



Full length article

A versatile smoothed particle hydrodynamics code for graphic cards[☆]C.M. Schäfer^{a,*}, O.J. Wandel^a, C. Burger^{a,b}, T.I. Maindl^b, U. Malamud^c, S.K. Buruchenko^d, R. Sfai^e, H. Audiffren^{a,f}, E. Vavilina^a, P.M. Winter^g^a Institut für Astronomie und Astrophysik, Auf der Morgenstelle 10, 72076 Tübingen, Germany^b Department of Astrophysics, University of Vienna, Tuerkenschanzstr. 17, 1180 Vienna, Austria^c Physics Department, Technion - Israel Institute of Technology, Haifa, 3200002, Israel^d South Ural State University, 456776 Zababachina 9-115 Snezhinsk, Chelyabinsk Region, Russia^e Sao Paulo State University, Guaratingueta, Sao Paulo, CEP 01049-010, Brazil^f Aix-Marseille University, 3 Place Victor Hugo, 13003 Marseille, France^g Institute for Machine Learning, Johannes Kepler University Linz, Altenberger Straße 69, 4040 Linz, Austria

ARTICLE INFO

Article history:

Received 14 February 2020

Accepted 10 August 2020

Available online 19 August 2020

Keywords:

Smoothed particle hydrodynamics

GPU-computing

Hydrodynamics

Continuum mechanics

ABSTRACT

We present the second release of the now open source smoothed particle hydrodynamics code *miluphcuda*. The code is designed to run on Nvidia CUDA capable devices. It handles one to three dimensional problems and includes modules to solve the equations for viscous and inviscid hydrodynamical flows, the equations of continuum mechanics using SPH, and self-gravity with a Barnes–Hut tree. The covered material models include different porosity and plasticity models. Several equations of states, especially for impact physics, are implemented. The basic ideas of the numerical scheme are presented, the usage of the code is explained and its versatility is shown by means of different applications. The code is hereby publicly available.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

The Lagrangian meshfree particle method smoothed particle hydrodynamics (SPH) was invented by Lucy (1977) and Gingold and Monaghan (1977) to solve the hydrodynamic equations for compressible flows in astrophysical applications. During the last 40 years, it has been widely used in astrophysics to tackle a substantial number of problems. In the last two decades the field of applications have expanded vividly: SPH has been established as one of the major concepts for fluid animation and rigid–fluid coupling in computer graphics (Ihmsen et al., 2014; Akinci et al., 2012), it was extended to continuum mechanics to model high-velocity impacts (Libersky and Petschek, 1991; Stellingwerf and Wingate, 1994), it is widely used in geosciences to model landslides and granular media (Yu et al., 2018; Bui et al., 2008), and the method has been improved and extended substantially to model phase separating fluid mixtures (Zander et al., 2018), to solve the magnetohydrodynamical equations (Price, 2012) and to model relativistic flows (Rosswog, 2015). It has also been used to solve the non-linear Schrödinger equation (Mocz and Succi, 2015). The SPH scheme has proven to be an advantageous method to solve all kind of different partial differential equations (PDEs).

Since the most astrophysical applications have to include gravity as the most dominating force, gravitational interactions had been implemented into the SPH scheme from the early development on. Moreover, not only external gravitational sources but also the self-gravity of the simulated flow or solid body can be considered. Especially tree algorithms for the treatment of the long range gravitational force have been combined with SPH (Hernquist and Katz, 1989).

Several SPH codes have been developed in the recent years and we present in the following the codes that have been published under the GNU General Public License. The list is most certainly incomplete and we want to apologize any omitted code. The first popular code GADGET-2 by Springel (2000) is a massively parallel TreeSPH code, which is still widely used in the astrophysical community. A more recent and modern code from the astrophysical community is the PHANTOM code by Price et al. (2017). It is a parallel, modular and low-memory smoothed particle hydrodynamics and magnetohydrodynamics code with focus on stellar, galactic, planetary and high energy astrophysics. The second version of the GASOLINE code, GASOLINE2, has also been published under the GPLv2 license (Wadsley et al., 2017). The code concentrates also on astrophysical problems with a special focus on cosmology. Another versatile SPH code written in C++ is SPHERAL++ by Owen (2001). Since the field of applications of the scheme has been broadened thoroughly in the recent decade, modern and versatile codes outside of the astrophysical community have emerged. PySPH is reproducible and high-performance

[☆] This code is registered at the ASCL with the code entry ascl:1911.023.^{*} Corresponding author.E-mail address: ch.schaefer@uni-tuebingen.de (C.M. Schäfer).

framework for smoothed particle hydrodynamics is developed by Ramachandran (2016). It is implemented in a mix of Python and Cython. The user writes pure Python code and the high-performance computing (HPC) code is generated automatically. PySPH includes solver for fluid and solid mechanics and supports both OpenMP and MPI. Another code to study free-surface flows is DualSPHysics (Crespo et al., 2015), which is the very first SPH framework featuring a modern design and GUI interface (DesignSPHysics). AQUAplusph (Cercos-Pita, 2015) is a 3D, OpenCL accelerated SPH solver for GPUs. GPUSPH by Hérault et al. (2010), which has been originally a port of the freely available SPHysics Fortran code to Nvidia's CUDA library, is now a standalone project and supports multiple GPUs. PersianSPH is an open-source code and features elastic-plastic soil behaviour and elastic-plastic solid behaviour (Korzani et al., 2017). SPHERA (Amicarelli et al., 2017) is also publicly available and supports the simulation of flooding including the transport of solid bodies, bed-load transport, damage on electrical substations and fast landslides in rocks and solids. Another versatile code within the astrocommunity is SWIFT by Schaller et al. (2018). Recent simulations with SWIFT included up to 100 million SPH particles (Kegerreis et al., 2019). It can directly read GADGET-2 output files and shows tremendous scaling.

Here, we present the updated version of our GPU SPH code miluphcuda.¹ The new version is published under the GPLv3 license and available on github via <https://github.com/christophmschaefer/miluphcuda>. Originally, the code has been developed to model collisions between solid, self-gravitating objects in an astrophysical context. The basic implementation is described in Schäfer et al. (2016), and until today, the code is solely applied in the astronomy community. The code handles one-, two- and three-dimensional problems. It has been used to study the activation of porous and non-porous asteroids (Haghighipour et al., 2018, 2016), collisions between Earth and its past moons (Malamud et al., 2018), collisional formation of moons (Malamud et al., 2020), impact cratering (Wandel et al., 2017), the formation of Arrokoth (Grishin et al., 2020), transfer and loss of water in hit-and-run collisions (Burger and Schäfer, 2017; Burger et al., 2018) and sampling of granular material (Schäfer et al., 2017) among other applications. Recent extensions include further porosity models, a module to solve the Navier–Stokes equations, and more sophisticated equation of states like ANEOS.

The outline of this paper is as follows. The next section summarizes the characteristics of the code as presented in Schäfer et al. (2016) and provides a comprehensive description of the new extensions. Section 3 shows some of the recent applications and numerical tests to provide an overview of the possibilities of the code. We will conclude in the final section and present details about the code and its structure in the appendix.

2. Physical and numerical method

In this section, we present the different physical models, their numerical implementations, and further aspects of miluphcuda.

The basic idea of the SPH scheme is to transform a system of partial differential equations to a system of ordinary differential equations, which can be solved by standard integrators. For this transformation the partial differential equations are approximated in two steps: First, with an average over all spatial field quantities by the help of the convolution with the kernel function and second by a discretization of this average. For an introduction to the fundamental ideas of SPH, we refer to the comprehensive review articles by Benz (1990), Monaghan (1992), Monaghan

(2005), Liu and Liu (2010) and Monaghan (2012). In the following, we present the SPH equations for viscid and inviscid hydrodynamics and continuum mechanics as implemented in the code. The different modules implemented in the code allow for the simulation of various different astrophysical objects: By the use of the sole hydrodynamical module, one can model a gaseous accretion disc, both viscous and inviscid. By adding the self-gravity module, the gravitational field produced by the disc is also included. The solid module allows to model brittle, rocky materials, such as smaller, non-porous planetesimals. Adding the porosity module, the code can be used to simulate porous objects like porous asteroids, comets and Kuiper belt objects. Larger planetesimals or dwarf planets can be simulated with a combination of the solid or hydro module with the self-gravity module. The different strength models included in the solid module allow to model brittle or granular materials, e.g., a layer of granular regolith on a brittle and porous asteroid.

2.1. Hydrodynamics

The flow of a fluid is completely described by its velocity field $\mathbf{v}(\mathbf{x}, t)$, and two thermodynamic field quantities, where usually the density $\varrho(\mathbf{x}, t)$ and the pressure $p(\mathbf{x}, t)$ are chosen. The conservation laws for mass, momentum and energy determine the dynamics and lead to a set of PDEs which can be solved with SPH for all locations \mathbf{x} and times t .

