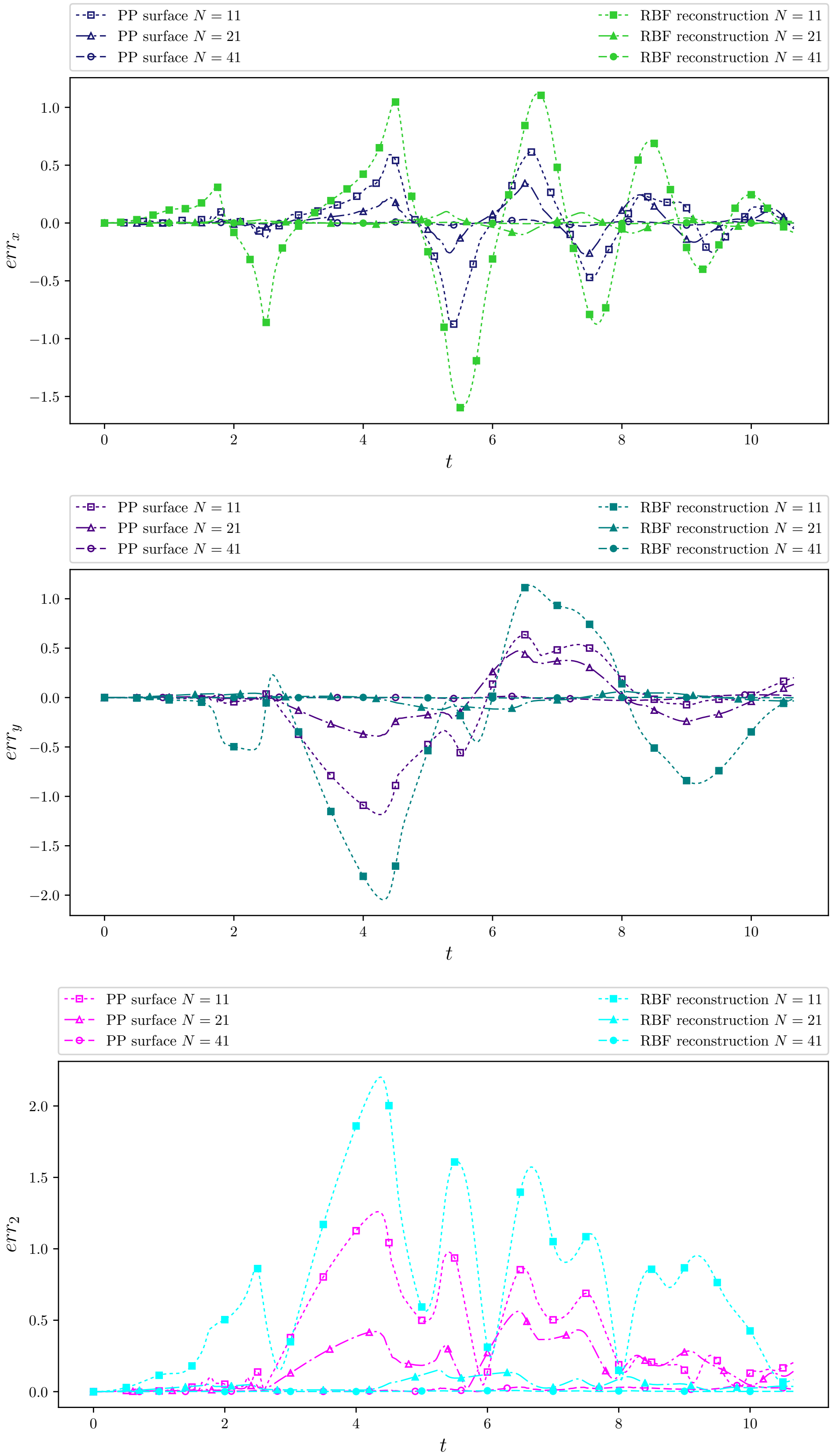


FIGURE 5.33: Test case 3: Absolute errors between the trajectories and the reference solution for $\mu = 0.15$, split in x- and y-coordinates (top and middle). Euclidean distances between reference and approximated solutions (bottom).

Results of test case 4: A very similar picture is seen also in the last test case. Again the solutions for the coarser surface reconstructions perform less good than the ones for the finer reconstructions. However, the solution computed on the piecewise planar surface for $N = 11$ varies strongly from the remaining solutions as the motion stops much earlier in time.

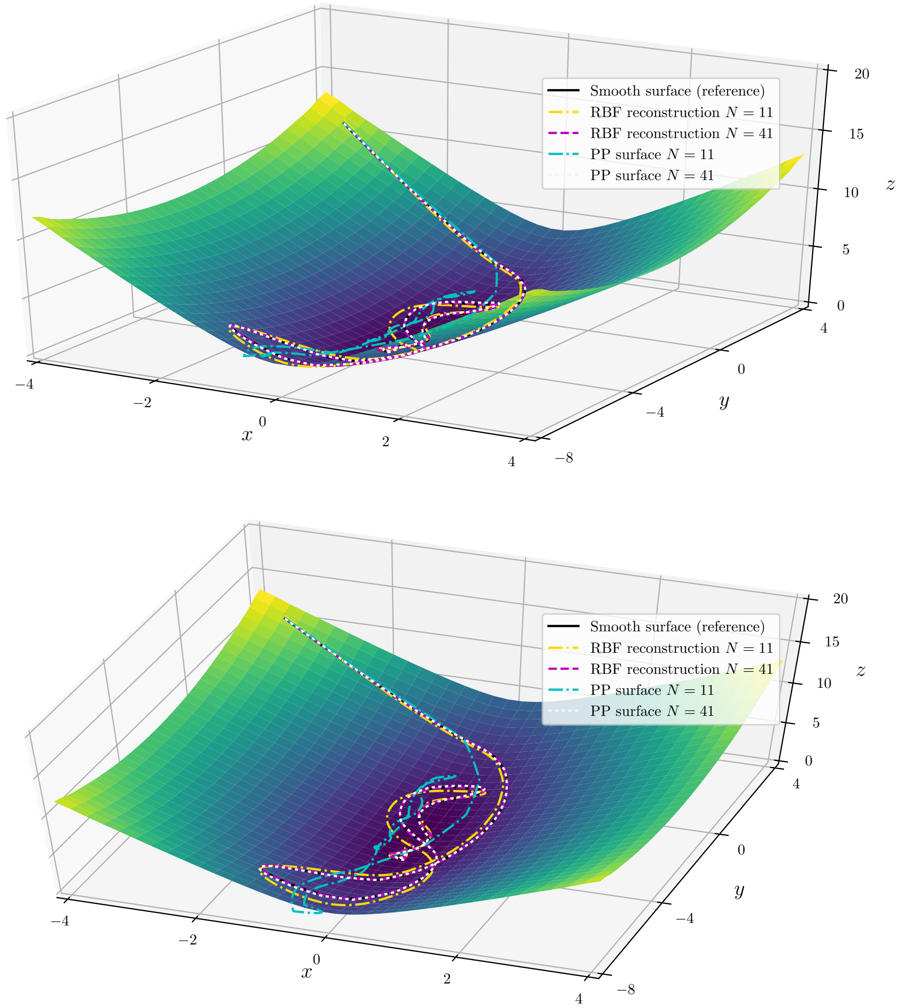


FIGURE 5.34: Test case 4: Surface and trajectories for friction coefficient $\mu = 0.15$ from two different angles. Only the solutions for $N = 11$ and $N = 21$ as well as the reference solution are plotted.

This is seen particularly well in the following figures. Figure 5.35 illustrates the much faster decay of the PP solution for $N = 11$ in x-direction as well as the stronger deflection in negative y-direction (it is the only solution that slightly exists from the given domain).

Finally figure 5.36 confirms that the best three solutions are those computed on the smooth reconstructed surfaces for $N = 21$ and $N = 41$ and the one computed on the finest picewise planar reconstruction ($N = 41$). The three worst solutions are

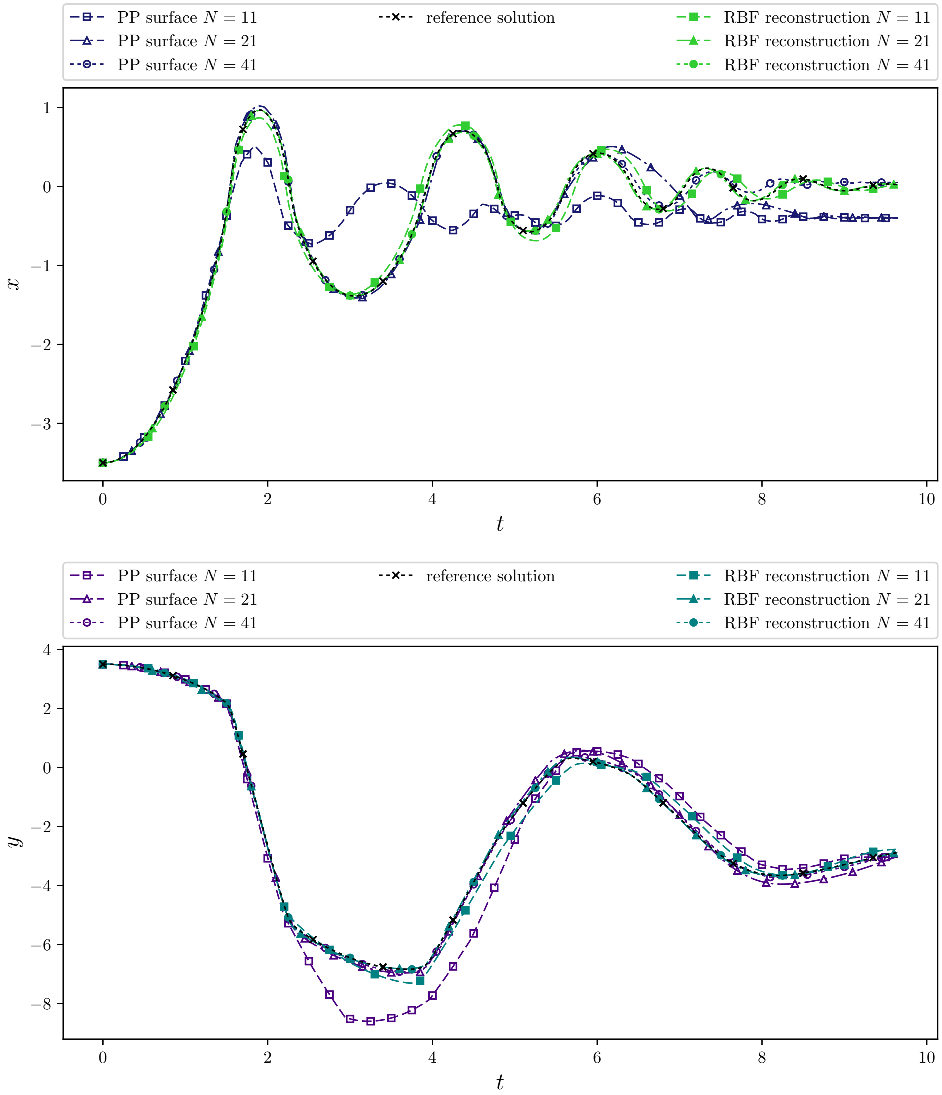


FIGURE 5.35: Test case 4: x- and y-coordinates of the trajectories for friction coefficient $\mu = 0.15$. The solutions for $N = 11, 21, 41$ as well as the reference solution are shown.

in this order the already discussed one one the coarsest piecewise planar solution, followed by the solution on the coarsest RBF surface and the one one the piecewise planar solution with $N = 21$.

Altogether the solutions computed on the RBF surfaces perform a little better in this test case, even though for a sufficiently fine discretization the solutions of both methods are comparable.

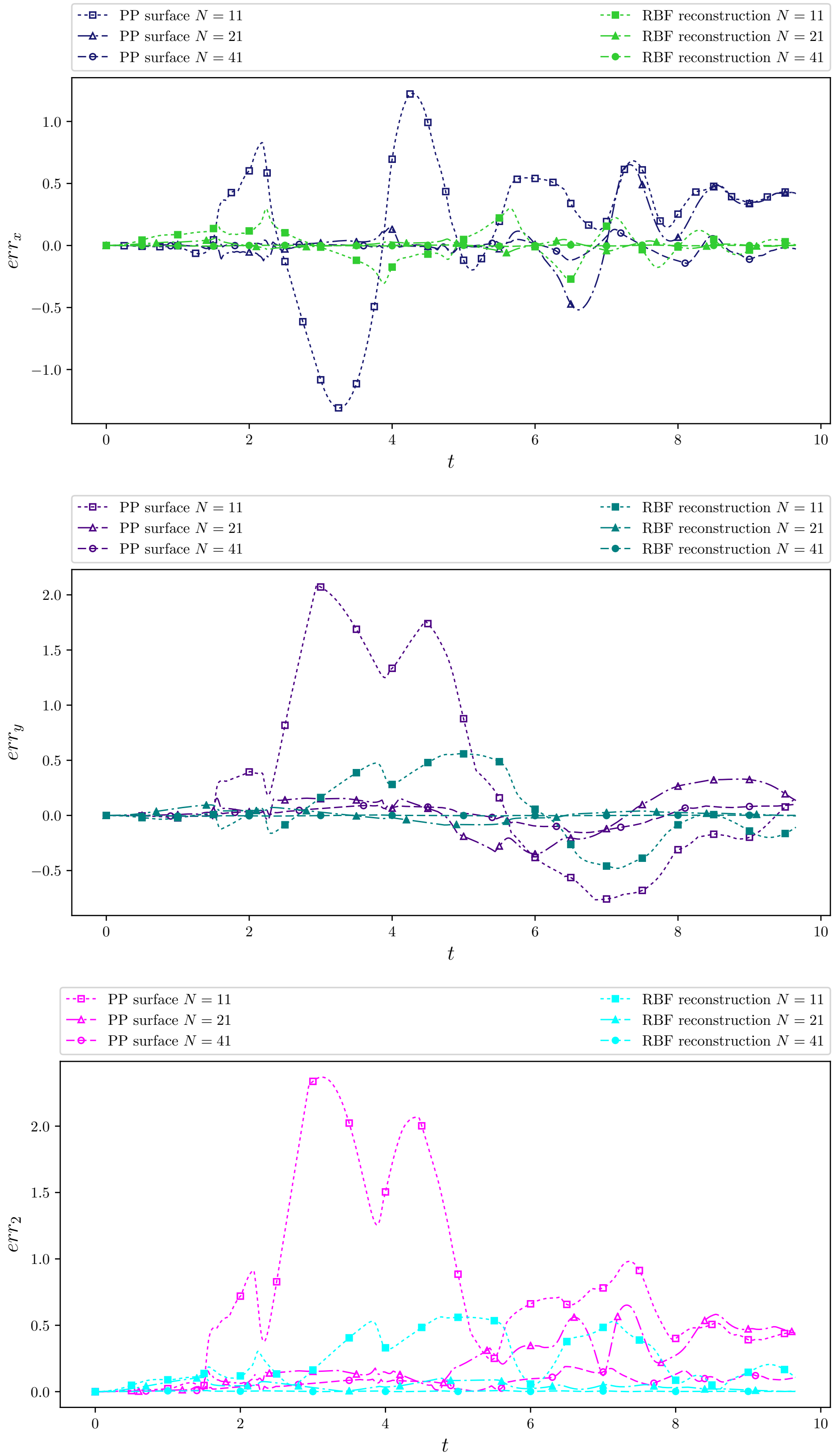


FIGURE 5.36: Test case 4: Absolute errors between the trajectories and the reference solution for $\mu = 0.15$, split in x- and y-coordinates (top and middle). Euclidean distances between reference and approximated solutions (bottom).

5.3.4 Short evaluation of the two presented methods

The test cases performed in the previous two subsections showed that – like in the 1-dimensional case – both methods have their advantages and disadvantages. Both perform well in terms of accuracy and both are able to conserve the total energy in the frictionless case. For larger surfaces however, the usage of radial basis functions for the reconstruction is manageable only if the surface is defined piecewise as otherwise the function would become to heavy. Also other reconstruction methods, like bicubic splines could be an alternative if a higher order reconstruction is favoured but also these bust be defined piecewise.

In therms of computational speed the computations on piecewise planar surfaces were significantly faster, in particular for finer grids.

As the semi-analytical solution on piecewise defined grids did not exhibit serious disadvantages, performs equally well in terms of accuracy (compared to the combination of RBF surfaces and the LSODE solver) and is much faster in terms of computational time, this method is – like already in the 1-dimensional case – the preferred one here.

5.4 Numerical results 2: Convergence test and more complex surfaces

In the current section only the qualitative behaviour of the two methods was investigated and compared with each other. The following section is now dedicated to a quantitative investigation of the method used to solve the equations of motion on piecewise planar surfaces, using a semi-analytical time-integration scheme. It shall be shown that the solution on piecewise planar surfaces really converges and that it does this with a reasonable rate. Moreover a consistency test will be performed, thus showing that the solution on piecewise planar surfaces not only converges, but converges to the desired reference solution, i.e. it is physically meaningful.

5.4.1 Description of the test problem

Usually, in the numerical solution of time-dependent ODEs, only the time-step $\triangle t$ is varied but (by definition of an ODE) there is no spatial discretization. This is the case if equation (5.1) is solved numerically (using e.g. the Livermore Solver for stiff ODEs) on an analytically given, smooth surface f. If instead a discretized surface is used for the solution of equation (5.1), there is more than one possible triangulation of f such that, in some sense, also the spatial discretization is involved in the solution (even though not in the sense of a PDE). Therefore two different convergence tests are performed in the following. First we will check if, for a given discretization, the solution of equation (5.1) is converging to some reference solution. Afterwards it will be check if the solution is also consistent, respectively if refining both, the time-step and the spatial discretization at the same time, the solutions are converging towards the reference solution on the corresponding analytical solution. If the method is consistent, the solution on the discretized surface is converging towards the solution on the analytical surface if the discretized surface converges to the analytical one. However, it is important to keep in mind that two different problems are considered here. One is the solution of equation (5.1) on a discretized surface using the method developed here, the second is the question if this method is producing results comparable to the numerical solution on analytical surfaces using standard solvers for (stiff) ODEs.

The test-surface $f(x,y)$ is a paraboloid where

$$f(x,y) = x^2 + y^2.$$

The initial conditions in each test are given by $(x, y, v_x, v_y) = (2, 1, 0, 1)$.

5.4.2 Convergence test 1:

Here, the discretization is kept constant and only the step-size $\triangle t$ is varying. The test is performed several times with a different spatial refinement each where N is giving the number of data points per coordinate direction. The first simulation was run with some step-size $\triangle t$ that can vary for different discretizations. In each of the following simulations the step-size was divided by two. The finest resolution $\triangle t/1024$ was used as reference solution. A standard four-stage Runge-Kutta (RK) scheme was chosen for the numerical time-integration along the planar elements of the surface. Table 5.1 shows the convergence rates for errors measured in the L_2-norm for all discretizations. The results for errors measured in L_1- and L_∞-norm vary little from this. Little more than linear convergence can be achieved. In comparison, using the same RK scheme to solve the system on an analytically given surface, third order of convergence is achieved on the paraboloid. This result is not surprising as in the first case the solution is neither smooth (trajectory) nor continuous (velocity).

5.4.3 Convergence test 2:

In order to verify if the method also converges towards the solution obtained on smooth grids, we solve equation 5.1 on discretized surfaces, gradually refining both, the time-step and the grid-width of the triangulation. By $\triangle x$ and $\triangle y$ we denote the average diameter of a triangle measured along the x- and y-axis (grid-width). Starting

step-size/N	512	1024	2048
$\triangle t/2$	0.790	0.680	0.593
$\triangle t/4$	1.134	0.758	0.724
$\triangle t/8$	0.904	0.825	0.932
$\triangle t/16$	0.929	0.974	1.126
$\triangle t/32$	0.969	1.049	0.985
$\triangle t/64$	1.019	1.037	1.014
$\triangle t/128$	1.174	1.114	1.111
$\triangle t/256$	0.960	1.192	1.184
$\triangle t/512$	1.219	1.507	1.777

TABLE 5.1: L_2 convergence rates for three different discretizations (512, 1024, 2048 points) of the paraboloid.

N	L_1-error	L_1-order	L_2-error	L_2-order	L_∞-error	L_∞-order
64	1.33e-1	-	1.78e-1	-	4.10e-1	-
128	6.69e-02	0.99	8.24e-02	1.11	1.77e-01	1.21
256	4.33e-02	0.63	5.65e-02	0.55	1.27e-01	0.48
512	6.98e-03	2.63	9.47e-02	2.58	2.30e-02	2.46
1024	2.71e-03	1.36	3.46e-03	1.45	8.02e-03	1.52
2048	9.00e-04	1.59	1.13e-03	1.62	2.62e-03	1.61

TABLE 5.2: Errors and convergence rates for the consistency test

with $N = 64$, the number of points is doubled with each refinement step. Hence, $\triangle x$ and $\triangle y$ are each divided by two in every consecutive simulation. In order to adjust the time-step $\triangle t$ accordingly, the Courant number $C = \triangle t \left(\frac{u_x}{\triangle x} + \frac{u_y}{\triangle y} \right)$ is used. This number usually is applicable only for PDEs, but in our case provides a meaningful constraint to the choice of the time-step during the spatial refinement. Assuming that for different triangulations the velocity of the point at a certain position on the surface does not vary critically, in order to keep the Courant number constant, also the number of time-steps must be doubled, meaning that the step-size $\triangle t$ has to be divided by two. The reference solution in this case is the numerical solution on the corresponding analytical surface. For the time integration a four-stage Runge-Kutta scheme was used in both cases. Table 5.2 shows the L_1-, L_2- and L_∞-errors and the corresponding orders of convergence. The different trajectories (denoted by D_{64}, D_{128}, etc.) are shown in figure 5.37.

5.4.4 Comparison of results under rotation of the initial conditions

One interesting question to ask is how the solution is changing if the initial conditions are rotated by some angle δ. Therefore from the original initial conditions

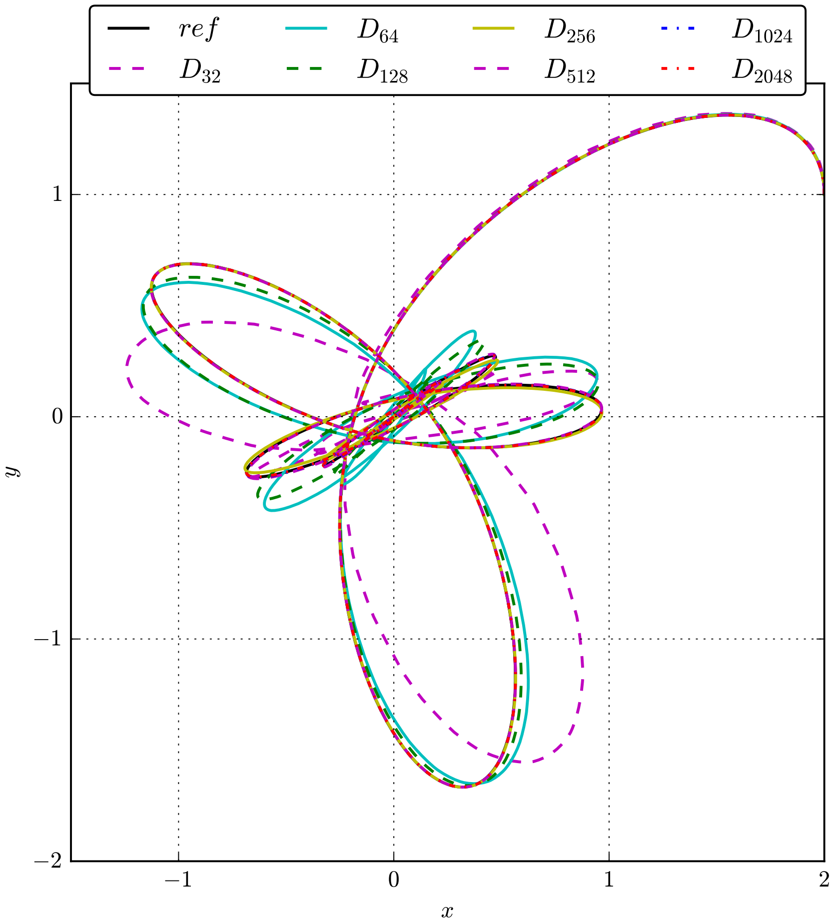


FIGURE 5.37: Trajectories for the consistency test, the trajectories for $N = 256, 512, 1024$ and 2048 practically overlap with the reference solution

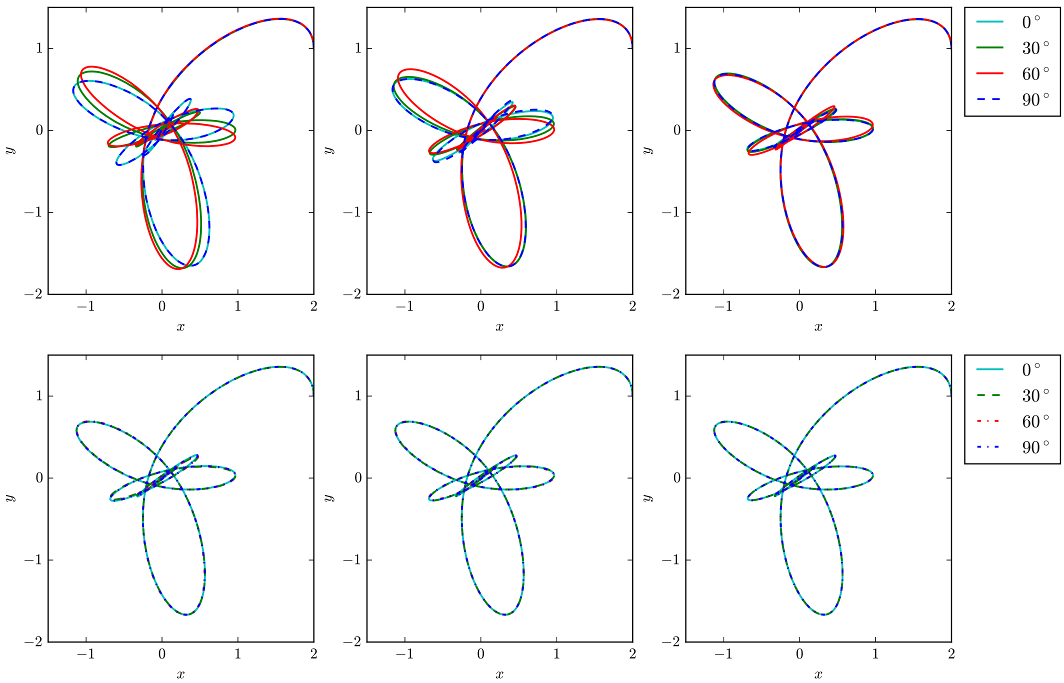


FIGURE 5.38: Backwards rotated trajectories for $N = 64, 128, 256$ (upper row) and $N = 512, 1024, 2048$ (lower row, each from left to right) and $\theta = 0°, 30°, 60°, 90°$

$(x_0, y_0, v_{x,0}, v_{y,0})$ new initial conditions $(\tilde{x}_0, \tilde{y}_0, \tilde{v}_{x,0}, \tilde{v}_{y,0})$ are computed, where:

$$(\tilde{x}_0, \tilde{y}_0) = (x_0 \cos\theta - y_0 \sin\theta, x_0 \sin\theta + y_0 \cos\theta).$$
$$(\tilde{v}_{x,0}, \tilde{v}_{y,0}) = (v_{x,0} \cos\theta - v_{y,0} \sin\theta, v_{x,0} \sin\theta + v_{y,0} \cos\theta).$$

As the original (analytical) surface is axis-symmetric around the z-axis, a rotation of the initial conditions around that point should not change the result but just rotate it around the same angle. Instead, the discretized surface is not symmetric and therefore more or less big changes (depending on the stage of refinement) are to expect. To test this, the system was solved for different angles $\delta = 0°, 30°, 60°, 90°$ for $N = 64, \ldots, 2048$, each with the finest time-resolution used in the previous test. After the simulation, the trajectory is rotated backwards with the same angle θ. The results are displayed in figures 5.38, showing in the upper row the solutions for $N = 64, 128, 256$ and in the lower one the solutions for $N = 512, 1024, 2048$, each from left to right. The strongest differences between the solutions is found for $\theta = 60°$ but also for $\theta = 30°$ differences are still significant. The solution for $\theta = 90°$ nearly coincides with the original one. As one would expect, the biggest differences are found in the solutions for the coarsest grid whereas they converge towards the 0°-solution with an increasing number of grid-points. From $N = 512$ on, differences are basically only seen in the error plots but not in the trajectories itself.