The Euler equation describes the flow of an inviscid, ideal fluid. It reads in Lagrangian representation

$$\varrho \frac{d\mathbf{v}}{dt} = -\nabla p, \quad (1)$$

with the velocity \mathbf{v} , the density ϱ and the pressure p . By the use of the identity

$$\frac{1}{\varrho} \nabla p = \frac{1}{\varrho^{2-\lambda}} \nabla \left(\frac{p}{\varrho^{\lambda-1}} \right) + \frac{p}{\varrho^{\lambda}} \nabla \frac{1}{\varrho^{1-\lambda}}, \quad (2)$$

one can derive the following two equivalent SPH representations,

$$\frac{d\mathbf{v}_a}{dt} = - \sum_b m_b \frac{p_a + p_b}{\varrho_a \varrho_b} \nabla W_{ab} \quad (3)$$

for $\lambda = 1$, and

$$\frac{d\mathbf{v}_a}{dt} = - \sum_b m_b \left[\frac{p_a}{\varrho_a^2} + \frac{p_b}{\varrho_b^2} \right] \nabla W_{ab} \quad (4)$$

for $\lambda = 2$.

2.2. Viscous flows

The extension of the Euler equation for viscous, Newtonian fluids is the Navier–Stokes equation. It reads in Lagrangian representation

$$\varrho \frac{d\mathbf{v}}{dt} = -\nabla p + \nabla \cdot \mathbf{T}, \quad (5)$$

where \mathbf{T} denotes the viscous stress tensor. The viscous stress tensor accounts for the internal friction of the flow. For a Newtonian fluid, where the viscosity depends linearly on the gradient of the velocity field, it is given by

$$T^{\alpha\beta} = \eta \left[\frac{\partial v^\alpha}{\partial x^\beta} + \frac{\partial v^\beta}{\partial x^\alpha} - \frac{2}{3} \delta^{\alpha\beta} \frac{\partial v^\gamma}{\partial x^\gamma} \right] + \zeta \delta^{\alpha\beta} \frac{\partial v^\gamma}{\partial x^\gamma} \quad (6)$$

with the dynamic viscosity coefficient η and the bulk viscosity coefficient ζ . The kinematic viscosity coefficient ν is the ratio between the dynamic viscosity coefficient and the density of the fluid, $\nu = \eta/\varrho$. The term in the parentheses in Eq. (6)

¹ miluph is pronounced [maɪlʌv]

represents the traceless part of the viscous stress tensor and is called viscous shear stress tensor. The numerical treatment and solution of the Navier–Stokes equation is demanding because of the second derivatives of the velocities. We follow the approach by [Flebbe et al. \(1994\)](#) and calculate the derivatives of the velocity components for particle a according to

$$\frac{\partial v_a^\alpha}{\partial x^\beta} = \sum_b \frac{m_b}{Q_b} (v_b^\alpha - v_a^\alpha) \frac{\partial W_{ab}}{\partial x^\beta} \equiv v_a^{\alpha\beta}. \quad (7)$$

The viscous stress tensor for particle a can be calculated via

$$T_a^{\alpha\beta} = \eta_a [v_a^{\alpha\beta} + v_a^{\beta\alpha}] - \frac{2}{3} \delta^{\alpha\beta} v_a^{\gamma\gamma}. \quad (8)$$

In a second step, the accelerations due to the viscous stress are determined via

$$\frac{dv_a^\alpha}{dt} = \sum_b m_b \frac{T_b^{\alpha\beta} + T_b^{\beta\alpha}}{Q_a Q_b} \frac{\partial W_{ab}}{\partial x^\beta}. \quad (9)$$

If the partial derivatives of the velocities are not calculated with Eq. (7), but with the numerical equivalent SPH sum

$$\frac{\partial v_a^\alpha}{\partial x^\beta} = \frac{1}{Q_a} \sum_b m_b (v_b^\alpha - v_a^\alpha) \frac{\partial W_{ab}}{\partial x^\beta}, \quad (10)$$

the accelerations are determined via

$$\frac{dv_a^\alpha}{dt} = \sum_b m_b \left[\frac{T_b^{\alpha\beta}}{Q_a^2} + \frac{T_b^{\beta\alpha}}{Q_b^2} \right] \frac{\partial W_{ab}}{\partial x^\beta}. \quad (11)$$

The bulk part of the Navier–Stokes equation can also be used to define an artificial viscosity that prevents mutual particle penetration and does not yield unwanted spurious shear. A possible ansatz for an artificial bulk viscosity was successfully applied in planet–disc interaction simulations in [Schäfer et al. \(2004\)](#): the bulk part of the viscous stress tensor is $\zeta \nabla \cdot \mathbf{v}$, which leads to an additional acceleration for particle a

$$\left. \frac{d\mathbf{v}_a}{dt} \right|_{\text{bulk}} = \sum_b m_b \zeta_{ab} \frac{(\nabla \cdot \mathbf{v})_a + (\nabla \cdot \mathbf{v})_b}{Q_a Q_b} \nabla_a W_{ab}, \quad (12)$$

or

$$\left. \frac{d\mathbf{v}_a}{dt} \right|_{\text{bulk}} = \sum_b m_b \zeta_{ab} \left[\frac{(\nabla \cdot \mathbf{v})_a}{Q_a^2} + \frac{(\nabla \cdot \mathbf{v})_b}{Q_b^2} \right] \nabla_a W_{ab}, \quad (13)$$

in case of the other SPH representation, respectively. The bulk viscosity coefficient ζ_{ab} for the interaction between particles a and b is

$$\zeta_{ab} = \begin{cases} -f \bar{h}_{ab}^2 \overline{(\nabla \cdot \mathbf{v})_{ab}} \bar{Q}_{ab} & \text{for } (\mathbf{v}_a - \mathbf{v}_b) \cdot (\mathbf{x}_a - \mathbf{x}_b) \\ 0 & \text{else.} \end{cases} \quad (14)$$

The term \bar{q}_{ab} denotes an abbreviation for the average $\frac{1}{2}(q_a + q_b)$ for any quantity q and f a parameter to scale the effect of the artificial bulk viscosity, usually set to 1.

2.3. Solid body mechanics

The implementation of solid body mechanics with fragmentation in our code follows the ideas and concepts of [Stellingwerf and Wingate \(1994\)](#) and [Benz and Asphaug \(1994\)](#). We refer for the thorough description of our implementation in [Schäfer et al. \(2016\)](#) and present the basic equations in the following.

For a perfectly elastic material, the conservation of momentum reads

$$Q \frac{dv^\alpha}{dt} = \frac{\partial \sigma^{\alpha\beta}}{\partial x^\beta}, \quad (15)$$

where the stress tensor $\sigma^{\alpha\beta}$ is built up by the pressure p and the traceless deviatoric stress tensor $S^{\alpha\beta}$

$$\sigma^{\alpha\beta} = -p \delta^{\alpha\beta} + S^{\alpha\beta}. \quad (16)$$

The conservation of the internal energy u is given by

$$\frac{du}{dt} = -\frac{p}{Q} \frac{\partial v^\alpha}{\partial x^\alpha} + \frac{1}{Q} S^{\alpha\beta} \dot{\epsilon}^{\alpha\beta}. \quad (17)$$

Here, $\dot{\epsilon}^{\alpha\beta}$ denotes the strain rate tensor for small deformations

$$\dot{\epsilon}^{\alpha\beta} = \frac{1}{2} \left(\frac{\partial v^\alpha}{\partial x^\beta} + \frac{\partial v^\beta}{\partial x^\alpha} \right). \quad (18)$$

Additionally, an equation for the time evolution of the deviatoric stress tensor $S^{\alpha\beta}$ is required to describe the dynamics of the solid body, which is called the constitutive equation. For a perfectly elastic behaviour, we choose Hooke's law, where the stress depends linearly on the strain, which yields using the Jaumann rate eventually

$$\frac{dS^{\alpha\beta}}{dt} = 2\mu \left(\dot{\epsilon}^{\alpha\beta} - \frac{1}{3} \delta^{\alpha\beta} \dot{\epsilon}^{\gamma\gamma} \right) + S^{\alpha\gamma} R^{\gamma\beta} - R^{\alpha\gamma} S^{\gamma\beta}, \quad (19)$$

with the shear modulus μ and the rotation rate tensor $R^{\alpha\beta}$

$$R^{\alpha\beta} = \frac{1}{2} \left(\frac{\partial v^\alpha}{\partial x^\beta} - \frac{\partial v^\beta}{\partial x^\alpha} \right). \quad (20)$$

2.4. Strength models

A solid material does not deform perfectly elastic for all experienced strains. Once the applied strain exceeds the elastic limit, the deformation becomes plastic. The transition from an elastic to a plastic state of a substance is characterized by a stress state. The stress at which the material starts deforming plastically is called yield stress, and the condition for plastic deformation is the yield criterion. For an isotropic medium, the yield criterion is a symmetric function of the principle stresses σ_i

$$f(\sigma_1, \sigma_2, \sigma_3) = \text{const.} \quad (21)$$

The constant on the righthandside is material dependent and related to the particular yield limit. Since f is a symmetric function of the principle stresses, the stress components can be expressed by the invariants of the stress tensor

$$f(I_1, I_2, I_3) = \text{const.} \quad (22)$$

In the original code, we have implemented the elementary von Mises yield strength to model plastic behaviour of solid bodies, and noted already the shortcoming of this simplistic model. Following [Collins et al. \(2004\)](#) and [Jutzi \(2015\)](#), we have added two additional strength models suitable for the modelling of granular material and a pressure dependent yield strength model, which is applicable to rocks and interplays with the fragmentation model for brittle materials by Grady and Kipp ([Schäfer et al., 2016](#)).

2.4.1. Von Mises yield stress

In the von Mises yield stress model ([von Mises, 1913](#)), the deviatoric stress is limited once it reaches a constant yield strength limit Y_M . At first, the second invariant of the stress tensor is computed via

$$J_2 = \frac{1}{2} S^{\alpha\beta} S_{\alpha\beta}, \quad (23)$$

for each individual particle, and the limiting factor f_Y is calculated according to

$$f_Y = \min \left[Y_M^2 / 3J_2, 1 \right]. \quad (24)$$

Then, the deviatoric stress, which was integrated using Eq. (19), is limited

$$S^{\alpha\beta} \rightarrow f_Y S^{\alpha\beta}. \quad (25)$$

2.4.2. Drucker–Prager and Mohr–Coulomb yield criterion

The Drucker–Prager yield strength depends on the pressure and is applicable to model rocky materials. The condition for plasticity reads

$$\sqrt{J_2} = A - BI_1, \quad (26)$$

where I_1 denotes the first invariant of the stress tensor ($I_1 = \text{tr}[\sigma^{\alpha\beta}]$) and J_2 the second invariant given by Eq. (23). The two material constants A and B are the so-called Drucker–Prager constants. The criterion is related to the Mohr–Coulomb yield criterion, and the constants A , B can also be expressed in terms of cohesion c and angle of internal friction φ of the material via

$$A = \frac{6c \cos \varphi}{\sqrt{3}(3 - \sin \varphi)}, \quad B = \frac{2 \sin \varphi}{\sqrt{3}(3 - \sin \varphi)}. \quad (27)$$

2.4.3. Pressure dependent yield criterion in combination with damage model

In our fracture model (Schäfer et al., 2016), the reduction in strength under tensile loading is expressed via a scalar damage variable $0 \leq D \leq 1$, where $D = 0$ corresponds to intact material and $D = 1$ to a totally fractured material that cannot undergo any tension or deviatoric stress. In this sense, a material with $D = 0.5$ experiences half the deviatoric stress and tension of an undamaged material with $D = 0$. In order to relate the fragmentation model with the yield strength for plastic flow, Collins et al. (2004) calculate a damage dependent yield strength in the following way. Undamaged, intact substances have a pressure dependent yield strength of

$$Y_p = c + \frac{\mu p}{1 + \mu p / (Y_M - c)}, \quad (28)$$

where μ is the coefficient of internal friction, given by $\mu = \tan \varphi$, c is the cohesion and Y_M the shear strength at infinite pressure. For totally damaged materials, the deviatoric stress is usually set to zero. However, as pointed out by Jutzi (2015), it is important to take into account the friction in totally damaged material, e.g., shattered rocks. Therefore, for totally damaged materials with $D = 1$, the yield strength is not vanishing but dominated by the pressure and given by

$$Y_D = \mu_D p, \quad (29)$$

where μ_D now denotes the coefficient of internal friction of the totally damaged, granular material. For partially damaged materials, the yield strength is determined according to

$$Y = (1 - D)Y_p + DY_D, \quad (30)$$

while $Y \leq Y_p$ is ensured. As soon as one particle exceeds the yield strength, the deviatoric stress is reduced according to Eq. (25), with the limiting factor

$$f_Y = \min \left[Y / \sqrt{J_2}, 1 \right]. \quad (31)$$

2.5. Porosity models

Macroscopic porosity can be easily modelled with SPH by simply adding macroscopic voids on the scale of the particle volume to the initial distribution. However, sub-resolution porosity can be troublesome. Various sub-resolution porosity models exist in the literature and are applied to porous materials from metals to rocks. Although originally the $p - \alpha$ -model was developed to study impacts into metal shielding, we have first implemented it for the simulation of porous, rocky objects such as comets and asteroids.

2.5.1. The $p - \alpha$ -model

The $p - \alpha$ -model by Herrmann (1969), which was modified by Jutzi et al. (2009), relates the porous pressure p of a material to the solid pressure p_s and the distention α . The main idea is to separate the volume change in a porous material into the pore collapse of the porous material and the compression of the matrix material. The distention is defined by

$$\alpha = \frac{\varrho_s}{\varrho} = \frac{1}{1 - \phi}, \quad (32)$$

with the solid density ϱ_s , the porous density ϱ and the porosity ϕ . The porous pressure is calculated with the empirically found equation using the distention α and the pressure of the solid material p_s (Carroll and Holt, 1972)

$$p = \frac{1}{\alpha} p_s(\varrho_s, u_s) = \frac{1}{\alpha} p_s(\alpha \varrho, u), \quad (33)$$

where ϱ_s and u_s are the density and the energy of the solid, and ϱ and u of the porous material. The crush-curve relates the pressure to the porosity and yields a maximum pressure which the porous material can undergo before crushing. This gives an additional boundary condition for the pressure in the simulation. Two crush-curves are currently implemented. One being quadratic and the other having two regimes with parameters gained by fitting function (34) to the crush-curve of porous pumice, using a compression testing machine. In the plastic regime the distention α is computed as follows

$$\alpha(p) = \begin{cases} (\alpha_e - \alpha_t) \frac{(p_t - p)^{n_1}}{(p_t - p_e)^{n_1}} + (\alpha_t - 1) \frac{(p_s - p)^{n_2}}{(p_s - p_e)^{n_2}} + 1 & \text{if } p_e < p < p_t \\ (\alpha_t - 1) \frac{(p_s - p)^{n_2}}{(p_s - p_e)^{n_2}} + 1 & \text{if } p_t \leq p < p_s \\ 1 & \text{if } p \geq p_s, \end{cases} \quad (34)$$

where p_e is the pressure where the transition from the elastic to the plastic regime happens, p_s the pressure where the porous material is fully compacted and the density equals the matrix density ($\alpha = 1$) and the pressure p_t indicates a transition between the two regimes with different slopes n_1 and n_2 (Jutzi et al., 2009).

The distention is evolved in time depending on the pressure as follows

$$\dot{\alpha} = \frac{d\alpha}{dp} \frac{dp}{dt}. \quad (35)$$

One of the most important aspects of the $p - \alpha$ -model implementation is the check function for the timestep. The relation between the pressure and the distention is an implicit. However, to solve it explicitly, one has to make sure that the distention is resolved correctly. Hence, the distention has to follow the crush-curve since each distention value relates to a specific maximum pressure value. If the integrated distention yields a pressure beyond the maximum pressure for this specific distention value, the time step has been too coarse, the crush-curve is not reproduced correctly and the timestep has to be reduced, the integration step has to be repeated. To ensure this critical boundary condition our integrators have either an distention or a pressure check function where the change of the previous to the current value is calculated and if these values are too large the timestep is decreased accordingly, and the current step is recalculated with the lowered timestep. In order to validate the porous model, we have simulated the impact of a nylon, spherical projectile into a cube of pumice in three dimensions. Fig. 1 shows the pressure-distention values of the SPH particles from the impact. The distention α lies exactly on the crush-curve and thus resolving the timestep accordingly with crush-curve parameters shown in Table 1. The simulation was a nylon sphere impacting into a pumice cube. The parameters for the Tillotson equation of state (EOS) are those of

Table 1

Parameters for the crush-curve which were used for the plot and the simulation in Fig. 1.

p_e [Pa]	p_t [Pa]	p_s [Pa]	α_0	α_t	n_1	n_2
10^6	6.8×10^7	2.13×10^8	3.45	1.90	12	3

Table 2

Parameters for the crush-curve which were used for the plot and the simulation in Fig. 2.