5.4.5 Frictionless circular motion in a horizontal plane:

The test-surface in this case is given by a paraboloid: $f(x,y) = x^2 + y^2$ with the domain restricted to $(x,y) \in [-4,4] \times [-4,4]$. In general, motion takes place in a horizontal plane (meaning horizontal to the z-plane) if a_z as well as v_z are *zero*. This means the following must be fulfilled:

$$\begin{aligned} a_z &= 0 \qquad \Leftrightarrow \qquad g(f_x^2 + f_y^2) = f_{xx}v_x^2 + 2f_{xy}v_xv_y + f_{yy}v_y^2 \\ v_{z,0} &= 0. \end{aligned}$$

In case of the above paraboloid we have:

$$\begin{aligned} &f_x = 2x, \qquad f_y = 2y, \\ &f_{xx} = 2, \qquad f_{yy} = 2, \qquad f_{xy} = 0, \end{aligned}$$

and therefore:

$$2g(x^2 + y^2) = v_x^2 + v_y^2$$

and with $(x^2 + y^2) = z$, $(v_x^2 + v_y^2 + v_z^2) = v^2$ and $v_z = 0$ the following condition for horizontal circular motion on a paraboloid is found:

$$v^2 = 2gz.$$

As the total energy must be conserved, it can directly be followed that also v^2 must be conserved. Therefore this test case is of special interest for two reasons:

- the analytical solution is known,
- the solution is periodic and therefore well suitable to test the accuracy of the numerical method.

The initial conditions for this test are:

$$\begin{aligned} &\text{a)} \quad (x_0, y_0) = (1,0) \quad &&\Rightarrow \quad z_0 = 1 \quad &&\Rightarrow \quad (v_{x,0}, v_{y,0}) = (0, \sqrt{2g}), \\ &\text{b)} \quad (x_0, y_0) = (\sqrt{2},0) \quad &&\Rightarrow \quad z_0 = 2 \quad &&\Rightarrow \quad (v_{x,0}, v_{y,0}) = (0, 2\sqrt{g}), \\ &\text{c)} \quad (x_0, y_0) = (2,0) \quad &&\Rightarrow \quad z_0 = 4 \quad &&\Rightarrow \quad (v_{x,0}, v_{y,0}) = (0, 2\sqrt{2g}), \end{aligned}$$

where g denotes the gravitational constant. All three simulations were run for a time $T = 1000s$. The circumference of the slice plane of $f(x,y)$ for a given height z is given by $L = 2\pi r$, where $r = \sqrt{z}$ from which it is followed that the time needed for a single revolution is given by $\tau = \sqrt{\frac{2}{g}}\pi$ resulting in a total number of 144 revolutions of the point. All three simulation were run on grids with $N = 64, 128, 256, 512$ grid points per coordinate-direction. Tables 5.3-5.6 show the results of the three test cases

z_0	mean	min	max	std
1.0078	1.0035	0.9737	1.0323	0.01074
2.0176	2.0168	2.0157	2.0177	0.00049
4.0070	3.9931	3.9931	4.0070	0.00965

TABLE 5.3: Results fot the test with $N = 64$ grid points.

z_0	mean	min	max	std
1.0019	1.0006	0.9839	1.0157	0.00542
2.0110	2.0141	2.0109	2.0174	0.00220
4.0017	4.0012	4.0007	4.0017	0.00025

TABLE 5.4: Results fot the test with $N = 128$ grid points.

for the four different grids from $N = 64$ to $N = 512$. Hereby z_0 denotes the initial height of the point on the triangulated surface. As the initial position of the point is given in terms of x and y, the z-coordinate slightly differs from the value on the analytical function. The initial velocities were therefore adjusted to the height on the discretized surface.

Furthermore, ”mean” denotes the mean value of the points height throughout the simulation whereas ”min” and ”max” denote the minimal and maximal height of the point. The standard-deviation of the height is given by ”std”.

z_0	mean	min	max	std
1.0005	1.0000	0.9996	1.0004	0.00026
2.0104	2.0111	2.0104	2.0119	0.00051
4.0004	3.9985	3.9969	4.0010	0.00028

TABLE 5.5: Results fot the test with $N = 256$ grid points.

z_0	mean	min	max	std
1.0001	0.9999	0.9996	1.0001	0.00016
2.0101	2.0102	2.0089	2.0114	0.00050
4.0001	3.9997	3.9993	4.0001	0.00029

TABLE 5.6: Results fot the test with $N = 512$ grid points.

5.5 Conclusions

In this chapter, the method presented in chapter 4 was extended to a semi-analytical solution for the equations of motion in 2D. While the gravity-dependent part of the equations was solved numerically using a fourth order explicit Runge-Kutta scheme, the curvature-dependent part can – as in the 1-dimensional case – be solved analytically. Therefore a smoothing of the trajectory in vicinity of the transition point between two triangles was used. Also in 2D, the method is very efficient, not least as no implicit solver is necessary for the time-integration and no additional approximations (e.g. concerning the partial derivatives) need to be done.

In the examples it could be shown that the method works well, leading to results fully comparable to the reference solution in terms of accuracy, even though the convergence rates drop to first order. The total energy is conserved in all frictionless tests. The high accuracy can particularly well be observed in the last test where even after nearly 144 revolutions on a horizontal circular path (inside a paraboloid) the total energy was conserved and errors in the point's height were little.

Chapter 6

Monte Carlo Simulation

Throughout the development-process of the model and the respective code, it quickly turned out that the problem is not very stable. Small changes, e.g. in the initial conditions, the sliding surface, or the model/program parameters lead to significant changes in the final results. In order to get more information on the behaviour of the problem, a Monte Carlo simulation for different realizations of the surface was done. The results are presented in the following.

6.1 Monte Carlo simulations: theory

Monte Carlo Simulations are used in a wide range of applications (especially physics and mathematics but also economics and biology) in order to obtain numerical results to problems that are too complicated or even impossible to be solved analytically (e.g simulation of systems with many degrees of freedom, such as fluids, disordered materials or strongly coupled solids). Often those problems contain one or more

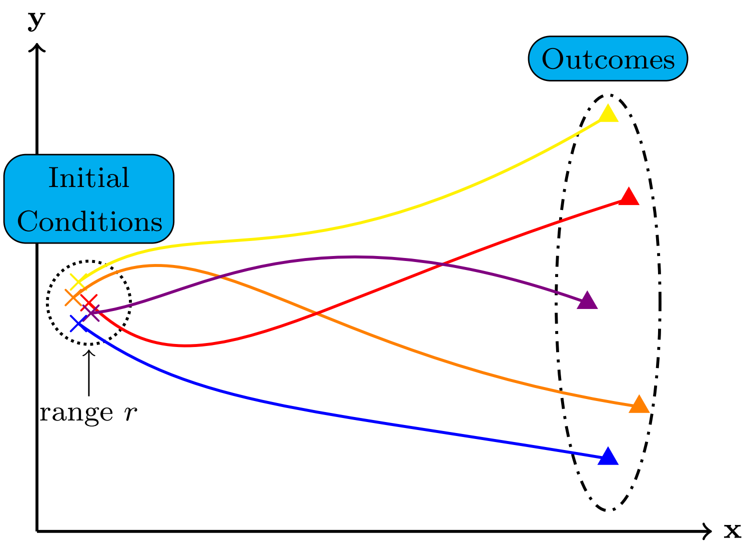


FIGURE 6.1: Sketch of the Monte Carlo approach for the problem of a point moving in the (x, y)-plane

parameters p that are not known exactly but are supposed to lie in a certain range r

of values. Modelling these parameters as random variables instead, and repeatedly performing a certain simulation finally gives a range of possible outcomes of the simulation and therefore more information than a single simulation could do. Of course, the results have to be post-processed in an appropriate way. In figure 6.1 the Monte Carlo approach is sketched for the problem of a point moving in the (x,y)-plane. The initial conditions are in the range r, indicated by a dotted black circle, where the parameter p is the initial position given by a tuple (x,y). The different realizations of p are indicated by coloured crosses. The corresponding lines denote the trajectories of a moving point while the endpoints are given by triangles in the respective colours. The dash-dotted ellipse indicates the range of those endpoints.

Generally, the following approach is followed when computing a Monte Carlo Simulation:

1. Define the domain of possible input values (the parameters p);
2. Choose a probability distribution function that best fits the parameters (e.g. a Gaussian distribution);
3. Compute a set of random numbers within the given domain;
4. Run the simulation, replacing one by one the parameter by a random number from the given set;
5. Evaluate and post-process the results using for example statistical methods (e.g. computation of mean values, variance and standard deviation, distribution of results, ...).

6.2 Monte Carlo applied to the landslide simulation

One big uncertainty in the simulation of landslides is the sliding surface. It is not only an approximation that in contrast to the real surface is approximated by planar triangles (whereas the real surface could theoretically even be smooth) but also the data points used to build the discretization are prone to measurement errors. Usually these data points are not distributed regularly over the whole domain, but they are given in the form of isolines. Then, different smoothing techniques – like the ones presented in section 3.1 – are used to obtain a surface representation based on a more regular grid, e.g. a Cartesian grid. Those new points though are basically chosen randomly such that local and global extrema of the real surface might be neglected. But, even small changes in the sliding surface may lead to significant changes in the final results, such that under these aspects, the outcome of a single simulation is of limited significance. In this context, a Monte Carlo simulation can provide valuable

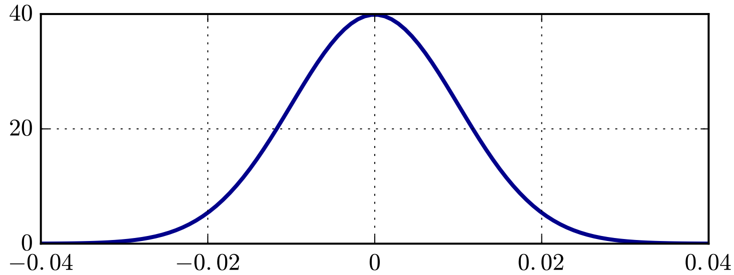


FIGURE 6.2: Normal distribution with $\mu = 0$ and $\sigma = 0.01$

information concerning the influence of the discretization but can also give a better and more detailed picture of the final result.

6.2.1 Randomization of the surface

In the following we will denote the data points by $\mathbf{P}_i = (x_i, y_i, z_i),\ \ i = 1, ..., N_p$, where the index i denotes the i-th data point and N_p is the total number of data points. There are two different kinds of surface randomization that will be considered in the following, namely:

1. Randomization of elevation; this can be seen as adding roughness, the parameter p that is replaced by a random variable is the z-component of the data points, i.e. $p = z$); the x/y-coordinates of the data points are not changed;

2. Randomization of x/y-coordinates; we now have two parameters p, namely p_1 and p_2 which are replaced by random variable $\Rightarrow p_1 = x$, $p_2 = y$); in this case also the elevation is recomputed. Generally the surface should not become more rough but just different.

In the first case the random variables are denoted by $\varepsilon_i,\ \ i = 1, ..., N_p$, where ε_i is in the range $\varepsilon_i = [\varepsilon_{i,\min}, \varepsilon_{i,\max}]$. Accordingly, in the second case the random variables for the x/y-coordinates are denoted by ξ and η with $\xi_i = [\xi_{i,\min}, \xi_{i,\max}]$ and $\eta_i = [\eta_{i,\min}, \eta_{i,\max}]$. In both cases the random variables are assumed to follow a Gaussian distribution:

$$\phi_G(x) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where μ and σ are the mean value and standard deviation of the distribution. If a random variable X follows a Gaussian distribution with mean μ and standard deviation σ, this is often denoted by

$$X \sim \mathcal{N}(\mu, \sigma^2).$$

In figure 6.2 a Gaussian distribution for $\mu = 0$ and $\sigma = 0.01$, a typical value used in the test-cases later, is shown. For a Gaussian distribution the values are distributed

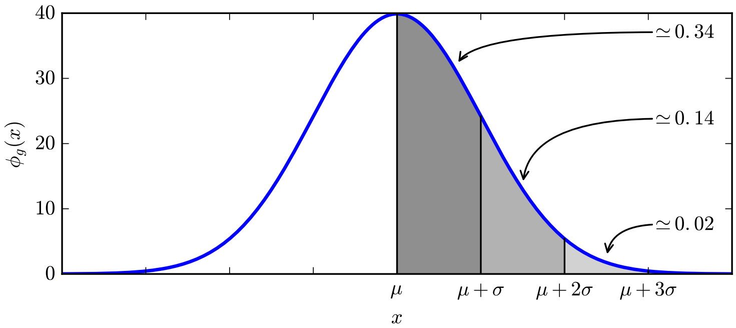


FIGURE 6.3: Normal distribution with $\mu = 0$ and $\sigma = 0.01$

as follows:

- 68.27% of the values are in the range $[\mu - \sigma, \mu + \sigma]$,
- 95.45% of the values are in the range $[\mu - 2\sigma, \mu + 2\sigma]$,
- 99.83% of the values are in the range $[\mu - 3\sigma, \mu + 3\sigma]$,

and is illustrated also in Fig. 6.3.

In order not to confuse the friction coefficient μ with the mean value of the Gaussian distribution, the parameters of the Gaussian will be denoted by μ_x, μ_y and μ_z and σ_x, σ_y and σ_z in the following.

The different realizations of the surface are then given by adding random numbers from the given range to the elevation value. Given a surface where N_P is the total number of points and letting N_S be the total number of simulations, the points of the randomized surfaces will be denoted by $\bar{\mathbf{P}}_i^k$ where the subscript $i = 1, \ldots, N_P$ denotes a certain point of the surface and the superscript $k = 1, \ldots, N_S$ the number of the current simulation. In case of adding roughness to the surface this would mean: for each simulation we compute as many random numbers as there are data points and (in a given order) add one random number to the z-coordinate of each data-point. This set of random numbers is denoted by $E^k = \{\varepsilon_1^k, \varepsilon_2^k, \ldots, \varepsilon_i^k, \ldots, \varepsilon_{N_P}^k\}$, $k = 1, \ldots, N_S$ where $E^k \sim \mathscr{N}(\mu_z = 0, \sigma_z^2)\ \forall k \in [1, N_S]$ with the lower index (i) identifying the data point and the upper (k) a particular simulation. Letting P_i be the i-th data point of the original surface and ε_i^k the i-th random number in the k-th simulation we have:

$$\bar{\mathbf{P}}_i^k = \mathbf{P}_i + \varepsilon_i^k.$$

In the same way one can obtain the randomization of the x/y-coordinates, just that in this case two sets of random numbers have to be computed.

6.2.2 Implementation

The whole code was implemented in Fortran 2003 which however does not contain an intrinsic function for the computation of normally but only uniformly distributed random numbers. There are different methods to compute normally distributed random numbers from uniformly distributed ones; here the ziggurat method was used Marsaglia and Tsang, 2007. While the code for a single surface simulation cannot be parallelized, this is very well possible for a Monte Carlo simulation. Hereby several realizations of a simulation can be run in parallel. One has to make sure though, that the created random numbers are thread safe, which means that the results of the parallel code-execution must be equal to the serial ones. Therefore a module was written that, computes thread safe normally distributed random numbers. The code was parallelized using Open-MP.

In order to avoid too distorted grid cells (triangles) and in particular to prevent grid cells from collapsing to a line, the following limitation was added in case if varied (x,y)-coordinates. As long as a random number (for the x-coordinates) is larger than $\triangle x/4$, it is divided by two. The same was applied for the according random numbers that are applied for the y-coordinates, where $\triangle x$ and $\triangle y$ are the diameter of a triangle in x- and y-direction.

6.2.3 Evaluation of numerical results

In the following the techniques used for the evaluation of the Monte Carlo simulations are presented. As reference solution we will consider the solution obtained on the unmodified piecewise planar surface, using the semi-analytical time-integration. It is certainly not meaningful to store and/or plot all trajectories of a single Monte Carlo simulation. Instead, special methods for the post-processing and visualization of the data are required. The three methods used throughout this work are presented in the following.

Statistical Parameters: We will denote by $x_i(t)$ the x-coordinate of the point during a certain simulation i at time t, the same applies to y, z, v_x, v_y and v_z. In order to evaluate the different solutions, mean values ($\bar{x}$, $\bar{y}$, $\bar{z}$, $\bar{v_x}$, $\bar{v_y}$, $\bar{v_z}$), standard deviation ($s_{x/y/z}$, $s_{v_x/v_y/v_z}$), and extrema were computed for the components of the points position and velocity for each time t. For the x-component of the position e.g. this can be computed as follows:

$$\bar{x}(t) = \frac{1}{N_s}\sum_{i=1}^{N_s} x_i(t),$$
$$s_x(t) = \sqrt{\mathrm{Var}(x(t))},$$

where the variance $\mathrm{Var}(x(t))$ is:

$$\mathrm{Var}(x(t)) = \mathrm{E}[(X - \mathrm{E}[X])^2] = \frac{1}{N_s}\sum_{i=1}^{N_s}(x_i(t) - \bar{x}(t))^2.$$

Where E is called the expected value. It is often more convenient though to compute the variance using an alternative formulation:

$$\mathrm{Var}(x(t)) = \mathrm{E}[X^2] - (\mathrm{E}[X])^2 = \frac{1}{N_s}\sum_{i=1}^{N_s}[x_i(t)]^2 - \left[\frac{1}{N_s}\sum_{i=1}^{N_s} x_i(t)\right]^2,$$

as it does not require storing the data in order to first compute the mean value $\bar{x}$.

Convex hulls: For the 2-dimensional problems the endpoints of all trajectories are plotted (endpoints are the final positions of the points, i.e. the position of a point at $t = T_{End}$). By "*reference-endpoint*" the endpoint on the unmodified surface is denoted. The areas in which the endpoints closest to the reference-endpoint are lying are given for different percentages with respect to the total number of runs, that means e.g. those 25% of the points that are closest to the reference-endpoint.

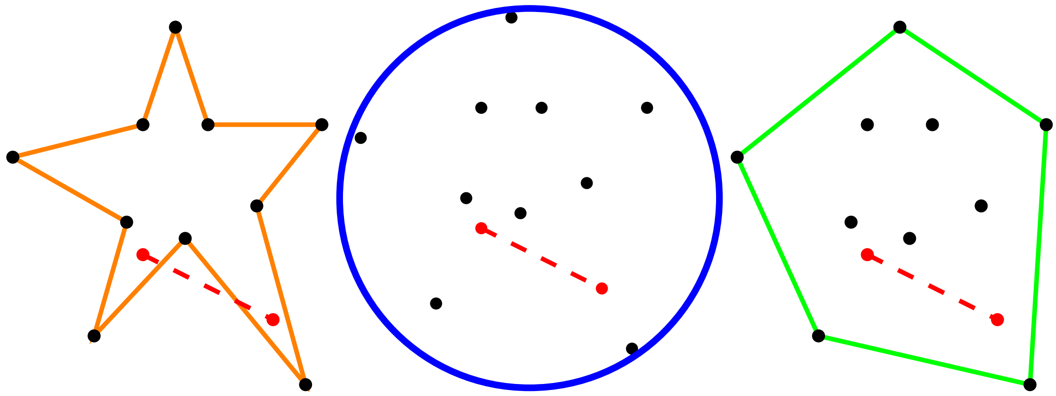
The areas itself were computed as follows:

1. Compute for all endpoints on randomized surfaces their euclidean distance to the endpoint on the unmodified surface;
2. Order those distances from smallest to largest;
3. Get the indices of those $k\%$ endpoints that are closest to the initial endpoint;
4. Compute the convex hull of those points and plot it.

The definition of a convex hull is based on convex sets. Simply spoken, a convex set is a region such that, for every pair of points within that region, every point on their (straight) connecting line is also within that region. (Note that this applies to Euclidean geometry only!) For a finite set of points, say X, the convex hull $\mathrm{Conv}(X)$ is the smallest convex set containing all those points. Or, in a more mathematical way:

$$\mathrm{Conv}(X) = \left\{ \sum_{i=1}^{|X|} \alpha_i x_i \,\middle|\, (\forall i : \alpha_i \geq 0) \wedge \sum_{i=1}^{|X|} \alpha_i = 1 \right\}.$$

An illustration of a non-convex and a convex set as well as the corresponding convex hull for a given set of points is given in figure 6.4. Fig. 6.4a shows a non-convex set for the given points, the line connecting the two red points is not fully contained in the set. Fig. 6.4b shows a convex set for the given points, but it is not the minimal



(A) Non-convex set (B) Convex set (C) Convex hull

FIGURE 6.4: Non-convex and convex set as well as convex hull for a given set of points.

set. Fig. 6.4c finally shows the convex hull for the given points. No smaller convex set can be found.

Evaluation of visited triangles: The two methods presented above give information that on one hand are easy to compare throughout different simulations (statistical parameters) and on the other hand give a very good picture on the distribution of the endpoints of a certain simulation. But neither do they provide information on the behaviour of the single trajectories of a simulation nor do they indicate the area of the surface covered by the trajectories. Applied to real landslides this would correlate to the area of the surface possibly influences by the moving mass.

This is treated using the last method which draws attention on the visited triangles. Throughout a certain simulation, the visiting times (=number of times the point passes through a certain triangle) for each triangle were counted and stored. Afterwards the triangles of the discretization are coloured according to their visiting times. In the following the matplotlib ”viridis” colormap is used, i.e. the lower the value, the darker the colour (here purple) and the higher the value, the brighter the colour (here yellow). For a better distinction, non-visited triangles are left white. In this way we know not only which triangles have been visited, but also have information on the frequency. Applied again to real landslides, these frequencies allow the computation of the probability that a certain part of the surface will – at any time – be covered by the sliding mass. In contrast, the convex hull of the endpoints contains this information only for the final time.

In a similar way the average velocity and/or acceleration could be computed for every triangle and be visualized using a vector field.

6.3 Numerical results

6.3.1 1D test case: Adding roughness to a polynomial

The first Monte Carlo simulation was performed for the 1-dimensional problem, with a test curve given by the 8-th order polynomial from section 4.4.3. Denoting by N_d the number of data points of the discretized surface, and by N_s the number of simulations, $N_d \times N_s$ random numbers have to be computed. A certain random number is denoted by $rn_{i,j},\ i = 1, \ldots N_s,\ j = 1, \ldots N_d$. The simulation was run for 100.000 different realizations of the surface with $\sigma = 0.01$.

Figure 6.5 shows the results plotted versus time t. Note that the two upper rows share a common axis (t). x, z, v_x and v_z denote the solutions on the unmodified surface. Note that $\max_i x(t)$, $\min_i x(t)$, and so on are the extrema of the solutions at each time t, they are not possible trajectories in itself! On average the motion decays faster than on the original test surface. Moreover one can observe that the standard deviations of the velocity, $\sigma_{v_{x/y}}$ are larger than $\sigma_{x/y}$ which in turn are larger than the standard deviation of the normal distribution used to vary the surface. The strongest values for σ are found when the point moves through the minimum of the surface, which obviously is also the point in which its velocity reaches its highest values.

One interesting aspect in running a Monte Carlo simulation with respect to the landslide-model is to find out more about the stability of the problem. In stable problems, small perturbations of the initial data lead to small changes in the final results. In contrast, if small perturbations lead to strong differences in the final results, the program in unstable. In the following, some test-cases for the landslide model and its results will be presented and discussed. In the first test-case, additional definitions and explanations are given that will also apply to the further test-cases presented in this report. They will not be repeated later.

6.3.2 Valley with decreasing slope

The first test surface is a valley with decreasing slope, given by the following function $f : D \to \mathbb{R}$ where $D = [-4, 44] \times [-4, 4]$ and

$$f(x,y) = e^{-0.3x} + 0.5y^2.$$

The surface is shown in figure 6.6. The surface was discretized with $n_x = 201$ and $n_y = 601$ grid points. The initial conditions of the point are:

$$\mathbf{x}_0 = (-3, -3),$$
$$\mathbf{v}_0 = (0, 0.5).$$

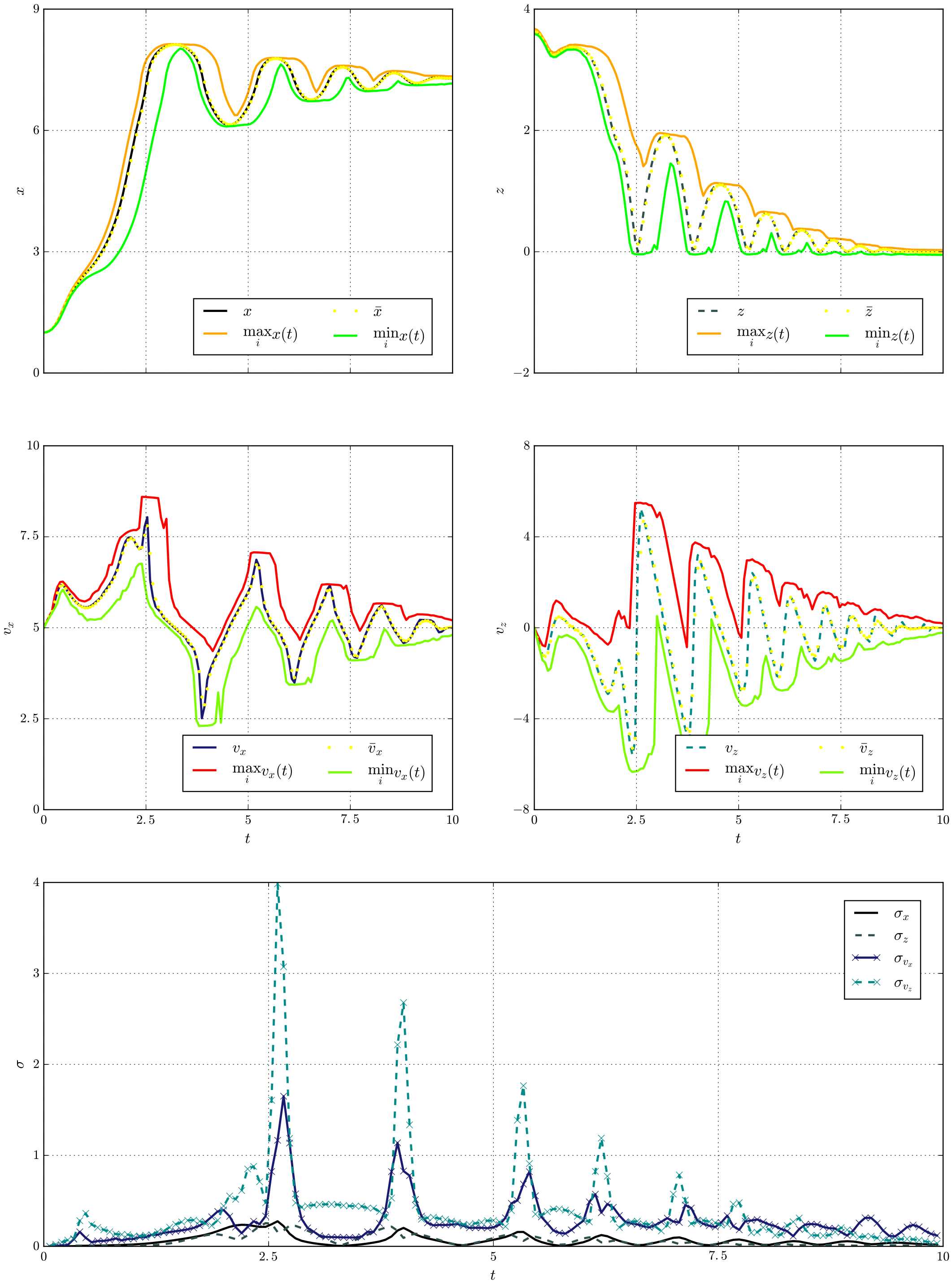


FIGURE 6.5: Result of the Monte Carlo simulation for an 8th order polynomial

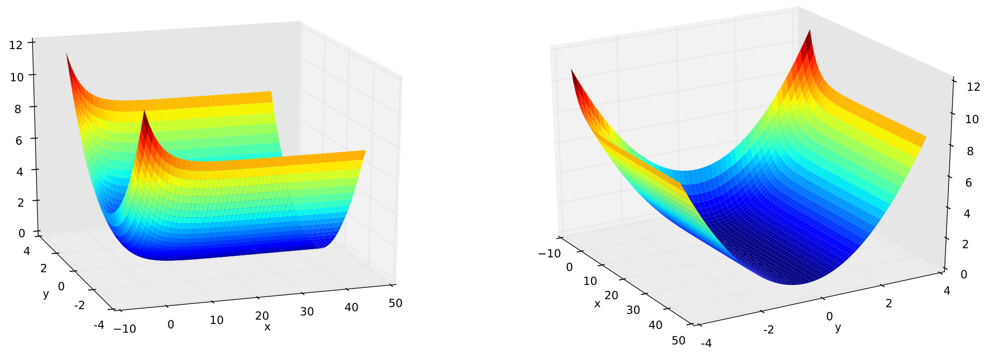


FIGURE 6.6: Valley with decreasing slope

Altogether five different (combined) tests are run for this surface, two of them adding roughness to the surface, and another three changing the (x,y)-coordinates of the grid.