p_e [Pa]	p_s [Pa]	c_0 [ms ⁻¹]	c_e [ms ⁻¹]
8×10^7	7×10^8	5.35×10^3	4.11×10^3

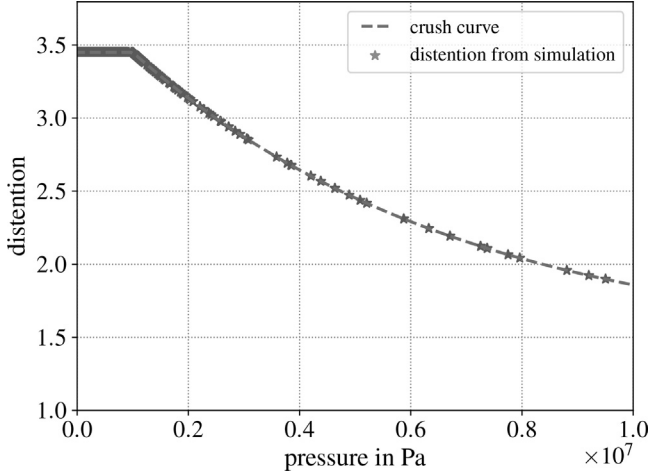


Fig. 1. Relation between the distention α and the pressure p . The initial distention is $\alpha_0 = 3.45$ and until p_e is reached the deformation is elastic and $\frac{d\alpha}{dp} = 0$.

solid rock with a volume $V = 216 \text{ cm}^3$ and a matrix density $\rho_s = 2325.3 \text{ kg/m}^3$. The impactor sphere has the Tillotson EOS parameters of nylon with a radius $r = 7 \text{ mm}$, a density $\rho_s = 1185 \text{ kg/m}^3$ and an impact velocity $v_{\text{imp}} = 2.58 \text{ km s}^{-1}$.

A shock wave test was performed which travels in one spatial dimension. The material was porous aluminium with an initial distention of $\alpha_0 = 1.275$. The crush-curve parameters are shown in Table 2 and the Tillotson EOS parameters are those of aluminium. The cylinder has a height $h = 2 \text{ cm}$ and a radius of $r = 0.2 \text{ cm}$. A constant velocity is set in the z -direction with $v_z = -45.8 \text{ cm s}^{-1}$. The quadratic crush-curve

$$\alpha = 1 + (\alpha_e - 1) \frac{(p_s - p)^2}{(p_s - p_e)^2} \quad (36)$$

was used and $\left[\frac{d\alpha}{dp} \neq 0 \right]_{\text{elastic}}$ is calculated using

$$\left[\frac{d\alpha}{dp} \right]_{\text{elastic}} = \frac{\alpha^2}{K_0} \left[1 - \left(\frac{1}{h(\alpha)^2} \right) \right] \quad (37)$$

where $K_0 = c_0^2 \rho_0$ and

$$h(\alpha) = 1 + (\alpha - 1) \frac{c_e - c_0}{c_0(\alpha_e - 1)}, \quad (38)$$

with $\alpha_e = \alpha(p_e)$. The results are shown in Fig. 2. We find perfect agreement with the simulation presented by Jutzi et al. (2009).

2.5.2. The ε - α -model

Another porosity model has been implemented into the code which is the so called ε - α -model (Collins et al., 2011). The main difference between this and the above mentioned p - α -model is

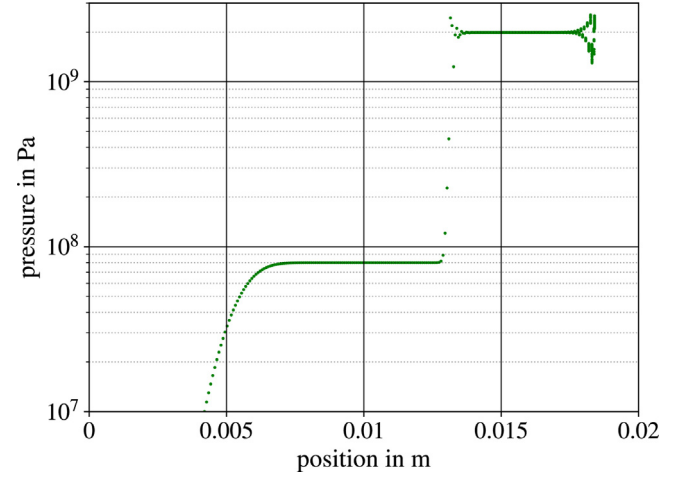


Fig. 2. One dimensional shock wave in a three dimensional porous aluminium cylinder. The pressure is plotted as a function of the vertical position of the SPH particle in the cylinder. The plot shows two waves with an elastic precursor followed by the actual shock wave.

the calculation of the distention. In this model the distention is calculated using the volumetric strain ε_V which is given by the change in volume of an element divided by the initial volume leading to following expression:

$$\varepsilon_V = \int_{V_0}^{V'} \frac{dV}{V} = \ln \left(\frac{V'}{V_0} \right), \quad (39)$$

with V_0 as the initial and V' as the updated volume. The volumetric strain can be related to the distention α

$$\varepsilon_V = \ln \left(\frac{V'}{V_0} \right) = \ln \left(\frac{V' V_s}{V_s V_0} \right) = \ln \left(\frac{\alpha}{\alpha_0} \right), \quad (40)$$

with an ideally constant matrix volume V_s . The compaction function is defined as

$$\alpha(p) = \begin{cases} \alpha_0 e^{\kappa(\varepsilon_V - \varepsilon_e)} & \text{if } \varepsilon_e > \varepsilon_V \geq \varepsilon_x \\ 1 + (\alpha_0 e^{\kappa(\varepsilon_x - \varepsilon_e)} - 1) \left(\frac{\varepsilon_c - \varepsilon_V}{\varepsilon_c - \varepsilon_x} \right)^2 & \text{if } \varepsilon_x > \varepsilon_V > \varepsilon_c \\ 1 & \text{if } \varepsilon_V \leq \varepsilon_c, \end{cases} \quad (41)$$

with ε_e as the elastic-plastic transition strain, ε_x the threshold strain for the transition between the two compaction regimes, ε_c the threshold for full compaction, κ the exponential compaction rate and α_0 the initial distention. The parameter ε_c has to be calculated beforehand with following equation

$$\varepsilon_c = \frac{2 - 2\alpha_0 e^{\kappa(\varepsilon_x - \varepsilon_e)}}{\kappa \alpha_0 e^{\kappa(\varepsilon_x - \varepsilon_e)}} + \varepsilon_x. \quad (42)$$

In this model the distention is not calculated directly using the pressure. The change in the distention is calculated as follows

$$\dot{\alpha} = \frac{d\alpha}{d\varepsilon_V} \frac{d\varepsilon_V}{dt}. \quad (43)$$

2.5.3. Sirono-model

The Sirono porosity model implemented in the code is by Sirono (2004) which was later modified by Schäfer et al. (2007), and extended to different material by Geretschauser et al. (2010). It is an isothermal model, which is the main difference to the two above mentioned models. It uses a slightly modified Murnaghan EOS which is an extension of the liquid EOS with a non-linear

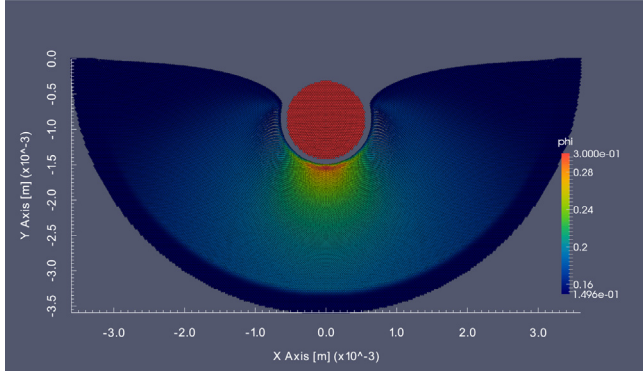


Fig. 3. Crater depth of the glass bead falling into the porous dust cake. The colour mapped is the filling factor $\phi = \varrho/\varrho_s$.

dependency of the pressure on the density

$$p(\phi) = K(\phi'_0) \left(\frac{\phi}{\phi'_0} - 1 \right), \quad (44)$$

where $\phi'_0 = \frac{\varrho'_0}{\varrho_s}$ and ϱ'_0 is the reference density, i.e. the density at $p = 0$. To calculate the pressure at a given density we only need the bulk modulus $K(\phi)$ which is given by the power law

$$K(\phi) = 2\mu(\phi) = K_0 \left(\frac{\phi}{\phi_i} \right)^\gamma, \quad (45)$$

with the shear modulus $\mu(\phi)$ and $\phi_i = \varrho_i/\varrho_s$. Since there is no energy dependency in this EOS it is only used for the low velocity regime. The pressure is restricted by the compressive strength $\Sigma(\phi)$ and tensile strength $T(\phi)$. Until the pressure reaches the upper limit (the compressive strength) the deformation is elastic and once it surpasses the compressive strength the deformation process will be plastic. This changes the slope for the calculation of the pressure Eq. (44) and thus deforming the material irreversibly. The pressure is limited in following way

$$p(\phi) = \begin{cases} \Sigma(\phi) & \text{if } \phi > \phi_c^+ \\ T(\phi) & \text{if } \phi < \phi_c^- \end{cases} \quad (46)$$

The compressive strength is calculated using this equation

$$\Sigma(\phi) = \begin{cases} p_m \left(\frac{\phi_{\max} - \phi_{\min}}{\phi_{\max} - \phi_{\min} - \epsilon} - 1 \right)^{\Delta \ln 10} & \text{if } \phi \leq \phi_{\min} + \epsilon \\ p_m \left(\frac{\phi_{\max} - \phi_{\min}}{\phi_{\max} - \phi} - 1 \right)^{\Delta \ln 10} & \text{if } \phi_{\min} + \epsilon < \phi < \phi_{\max} \\ \infty & \text{if } \phi \leq \phi_{\max}, \end{cases} \quad (47)$$

with the material dependent parameters p_m , Δ , ϵ and the minimum and maximum filling factors ϕ_{\min} and ϕ_{\max} . For the test calculation presented in Fig. 3, the following values have been calibrated using (Güttler et al., 2009) the mean pressure $p_m = 260$ Pa, $\Delta = 0.58$, $\epsilon = 0.005$, $\phi_{\min} = 0.12$ and $\phi_{\max} = 0.58$. The tensile strength is given by

$$T(\phi) = -10^{a+b\phi} \text{ Pa}, \quad (48)$$

with $a = 2.8$ and $b = 1.48$. Fig. 3 shows one of these (in this case two dimensional) simulations where a glass bead falls into the porous dust cake (SiO_2). The used values for the setup are shown in Table 3.

2.6. Equation of states

The equation of state relates the thermodynamic state variables and closes the system of PDEs. It is normally defined as

Table 3

Parameter table of the two dimensional crater depth validation simulations for the Sirono model.

Physical Quantity	Value with Unit
Glass bead	
Matrix density ϱ_0	$2.54 \times 10^3 \text{ kg/m}^3$
Bulk modulus K_0	$5 \times 10^9 \text{ Pa}$
Murnaghan exponent n	4
Radius r	$0.55 \times 10^{-3} \text{ m}$
Impact velocity v_0	0.65 ms^{-1}
Dust target	
Initial filling factor ϕ_0	0.15
Matrix density ϱ_s	2000 kg/m^3
Reference density ϱ'_0	300 kg/m^3
Bulk modulus K_0	$3 \times 10^5 \text{ Pa}$
Radius r_{dust}	$3.3 \times 10^{-3} \text{ m}$
Other parameters	
Artificial viscosity α, β	0.1, 0

a relation between the density, pressure and temperature or internal energy of a specific material. Depending on the physical regimes, different EOS are applied. There is no universal EOS, which can be used for all materials under all conditions, liquid, gaseous or solid. Therefore, various EOS are implemented and available in the code. In the original version, the available EOS were liquid EOS, Murnaghan EOS, and Tillotson EOS. In the new version, we have added the EOS for polytropic gas, isothermal gas, ideal gas, and ANEOS. The use of porosity models require additional porosity-dependent pressure relations, which are implemented via special EOS for these porous materials. Hence, there are several additional versions for porous materials with Tillotson EOS for the p - α -, the Sirono-, and the ϵ - α -model. Additional EOS may be included to the code straightforwardly by adding another EOS_TYPE_* to pressure.h and the calculation of the pressure to the kernel function in pressure.cu.

2.6.1. Murnaghan EOS

The Murnaghan EOS can be derived from the assumption that the bulk modulus varies linearly with pressure (Murnaghan, 1937). It reads

$$p = \frac{K_0}{n} \left(\left(\frac{\varrho}{\varrho_0} \right)^n - 1 \right), \quad (49)$$

where K_0 is the bulk modulus and ϱ_0 the density in the relaxed state. The Murnaghan EOS can be used to model weakly compressible materials. In this publication, it was applied to model the gravitational collapse of a granular column, see Section 3.4.

2.6.2. Tillotson EOS

A widely used non-linear EOS for simulating high-velocity impacts was introduced by Tillotson (1962) and can be applied over a wide range of physical conditions, while being computationally simple. It provides rudimentary distinction between the solid and vapour phase, but lacks a consistent treatment of phase changes. In the case of compressed regions ($\varrho \geq \varrho_0$) and u lower than the energy of incipient vaporization u_{iv} the equation of state reads

$$p = \left[a_T + \frac{b_T}{1 + u/(u_0 \eta^2)} \right] \varrho u + A_T \xi + B_T \xi^2, \quad (50)$$

with $\eta = \varrho/\varrho_0$ and $\xi = \eta - 1$. In case of expanded material (u greater than the energy of complete vaporization u_{cv}) the

equation of state takes the form

$$p = a_T \varrho u + \left[\frac{b_T \varrho u}{1 + u/(u_0 \eta^2)} + A_T \xi \exp[-\beta_T (\eta^{-1} - 1)] \right] \quad (51)$$

$$\times \exp[-\alpha_T (\eta^{-1} - 1)^2]. \quad (52)$$

The symbols ϱ_0 , A_T , B_T , u_0 , a_T , b_T , α_T , and β_T are material dependent parameters. In the partial vaporization $u_{iv} < u < u_{cv}$, the pressure is linearly interpolated between the two values obtained via (50) and (51), respectively. The Tillotson EOS has the advantage of being computational very simple, and still sophisticated enough for the application over a wide regime of physical conditions in high pressure ranges, such as high-velocity collisions and impact cratering.

Note, that the Tillotson EOS can also yield negative pressure values for expanded ($\eta < 1$) and cold ($u < u_{iv}$) states. This corresponds to a tension in the solid material. As stated by Melosh (1996), a fluid will fragment into smaller droplets at lower densities, limiting the tension. In order to consider this behaviour, a density limit parameter can be defined for each material type in the simulation. As soon as the ratio η falls below this limit, the pressure is set to zero instead of becoming negative. Additionally, there is also a compile time option which prevents negative pressure values at all.

2.6.3. ANEOS

A more sophisticated treatment of phase changes is possible with the analytic equations of state (ANEOS) package (see, e.g., Melosh, 2007). ANEOS is based on calculating the Helmholtz free energy from first principles, where all relevant quantities can then be deduced from. This also holds for the temperature, which is not directly possible for the Tillotson EOS. However, ANEOS requires more than 40 input parameters (10 for the Tillotson EOS), and is not freely available. It consists of several thousand lines of FORTRAN code, which has to be utilized to produce lookup-tables, which can then be interpolated from. To enhance the thermodynamical aspects of our SPH code's material model, we have recently finished implementing ANEOS to overcome the limitations of the Tillotson EOS (Malamud et al., 2020).

2.7. Fixed and variable smoothing

The code supports three different methods for SPH smoothing: Either a fixed smoothing length for each material type is set or the smoothing length varies for each particle separately. In the latter case, a fixed number of interaction partners can be set individually for each material type or the smoothing length is integrated according to the following evolution equation

$$\frac{dh}{dt} = \frac{h}{d} \nabla \cdot \mathbf{v}, \quad (53)$$

where d denotes the dimension. Additionally, an upper and lower limiter can be chosen for each material type. If two particles with differing smoothing lengths h_a and h_b interact, the kernel is calculated with their mean $\frac{1}{2}(h_a + h_b)$. The search for the interaction partners is conducted by the help of the Barnes–Hut tree. For each particle, a tree walk starting at the root of the tree is performed. The search continuous descending only into nodes that intersect with the sphere with the radius of the particle's smoothing length. Finally, individual particles in the lowermost nodes (the leaves) are checked for interaction by calculating the distance and comparing to the smoothing length. Since the interaction partner search for one particle is independent from the search for the partners of another particle, the search algorithm parallelizes perfectly.