Test 1: adding roughness In the first test, roughness was be added to the surface. For the randomization of the surface a normally distributed random variable with $\mu_z = 0$ and $\sigma_z = 0.0025$ was used. The friction coefficient is $\mu = 0.1$. Altogether 1000 runs were computed for the simulation.

The results of the MC simulation are depicted below. In figure 6.7, the trajectory (x vs. y) is plotted for the initial (not randomized) surface (=reference solution, blue curve) as well as for the mean value of the remaining runs (magenta). The visited triangles are coloured according to the colourbar on the right. In addition, the convex hull of all endpoints is given as black dashed polygon. Figure 6.8 shows a zoom on the endpoints and respective areas. The 25%-area is depicted in green, followed by 50, 75, 95 and 99% in blue, cyan, red and magenta. The black dashed polygon is again the convex hull for all endpoints. The endpoint of the reference solution is marked with a blue cross, the averaged endpoint is marked with a magenta cross.

It immediately stands out that even though the trajectory on the original surface and the mean trajectory of the Monte Carlo trajectories (those on the randomized surfaces) are very close to each other, the endpoints are quite widespread (= large standard deviation). While the differences in y-direction (perpendicular to the main direction of the motion) are comparably small (y-components are in the range of $\approx [-0.29, +0.5]$) those in x-direction are exactly the opposite (they are in the range of $\approx [29.5, 38.5]$) which in turn means that the points' velocities vary a lot from one simulation to another. On the other hand we find that the 50%-area is still comparably small. As can be expected, there is a concentration of endpoints around $y = 0$, that means along the minimum of the surface (in y-direction).

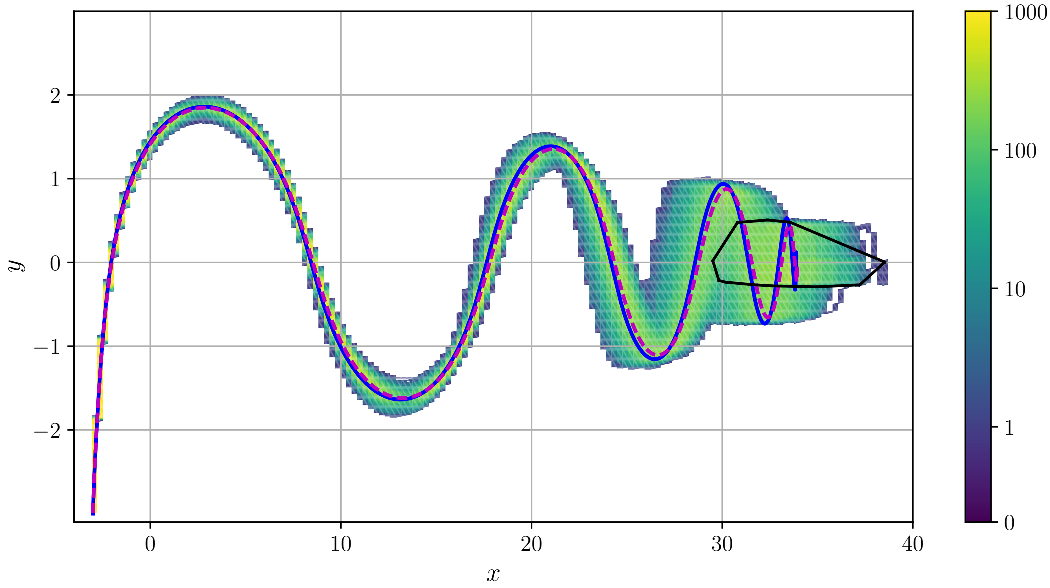


FIGURE 6.7: Reference (blue) and averaged (magenta) trajectories, visited triangles (according to the colourbar on the right) and convex hull of all endpoints (black) for the simulation run on the valley with decreasing slope (with roughness added)

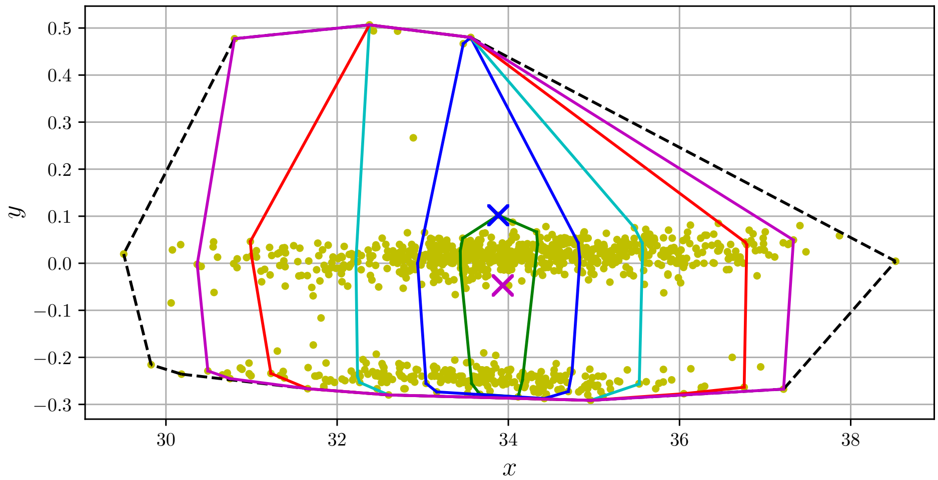


FIGURE 6.8: Endpoints on the unmodified surface and average of all endpoints (blue and magenta coloured crosses), and respective areas indicating vicinity to the original endpoint for the simulations run on the valley with decreasing slope (with added roughness)

The standard deviation of the position, velocity and acceleration are depicted in the following. Figure 6.9 shows the standard deviation for the x- and y-coordinates, marked by blue-dashed and red-solid lines, respectively. The thin black dashed and dotted lines represent the respective x- and y-coordinates, but are scaled such as to lie in the same range as $\sigma(x/y)$. $\sigma(x)$ is increasing monotonically with t, reaching a maximum value of approximately 1.45; the black dashed line shows that x increases likewise. $\sigma(y)$ instead undergoes an oscillation that reaches a maximum each time x reaches a minimum ($y = 0$), the point in which the velocity has a local maximum. The maximum $\sigma(y)$ is approximately 0.3 and therefore several times smaller than the maximum of $\sigma(x)$. This corresponds to the results found in figures 6.7 and 6.8.

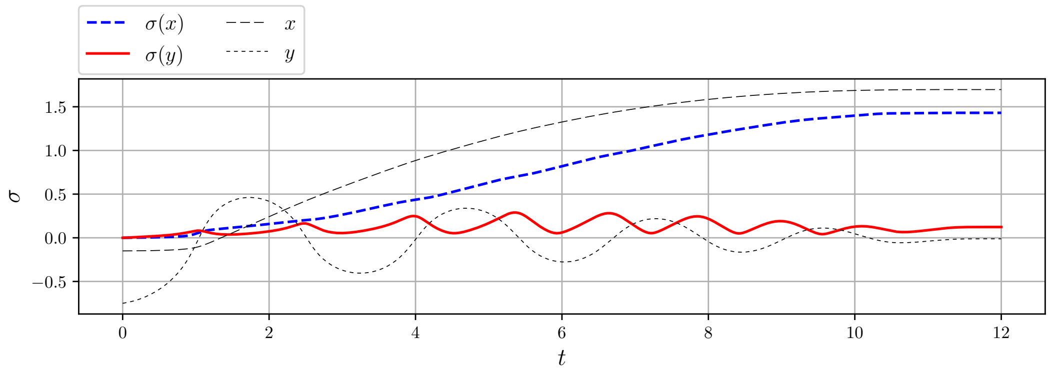


FIGURE 6.9: Standard deviation of the x- and y-coordinates for the simulation with $\sigma_z = 0.0025$ and $\mu = 0.1$, run on the valley with decreasing slope (with added roughness)

Figure 6.10 shows the standard deviation for the components of the velocity, v_x and v_y and – scaled to fit the range of $\sigma(v_x/v_y)$) – the velocities itself. In contrast to $\sigma(x/y)$, $\sigma(v_x)$ is generally larger than $\sigma(v_x)$. $\sigma(v_x)$ increases quickly at the beginning, as also v_x increases, and then drops as fast as well while v_x reached its maximum. $\sigma(v_x)$ raises slightly each time the point passes through $y = 0$ and then drops again before finally it becomes *zero* as also v_x becomes *zero*. The variations in $\sigma(v_y)$ in contrast are – besides the beginning – significantly stronger, and reaches a maximum as well, each time the point passes through $y = 0$. Obviously, it also becomes *zero* when v_y becomes *zero*.

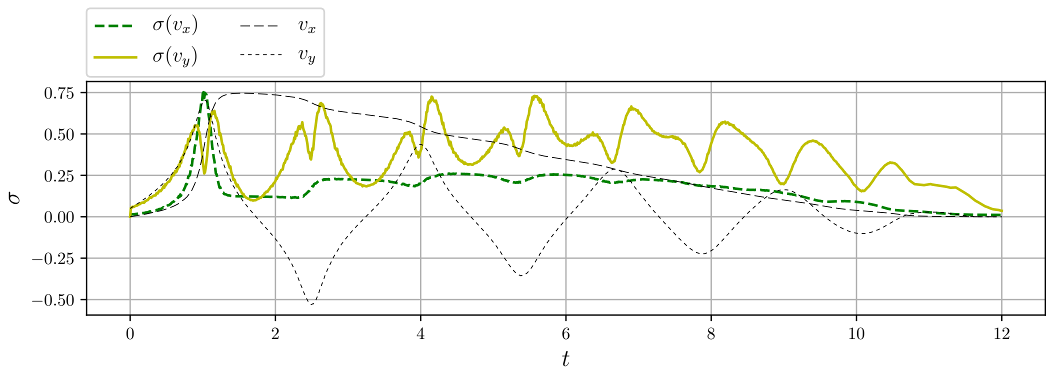


FIGURE 6.10: Standard deviation of the velocity components v_x and v_y for the simulation with $\sigma_z = 0.0025$ and $\mu = 0.1$, run on the valley with decreasing slope (with added roughness)

Finally, figure 6.11 shows the standard deviations $\sigma(a_x)$ and $\sigma(a_y)$. The trajectory in x-direction is monotonous, moreover, at the beginning of the trajectory the surface is much steeper as later on. This is reflected also in $\sigma(a_x)$ which reaches a maximum around $t = 1$ and then stays in the range of about $[0.1, 0.2]$. In y-direction instead, the trajectory shows an oscillating behaviour that is mirrored by $\sigma(a_y)$. $\sigma(a_y)$ also illustrates very well (in an indirect way) the roughness of the surface, showing small-scale oscillations along the whole curve. They are however best seen for large t as they are masked by the larger oscillations earlier on.

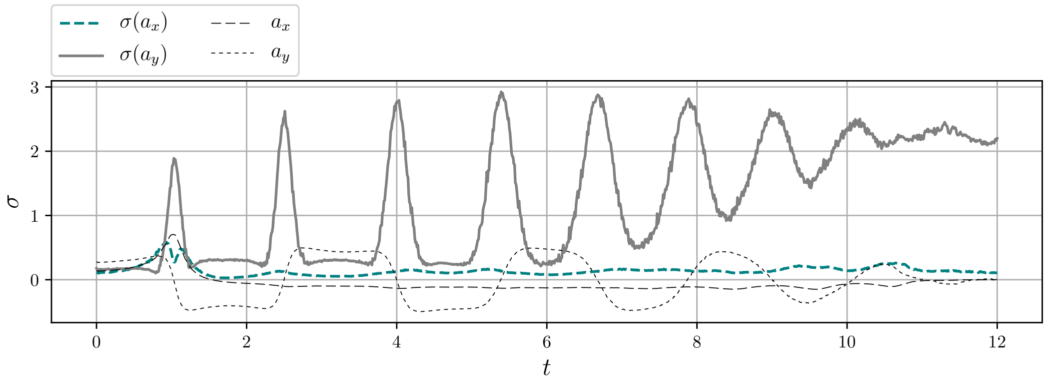


FIGURE 6.11: Standard deviation of the acceleration components a_x and a_y for the simulation with $\sigma_z = 0.0025$ and $\mu = 0.1$, run on the valley with decreasing slope (with added roughness)

Test 2: adding roughness + change of initial position In addition it is interesting to see, how the solution changes, if the initial position of the point is changed. This can of course be subject to a new Monte Carlo model, or even a combined problem, but one can also compute the solutions for a few, slightly varying, initial positions and get an idea on the influence on the final solution. This is done in the following, where the varied initial conditions were chosen as follows:

$$
\begin{aligned}
\mathbf{x}_0^1 &= (-3.05, -3.05),\\
\mathbf{x}_0^2 &= (-3.05, -2.95),\\
\mathbf{x}_0^3 &= (-2.95, -3.05),\\
\mathbf{x}_0^4 &= (-2.95, -2.95).
\end{aligned}
$$

The original starting point is denoted by $\mathbf{x}_0^0$; the initial velocity is kept the same, meaning that for all the points we have $\mathbf{v}_0 = (0, -0.5)$. Again, 1000 runs were computed for each simulation. All remaining parameters were not changed with respect to the previous test. In figure 6.12, the solutions (trajectories) for the 5 different initial positions are displayed (colours according to the legend), in addition the convex hull of the respective endpoints are shown (again in the according colours). The con-

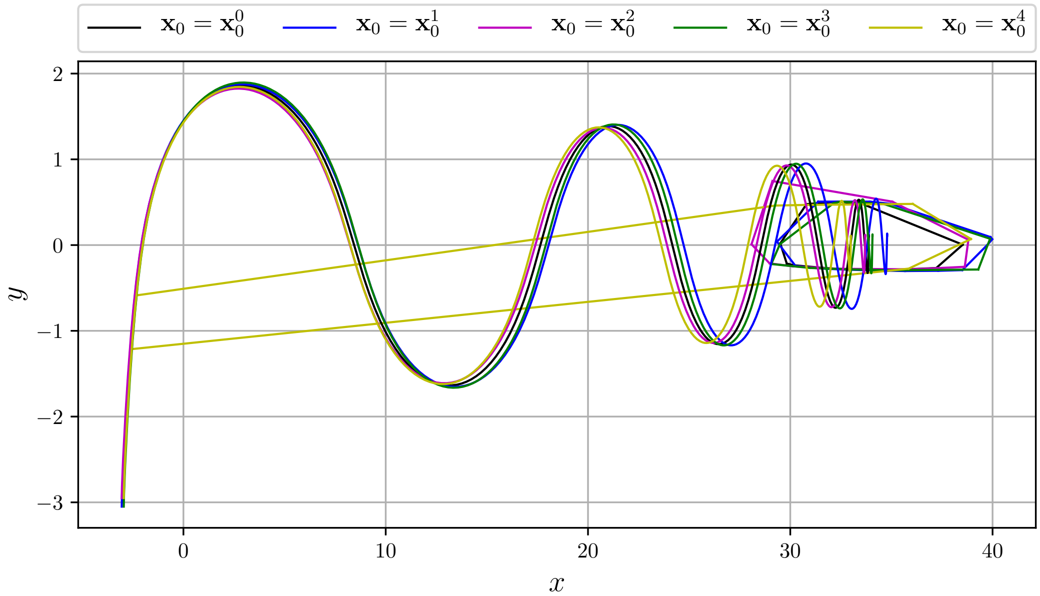


FIGURE 6.12: Reference solutions for the 5 different initial positions including convex hulls of the respective endpoints for the simulations run on the valley with decreasing slope and added roughness

vex hulls including the 50%-areas are shown in figure 6.13. As expected, the solution

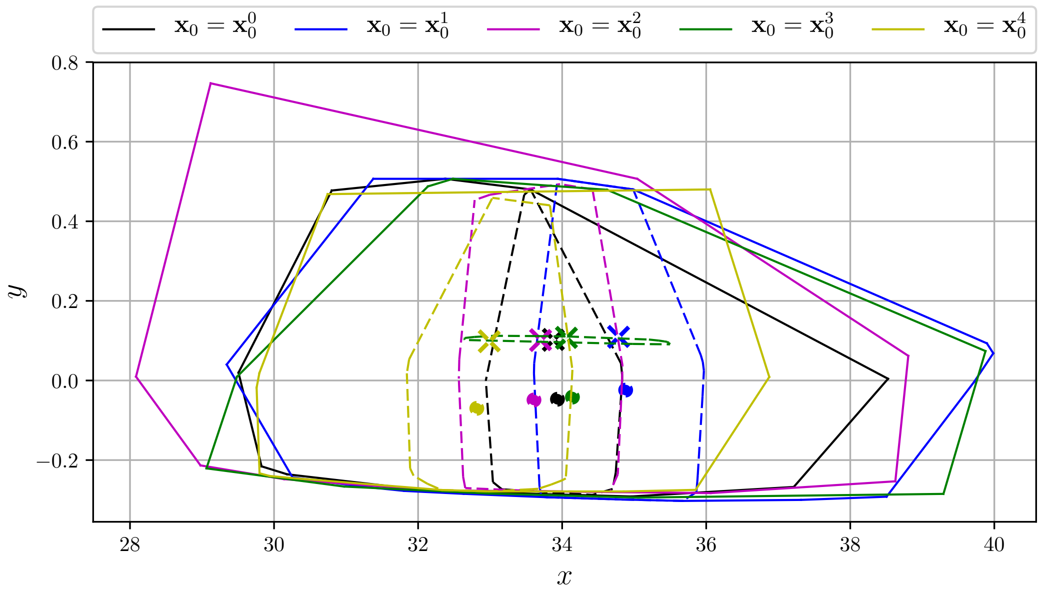


FIGURE 6.13: Convex hulls for the 5 different initial positions including the 50%-areas for the simulations run on the valley with decreasing slope

noticeably depends on the initial position of the point. We can see a clear shift along the x-axis in its final position, where a higher elevation at the initial time leads to a shift to the right (point moves longer) and lower initial elevation leads to a shift to the left (point stops earlier). An equivalent shift is found also in the convex hulls of the endpoints where we also see some outlier endpoints for the case $\mathbf{x}_0 = \mathbf{x}_0^4$. Generally however, the convex hulls are of very similar shape for all five initial positions. Note that all variations are only due to the change in the initial position, as the surfaces are the same throughout the simulations.

The standard deviations of position, velocity and acceleration are very similar to the previous case and thus are not shown here. This is not surprising, as the surfaces as well as the parameters are the same and only the initial position is slightly varied.

Test 3: changing (x,y)-coordinates For this test the standard deviation of the random variable was chosen $\sigma_{x/y} = 0.01$, the friction coefficient is $\mu = 0.1$. As in the previous two tests, 1000 runs were computed for the simulation. Note that the standard deviation used in this test is not comparable to the one in the previous tests, as they influence the surface in a very different way.

The results of the simulation are depicted below. In figure 6.14, the trajectory (x vs. y) is plotted for the initial (not randomized) surface (=reference solution, blue curve) as well as for the mean value of the remaining runs (magenta). Again, the visited triangles are coloured according to the colourbar on the right. In addition, the convex hull of all endpoints is given as black dashed polygon. Figure 6.15 shows a zoom on the endpoints and respective areas. The 25%-area is depicted in green, followed by 50, 75, 95 and 99% in blue, cyan, red and magenta. The black dashed polygon is again the convex hull for all endpoints. The endpoint of the reference solution is marked with a blue cross, the averaged endpoint is marked with a magenta cross.

While in x-direction the endpoints are distributed similarly to the first test, (their range is $\approx [26.5, 38.5]$), they are lying in a much smaller, curved band in y-direction. The values in y-direction are in the range of $\approx [0.027, 0.11]$), where the width of the band in which they lie is about 0.01.

The standard deviations for position, velocity and acceleration, are depicted in the following three figures, 6.16, 6.17 and 6.18. Like in the first test case, position, velocity and acceleration are plotted qualitatively to allow an easier comparison to the data. It is easily seen that the qualitative behaviour of the standard deviations is equivalent to test 1. However, as there is no artificial roughness in the surface, there are also no additional small-scale oscillations in $\sigma(a_y)$. Moreover, in contrast to test 1 again, $\sigma(a_y)$ decreases significantly for large t which mirrors the fact that all endpoints are located in a small band while on the more rough surface, the endpoints are spread wider in y-direction.

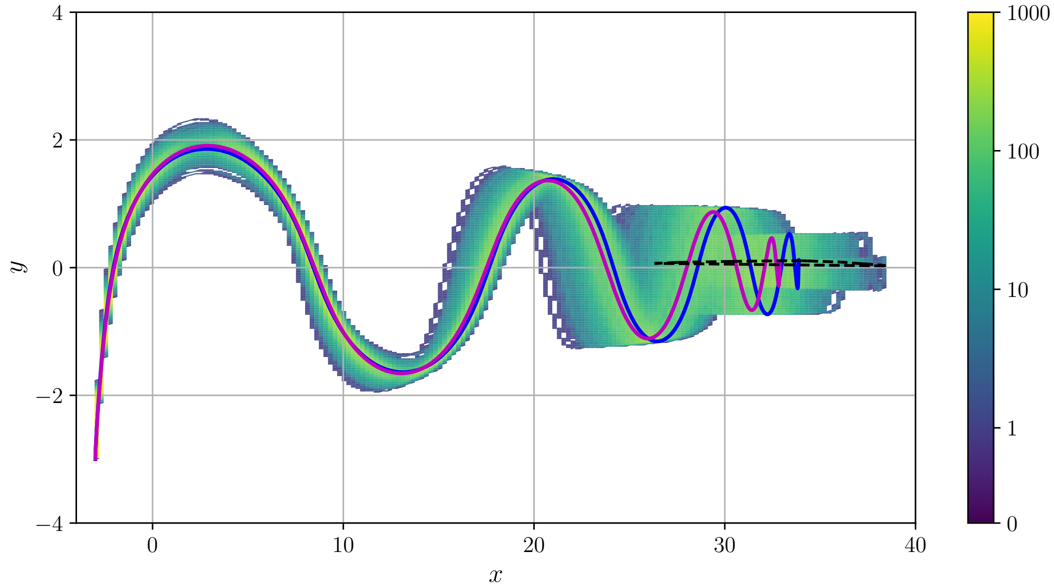


FIGURE 6.14: Reference (blue) and averaged (magenta) trajectories, visited triangles (according to the colourbar on the right) and convex hull of all endpoints (black) for the simulation run on the valley with decreasing slope (under change of (x,y)-coordinates)

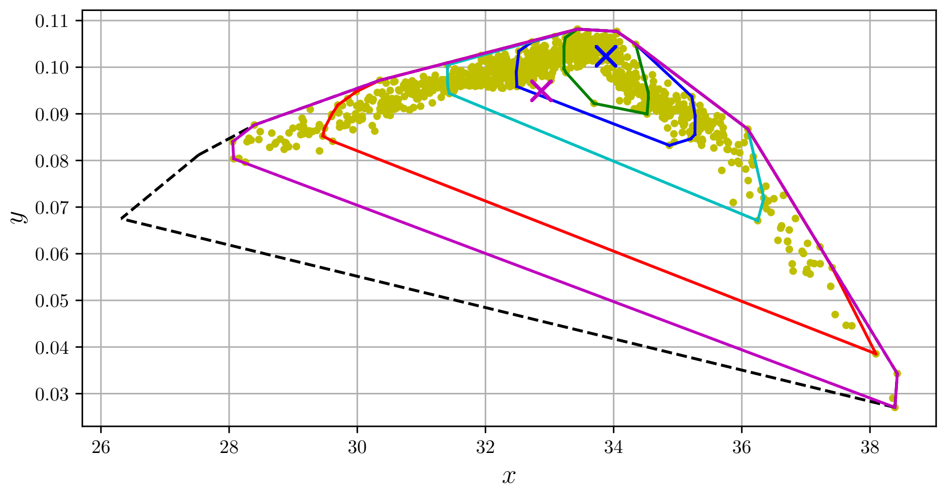


FIGURE 6.15: Endpoint on the unmodified surface and average of all endpoints (blue and magenta coloured crosses), endpoints on the randomized surfaces (yellow dots) and respective areas indicating vicinity to the original endpoint for the simulations run on the valley with decreasing slope (under change of (x,y)-coordinates)

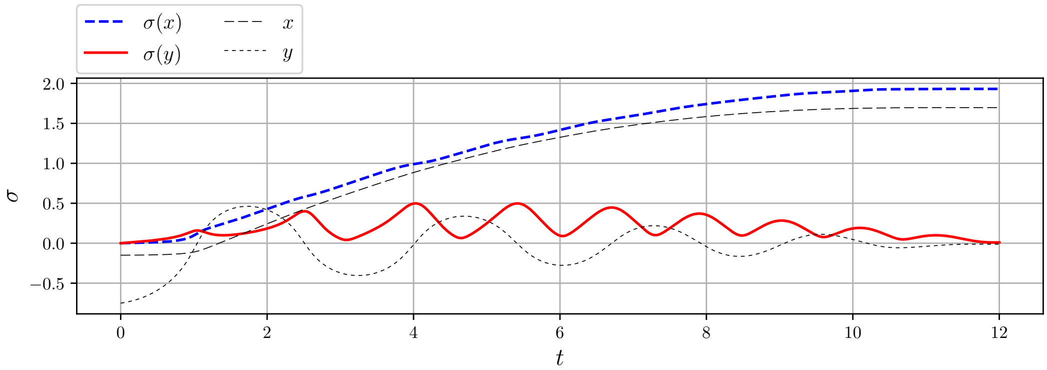


FIGURE 6.16: Standard deviation of the x- and y-coordinates for the simulation with $\sigma_{x/y} = 0.01$ and $\mu = 0.1$, run on the valley with decreasing slope (under variation of (x, y)-coord.)