2.8. Time integrator

The default time integrator is an embedded second order Runge–Kutta integrator, which uses an additional third step to determine an adaptive time step. Its implementation is based on the description in Butcher (1987). Four more integrators are implemented: a simple Euler integrator for test purposes during new developments, two different predictor–corrector schemes, one with a prediction step of half the time step and a correction step with the whole time step and vice versa, and a special purpose coupled Heun and fourth order Runge–Kutta integrator. The time step size for all integrators is limited by the Courant condition and the artificial viscosity is also taken into account (Monaghan, 2012)

$$\Delta t \leq \text{CFL} \frac{h}{c_s + 1.2(c_s \alpha + \mu_{\max} \beta)}. \quad (54)$$

Here, CFL is the Courant number, c_s denotes the sound speed, h is the smoothing length, α and β are the artificial viscosity parameters and μ_{\max} is the maximum value of μ_{ab} from all interaction pairs. The term μ_{ab} is an approximation for the velocity divergence and calculated according to

$$\mu_{ab} = \frac{\bar{h}_{ab}(\mathbf{v}_a - \mathbf{v}_b) \cdot (\mathbf{x}_a - \mathbf{x}_b)}{(\mathbf{x}_a - \mathbf{x}_b)^2 + \varepsilon_v \bar{h}_{ab}^2}. \quad (55)$$

The term \bar{h}_{ab}^2 is the average of the smoothing lengths of particle a and b . and the factor $\varepsilon_v \bar{h}_{ab}^2$ in the denominator prevents divergence of μ_{ab} . Apart from the Courant time limit, there exist additional limit check functions to ensure convergence of the crush-curve for simulations including porosity, see Section 2.5, and upper limits for an absolute change of specific values like the damage variable in simulations including brittle fracture can be specified to ensure convergence, i.e.

$$\Delta t \leq \begin{cases} \varepsilon \frac{|\xi| + \xi_{\min}}{\left| \frac{d\xi}{dt} \right|} & \text{for } \left| \frac{d\xi}{dt} \right| > 0 \\ \Delta t_{\max} & \text{else,} \end{cases} \quad (56)$$

for all quantities ξ which are integrated, where $\varepsilon < 1$ and ξ_{\min} is a measure to set the minimum time step.

In addition, a combined Heun and fourth order Runge–Kutta (RK4) integrator is implemented with intended application for circumbinary discs following the method by Thun and Kley (2018). The Heun part is used to solve the hydrodynamical equations, i.e. to integrate the SPH equations and the RK4 part is used to integrate the orbits of the gravitating objects. For each complete timestep in the Heun integrator, the four steps of the RK4 are performed. In every substep of the RK4 the accelerations for all gravitational point masses are calculated. At the end of timestep, both integrators are at the new time $t + \Delta t$ and the cycle restarts.

2.9. Additional gravitating bodies

An arbitrary number of purely gravitationally interacting (point-)masses may be included in the simulation. The gravitational force from these objects leads to an additional acceleration of the particles. These masses also interact mutually and their motions are integrated with the same integrator that integrates the fluid or solid body. This renders simulations possible that include more than one central object, e.g., collisions of smaller objects in a planetary system or the simulation of accretion discs around circumbinaries with one or more potentially embedded planets. The point masses are stored in additional data files and a minimum and maximum distance for the particle can be specified individually for each point mass. Once a particle comes too close or too far, it is deactivated in the simulation.

2.10. Long-range forces

Currently, *miluphcuda* allows to simulate the self-gravity of the mass distribution given by the particle distribution. The long-range forces due to the mutually gravitational interaction of the particles is either calculated directly via a N^2 algorithm or approximately using the Barnes–Hut tree (Barnes and Hut, 1986) that is also used for the interaction partner search. The latter one may be also a starting point to implement other long-range forces like Coulomb's law. The additional acceleration due to the gravitational interaction is added to the momentum equation of the applied module, i.e. one of Eqs. (1),(5),(15).

The self-gravity module has been validated by the simulation of an isothermal, spherical and initially uniformly rotating collapsing molecular cloud by Schäfer et al. (2016). The module is used in the three applications presented in Sections 3.1 to 3.3.

3. Applications

In this section we intent to display the versatility of *miluphcuda*. We have adapted the code to various differing physical problems including hydro- and solid mechanics, with and without gravitational forces, self-gravity, strength-dominated and strengthless bodies, inviscid and viscous flows. The subsections contain exemplary simulations of various projects which make use of our code. In practice, each numerical investigation with SPH has to include a convergence study, i.e. varying number of particle numbers, varying smoothing lengths, and different values for other parameters of the method such as the artificial viscosity. Since our intention is to present many different simulations with all available modules, we do not present these convergence studies here, and focus only on variety of applications.

3.1. Embedding realistic collision outcomes into long-term planet formation simulations

The formation of terrestrial planets is believed to proceed in several relatively distinct steps, where during the last and final phase, often termed 'late-stage accretion', planetary embryos (\sim Moon- to Mars-sized objects) and remaining smaller planetesimals eventually accrete into planets. This phase is characterized by chaotic interaction and giant collisions among large, similar-sized, gravity-dominated bodies, which shape many of a planet's final characteristics – its orbital parameters, spin and obliquity, and not least its composition, especially also volatile/water contents. Since (giant) collisions are naturally the core agent of accretion, it is crucial to study those events individually in detail. However, to obtain truly self-consistent results especially in the context of a chaotic planet formation environment it is eventually the combination of the long-term dynamical evolution of protoplanets/planetesimals with realistic outcomes of individual encounters that is necessary. We follow two different approaches for this problem, both centred around physically accurate collision simulations with *miluphcuda*. The first one is to use a catalog of collision outcomes, spanning the most relevant parts of parameter space, where the outcome of a particular collision scenario is then interpolated from (Section 3.1.1). The second approach is to run a dedicated SPH simulation for each occurring collision event 'on the fly', whose results are directly re-inserted into the further underlying N-body evolution (Section 3.1.2).

Our main aim in these applications is to study volatile/water transport, which may have been delivered to the accretion zone of Earth by successive collisions among water-carrying planetesimals and planetary embryos from the outer asteroid belt region.

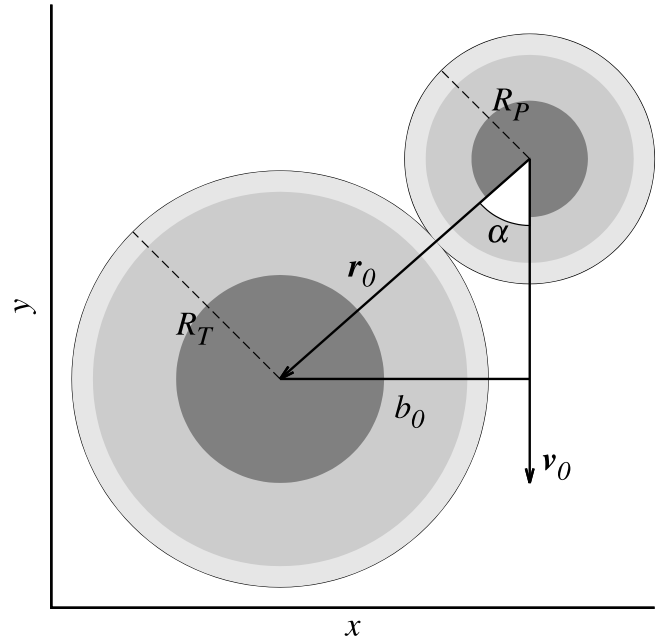


Fig. 4. Collision geometry for similar-sized, differentiated bodies in a target-centric frame. $|v_0|$ is the impact velocity, α the impact angle (0° for head-on).

Even though semi-analytical models (scaling laws) to approximately predict collision outcomes have been developed in recent years (Leinhardt and Stewart, 2012; Leinhardt et al., 2015; Marcus et al., 2010b), they are currently not able to reliably predict the more subtle consequences for water inventories (Burger et al., 2018), where existing studies have almost exclusively assumed oversimplified perfectly inelastic merging of water-bearing bodies. This is despite the fact that water and other volatile material are particularly susceptible to collisional fragmentation and loss processes, and thus masses as well as water contents of final terrestrial planets are considerably overestimated in most cases.

Giant collisions of similar-sized bodies (cf. Fig. 4) are complex and generally super-sonic events, where strong shocks dissipate large amounts of energy, which can lead to large-scale melting and vaporization, or even eject outer layers mechanically. In addition material initially under high pressure can be released from the deep interior when bodies are (partly) disrupted (cf. Fig. 6). Correct modelling of the thermodynamic response with a suitable equation of state is therefore crucial (Section 2.6), while the precise material rheology becomes typically less important once in the gravity-dominated regime. The majority of such encounters is more or less oblique, which makes them naturally fully 3-dimensional events, and gives rise to frequent hit-and-run outcomes (e.g., Fig. 6), where the colliding bodies separate again (gravitationally) after the interaction phase.