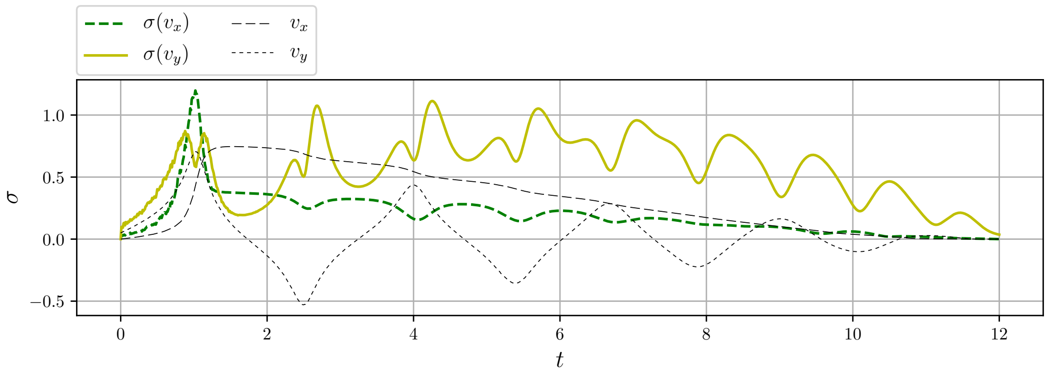


FIGURE 6.17: Standard deviation of v_x and v_y for the simulation with $\sigma_{x/y} = 0.01$ and $\mu = 0.1$, run on the valley with decreasing slope (under variation of (x, y)-coord.)

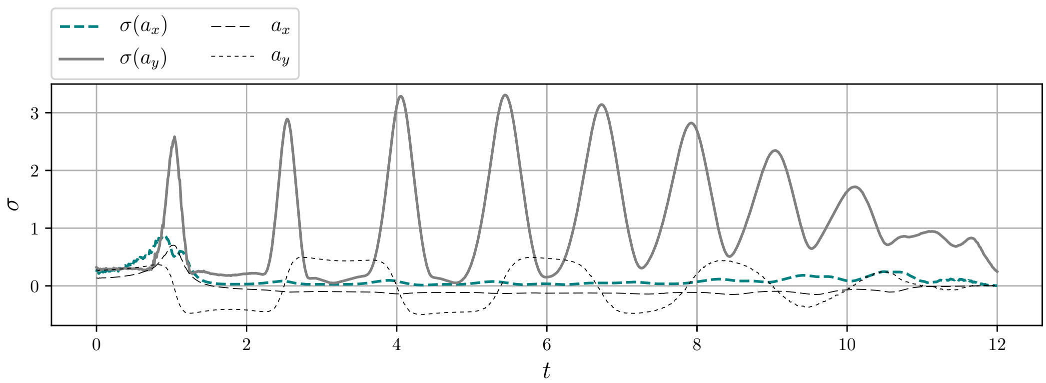


FIGURE 6.18: Standard deviation of a_x and a_y for the simulation with $\sigma_z = 0.01$ and $\mu = 0.1$, run on the valley with decreasing slope (under variation of (x, y)-coord.)

Test 4: changing (x,y)-coordinates + change of initial position For the variation of initial positions, the coordinates are again chosen as follows:

$$\begin{aligned}\mathbf{x}_0^1 &= (-3.05, -3.05),\\ \mathbf{x}_0^2 &= (-3.05, -2.95),\\ \mathbf{x}_0^3 &= (-2.95, -3.05),\\ \mathbf{x}_0^4 &= (-2.95, -2.95).\end{aligned}$$

The original starting point is denoted by $\mathbf{x}_0^0$; the initial velocity is kept the same meaning for all the points we have $\mathbf{v}_0 = (0, -0.5)$. All remaining parameters were not changed with respect to test 3. The five averaged trajectories are shown in figure 6.19 together with the convex hulls of all endpoints. A picture similar to test 2 is found. There is a noticeable dependency of the results on the initial conditions; while at the beginning the trajectories practically overlap, the differences between them increase significantly each time the point is in vicinity of a maximum turning point of the trajectory. In between, the differences increase only slightly.

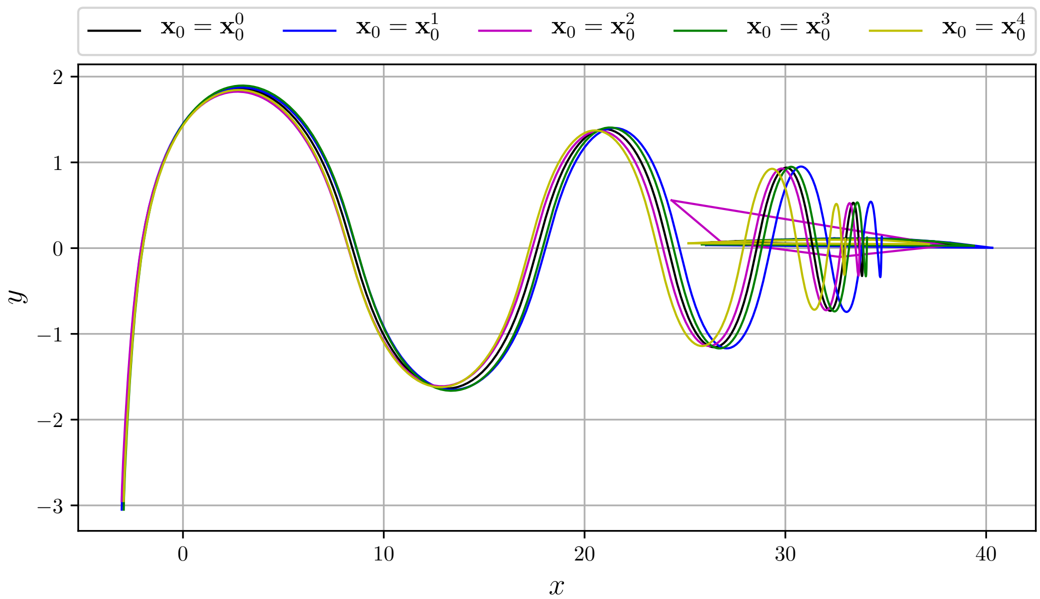


FIGURE 6.19: Reference solutions for the 5 different initial positions including convex hulls of the respective endpoints for the simulations run on the valley with decreasing slope and added roughness

The shapes of the convex hulls are comparable again. Only for $\mathbf{x}_0 = \mathbf{x}_0^2$ there are some outlier endpoints as can be seen in figure 6.20 (top picture). The bottom picture depicts the same situation but with the 95%-are for the simulation with initial condition $\mathbf{x}_0 = \mathbf{x}_0^2$ (magenta) in order to visualize better also the remaining convex

TABLE 6.1: Initial height of the five initial positions and differences in height of the varied initial positions with the height of the original initial position.

$\mathbf{x}_0$	z_0	$diff$
$\mathbf{x}_0^0$	6.96047	–
$\mathbf{x}_0^1$	7.14834	-0.18787
$\mathbf{x}_0^2$	6.84834	$+0.11213$
$\mathbf{x}_0^3$	7.07575	-0.11528
$\mathbf{x}_0^4$	6.77575	$+0.18472$

hulls. The crosses depict the endpoints of the reference solutions, the dots represent the averaged endpoints. For both, the reference and the averaged solution, the trajectories for the initial positions $\mathbf{x}_0^2$ and $\mathbf{x}_0^3$ lie much closer to the original initial position $\mathbf{x}_0^0$ then the remaining two. This is explained by the initial height and therefore the potential energy of the points, the former is given in table 6.1 together with the difference between the heights of $\mathbf{x}_0^0$ and the other initial positions.

The point with initial position $\mathbf{x}_0^1$ has greater initial height than the original point $\mathbf{x}_0^0$ and stops significantly later (in terms of larger values of x). The point with initial position $\mathbf{x}_0^4$ has an initial height that is lower with the same ratio as z_0^1 is higher with respect to z_0^0. The point stops significantly earlier with respect to the original point, but the distance to the original point at final time is about the same as for the point with initial position $\mathbf{x}_0^1$. The same behaviour is true for the two remaining points, just that the initial heights differ less strong from the original height and also the final positions are much closer to the final position of the point with the original (unmodified) initial position.

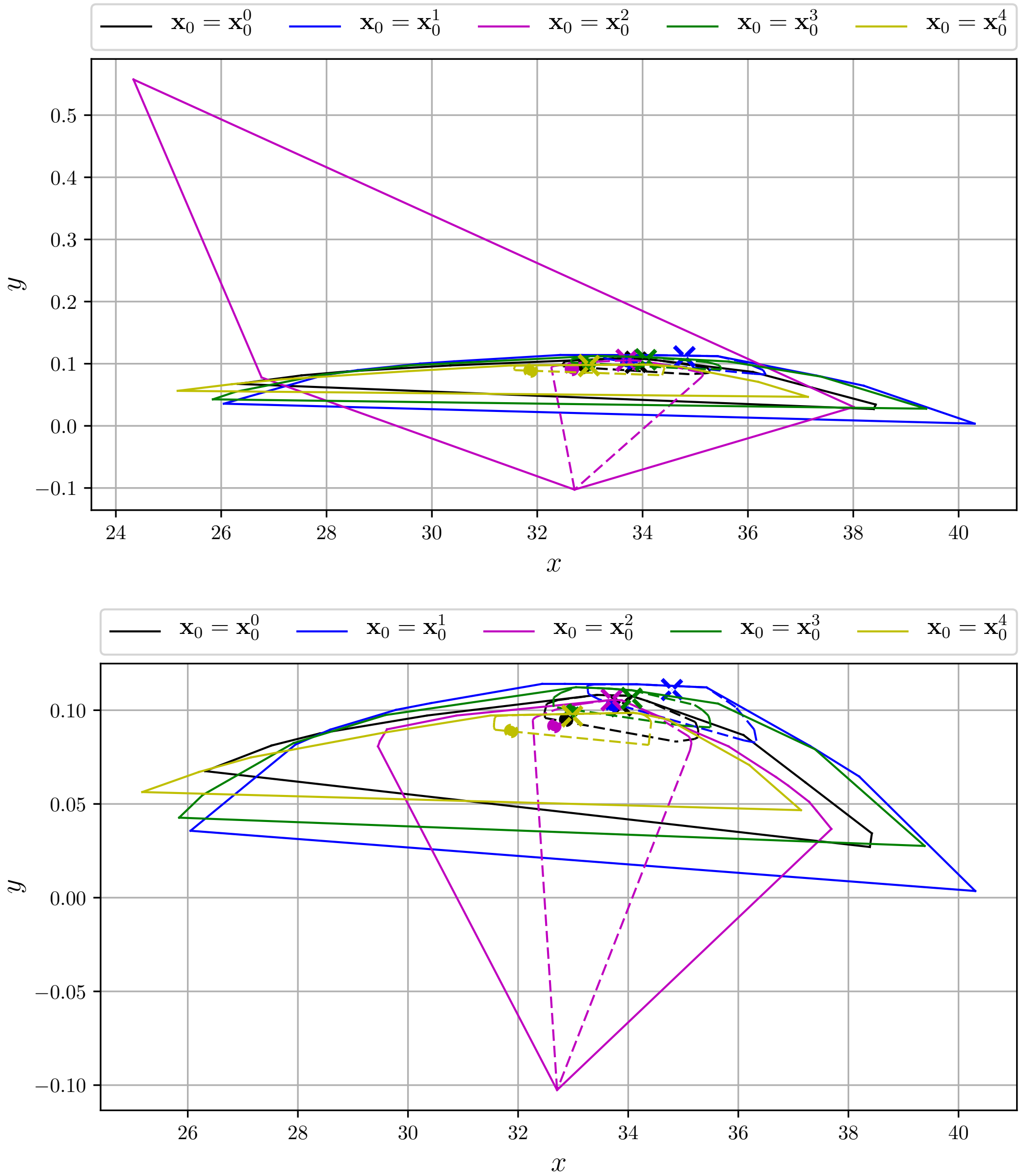


FIGURE 6.20: Convex hulls for the 5 different initial positions including the 50%-areas for the simulations run on the valley with decreasing slope (top picture). The bottom picture depicts the same situation but with the 95%-are for the simulation with initial condition $\mathbf{x}_0 = \mathbf{x}_0^2$ in order to visualize better also the remaining convex hulls.

Test 5: Changing (x, y)-coordinates + variation of standard deviation and friction The last test is dedicated to the variation of friction and standard deviation. Also here, the (x, y)-coordinates are changed and the z-coordinate is corrected according to the underlying function. The initial conditions are not changed, i.e.

$$\mathbf{x}_0 = (-3, -3),$$
$$\mathbf{v}_0 = (0, 0.5).$$

For the friction coefficient $\mu = 0.15$ and $\mu = 0.2$ are used in addition to $\mu = 0.1$, for the standard deviation $\sigma = 0.02$ is used in addition to $\sigma = 0.01$. Hence, in addition to the already known test 3 for which $\mu = 0.1$ and $\sigma = 0.01$, five different simulations are presented in the following. In this series of test, test 3 will be also be called test 5.1 in order to avoid unnecessary repetition of information.

Test 5.2: $\mu = 0.1$, $\sigma_{x/y} = 0.02$ Figure 6.21 shows the reference (blue) and averaged (magenta) trajectory as well as the visited triangles and the convex hull of the endpoints for $\mu = 0.1$ and $\sigma_{x/y} = 0.02$. Compared to the test with $\sigma_{x/y} = 0.01$ a noticeable larger variation of the endpoints is indicated by the expanded convex hull on one side and the number of visited triangles on the other hand. The latter is significantly higher as for $\sigma_{x/y} = 0.01$, furthermore, the trajectories start to diverge considerably early.

While for $\sigma_{x/y} = 0.01$ the x-components of the endpoints are in the approximate range $[26.5, 38.5]$, this range is extended to approximately $[16.5, 47.5]$ for the larger $\sigma_{x/y}$ (compare figure 6.22, bottom). Figure 6.22 (bottom) reveals that the qualitative distribution of the endpoints resembles the one for $\sigma_{x/y} = 0.01$, describing a bow that is open towards negative y, while both cases, the point stops while moving in positive y-direction. Moreover, also the relative sizes of the $k\%$ areas are very similar to the ones for the smaller σ.

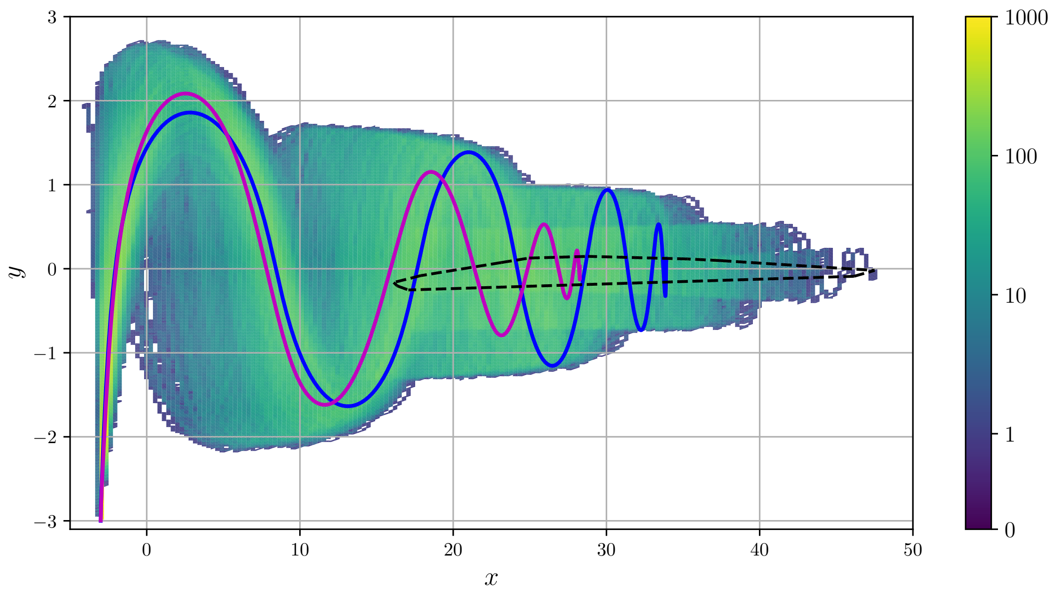


FIGURE 6.21: Reference and averaged trajectories (blue and magenta), convex hull for all endpoints and visited triangles for $\mu = 0.1$, $\sigma_{x/y} = 0.02$

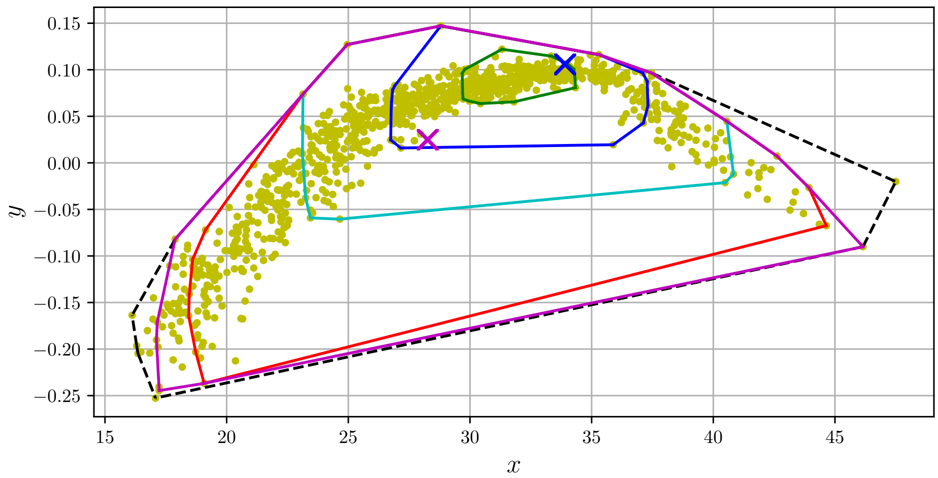


FIGURE 6.22: Zoom on the $k\%$ areas for $\mu = 0.1$, $\sigma_{x/y} = 0.02$

Test 5.3: $\mu = 0.15$, $\sigma = 0.01$ The results for $\mu = 0.15$ and $\sigma = 0.01$ are depicted below. Figure 6.23 shows the reference and averaged trajectories, the convex hull of the endpoints and the visited triangles. While, due to the increased friction coefficient, the motion stops earlier (with respect to x and t), the variance within the trajectories is comparable to the test with the same standard deviation $\sigma_{x/y}$ but $\mu = 0.1$.

The motion stops while the point is moving in negative y-direction, in the reference solution as well as in the averaged solution. The convex hull of all endpoints and the according $k\%$ areas are depicted in figure 6.24. The endpoints are distributed more loosely when compared to the test with $\sigma_{x/y} = 0.01$, but again form a bow that now is open in positive y-direction. The ratios between the $k\%$areas though are similar, too.

The increased friction coefficient strongly influence the final position, though. For $\mu = 0.1$ the final position of the reference solution was $(x,y) = (33.88, 0.10)$ while for $\mu = 0.15$ the final position was $(x,y) = (20.01, -0.14)$. In comparison, the averaged final positions are $(x,y) = (32.87, 0.09)$ and $(x,y) = (19.40, -0.14)$, respectively.

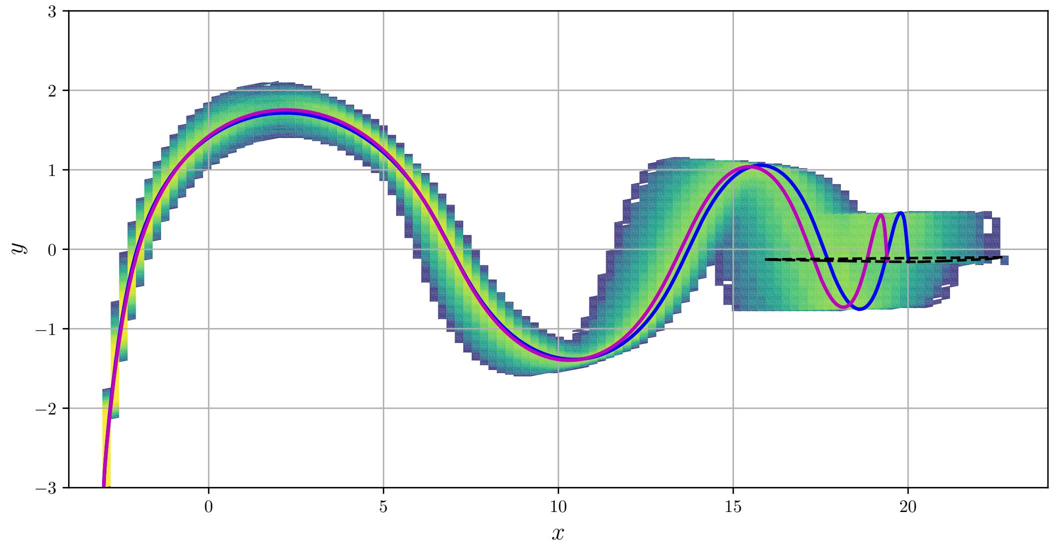


FIGURE 6.23: Reference and averaged trajectories (blue and magenta), convex hull for all endpoints and visited triangles for $\mu = 0.15$, $\sigma_{x/y} = 0.01$

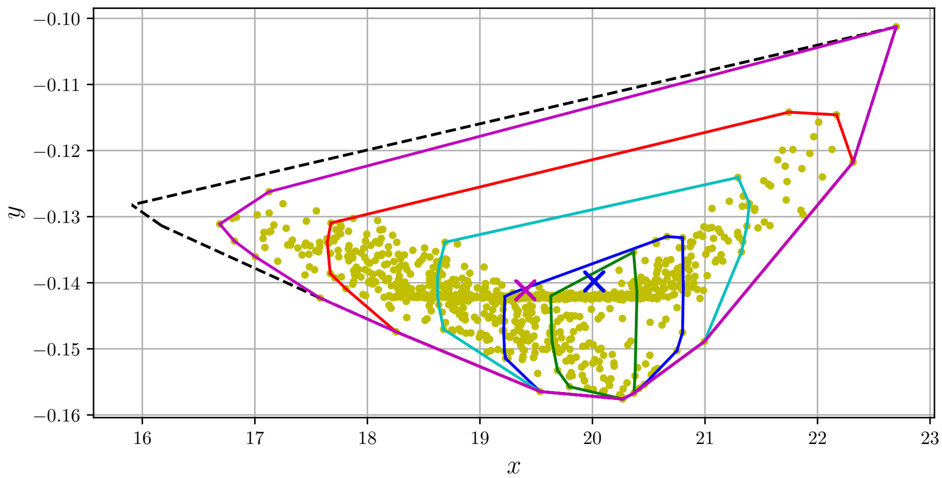


FIGURE 6.24: Zoom on the $k\%$ areas for $\mu = 0.15$, $\sigma_{x/y} = 0.01$

Test 5.4: $\mu = 0.15$, $\sigma = 0.02$ For $\mu = 0.15$ and $\sigma = 0.02$ the trajectories show a strong variance that resembles the variance found in the case $\mu = 0.1$, $\sigma = 0.02$ (compare figure 6.25). Likewise, the averaged trajectory ends earlier than the reference trajectory. The convex hull of the endpoints spreads over the whole domain, caused by an outlier that left the domain at $x = -4$.

The 99% area is significantly smaller though, and covers an approximate range of $[11, 22]$ in terms of x as can be seen in figure 6.26. The bow shape of the endpoints is much less distinct than in the previous cases, where the majority of points is lying in a line rather than along a bow. Therefore also the shapes and ratios of the $k\%$ areas changed. Nonetheless, the 25% area is still comparably small with respect to the remaining areas.

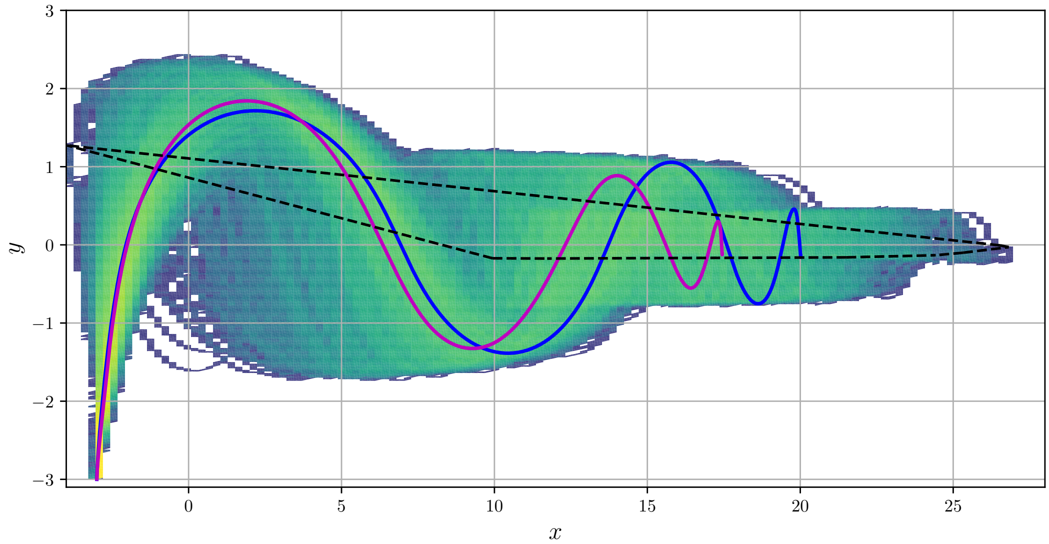


FIGURE 6.25: Reference and averaged trajectories (blue and magenta), convex hull for all endpoints and visited triangles for $\mu = 0.15$, $\sigma_{x/y} = 0.02$

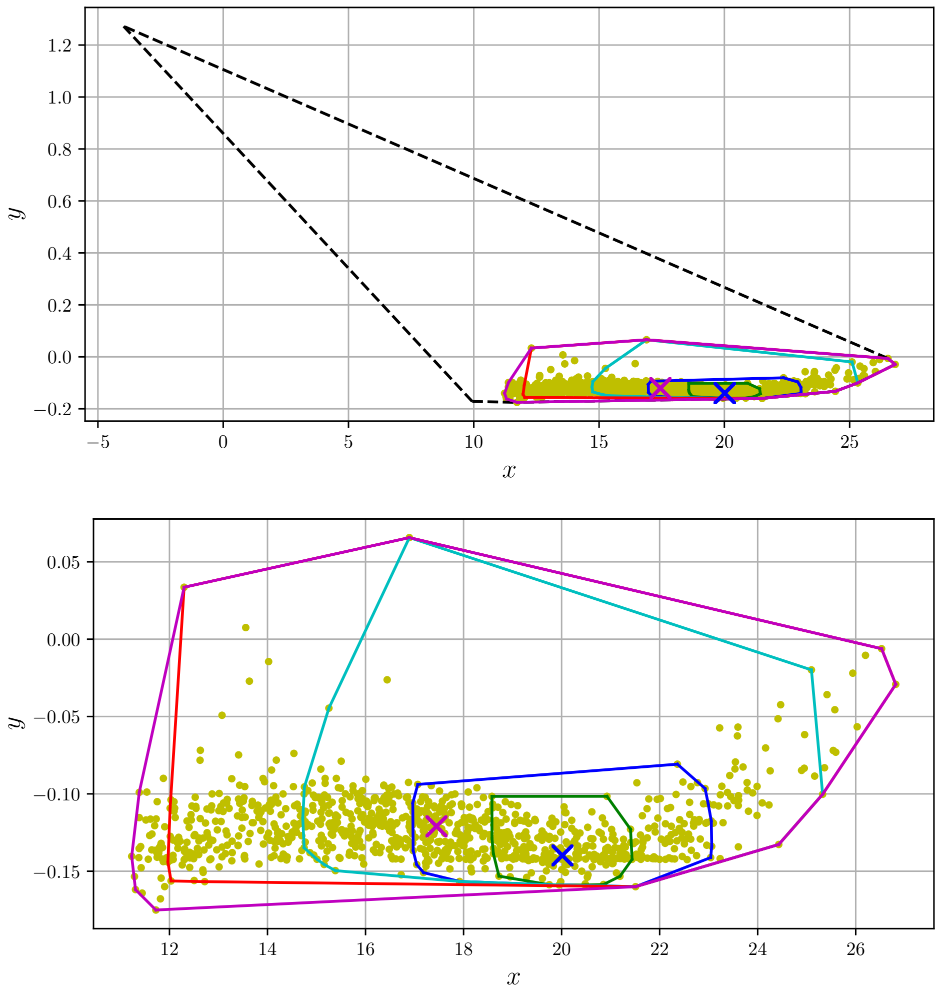


FIGURE 6.26: Zoom on the k% areas for $\mu = 0.15$, $\sigma_{x/y} = 0.02$ (top), without full convex hull (bottom)

Test 5.5: $\mu = 0.2$, $\sigma = 0.01$ The reference and averaged trajectories in figure 6.27 as well as the variance in the visited triangles and the elongation of the convex hull resemble those for $\mu = 0.1$ and $\mu = 0.15$ under the same value of $\sigma_{x/y}$, just that the motion stops earlier with respect to t and x.

The endpoints barely form a bow any more, only a slight opening in positive y-direction is suggested by the shape of the k% areas.

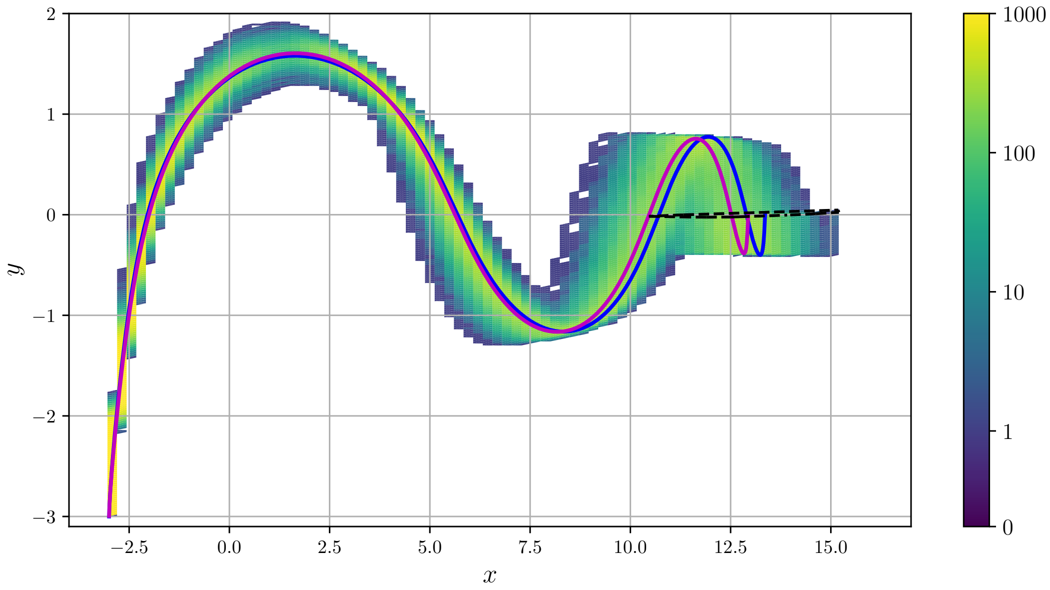


FIGURE 6.27: Reference and averaged trajectories (blue and magenta), convex hull for all endpoints and visited triangles for $\mu = 0.2$, $\sigma_{x/y} = 0.01$

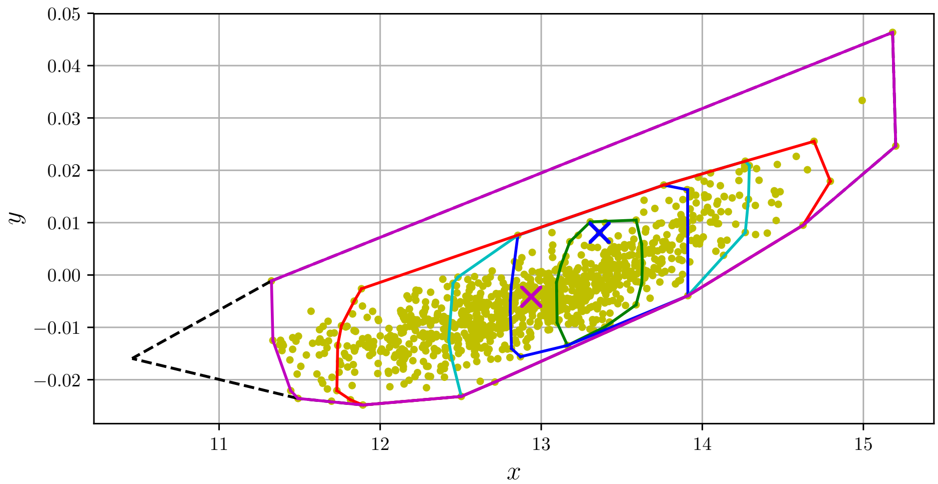


FIGURE 6.28: Zoom on the $k\%$ areas for $\mu = 0.2$, $\sigma_{x/y} = 0.01$

Test 5.6: $\mu = 0.2$, $\sigma = 0.02$ Like for the previous two tests with $\sigma_{x/y} = 0.02$, the variance between the trajectories is significantly larger than for $\sigma_{x/y} = 0.01$. The convex hull spreads over almost the whole x-range as can be seen in figure 6.29. The reference and averaged point stops while moving in positive y-direction.

The endpoints in the last test clearly form a bow again, having an opening in positive y-direction is suggested by the shape of the $k\%$ areas.

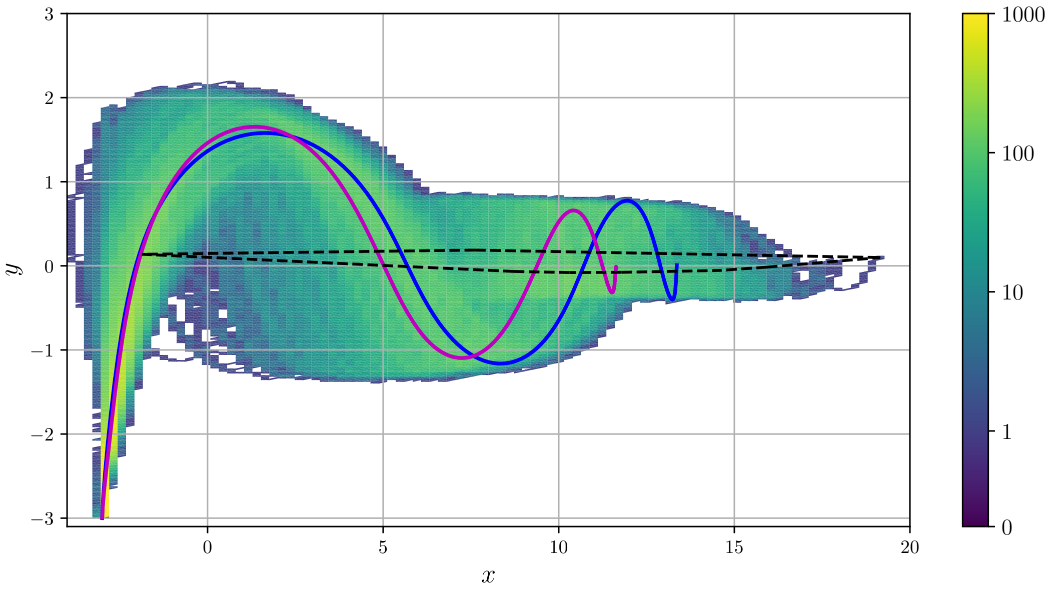


FIGURE 6.29: Reference and averaged trajectories (blue and magenta), convex hull for all endpoints and visited triangles for $\mu = 0.2$, $\sigma_{x/y} = 0.02$

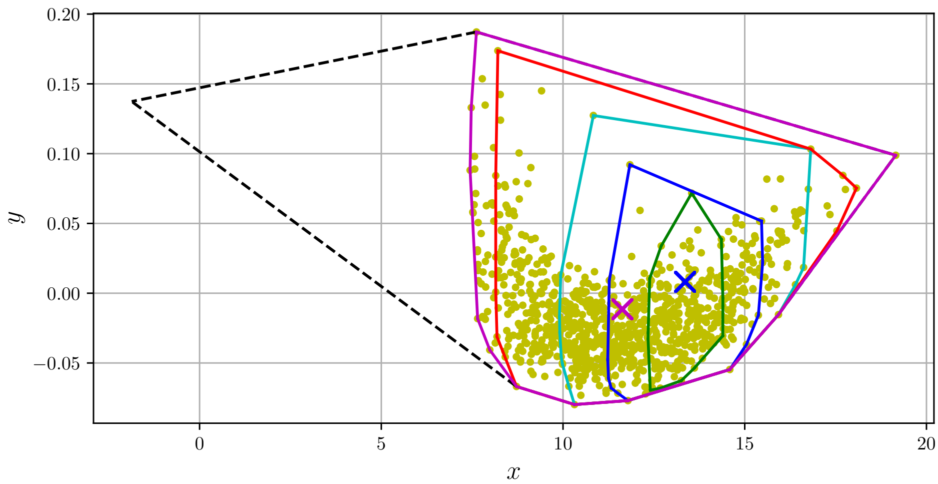


FIGURE 6.30: Zoom on the $k\%$ areas for $\mu = 0.2$, $\sigma_{x/y} = 0.02$

Direct comparison of the standard deviations for the results of test 5 In the following the standard deviations for position, velocity and acceleration of the previous test are directly compared with each other. The x- and y-components are evaluated separately. The standard deviations are plotted in t as long as not all runs of a certain simulation stopped (test 5 (red curves) stops after little less than $7s$). In the following figures $\sigma(x/y)$ denote the standard deviation of the trajectories with respect to the reference trajectory. $\sigma_{x/y}$ instead denotes the standard deviation used for the generation of the random variables.

Figure 6.31 shows on top the standard deviations of the x-coordinates and on the

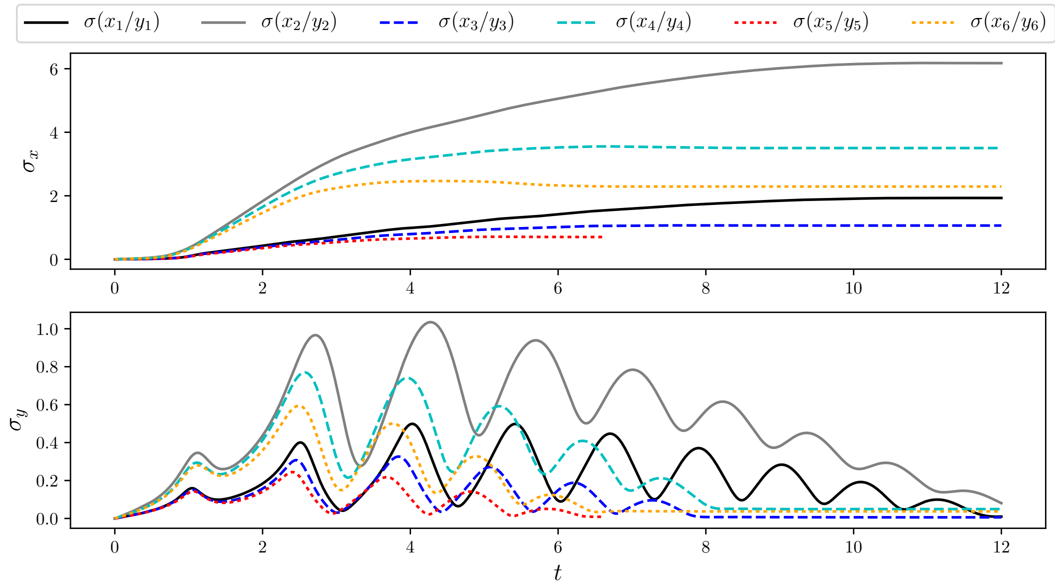


FIGURE 6.31: Standard deviation for the x- and y-coordinates of the trajectories of the previous results

TABLE 6.2: Comparison of the endpoints of the reference and averaged trajectories

$\mu \mid \sigma_{x/y}$	reference x	average x	rel. diff. x (%)
0.10 | 0.1	33.88	32.87	2.74
0.10 | 0.2	33.88	28.26	15.24
0.15 | 0.1	20.01	19.40	2.65
0.15 | 0.2	20.01	17.45	11.13
0.20 | 0.1	13.36	12.94	2.57
0.20 | 0.2	13.36	11.63	10.57

bottom those for the y-coordinates. In both cases it is easily seen that the standard deviation is larger for larger values of $\sigma_{x/y}$ (grey, cyan and yellow lines). Moreover we find that the standard deviation is larger for smaller friction coefficients μ as can be seen also in table 6.2 which shows the relative differences between the x-components of the reference and averaged endpoints. Therefore the differences between the reference and averaged solutions were divided by the x-range of the reference solution. The values are significantly larger for $\sigma_{x/y} = 0.02$. For increasing values of the friction coefficient μ one finds a slight decrease in the relative difference. $\sigma(x)$ increases in accordance to the rate of change in x-direction (of the trajectory), i.e. the changes are bigger at the beginning and less strong as the point reaches the final position. As already seen in test 1, *sigma*(y) undergoes oscillations that correspond to the oscillations of the trajectory in y-direction. These oscillations are noticeably stronger for larger values of σ_y and less strong for larger friction coefficients. The slightly lower values for larger values of μ are explained by the lower velocities of the point.

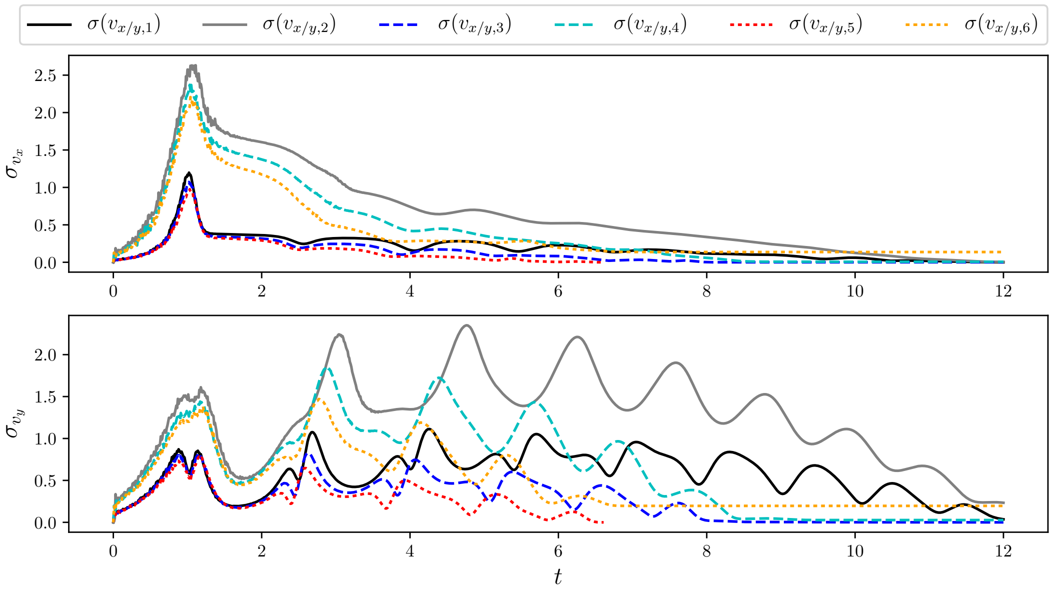


FIGURE 6.32: Standard deviation for v_x and v_y of the trajectories of the previous results

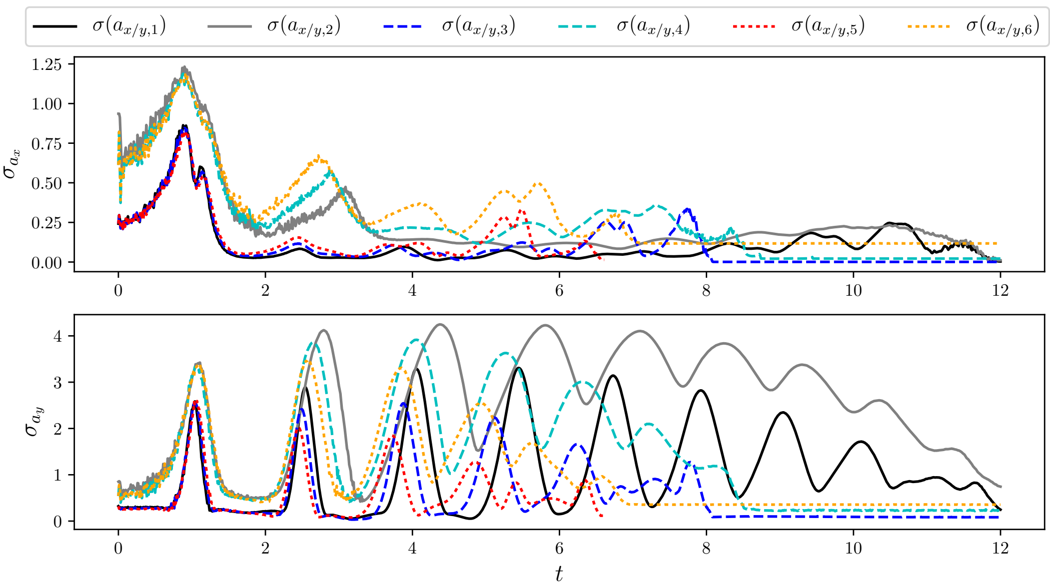


FIGURE 6.33: Standard deviation for a_x and a_y of the trajectories of the previous results

Just as well, the standard deviations for the velocities and accelerations follow those two rules. As can be seen in figures 6.32 and 6.33, $\sigma(v_{x/y})$ and $\sigma(a_{x/y})$ are larger for larger values of $\sigma_{x/y}$. Increasing the friction coefficient μ leads to lower values of $\sigma(v_{x/y})$ and $\sigma(a_{x/y})$.

First conclusion: Despite the smoothness and simplicity of the present surface, changing the standard deviation $\sigma_{x/y}$ of the random variables leads to significant changes in the outcome of the simulations. This is even more surprising as in this test no roughness was added to the surface but only the (x,y)-coordinates of the grid were varied. The z-coordinate was adjusted according to the underlying function. Therefore, all realizations of the surface represent the same smooth surface. This leads to the idea that the goodness of a simulation with respect to the reference solution might depend on the regularity of the discretization. Even though this is true in a general sense, throughout the simulations the solution on the most regular discretization was never the best one with respect to the reference.

This is not too surprising however, as the differences in regularity are marginal. This is easily explained by the so called ”regression to the mean”. If the surface is discretized using a sufficiently large amount of normally distributed random variables, all surface realizations will undergo an overall similar change in their coordinates. (As measure for the regularity of a triangle the ratio between the longest side of a triangle and its circumcircle was used.)

6.3.3 Valley with change of concavity

The next test-surface is composed (in negative x-direction) by an increasingly steep slope (concave function), followed by a change of concavity into convexity, resulting in a saddlepoint and finally ending in a convex area that quickly approaches a gradient nearly *zero* (asymptotically). In y-direction the surface builds a valley, determined by a simple parabola. As initial surface the following function $f : D \to \mathbb{R}$ with $D = [-40,8] \times [-8,8]$ and

$$f(x,y) = 0.9\arctan(1.5x))^3 + 0.1y^2.$$

The surface is shown in figure 6.34.

Test 1: adding roughness + variation of friction Here, the results shall in addition be compared for three different values of the friction coefficient μ, two different standard deviations σ_z and two different grid resolutions. In order to reduce the computational time (in contrast to the first two tests), only 500 simulations were run for each test combination. The parameters were chosen as follows: $\mu = 0.15,\ 0.175,\ 0.2$, $\sigma_z = 0.005,\ 0.01$ and $(n_x,n_y) = (129,129),\ (257,65)$ ($n_{x/y}$ are the number of grid-points in x- respectively y-direction), accounting for $(n_x-1)\times(n_y-1) = 16384$ grid

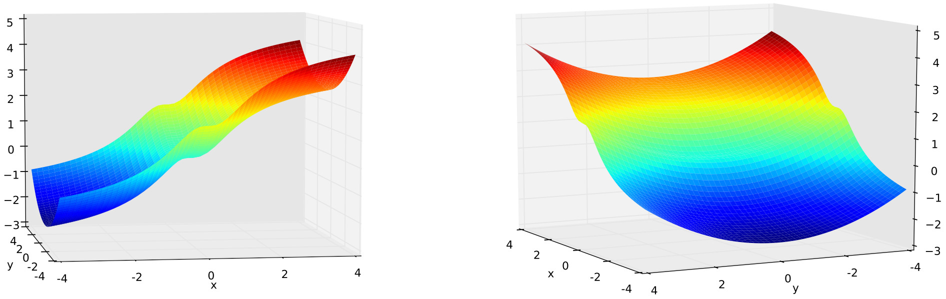


FIGURE 6.34: Test surface from different angles.

cells each. As the domain D is not squared but elongated, the non-symmetric resolution seems to be a better choice but we will go more into detail with this later.

In the first test we set $\sigma_z = 0.005$ and $(n_x, n_y) = (129, 129)$ and run 500 simulations each for the three friction coefficients mentioned above. The results are shown below; figure 6.35 shows the three trajectories and the averaged trajectory over all the simulations (top: $\mu = 0.15$, middle: $\mu = 0.175$, bottom: $\mu = 0.2$). It is easily seen that for each value of μ the original and the averaged solution nearly coincide, not just in terms of their shape but also concerning the endpoints. This indicates that on average the randomization of the surface with $\sigma_z = 0.005$ has relatively little impact on the final result.

Looking at different values of μ instead we find a totally different picture with strongly differing solutions. While for $\mu = 0.15$ the motion shows significant oscillations in y-direction and stops in $x \approx -31.5$, for $\mu = 0.175$ oscillations are damped after the point passed the saddlepoint while it stops moving in $x \approx -25$. A totally different picture is found for $\mu = 0.2$ as the point stops moving in vicinity of the saddlepoint at $x \approx 6$. As in the saddlepoint both partial derivatives f_x and f_y are *zero*, and also around the saddlepoint these derivatives are very small, there is no significant acceleration of the point and therefore the motion stops for too large values of μ.

The $k-\%$ areas for all three cases are depicted in figure 6.36. The 100% areas are indicated by thick dashed lines, while the 95% areas are indicated by solid, the 75% areas by dashed and the 50% areas by dash-dotted lines. 25% areas are indicated by filled areas in the respective colours. We find (few) outliers stopping around the saddlepoint in the first two cases $\mu = 0.15, 0.175$, but, while for $\mu = 0.15$ there are less than 1% of outliers, for $\mu = 0.175$ this number is larger, even though still less than 5%. However, it could be an indicator for the saddle point being a critical point for this case. This assumption is supported by the fact that around the saddlepoint we find a significant damping of oscillations, indicating a relatively low velocity of the point in this position.

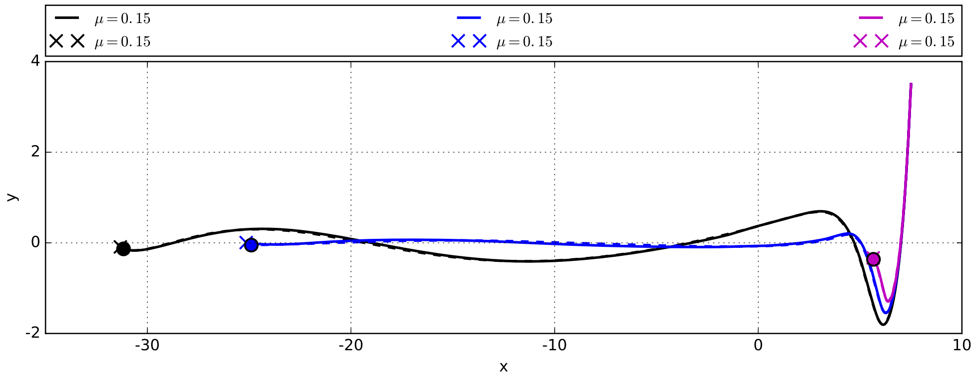


FIGURE 6.35: Trajectories and the averaged trajectory over all the simulations for $\sigma = 0.005$ and $(n_x, n_y) = (129, 129)$ (black: $\mu = 0.15$, blue: $\mu = 0.175$, magenta: $\mu = 0.2$)

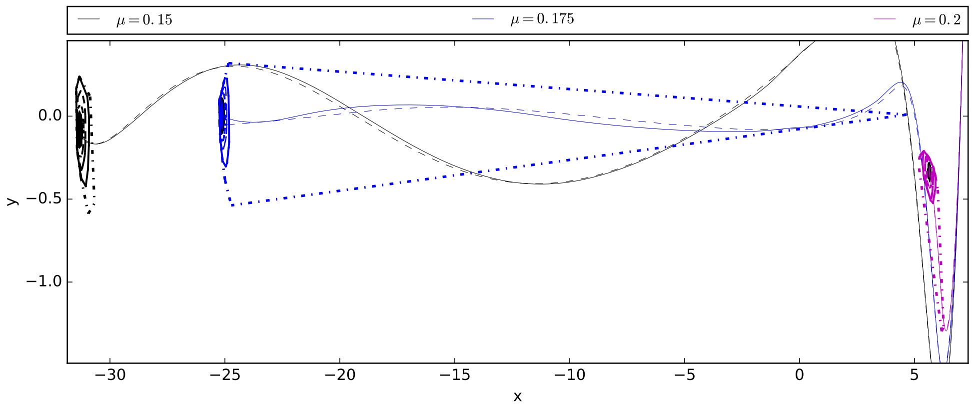


FIGURE 6.36: $k-\%$ areas for all three original and the averaged trajectory over all the simulations for $\sigma = 0.005$ and $(n_x, n_y) = (129, 129)$ (black: $\mu = 0.15$, blue: $\mu = 0.175$, magenta: $\mu = 0.2$)

Figure 6.37 shows a zoom on the $k-\%$ areas for the three cases (left: $\mu = 0.15$, middle: $\mu = 0.175$, right: $\mu = 0.2$).