3.1.1. Collisions catalog for protoplanetary bodies

In order to arrive at a more realistic estimate of the amount of water transported to Earth's accretion zone, we systematically investigated the rate of water transport in collisions by performing a suite of several hundred collision simulations with *miluphcuda*. Here, we present results in terms of the fraction of water retained after a collision for different parameters of the system including collision velocity, impact angle, and the masses and water-mass fractions of the projectile and target. The resulting catalog of collisional water loss focuses on planetary embryos of masses in the Ceres to Earth range and initial water content. Table 4 lists the parameters chosen for the catalog.

Table 4Scenario parameters of the collision catalog. The velocity v_{esc} denotes the two-body escape velocity.

Parameter	Values
Total mass $M_{\text{tot}} = M_{\text{projectile}} + M_{\text{target}}$	2 $M_{\text{Ceres}} (1.88 \times 10^{21} \text{ kg})$, 2 $M_{\text{C}} (1.47 \times 10^{23} \text{ kg})$, 2 $M_{\text{D}} (1.28 \times 10^{24} \text{ kg})$, 2 $M_{\text{E}} (1.19 \times 10^{25} \text{ kg})$
Mass ratio $\gamma = M_{\text{projectile}}/M_{\text{target}}$	0.1, 0.5, 1.0
Projectile water mass fraction w_p	0.1, 0.2
Target water mass fraction w_t	0.1, 0.2
Impact velocity $v [v_{\text{esc}}]$	1.0, 1.5, 2.0, 3.0, 5.0
Impact angle $\alpha [\text{deg}]$	0, 20, 40, 60

Table 5Tillotson EOS Parameters adopted from Benz and Asphaug (1999) used in the simulations for the collision catalog. The bulk modulus is set equal to A_T and B_T , shear modulus and the von Mises yield stress are denoted by μ and Y_M , respectively.

Material	$\rho_0 [\text{kg/m}^3]$	$A_T [\text{GPa}]$	$B_T [\text{GPa}]$	$E_0 [\text{MJ/kg}]$	$E_{iv} [\text{MJ/kg}]$	$E_{cv} [\text{MJ/kg}]$
Basalt	2700	26.7	26.7	487	4.72	18.2
Ice	917	9.47	9.47	10	0.773	3.04
Material	α_T	β_T	a_T	b_T	$\mu [\text{GPa}]$	$Y_M [\text{GPa}]$
Basalt	0.5	1.5	5	5	22.7	3.5
Ice	0.3	0.1	10	5	2.8	1

Table 6

Column description of the collision outcome data in the supplement.

Column	Description
v	collision velocity [v_{esc}]
alpha	collision angle [deg]
m	$M_{\text{tot}} = M_{\text{projectile}} + M_{\text{target}}$ [kg]
gamma	mass ratio $\gamma = M_{\text{projectile}}/M_{\text{target}}$
wp	projectile water mass fraction
wt	target water mass fraction
mS1	mass of survivor 1 [kg]
mS2	mass of survivor 2 [kg]
wmfS1	survivor 1 water mass fraction
wmfS2	survivor 2 water mass fraction
wretained	$1 - \frac{\text{water lost to debris [kg]}}{\text{total initial water [kg]}}$

In order to limit the parameter space we assumed a uniform structure consisting of a basaltic silicate rock interior covered by a water ice mantle for all bodies. Our material model is based on the Tillotson equation of state (see Melosh, 1996) and uses the parameters given in Table 4. The Weibull parameters for the damage model are $m = 16$ and $k = 10^{61} \text{ m}^{-3}$ (Nakamura et al., 2007) for basalt and $m = 9.1$ and $k = 10^{46} \text{ m}^{-3}$ (Lange et al., 1984) for water ice, respectively (see Table 5).

The outcome in terms of the masses of the largest two surviving fragments and the water remaining on these survivors is given in the supplementary data to this article. The data is given in xlsx format with columns as described in Table 6. Visualizing our high-dimensional data is difficult. In order to properly present our catalog, we thus apply the embedding algorithm UMAP (McInnes et al., 2018a,b), which allows an efficient, non-linear down-projection of our six-dimensional input space. We use the default hyperparameters for the UMAP mapping, except for $n_{\text{neighbours}}=15$ and $\text{min_dist}=0.99$. Fig. 5 visualizes the embedded dataset. Each of the original six dimensions is contributing some structure to the embedding. There are 17 clusters which are – each for itself – homogeneous in impact angle and projectile/target water mass fraction and composed of different impact velocities and masses. Although the plots are still difficult to interpret, a comparison of the colour gradients indicate that the water retention of the two largest fragments is correlated with both the impact velocity and the impact angle: while the gradients of the velocity closely describe the intra-cluster gradients of the water retention, the impact angle does so with the global gradient across all clusters.

In the near future we plan to create a scheme that allows us to integrate data from an extended catalog into N-body calculations that will use predicted collision outcomes instead of perfect merging. First experiments with multi-dimensional interpolation work satisfactorily regarding the water loss, but fail in terms of predicting correct fragment sizes, positions, and velocity vectors after the impact. Therefore, we currently run a simulation campaign of thousands of collisions – enough to apply machine learning methods to predict detailed collision outcomes as they occur in planet formation simulations (Winter, Winkler, Maindl, and Schäfer, in prep.).

3.1.2. Direct N-body – SPH hybrid simulations

The direct combination of an underlying N-body computation (realized with the versatile REBOUND package, Rein and Liu, 2012) with dedicated individual collision simulations has the advantage of full self-consistency and allows to precisely model all collisions at hand, including more subtle processes like water loss and transfer between colliding bodies in hit-and-run encounters. Such a direct hybrid approach has been applied only rarely and with limited resolution (Genda et al., 2011, 2017), and never to follow water or other volatiles to our knowledge.

In order to ensure the flawless handover from the N-body computation to an SPH collision and back, several well-synchronized steps are necessary, implemented via a Python interface, which calls the necessary subprocesses (including miluphcuda) and handles communication between them. Once bodies are found to be on colliding trajectories, the N-body integration continues until they have approached to a well-defined starting distance for the SPH simulation – several times the sum of their radii to allow for the build-up of potential pre-collision tidal deformations. The colliding bodies are initialized already with self-consistent hydrostatic profiles following Burger et al. (2018), which works practically instantaneously and eliminates the need for lengthy numerical relaxation. To ensure a self-consistent (gravity) environment, the star and all other bodies in the N-body system (besides the two colliding ones) are additionally included as point masses for the whole duration of the SPH simulation. Once the collision/interaction phase is over (potential post-collision fragments have clearly separated), we first identify spatially connected fragments via an octree/friends-of-friends algorithm, followed by additional search for gravitationally bound fragments, which are then each considered a single post-collision body. Eventually their (barycenters') positions and velocities are re-inserted into the N-body integration.

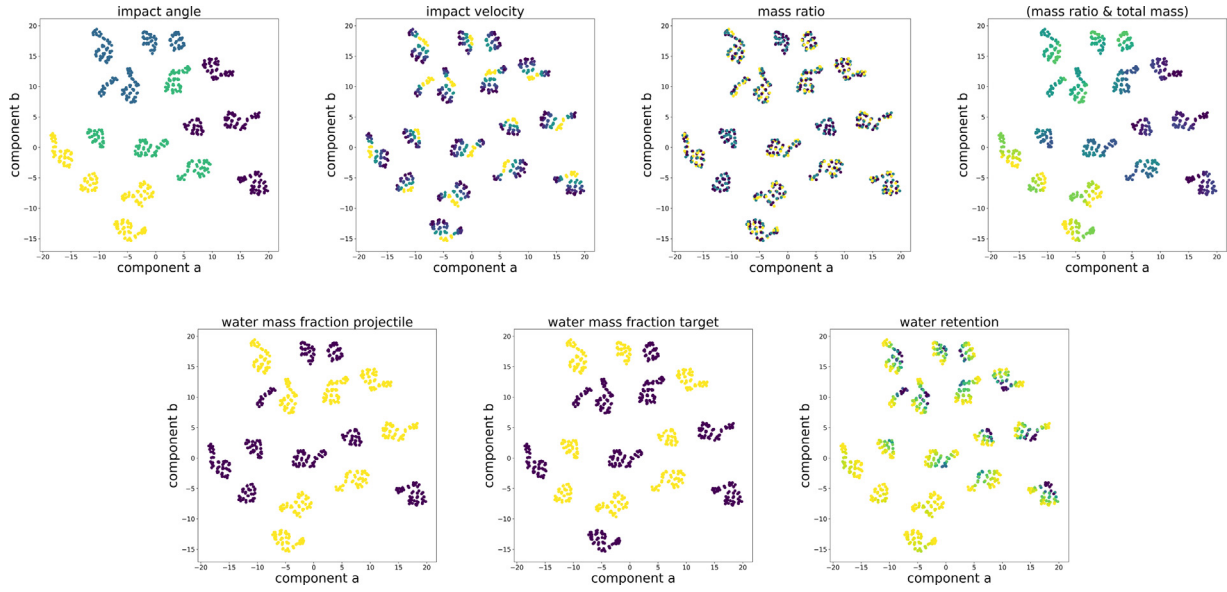


Fig. 5. For visualization, the six-dimensional input space was embedded into two dimensions *component a* and *component b* using UMAP. The titles of the plots indicate the respective colour codings. Note that only the water retention colour code represents a result after evaluating the collision outcomes, the others represent the initial conditions.