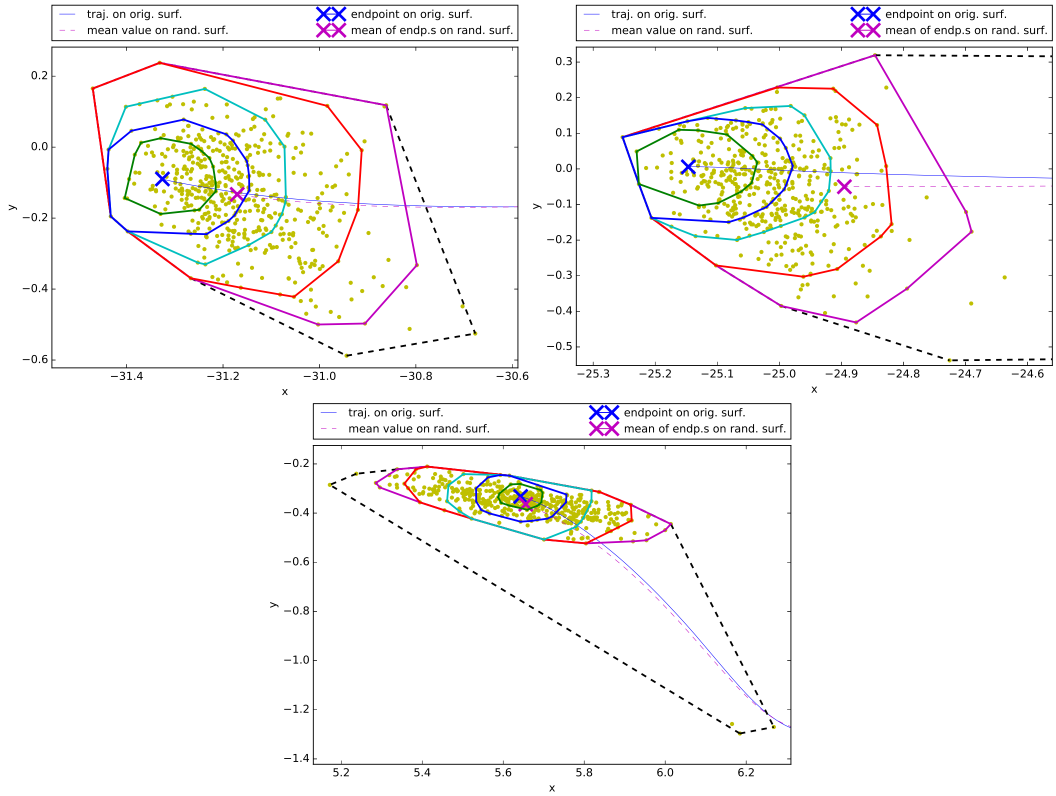


FIGURE 6.37: Zoom on the $k\%$ areas (despite 100%) for $(n_x, n_y) = (129, 129)$, $\sigma_{x/y} = 0.005$ and $\mu = 0.15$ (left) $\mu = 0.175$ (middle) and $\mu = 0.2$ (right)

Test 2: adding roughness + variation of grid resolution and friction coefficient For the third test we change the resolution of the grid. Keeping the total number of grid cells constant, the number of cells in x-direction is doubled while in y-direction it is divided by two. This leads to more regular cells in the x/y-plane, but does not consider the steepness of the surface in the different coordinate directions. (This will quickly be discussed later.) The standard deviation is again $\sigma_z = 0.005$ and tests were run for the already known friction coefficients. As in the previous two tests, figure 6.38 shows the trajectories for the three known friction coefficients. Compared to the first test case we find a stronger variation of the endpoints, especially for $\mu = 0.175$ the averaged endpoint differs significantly from the original one. The influence though is less strong than in case two where we increased the standard deviation of the randomization. This is confirmed by figure 6.39, indicating a smaller standard deviation of the endpoints (note that the scale on the y-axis is different!). There is no outlier for $\mu = 0.15$ and they are fewer for $\mu = 0.2$ (less than 1% compared to slightly less than 5% in the previous case). Besides for $\mu = 0.175$ the results concerning the standard deviation of the endpoints as well as their outliers look similar to the first case (compare also figures 6.40 and 6.41). Nonetheless, this case shows again the instability of the problem. Increasing the number of cells in x-direction – the main

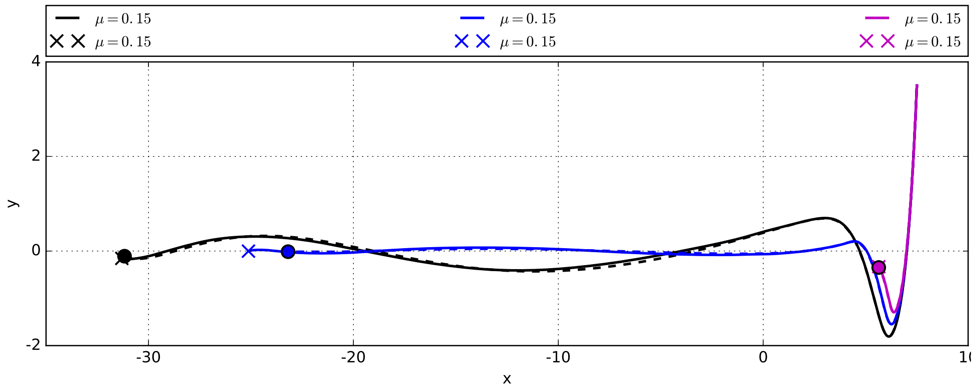


FIGURE 6.38: Trajectories and the averaged trajectory over all the simulations for $(n_x \times n_y) = (257 \times 65)$ and $\sigma = 0.005$ (black: $\mu = 0.15$, blue: $\mu = 0.175$, magenta: $\mu = 0.2$)

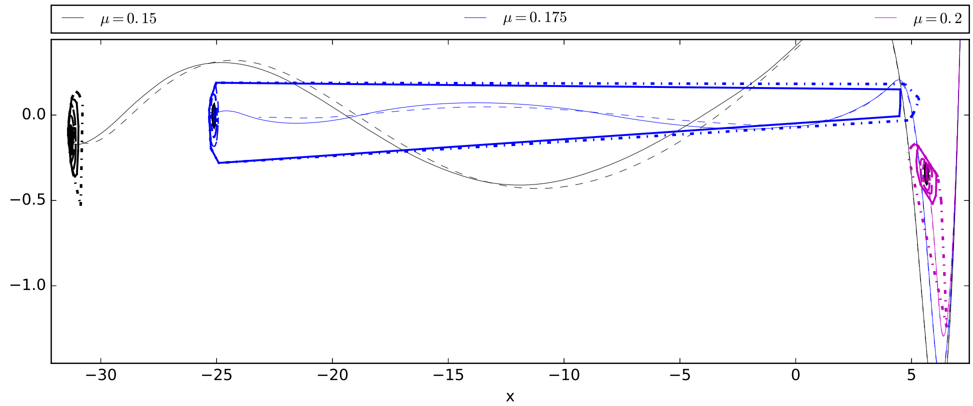


FIGURE 6.39: $k\%$ areas for all three cases (black: $\mu = 0.15$, blue: $\mu = 0.175$, magenta: $\mu = 0.2$) for $(n_x, n_y) = (257, 65)$ and $\sigma_z = 0.005$

direction of motion – generally leads to a slightly increased friction which in turn is enough to stop a significant number of points in the saddlepoint. In other words, missing just few information on the surface, but also adding few wrong information can have a strong impact on the results! This however cannot be avoided as the real surface is given by single data-points only. A Monte Carlo simulation cannot "repair" this of course, but using the principle of "regression to the mean", it can tell us more about the probability of a certain outcome.

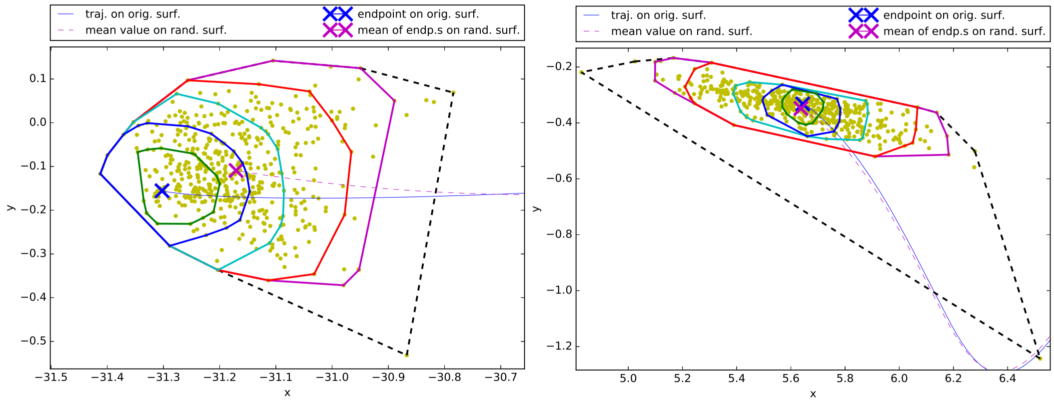


FIGURE 6.40: Zoom on the $k\%$ areas (despite 100%) for $(n_x, n_y) = (257, 65)$, $\sigma_z = 0.005$ and $\mu = 0.15$ (left) and $\mu = 0.2$ (middle and right)

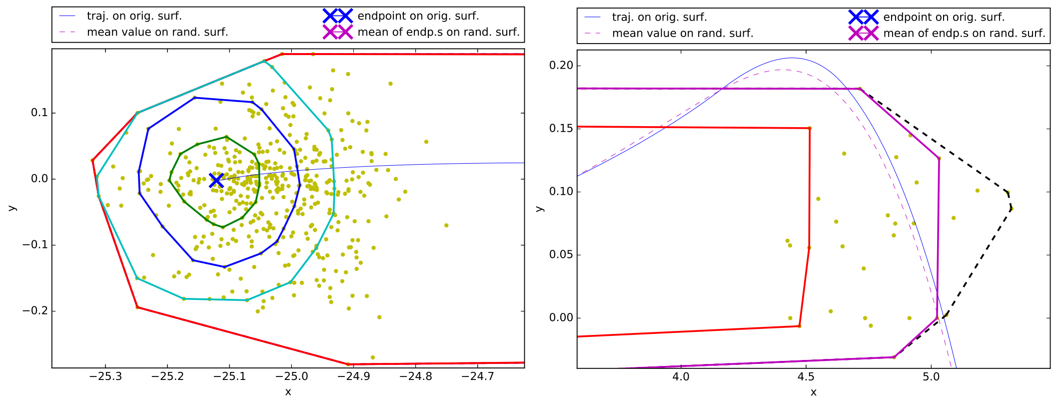


FIGURE 6.41: $k\%$ areas for $\mu = 0.175$ (for $(n_x, n_y) = (257, 65)$, $\sigma_{x/y} = 0.005$)

Test 3: adding roughness + change of grid resolution and distribution parameter σ_z Finally we also look at the results for $\sigma_z = 0.01$, $(nx, ny) = (257, 65)$ and the three known friction coefficients. Again, the trajectories are plotted in figure 6.42 and the respective $k\%$-areas are shown in figure 6.43. For this parameter configuration we find the strongest differences between the original and the averaged endpoint-values. This is most significant for $\mu = 0.175$ again, but in this case we finally also find at least one point for $\mu = 0.2$ that does not stop in the saddlepoint. Apparently, due to the larger standard deviation (used for the randomization of the surface) in combination with the coarser resolution in y-direction (=less friction in this direction than in the first two cases) in some cases ($< 1\%$) a slope is formed that allows a velocity that is just large enough to overcome the saddlepoint. On the other hand, the number of outliers for $\mu = 0.175$ that stop in vicinity of the saddlepoint increases a lot. Therefore also the difference between the original and the averaged endpoint is the largest for all the three cases. Overall, the standard deviation for the endpoint-position is increased significantly compared to the other three cases as can be seen in figures 6.44 and 6.45.

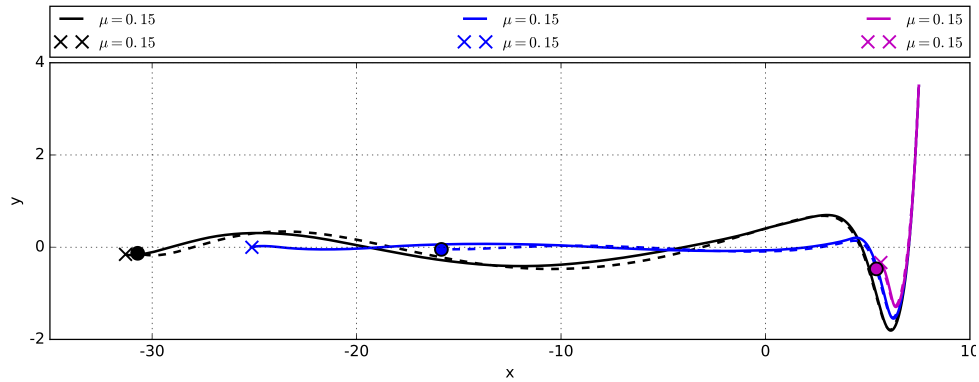


FIGURE 6.42: Trajectories and the averaged trajectory over all the simulations (top: $\mu = 0.15$, middle: $\mu = 0.175$, bottom: $\mu = 0.2$)

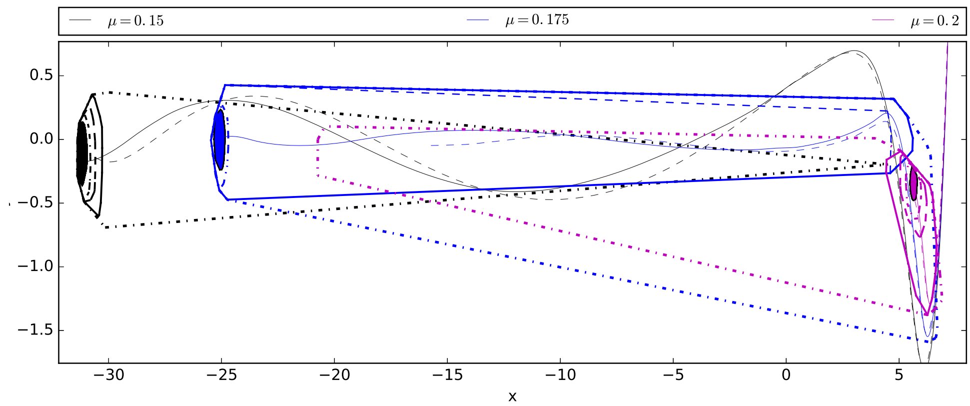


FIGURE 6.43: Trajectories and the averaged trajectory over all the simulations (top: $\mu = 0.15$, middle: $\mu = 0.175$, bottom: $\mu = 0.2$)

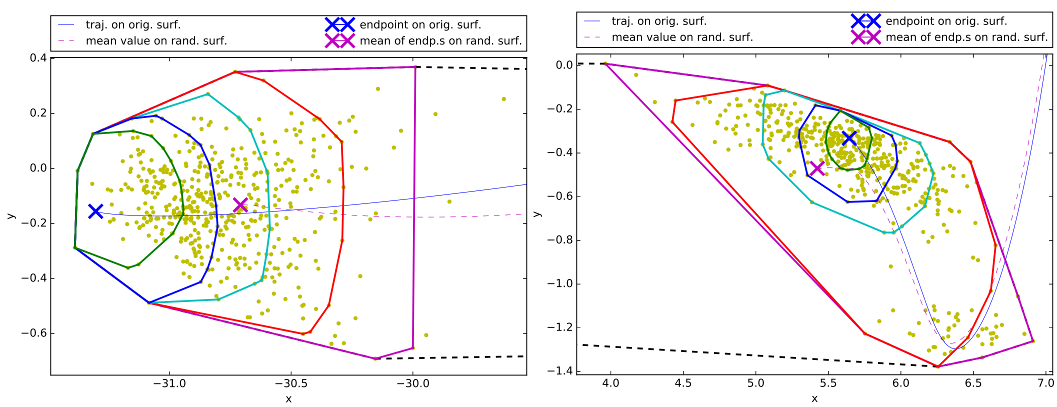


FIGURE 6.44: Zoom on the $k\%$ areas (despite 100%) for $\mu = 0.15$ (left) and $\mu = 0.2$ (right)

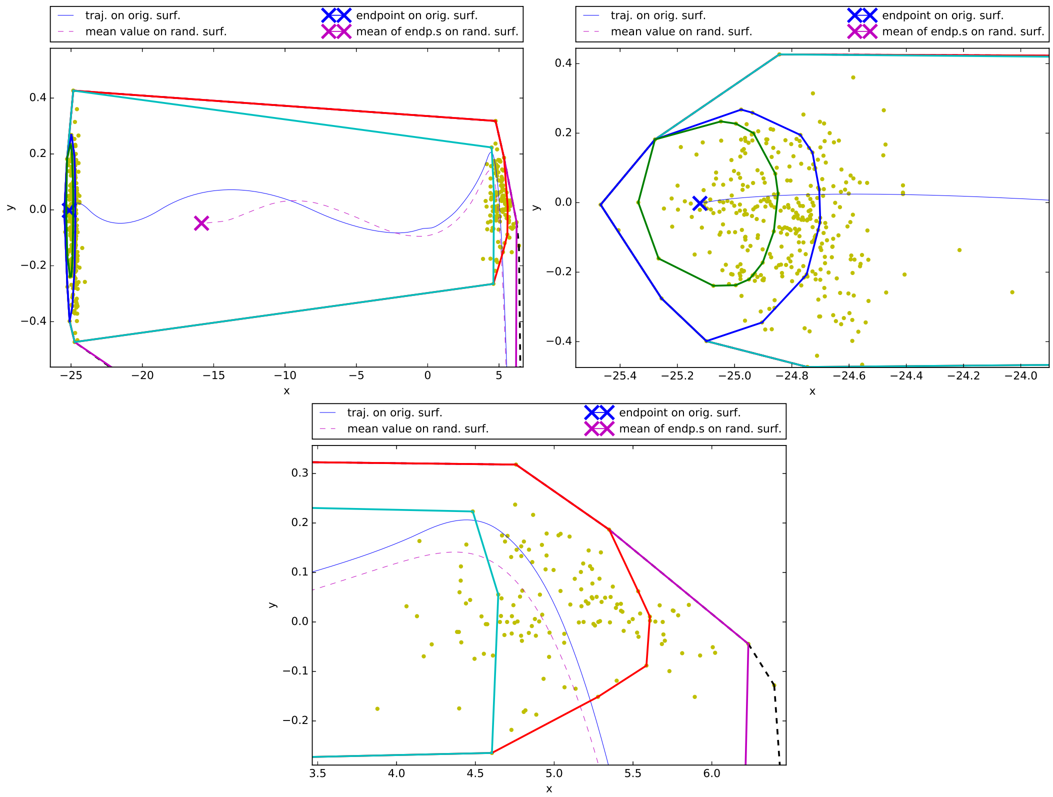


FIGURE 6.45: Zoom on the $k\%$ areas (despite 100%) $\mu = 0.175$

Test 4:changing (x,y)-coordinates + variation of friction and distribution parameter σ Keeping the other parameters constant, we only increase the standard deviation in this test and set $\sigma_{x/y} = 0.01$. Again 500 simulations are run for each of the three friction coefficients mentioned above. The results are depicted in the

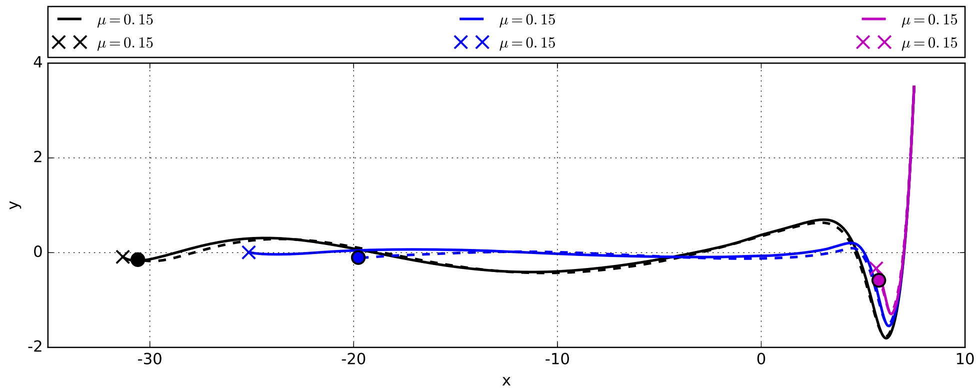


FIGURE 6.46: Trajectories (solid lines) and the averaged trajectories (dashed lines) over all the simulations for $(n_x, n_y) = (129, 129)$ and $\sigma = 0.01$ (black: $\mu = 0.15$, blue: $\mu = 0.175$, magenta: $\mu = 0.2$)

figures that are described below. Figure 6.46 shows the three original and averaged trajectories for $\mu = 0.15$, $\mu = 0.175$ and $\mu = 0.2$. Compared to the previous results with $\sigma = 0.005$ it immediately stands out that, while the shapes of the trajectories do

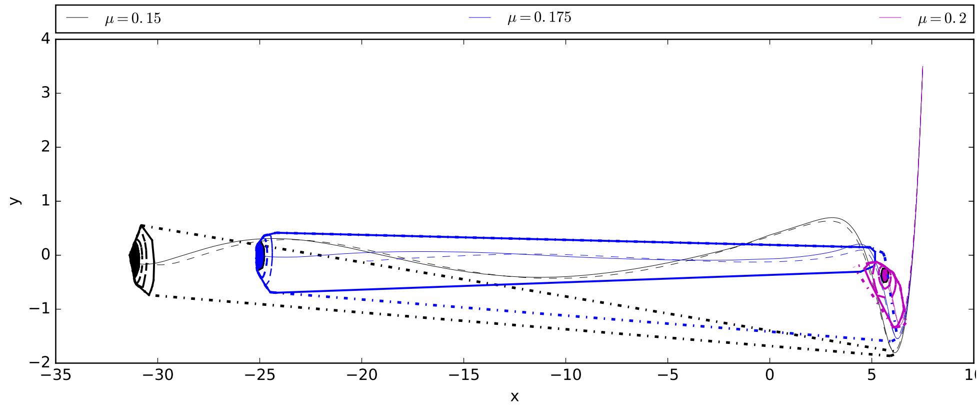


FIGURE 6.47: $k\%$ areas for all three cases (black: $\mu = 0.15$, blue: $\mu = 0.175$, magenta: $\mu = 0.2$) for $(n_x, n_y) = (129, 129)$ and $\sigma_{x/y} = 0.01$

not differ strongly, the endpoints do so. The strongest effect is found for $\mu = 0.175$, confirming that in this case the saddlepoint is critical for the following motion. While for $\mu = 0.15$ the velocity at the saddlepoint in almost all cases stays large enough to overcome this plateau and for $\mu = 0.2$ the velocity is never enough, both cases are possible for $\mu = 0.175$ such that there is a large range of possible endpoints. Besides that, the $k\%$ areas for all values of μ are larger than in the previous test.

Figure 6.47 depicts the $k\%$ areas for all three cases. Again, the 100% areas are indicated by thick dashed lines, while the 95% areas are indicated by solid, the 75% areas by dashed and the 50% areas by dash-dotted lines. 25% areas are indicated by filled areas in the respective colours.

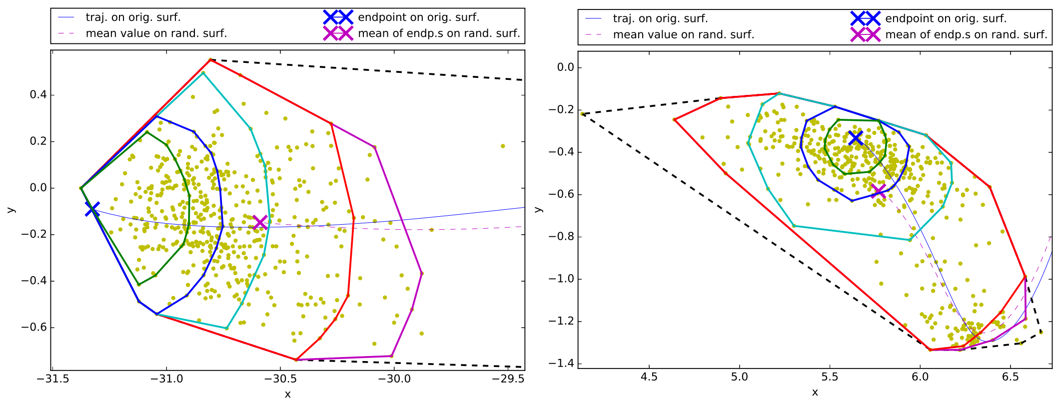


FIGURE 6.48: $k\%$ areas for the first and last case ($\mu = 0.15$ and $\mu = 0.2$)

Figure 6.48 shows a zoom on the $k - \%$ areas for the first and last case ($\mu = 0.15$ and $\mu = 0.2$), figure 6.49 shows the $k\%$ areas for the second case ($\mu = 0.175$). As already mentioned, it is special as the endpoints are not concentrated in basically one single area, but there is a second accumulation of points, located in the saddlepoint of the surface. The latter accounts for about 17% of all endpoints and thus these endpoints cannot longer be considered as outliers. All together we find that increas-

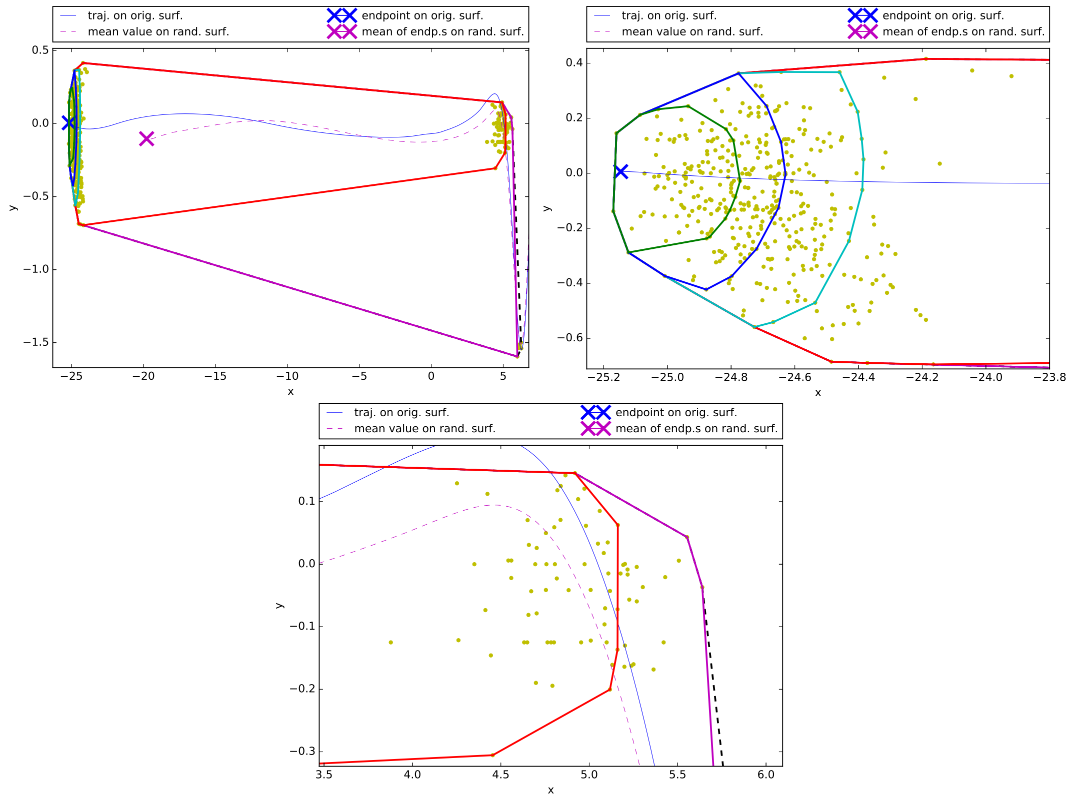


FIGURE 6.49: $k\%$ areas for the second case ($\mu = 0.175$)

ing the standard deviation for the randomization (not surprisingly) also increases the standard deviation of the results. The 25%-area nearly doubles its radius. Compared to the first test case (Valley with decreasing slope) it is interesting to see that for $\mu = 0.15$ and $\mu = 0.175$ nearly all points on the randomized surfaces stop earlier than the point on the original surface. In contrast, for $\mu = 0.2$ this is not the case. Like in the first test case the mean value is very close to the original value, both relatively centred within the point-cloud. This can be explained by the (generally) increased friction due to the increased roughness in the randomized surfaces. We already mentioned that the saddlepoint is a critical point of this surface. If the velocity of the point in the saddlepoint is barely enough to overcome it and go on moving, every small perturbation of the surface that additionally increases the friction has large effects on the continuing motion as it determines if the point stops or the velocity is just enough to keep the point moving. If the friction coefficient instead is already too large as to be enough to overcome the saddlepoint, or (which in principle is equivalent) there is no such point (like in the "Valley with decreasing slope"), small perturbations in the surface lead to small accelerations or decelerations of the point but cannot critically change the point's behaviour.

6.3.4 High order 2D polynomial with additional bumps

The last test surface is more complex, based on a high order 2-dimensional polynomial with additional bumps:

$$f(x,y) = 0.2\sum_{i=0}^{8} a_i(y+1.3)^i + 0.1\sum_{j=0}^{6} b_j(x+1.3)^j + 0.01\cos(x/4)\cos(y/2)$$

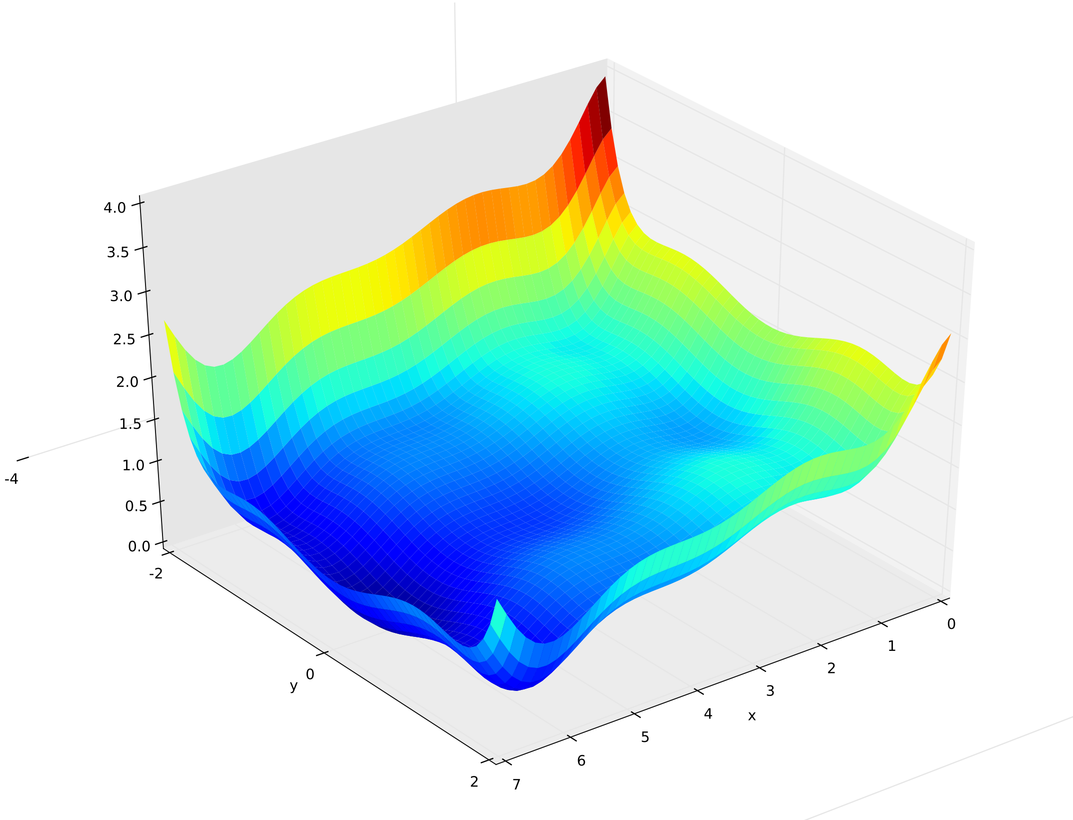
with parameters

$$\begin{aligned} a_i = [&-122,\ 353.531,\ -369.969,\ 200.962,\ -63.1981,\\ &11.9264,\ -1.3309,\ 0.0808532,\ -0.00205853], \quad i = 0,\dots,8 \\ b_j = [&-3.01077,\ -0.32076,\ 6.97421,\ 0.61875,\ -4.83125,\\ &-0.275,\ 1], \quad j = 0,\dots,6 \end{aligned}$$

Fig. 6.50 shows the resulting surface in the domain $(x,y) \in [0,7] \times [-2,2]$ from two different angles. The slope decreases (non-monotonically) in x-direction ending in a depression stretched out along y that itself consists of three minima. In y-direction the surface forms a bumpy, flat valley that steeply raises at the boundary of the domain. The grid width of the original grid was chosen $d_x = d_y = 0.04$. During the Monte Carlo simulation only the interior points of the grid are randomized such that the domain remains constant. The reference solution is always the solution on the original (non-randomized) surface.

In the following, three tests for different friction coefficient μ are shown. During each test the standard deviation $\sigma_{x/y}$ was varied using $\sigma_{x/y} = 0.0002, 0.0004, 0.0008,$ $0.0016, 0.0032$. Simulations were run with 500 different realizations of the surface each. The initial conditions for all test cases are: $\mathbf{x}_0 = (0.05, 1.95)$ and $\mathbf{v}_0 = (0,0)$. To evaluate the results, the visiting times (=number of times the point passes through a certain triangle) for each triangle were counted. Fig.s 6.51 to 6.53 show the domain $[0,8] \times [-2,2]$ where the triangles of the discretization are coloured according to their visiting time. For a better distinction, non-visited triangles are left white. The reference solution is indicated by a red line while the black dashed line is the averaged trajectory.

Test 1: $\mu = 0.15$ For $\sigma_{x/y} = 0.0002$ the trajectories vary little from the reference solution which therefore matches the averaged solution. Until $\sigma_{x/y} = 0.0008$ the reference and averaged trajectory basically coincide, even though single trajectories start to diverge from the reference solution. For $\sigma_{x/y} = 0.0016$ and even more for $\sigma_{x/y} = 0.0032$ we finally see a wide range of possible trajectories with three accumulation points which are actually the three minima of the surface's depression. In



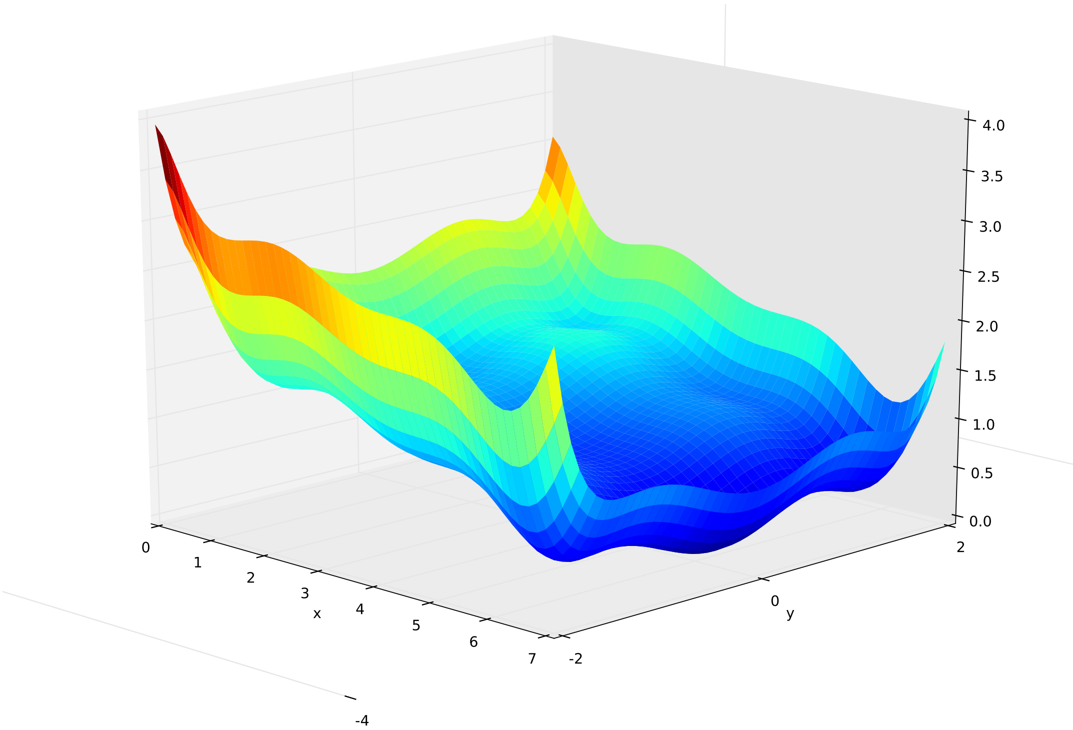


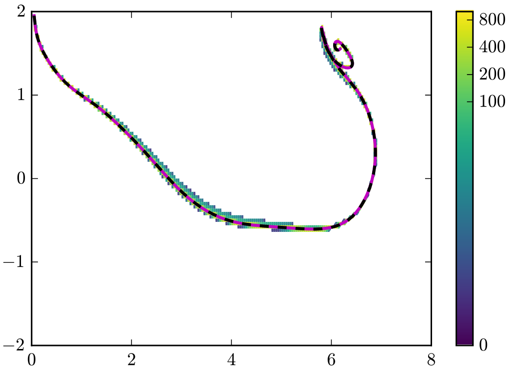
FIGURE 6.50: Surface shown from two different angles.

the last case also the averaged trajectory is clearly different from the reference solution with its endpoint located in vicinity of the deepest point of the surface (global minimum with respect to the given domain).

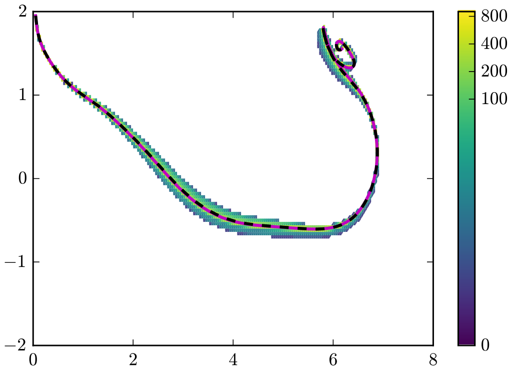
Test 2: $\mu = 0.20$ Increasing the friction coefficient results in a shift of the reference endpoint towards the global minimum and less strong variation in the trajectories. The reference and averaged trajectory basically coincide until $\sigma_{x/y} = 0.0016$ and are still very similar also for $\sigma_{x/y} = 0.0032$. Overall, the reference trajectory is relatively similar to the averaged trajectory for the case $\mu_{x/y} = 0.15$, $\sigma_{x/y} = 0.0032$ (cf. Fig. 6.51e).

Test 3: $\mu = 0.25$ As in the previous case, the averaged and reference solution coincide until $\sigma_{x/y} = 0.0016$ and still are similar for $\sigma_{x/y} = 0.0032$. The variation of the trajectories slowly increases between $\sigma_{x/y} = 0.0002$ and $\sigma_{x/y} = 0.0016$ and rises significantly for $\sigma_{x/y} = 0.0032$ even though it is still less strong than in the case $\mu = 0.2$, $\sigma = 0.0032$. The number of outliers however is higher than in test 2.

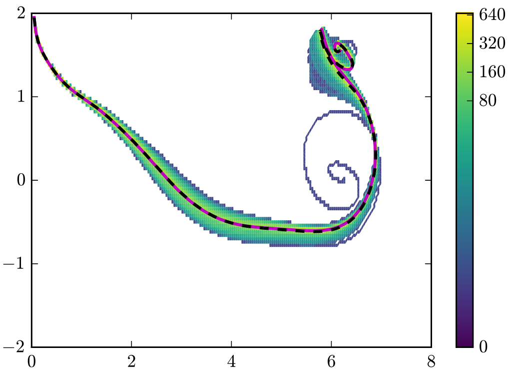
Altogether, also for these tests the strong variation of the trajectories is noticeable where the variation increases with $\sigma_{x/y}$ and decreases with μ. Especially for $\mu = 0.15$ and $\sigma_{x/y} = 0.0016,\ 0.0032$ it sticks out that the trajectories cover the whole depression, a result that is not a priori expected from the reference solution. It clearly illustrates the dependency of the motion on the surface. Certainly it is expectable that for larger mass-blocks the variation is smaller than for a single mass point. Nonetheless, Monte Carlo simulations are likely to add valuable information also in case of real landslide simulations as usually many parameters (like the surface or friction coefficient) are not known exactly. In this way, a whole range of possible outcomes, including their likelihood, can be computed instead of a single result. Thus, the most probable scenario, but also extreme outcomes can be estimated, leading to a more complete picture of the problem.



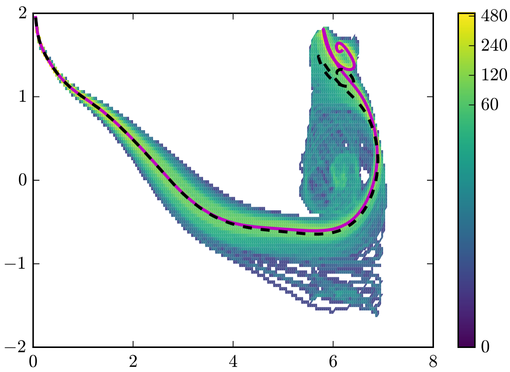
(A) $\mu = 0.15$, $\sigma = 0.0002$



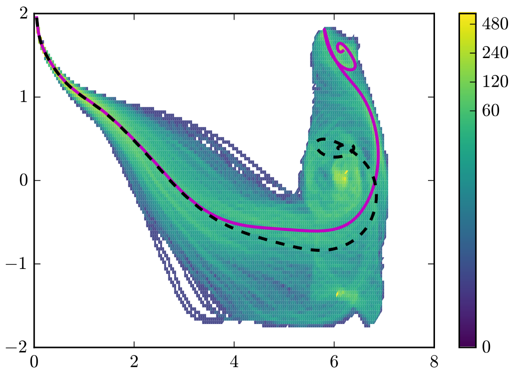
(B) $\mu = 0.15$, $\sigma = 0.0004$



(C) $\mu = 0.15$, $\sigma = 0.0008$

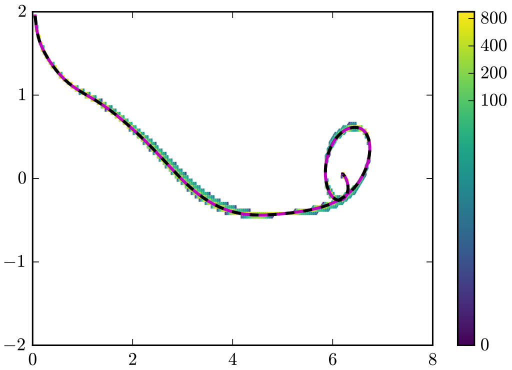


(D) $\mu = 0.15$, $\sigma = 0.0016$

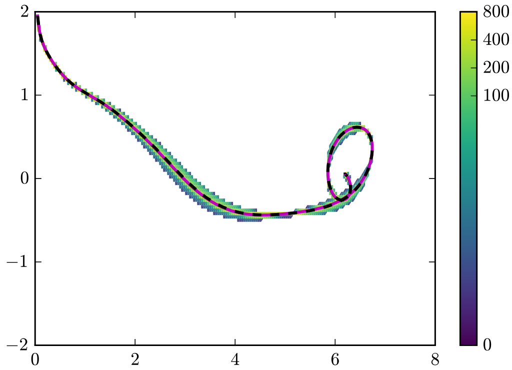


(E) $\mu = 0.15$, $\sigma = 0.0032$

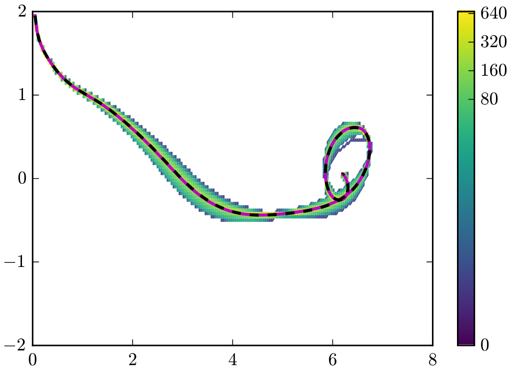
FIGURE 6.51: Results of the Monte Carlo simulation for $\mu = 0.15$



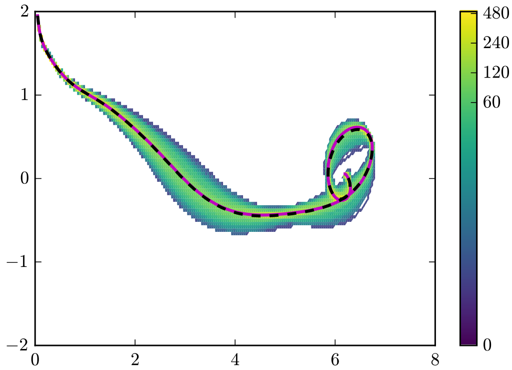
(A) $\mu = 0.2$, $\sigma = 0.0002$



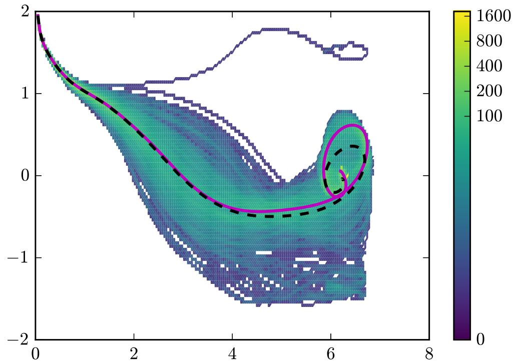
(B) $\mu = 0.2$, $\sigma = 0.0004$



(C) $\mu = 0.2$, $\sigma = 0.0008$

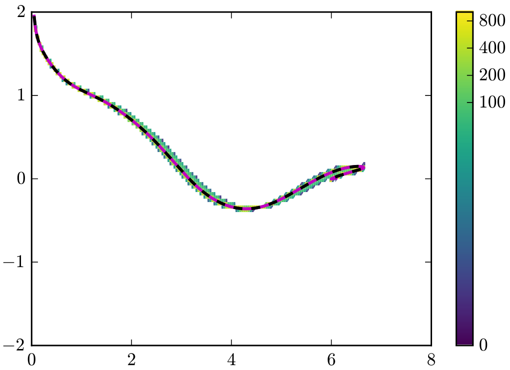


(D) $\mu = 0.2$, $\sigma = 0.0016$

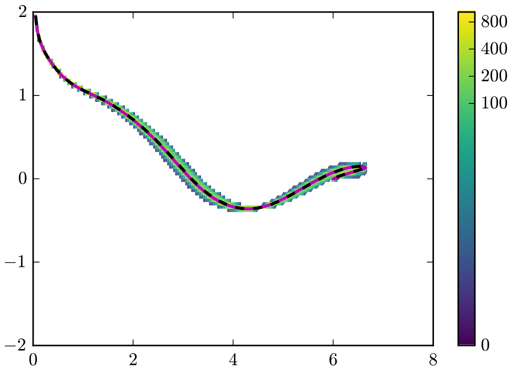


(E) $\mu = 0.2$, $\sigma = 0.0032$

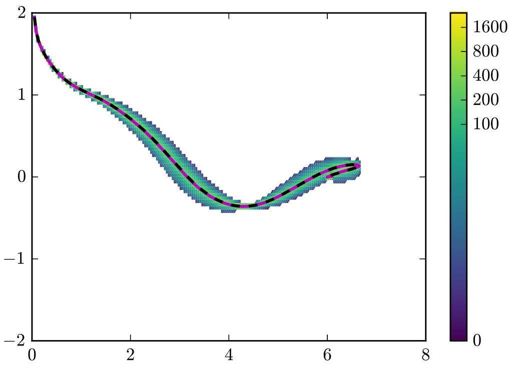
FIGURE 6.52: Results of the Monte Carlo simulation for $\mu = 0.2$



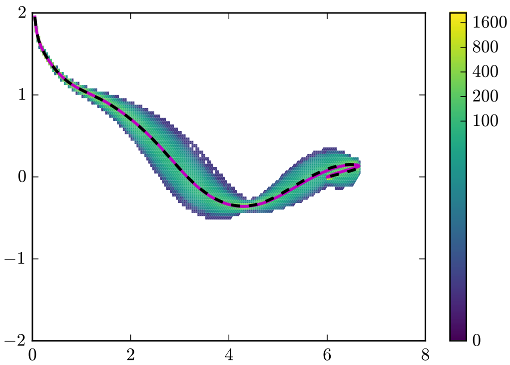
(A) $\mu = 0.25$, $\sigma = 0.0002$



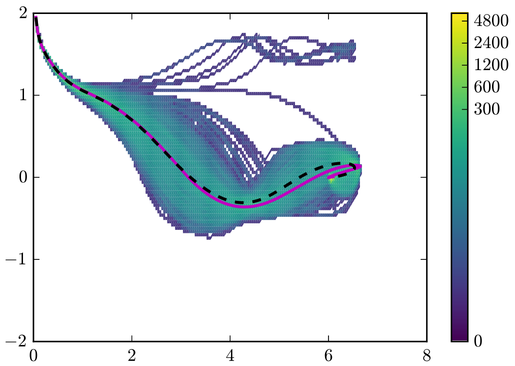
(B) $\mu = 0.25$, $\sigma = 0.0004$



(C) $\mu = 0.25$, $\sigma = 0.0008$



(D) $\mu = 0.25$, $\sigma = 0.0016$



(E) $\mu = 0.25$, $\sigma = 0.0032$

FIGURE 6.53: Results of the Monte Carlo simulation for $\mu = 0.25$

6.4 Conclusion:

Overall, the presented test-cases clearly demonstrate the dependency of the solution on the initial conditions and the sliding surface. It could be shown that even for very simple and smooth surfaces (valley with decreasing slope), increasing the standard deviation $\sigma_{x/y}$ leads to significant changes in the final solution and the averaged trajectory differs noticeably from the reference trajectory. This does not necessarily imply that in real landslide simulations the solutions will differ that much, instead, the volume and weight of real blocks compared to single points will act as a stabilizer, on the other hand, the results clearly show that the problem in general is unstable and thus, results have to be taken with care. The simulations should later be repeated with the full program, including several blocks and interaction between blocks.

In chapter 3, different methods for the reconstruction of surfaces from a given set of points were presented. As it could be shown that even for very simple, smooth surfaces the reconstruction/grid plays a significant role, it seems recommended to run simulations not only for different realizations of one reconstruction, but for different types of reconstructions as they usually do not reconstruct particular surface characteristics in the same way.

One big advantage of the new method that was presented in previous chapters is its efficiency. Together with an optimized implementation, several thousand full simulations can be run in an appropriate amount of time, even on a home computer/laptop. Not all the results of course are stored as this would require to much storage, instead the statistics are computed "on the fly" (as it is already done now). In this way, instead of getting a single result (from a single simulation) we get an averaged result (from simulations on a wide range of possible realizations of the same surface) plus a best/worst case scenario (extrema over all the simulations) and information on the standard deviation.

It should in addition be checked, if the results of the simulations follow a certain distribution (e.g. are they again distributed normally?). All together this will give a much more complete picture for a certain problem.

Certainly, for real world simulations one should definitely consider using correlated random numbers instead of uncorrelated ones when adding roughness to the surface in order to avoid an unnatural level of roughness. It is definitely an interesting observation that also changing the grid resolution, despite keeping the overall amount of cells constant and actually thereby creating more regular cells, can significantly change the results. While for adding roughness on might expect a larger variety in the Monte Carlo simulations concerning the trajectories, the strong variation of the results when only slightly changing the distribution of the data-points while keeping the underlying surface constant, are definitely surprising and should be further investigated for real landslides.

Chapter 7

Possible Application of the Results in Real World Landslide Simulations

Up to now only a single mass point was considered in the computations. In this case, the triangle in which the point is located, is unique and acceleration due to gravity is computed easily. If instead, the moving mass is represented by a larger block, it will generally spread over several triangles that all influence its acceleration which therefore is more difficult to determine. It therefore seems appropriate to compute the acceleration, taking into account the slopes of all triangles covered by the block.

This however leads to one problem: the method presented throughout this work is based on the re-computation of the acceleration each time a point leaves a certain triangle and moves to another one. For larger blocks, whose motion (not acceleration) is computed using few points (e.g. the barycentre or the nodes of the block), the acceleration due to gravity can be recomputed when one of those points leaves its triangle, while there is no information on the rest of the block. It follows that the acceleration theoretically can change significantly without being considered in the computations. The discrepancy between the real and the considered change in acceleration obviously depends on the size of the blocks with respect to the size of the triangles building the surface.

For a single block this problem can be reduced using several representative points laying inside the block. During time-integration all those points are moved (having equal acceleration and velocity) and the acceleration is recomputed as soon as one of the representatives leaves its current triangle. This is possible as the shape of the whole blocks is unchanged during motion. This is displayed in fig. 7.1. The blue rectangle indicates the initial position of the block, its barycentre in which the forces are applied is given by a red dot and its representative points are shown in blue again. The motion of the block is indicated by black arrows. Finally the new position of the block and its representative points are shown in green, the barycentre instead is still indicated by a red dot. On can see several representative points having reached the border of their triangles while the barycentre did not yet reach the boundary of its triangle.

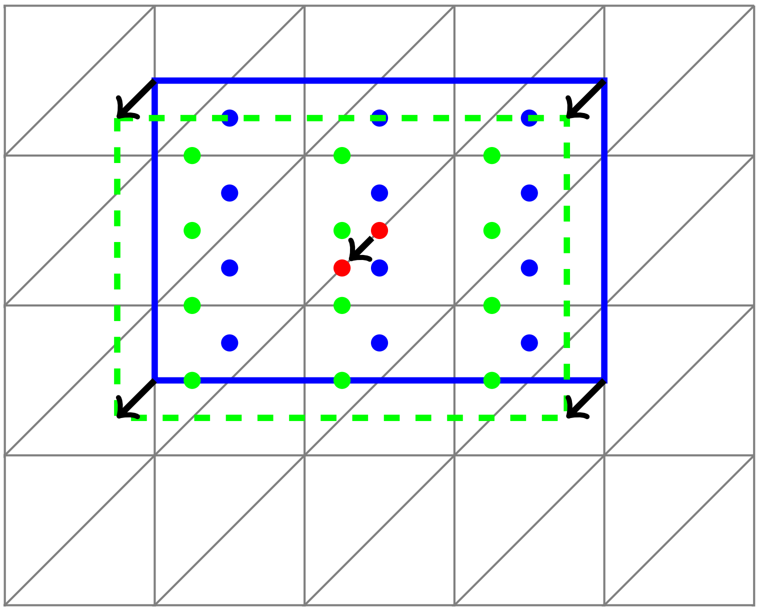


FIGURE 7.1: Motion of a single block using representative points

For more than one interacting block instead, this is not applied easily as also the representative points should all move with a different velocity. In any case, in contrast to the old model, the time-step used for the time-integration scheme can generally not longer be constant. Assuming that the motion of each block is determined by the motion of its four (possibly also more) vertices, the acceleration of each single block should be recomputed as soon as any of the vertices leaves its triangle. This requires kind of a double time-integration using a general time-step $\triangle t_g$ and an intermediate time-step $\triangle t_i$. For a certain vertex the time needed to reach the boundary of its triangle is the intersection time $\triangle t_{int}$ The time-integration scheme can then look as follows:

Step 1: Compute a full time-integration step for each vertex using $\triangle t_g$;

Step 2: Check if any of the vertices left its triangle;

Step 3a: If no: proceed with the next time-step (Step 1), no acceleration is recomputed;

Step 3b: If yes: compute the minimum of all intersection times over all vertices:

$$\triangle t_i = \min(\triangle t_{int})$$

Step 4: Recompute the current time-integration step with time-step $\triangle t_i$ and proceed with the next time-step (Step 1).

This routine is computationally more expensive than just recomputing the acceleration at the end of a full time-step. As however the implementation performs very well in terms of computational time this will not be an issue. It can be further optimized by a proper choice of the time-step that in the best case is adjusted to the velocity of the blocks.

Chapter 8

Summary and Future Work:

The main goal of the present work was to develop an efficient – and, in the frictionless energy-conserving – method for solving the equations of motion given in Eq. (2.9) on 1D and 2D curves and surfaces, respectively. Therefore two different approaches were compared: solving the equations numerically on smooth reconstructed surfaces and computing a (semi-)analytical solution on triangulated surfaces. As the problem generally is stiff, special care has to be taken concerning the choice of the time-integration method. While for smooth surfaces the LSODE package provides an efficient solver, particularly suited for stiff problems, no standard solver is available for piecewise planar (=non-smooth) surfaces.

Three different methods for the reconstruction of smooth surfaces using radial basis functions where presented and compared to each other. It could be shown that they all have specific advantages and disadvantages but generally perform well even in the reconstruction of sharp changes in slope. Moreover, the Livermore Solver for ODEs was presented.

For the 1-dimensional case a method with which the equations of motion on a discretized curve formed by straight segments can be solved fully analytically, using the approximation of infinitely small circular sectors to smooth discontinuities, was presented. It was shown that the gravity-containing terms of the equations of motion are negligible compared to the curvature-dependent terms as the point moves through one of those circular sectors of infinitely small radius. On the other hand, the curvature-dependent terms vanish in the linear regions of the curve. Solving the mentioned terms independently of each other leads to a very efficient way for the computation of centripetal acceleration and the corresponding friction term, that does not require any information on second derivatives.

In the examples it could be shown that the method works well, reaching about second order of convergence. The results obtained on discretized curves by splitting the equations of motion into its gravity-dependent and curvature-dependent components are fully comparable to those obtained by solving the equations of motions numerically on corresponding smooth curves.

Omitting curvature-dependent friction terms instead can lead to completely different results and significantly changes the behaviour of the sliding body. It could be shown in the examples that these terms influences the motion more than grid refinement. In the second example we saw that the maximum error was about 100 times larger than the maximum error in the solution of the full equations.

A big advantage of the presented method is its low computational cost as no approximations have to be made. All necessary computations can be done exactly, which also leads to a high accuracy. Moreover, the total energy is conserved in all cases in absence of friction. This is a direct result of the exact solution and high accuracy in the computations, since it is not directly imposed in the equations themselves.

Very similar results were obtained also for the 2-dimensional case in which the piecewise-linear curves were replaced by piecewise planar surfaces. With only little changes, the same technique as in 1D can be applied, even though, the implementation is significantly more complex.

To further examine the model, and the dependency of the solution of the equations on the surface and the initial conditions, a Monte Carlo simulation was implemented and used. The test that were run revealed a possibly strong dependency of both components, in particular however of the surface. Tests were run either adding roughness to the existing surface, or, just changing slightly the underlying grid (x-/y-components). In the latter case, all realizations of the surface represent the same original function (in contrast to adding roughness). But even in that case, the outcome of the simulation may vary greatly. This does not necessarily imply that in real landslide simulations the solutions will differ that much, instead, the volume and weight of real blocks compared to single points will act as a stabilizer, on the other hand, the results clearly show that the problem in general is unstable and thus, results have to be taken with care. The simulations should later be repeated with the full program, including several blocks and interaction between blocks.

Finally it was discussed how the results can be applied to real world landslide problems. Future work could now be dedicated to the integration of the new method in the full landslide model, i.e. including multiple blocks as well as block-block-interaction.

Appendix A

Implementation

A.1 Deciding if a point is located inside a specific triangle

In this section we will discuss how one can decide if a point P is located inside a specific triangle T or not. We will discuss here only the problem in a planar geometry and therefore a triangle in the following is always a planar one. There are different methods to solve this problem, however, using barycentric coordinates (respectively convex-coordinates) is among the most efficient (in terms of computational effort) techniques so far.
Each triangle in $\mathbb{R}^3$ (as well as in $\mathbb{R}^2$) can uniquely be defined by three vertices which in the following are denoted by $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$. As we are working in $\mathbb{R}^3$ each of these vertices is defined by three coordinates, meaning: $\mathbf{v}_i = \mathbf{v}_i(x,y,z),\ \ i = 1,2,3$. Let now S be the point set containing exactly these three vertices:

$$S = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\},$$

then the convex hull of S is forming exactly the triangle T:

$$T = \mathrm{Conv}(S).$$

Remark 1. The convex hull $\mathrm{Conv}(S)$ of a point set S is the smallest convex set of points containing S.

We know that a point $\mathbf{v}'$ lying in the plane that is span by the point set S can be expressed as a linear combination of three points $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$:

$$\mathbf{v}' = \lambda_1 \mathbf{v}_1 + \lambda_3 \mathbf{v}_3 + \lambda_3 \mathbf{v}_3 \tag{A.1}$$

We now want to know, if that point is located inside a certain triangle that is defined by those three points $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ or not. We say that a point is located inside a

triangle T if it lies either on the boundary of T or is contained in its interior. Otherwise we say that it lies outside. This problem is equivalent to determining if a point is contained in the convex hull of a set of three points.
By definition, a point $\mathbf{v}'$ is contained in the convex hull $\mathrm{Conv}(S)$ if it can be expressed as convex-combination of the elements of the point set S by which this convex hull is defined:

$$\mathbf{v}' \in S \;\Leftrightarrow\; \mathbf{v}' = \sum_{i=1}^{n} \mathbf{v}_i \alpha_i, \;\wedge\; \mathbf{v}_i \in S, n \in \mathbb{N}, \sum_{i=1}^{n} \alpha_i = 1, \alpha_i \geq 0.$$

Expressed via equation ((A.1)) this means that:

$$\mathbf{v}' \in T \Leftrightarrow \begin{cases} \lambda_1,\ \lambda_2,\ \lambda_3 \geq 0 \\ \lambda_1 + \lambda_2 + \lambda_3 = 1 \end{cases}$$

If exactly one of the λ_i is *zero*, the point is located on an edge of the triangle, if two are equal to *zero*, meaning that one of them must be *one*, the point is located on a vertex of the triangle. If at least one of the conditions is not fulfilled, the point is not located inside the triangle.

We will now determine the equations that need to be solved within the landslide-code. Assume that we are given the tree vertices $\mathbf{v}_i = \mathbf{v}_i(x, y, z),\ i = 1, 2, 3$ defining the triangle T and a point P whose position is specified by $\mathbf{v}' = \mathbf{v}'(x, y, z)$. Then the point $\mathbf{v}'$ can be expressed as a linear combination of the $\mathbf{v}_i$ as follows:

$$\mathbf{v}' = \lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2 + \lambda_3 \mathbf{v}_3.$$

This can be written component-wise as:

$$x' = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3, \tag{A.2a}$$

$$y' = \lambda_1 y_1 + \lambda_2 y_2 + \lambda_3 y_3,, \tag{A.2b}$$

$$z' = \lambda_1 z_1 + \lambda_2 z_2 + \lambda_3 z_3,. \tag{A.2c}$$

Moreover, from (A.1) it follows that for a point contained in the convex hull of a triangle the following must hold:

$$\lambda_3 = 1 - \lambda_1 - \lambda_2.$$

These are four conditions for three invariants but we can, without restrictions, reduce this problem to two dimensions by simply projecting all the point to the (x, y)-plane and then abandon (A.2c). This works as, in contrast to a rotation, a projection is invariant in the x- and y-components. Therefore it is enough to know whether the

projection of the point onto the (x,y)-plane is contained in the convex hull of the three vertices projected onto the same plane.
The remaining equations (A.2a) and (A.2b) of system (A.2) then become:

$$\begin{aligned}
x' &= \lambda_1 x_1 + \lambda_2 x_2 + (1 - \lambda_1 - \lambda_2) x_3 \\
&= \lambda_1 (x_1 - x_3) + \lambda_2 (x_2 - x_3) - x_3, \\
y' &= \lambda_1 y_1 + \lambda_2 y_2 + (1 - \lambda_1 - \lambda_2) y_3 \\
&= \lambda_1 (y_1 - y_3) + \lambda_2 (y_2 - y_3) - y_3,
\end{aligned}$$

which can be written as

$$\begin{pmatrix} (x_1 - x_3) & (x_2 - x_3) \\ (y_1 - y_3) & (y_2 - y_3) \end{pmatrix} \cdot \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} (x' - x_3) \\ (y' - y_3) \end{pmatrix} \tag{A.4}$$

Solving the system given in ((A.4)) gives the two parameters λ_1 and λ_2 while λ_3 follows directly from (5). If now all the three parameters are positive, the point is located inside the triangle, respectively on one its edges or nodes as it was already described above.
Moreover, by enumerating the edges and vertices of the triangle in a certain way, we immediately find the number of the node or vertex on which the point is located if necessary. The enumeration is therefore done counter-clockwise and the edges are enumerated in such a way that their indices correspond to the indices of the nodes that are opposite to them. This can be seen in figure (A.1).

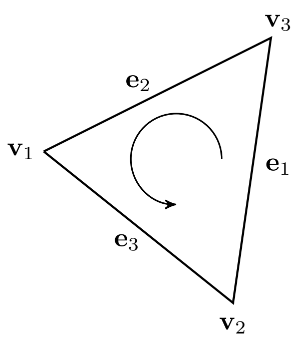


FIGURE A.1: Example of a triangle with the enumeration of its edges and nodes.

Remember that the linear-combination to express the point P was written as $\mathbf{v}' = \lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2 + \lambda_3 \mathbf{v}_3$. Thanks to our enumeration we immediately now from the parameters λ_i on which edge or node this point is located if so. In case that exactly one parameter is *zero*, let's say λ_3 and the other two are positive, then the point must be located on the edge e_3 excluding its two vertices, because the node $\mathbf{v}_1$ and the other two parameters are in the range $(0,1)$. If instead two of the parameters, e.g. λ_2 and λ_3, are *zero*, then the point is located on a vertex. In this case it would be $\mathbf{v}_1$. This can easily be shown:

Proof. Assume that $\lambda_3 = 0$ and $\lambda_1, \lambda_2 \in [0,1]$. Knowing that $\lambda_1 + \lambda_2 + \lambda_3 = 1$ it follows that $\lambda_2 = 1 - \lambda_1$. Therefore the equation for a point $\mathbf{v}'$ reduces to:

$$\mathbf{v}' = \lambda_1 \mathbf{v}_1 + \lambda_2 \mathbf{v}_2,$$

and we find:

$$\begin{aligned} \mathbf{v}' &= \lambda_1 \mathbf{v}_1 + (1 - \lambda_1)\mathbf{v}_2 \\ \Leftrightarrow \mathbf{v}' &= \lambda_1 \mathbf{v}_1 + \mathbf{v}_2 - \lambda_1 \mathbf{v}_2 \\ \Leftrightarrow \mathbf{v}' &= \mathbf{v}_2 + \lambda_1(\mathbf{v}_1 - \mathbf{v}_2) \end{aligned}$$

Equation (A.1) is a usual formulation for a line in the space. As $\lambda_1 \in [0,1]$ the point $\mathbf{v}'$ is always on the line bounded by $\mathbf{v}_1$ and $\mathbf{v}_2$. We can now distinguish two cases: if both the remaining parameters λ_1 and λ_2 are positive, the point $\mathbf{v}'$ is located on the interior of the line which means that it can never coincide with one of its vertices. As the line defined by the vertices $\mathbf{v}_1$ and $\mathbf{v}_2$ was defined as e_3 we find that for $\lambda_3 = 0$ the point $\mathbf{v}'$ is located on edge e_3 or in general: Iff only the parameter λ_i equals *zero* then the point is located on edge e_i, $i = 1,2,3$.
If instead one of the two parameters equals *one* then the point coincides with one of the two vertices. E.g. for $\lambda_2 = 0$ we find:

$$\mathbf{v}' = \mathbf{v}_2 + 1(\mathbf{v}_1 - \mathbf{v}_2) = \mathbf{v}_1.$$

Or in general: Iff the parameter λ_i equals *one* then the point is located on vertex $\mathbf{v}_i$, $i = 1,2,3$. □

A.2 Deciding if a point without initial velocity can move into a triangle or not

Now we will discuss the trajectory of a point that at initial time is located either on a common edge of two triangles or in a node that is shared by an arbitrary (but finite) number of triangles.
For each triangle we know the acceleration (due to gravity, in the following denoted by $\mathbf{a}_G$) acting on a point located in the triangle, the coordinates of its vertices $\mathbf{v}_i$, $i = 1,2,3$, the vector normal to the triangle T, pointing into the surface ($\mathbf{n}(T)$) and the

vectors normal to its edges e_i, lying in the plane defined by the triangle T and pointing into it ($\mathbf{n}_E^i(T)$). Here, the subscript E indicates that it is a normal to an edge, while the superscript i indicates the number of the edge. The variable T always denotes a particular triangle. If we are talking about more than one triangle, they are indicated by T_j, where j is the number of the triangle. In accordance, the acceleration $\mathbf{a}_G$ of triangle T_j is indicated by $\mathbf{a}_G(T_j)$.
For each edge we know its two vertices and triangles while for each node we know all its adjacent edges and triangles.

A.2.1 A point located on an edge

In case of a point located on an edge but not coinciding with one of its vertices we basically have to distinguish three cases. The two triangles adjacent to the edge can either form a valley, or a ridge or a monotonous increasing/decreasing surface. In figure (A.2) the last case is illustrated.
To decide whether a point can move from its initial position into a certain triangle or

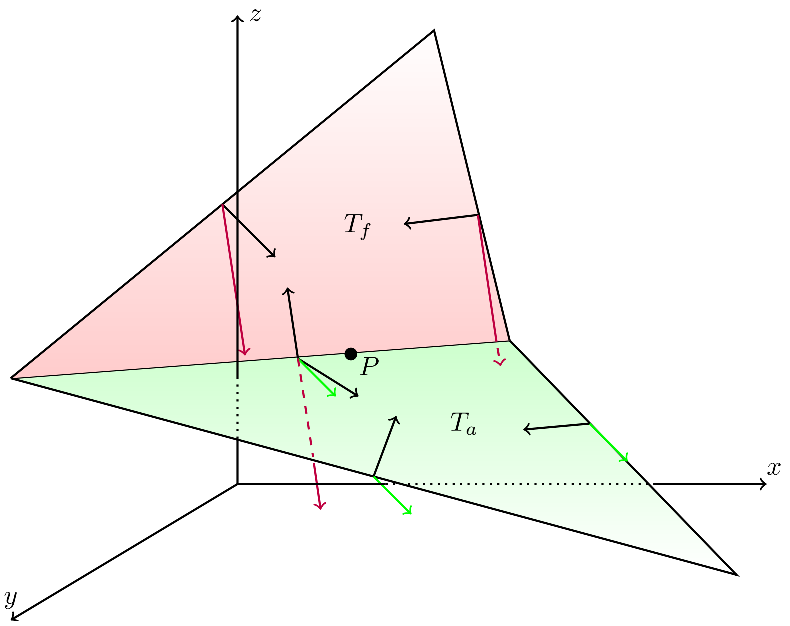


FIGURE A.2: Possible situation of two neighbouring triangles with different inclination in a 2-dimensional surface-triangulation. The inward directed normal-vectors to the edges are drawn in black, and the vectors indicating the acceleration (due to gravity) belonging to a triangle are drawn in the corresponding colours (red for the upper and green for the lower triangle). The initial position of the point is indicated by P. The triangle into which P is allowed to move is indicated by $T_{a(llowed)}$, the triangle in which it cannot move is indicated by $T_{f(orbidden)}$.

not, we will use the acceleration a_G and the normal vector of the edge. Before doing that let's define what it means that a vector is pointing into a triangle.

Definition 2. Given a point $\mathbf{p}$ that is located on the boundary of a triangle T, we say that a vector $\mathbf{v}$ is pointing into a triangle T if $\exists\ \varepsilon > 0$ such that

$$\mathbf{p}_{+} = \mathbf{p} + \varepsilon\mathbf{v}$$

is contained in the convex hull of T.

In addition we will define the positive and negative half-plane of a given plane H.

Definition 3. Given a plane H and an edge e that is defining an infinite strait line in that plane then this line is dividing the plane into two half-planes. Let now $\mathbf{p}$ be a point on the line and $\mathbf{v}$ be a vector in the plane H. Then we will denote by $H_e^{+}(\mathbf{v})$ (positive half-plane defined by e with respect to $\mathbf{v}$) the closed half plane for which the following condition is fulfilled $\forall\ t \in \mathbb{R}^{+}$:

$$\mathbf{p}_{+} = \mathbf{p} + t\mathbf{v}$$

By $H^{-}(\mathbf{v})$ (negative half-plane defined by e with respect to $\mathbf{v}$) instead we will define the complement $H \backslash H^{+}(\mathbf{v})$.
In other words: the positive half-plane with respect to $\mathbf{v}$ is that half plane into which the vector $\mathbf{v}$ points while the negative half plane is that one out of which it is pointing.

Thus we find the following result:

Proposition 1. *Let T be a (planar) triangle and let e denote one of its edges where $\mathbf{n}_E^i$ is the normal vector to this edge pointing into the triangle. By $H_e^{+} = H_e^{+}(\mathbf{n}_E^i)$ we denote the positive half-plane with respect to the normal vector of e. Then the following is true:*

- *Also the triangle T is contained in H_e^{+}*
- *Every vector $\mathbf{v}'$ for which $\alpha = \sphericalangle(\mathbf{v}', \mathbf{n}_E^i) \leq 90^\circ$ is pointing into H_e^{+}. Or, in other words: $\mathbf{v}'$ is pointing into H_e^{+} if*

$$\mathbf{v}' \cdot \mathbf{n}_E^i \geq 0.$$

- *And in particular: Every vector that is pointing into H_e^{+} is also pointing into the triangle T.*

***Note** that all these results are always true only with respect to one particular edge of the triangle.*

Following these results we can decide very easily if a point P that is located on the interior of an edge e that belongs to a triangle T can move into that triangle only due to gravity or not.
If the only force acting on a point is gravity, then this point can only move downwards (in the direction of the steepest gradient) or stay in its position (in absence of any slope). This force is given by the acceleration $\mathbf{a}_G$. The point P located on an edge e_i can move into a triangle T if $\mathbf{a}_G(T)$ is pointing into T with respect to that edge e_i. In other words: P can move into T from edge e_i if

$$\mathbf{a}_G(T) \cdot \mathbf{n}_E^i(T) \geq 0.$$

Note that this criterion is only a necessary one. It is not sufficient to decide about the direction of motion of the point. This also depends on the configuration of the second triangle adjacent to the edge e^i.
In order to investigate the different possible scenarios of the point's motion we will now compute the dot-product described above for the two triangles (denoted by T_1 and T_2) that are adjacent to one particular edge e. The normal vectors of e pointing into T_1 and T_2 are denoted by $\mathbf{n}_E^{T1}$ and $\mathbf{n}_E^{T2}$ while the accelerations are denoted by $\mathbf{a}_G^{T1}$ and $\mathbf{a}_G^{T2}$. The following three cases are possible:

Case 1: $(\mathbf{n}_E(T1) \cdot \mathbf{a}_G(T1)) \geq 0 \wedge (\mathbf{n}_E(T2) \cdot \mathbf{a}_G(T2)) \leq 0$ (and vice versa)
$\Rightarrow$ The two triangles are building a surface with monotone slope. The point will move into triangle T_1 (and vice versa).

Case 2: $(\mathbf{n}_E(T1) \cdot \mathbf{a}_G(T1)) \geq 0 \wedge (\mathbf{n}_E(T2) \cdot \mathbf{a}_G(T2)) \geq 0$
$\Rightarrow$ The two triangles are building a ridge. The point is in unstable equilibrium and will by definition move into the triangle with steeper slope. If the slopes are equal the triangle is chosen randomly. If both triangles have *zero* slope the point will not move.

Case 3: $(\mathbf{n}_E(T1) \cdot \mathbf{a}_G(T1)) \leq 0 \wedge (\mathbf{n}_E(T2) \cdot \mathbf{a}_G(T2)) \leq 0$
$\Rightarrow$ The two triangles are building a valley. The point will move downwards along the edge if it is inclined and does not move if the edge is horizontal.

A.2.2 A point located on a node

If instead the point P is located on one of the three nodes of the triangle, many more scenarios are possible. Moreover, the above criterion to determine the direction of motion of a point is not longer applicable as a node is not providing enough information to uniquely divide a plane into to half-planes. However, we can use two half-planes to uniquely describe a specific sub-plane that contains a particular triangle and whose boundaries coincide with the two edges shared by a particular node of the triangle.

Definition 4. Let T be a planar triangle, n be one particular node of T and e_1 and e_2 the two edges of T adjacent to n. By $\mathbf{n}_E^1$ and $\mathbf{n}_E^2$ we again denote the normal vectors of e_1 and e_2. Let H be the plane in which T is lying. Then we call the intersection of $H_{e_1}^+(\mathbf{n}_E^1)$ and $H_{e_1}^+(\mathbf{n}_E^2)$ the positive sub-plane $S_T^+(n)$ of T with respect to n:

$$S_T^+(n) = H_{e_1}^+(\mathbf{n}_E^1) \bigcap H_{e_1}^+(\mathbf{n}_E^2).$$

The complement of $S_T^+(n)$ is then called the negative sub-plane of T with respect to n:

$$S_T^-(n) = H \backslash S_T^+(n).$$

The above defined sub-plane is also shown in figure (A.3).

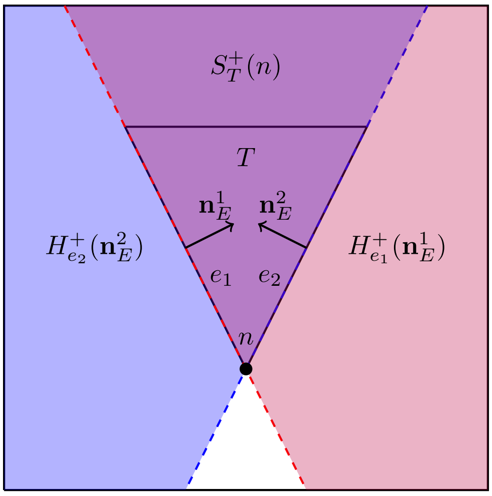


FIGURE A.3

This allows us to formulate a necessary and sufficient condition for a vector starting from n to point into $S_T^+(n)$.

Proposition 2. *Let T be a triangle, n be one particular node of T and e_1 and e_2 the two adjacent edges to n. The two normal vectors to e_1 and e_2 are denoted by $\mathbf{n}_E^1$ and $\mathbf{n}_E^2$. Then the following holds:*
A vector $\mathbf{v}$ is pointing from n into $S_T^+(n)$ if and only if it is pointing into $H_{e_1}^+(\mathbf{n}_E^1)$ and $H_{e_1}^+(\mathbf{n}_E^2)$. In other words: a vector starting from node n is pointing into $S_T^+(n)$ iff

$$\mathbf{v} \cdot \mathbf{n}_E^1 \geq 0 \wedge \mathbf{v} \cdot \mathbf{n}_E^2 \geq 0. \tag{A.5}$$

If for a triangle T condition ((A.5)) is fulfilled, where $\mathbf{v} = \mathbf{a}_G(T)$, is called an *allowed* triangle, otherwise it is called $forbidden$. Among all the allowed triangles we now need to find the triangle into which P will actually move.

Case 1: There is no allowed triangle: the point P is in a valley in a stable equilibrium and will not move.

Case 2: There is exactly one allowed triangle: the way of motion is defined uniquely, meaning that the point will start moving into that triangle.

Case 3: There is more than one allowed triangle: the point will start moving into the triangle with steepest gradient. If the maximum is shared by two or more triangles the point starts moving into one of them randomly.

Case 4: If none of the triangles has any slope (all triangles are parallel to the (x,y)-plane), the point will not move.

Appendix B

Quaternions

First described by W. R. Hamilton in 1843, quaternions are widely used e.g. in three-dimensional computer graphics and allow a simple formulation of rotations, especially avoiding the so called gimbal lock.
A quaternion is composed of a real part and a complex part where the latter has three components, written in the style of complex numbers using i, j and k:

$$x = x_0 + x_1 i + x_2 j + x_3 k,$$

where x_0, x_1, x_2 and x_3 are real numbers and $1,i,j,k$ is a standard basis over $\mathbb{R}$. Real and complex part of x can also be written as:

$$\begin{aligned}\Re\, x &= x_0,\\ \Im\, x &= x_1 i + x_2 j + x_3 k.\end{aligned}$$

Analogous to the complex numbers, generally every (non-real) quaternion x can be written in polar coordinates:

$$x = |x|(\cos\phi + \varepsilon \sin\phi)$$

with a polar angle $\phi = \arccos\left(\frac{\Re x}{|x|}\right) \in]0,\pi[$ and $\varepsilon = \frac{\Im x}{|x| \sin\phi}$, where ε is called a pure unit quaternion. For unit quaternions this reduces to $\phi = \arccos(\Re x)$ and $\varepsilon = \frac{\Im x}{\sin\phi}$.
Let now q be unit quaternion where $q \neq \pm 1$, then every rotation $\mathbb{R}^3$ can be expressed in polar form via an angle $0 < \alpha < 2\pi$ and a pure unit quaternion ε:

$$q = \cos\frac{\alpha}{2} + \varepsilon \sin\frac{\alpha}{2},$$

where α is the axis of rotation and $\varepsilon \in \mathbb{R}^3$ is the axis of rotation.

Bibliography

Abadie, S. et al. (2010). “Numerical simulation of waves generated by landslides using a multiplefluid Navier-Stokes model”. In: *Coast. Eng.* 57(9), pp. 779–794.

Billings, S.D., R.K. Beatson, and G.N. Newsam (2002a). “Interpolation of geophysical data using continuous global surfaces”. In: *GEOPHYSICS* 67, pp. 1810–1822.

— (2002b). “Smooth fitting of geophysical data using continuous global surfaces”. In: *GEOPHYSICS* 67, pp. 1823–1834.

Bortolucci, Elisabetta (2001). “Modelli dinamici di frane e dei maremoti indotti”. PhD thesis. University of Bologna.

Bouchut, F. et al. (2014). “A two-phase shallow debris flow model with energy balance”. In: *Math. Modell. Numer. Anal.* 49(1), pp. 101–140.

Bouchut F.and Westdickenberg, M. (2004). “Gravity driven shallow water models for arbitrary topography”. In: *Commun. Math. Sci.* 2(3), pp. 359–389.

Cheng, Y. P. et al. (2003). “Distinct element simulation of crushable soil”. In: *Geotechnique* 53, pp. 633–641.

Cruden, D.M. and D.J Varnes (1996). “Landslide Types and Processes”. In: *Special Report, Transportation Research Board, National Academy of Sciences* 247, pp. 36–75.

Cundall, P. A. and O. D. L. Strack (1979). “A discrete model for granular assemblies”. In: *Geotechnique* 29, pp. 47–65.

Highland, L.M. and P. Bobrowsky (2008). *The landslide handbook - A guide to understanding landslides*. Tech. rep. Reston, Virginia, U.S. Geological Survey.

Hindmarsh, A.C. and K. Radhakrishnan (1993). *Description and Use of LSODE, the Livermore Solver for Ordinary Differential Equations*. Tech. rep. Nasa Reference Publication, 1327.

Hungr, O. and S. McDougall (2009). “Two numerical models for landslide dynamic analysis”. In: *Comput. Geosci.* 35, pp. 978–992.

Hungr, O. et al. (2001). “A Review of the Classification of Landslides of the Flow Type”. In: *Environmental and Engineering Geosciences* VII, p. 221.238.

Lucas, A. et al. (2011). “Influence of the scar geometry on landslide dynamics and deposits: Application to Martian landslides”. In: *J. Geophys. Res.* 116.

Marsaglia, G. and W. W. Tsang (2007). “The Ziggurat Method for Generating Random Variables”. In: *Journal of Statistical Software* 5.

O'Sullivan, C. (2011). "Particle-Based Discrete Element Modeling: Geomechanics Perspective". In: *Int. J. Geomechan.* 11, pp. 449–464.

Pastor, M. et al. (2007). "A SPH depth integrated model with pore pressure coupling for fast landslides and related phenomena". In: *Ho, K., Li, L. (Eds.), 2007 International Forum on Landslides Disaster Management*, pp. 987–1014.

Pirulli, M. and G. Sorbino (2008). "Assessing potential debris flow runout: a comparison of two simulation models". In: *Nat. Hazards Earth Syst. Sci.* 8, pp. 961–971.

Pudasaini, S.P. (2012). "A general two-phase debris flow model". In: *J. Geophys. Res.* 117.

Tinti, S, E Bortolucci, and A Armigliato (1999). "Numerical simulation of the landslide-induced tsunami of 1988 on Vulcano Island, Italy". In: *Bulletin of Vulcanology* 61, pp. 121–137.

Tinti, S, E Bortolucci, and C Vannini (1997). "A Block-Based Theoretical Model Suited to Gravitational Sliding". In: *Natural Hazards* 16, pp. 1–28.

USGS (2016). *Landslide Classification*. http://pubs.usgs.gov/fs/2004/3072/fs-2004-3072.html.

Varnes, D. J. (1978). *Slope movement types and processes*. Tech. rep.

Warren, J et al. (2004). *Barycentric coordinates for convex sets*. Tech. rep. Advances in Computational and Applied Mathematics.