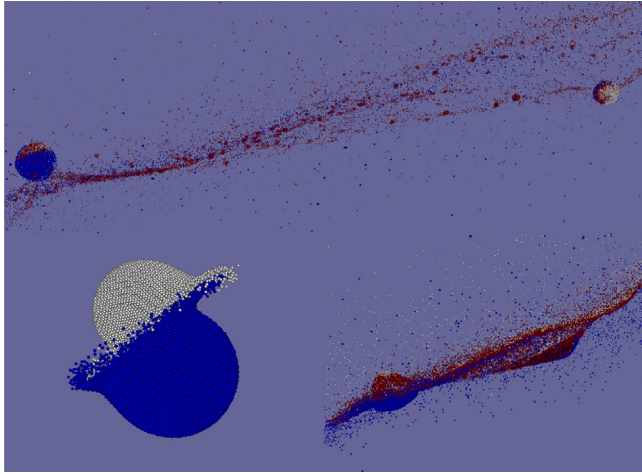


Fig. 6. Snapshots of a hit-and-run collision between protoplanet-sized bodies. The impactor (white) approaches from the left. Both objects are differentiated into an iron core (black), a silicate mantle (red) and a water shell, colour-coded in blue for the target and white for the impactor, to illustrate not only collisional losses but also water transfer between the bodies. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Source: Refer to [Burger et al. \(2018\)](#) for more details.

The vastly different time as well as size scales of individual collisions vs. long-term N-body evolution pose particular challenges. Individual collisions cover \sim hours to days at most, while the whole planetary accretion phase lasts for several 100 Myrs. The spatial domain of individual collisions is usually considerably below 10^6 km, while the whole system covers tens of au ($\sim 10^9$ km). The capabilities of *miluphcuda* allow us to run the necessary multi-material collision simulations (including surface water layers), where the various available strength/material models (Section 2.4) and equations of state (Section 2.6) can be utilized to account for the broad range of physical states of water in an active planet formation environment. The actual conditions at the onset of some currently simulated collision however, depend on many parameters, among them the distance to the star,

the body's (especially thermal) history, and the time since the last (large) collision/impact event. For a truly holistic picture all these processes (and likely more) will have to be included eventually, which is our ultimate but not yet accomplished goal, therefore a preliminary solution is to model water inventories as (strengthless) fluid, properties that likely resemble a surface ocean on average. Our SPH code provides the necessary fast and efficient computations to run typically \sim hundreds of dedicated collision simulations per underlying N-body scenario, where individual ones (with resolutions between 25 000 to 75 000 SPH particles) require on the order of an hour to complete (with all pre- and post-processing steps).

In order to model collisional water transport we apply this hybrid framework to an evolving system of several hundred planetary embryos and planetesimals which extends into the water-rich region beyond the ice condensation line (snow line). We include either no (very disruptive), one (accretion or erosion), or two (hit-and-run) post-collision bodies in the further N-body evolution. The rationale is that collisions between large, gravity-dominated bodies generally result in at most two large post-collision bodies besides orders-of-magnitude smaller additional debris, which has either been desiccated directly during the collision, or remaining water is otherwise likely often lost quickly, once vaporized by large-scale energy deposition during impact. An example of such a collision is illustrated in [Fig. 6](#).

3.2. Formation of massive exomoons of super-terrestrial exoplanets

Most simulation studies of giant impacts have focused on the collisional phase space conducive to the formation of Solar system planets and satellites ([Barr, 2016](#)). Despite an extensive collision simulation literature, there have only been a few studies that investigated giant impacts relevant to exoplanets that are more massive than the Earth ([Marcus et al., 2010a,b; Liu et al., 2015; Barr and Bruck Syal, 2017](#)), and in particular studies that focus on the formation of exosolar satellites ([Barr and Bruck Syal, 2017](#)). We also use *miluphcuda* in order to simulate novel scenarios which enable the formation of massive exomoons. We investigate the mass, long-term tidal-stability, composition and origin of material of such exomoons.

Since super-terrestrial planet collisions are often highly energetic, the debris discs which they generate typically require sensitive treatment which goes beyond the capabilities of the more common, Tillotson EOS. We therefore implement the M-ANEOS EOS. Our M-ANEOS parameter input files are derived from Melosh (2007). We consider differentiated impactors and targets composed of 30% iron and 70% dunite by mass. The initial setup of each simulation is calculated via a pre-processing step, in which both impactor and target are generated with relaxed internal structures, i.e. having hydrostatic density profiles and internal energy values from adiabatic compression, following the algorithm provided in appendix A of Burger et al. (2018). All our simulations have a resolution of 10^6 SPH particles.

An example of a particularly interesting simulation is shown in Fig. 7. While most collisions generate a massive disc, which later evolves to coagulate into a massive satellite, we also find this rare and illustrative case in which an exomoon forms through a graze & capture scenario between a super-Earth and an Earth-sized planet. The result is an intact, planet-sized exomoon, containing about half the mass of the Earth. At this size, an exomoon can be potentially detectable using our best available observing instruments. E.g., see the Hunt for Exomoons with Kepler (HEK) (Kipping et al., 2012) initiative. We note that to date, there has not been any confirmed detection of an exomoon, hence this finding may be important.

We validate our model by successfully reproducing the results obtained in a predecessor study (Barr and Bruck Syal, 2017) which uses a different, adaptive mesh refinement (AMR) code. Indeed our results are broadly compatible with theirs, and the data follows similar trends.

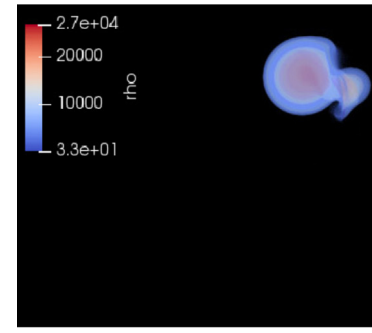
3.3. Ring structures around Chariklo and Haumea

All the four giant planets of our solar system host ring structures in equatorial orbits within their Roche radius, and recently at least two other bodies are found to host ring structures: the Centaur 10199 Chariklo (Braga-Ribas et al., 2014) and the dwarf planet Haumea (Ortiz et al., 2017). The characteristics of the rings vary widely, but in many systems it is possible to find dense and narrow rings mainly composed by particles with sizes ranging from centimetres to a tens of meter bodies. In such case the particles' interaction cannot be neglected, what poses a computational challenge for usual N-body approach usually applied to study the particles dynamics.

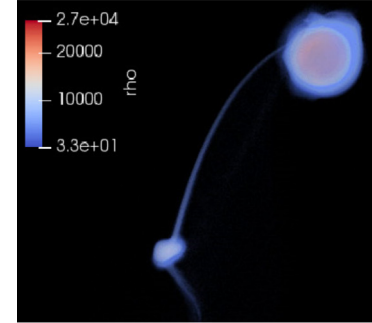
Furthermore, dense rings are optically thick and highly collisional systems. The typical impacts might happen at multiple times the metal escape velocity, so the collisions outcome cannot be safely determined from the extrapolation of the usual prescriptions (e.g., Leinhardt and Stewart, 2012).

Some of the issues mentioned above can be solved using SPH simulations. The parallel nature of the technique allows an efficient way to compute the mutual interaction among a large ensemble of particles. Besides being possible to take into account the material properties, SPH simulations also allow the inclusion of fragmentation and porosity effects, so the collisions are treated in a more realistic way.

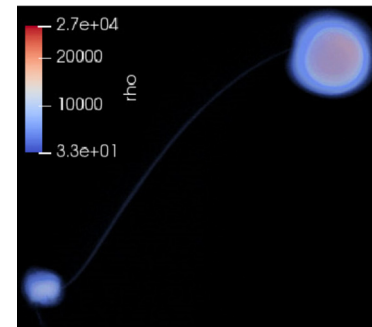
Our code has been successfully adapted to simulate the system of Chariklo aiming to explain the formation the region depleted of material between the two rings, and to investigate the interaction of the ring particles with a putative satellite that may inhabit this gap. We employed the Tillotson equation of state, and the initial setup consisted in a small section of 10×5 km of the rings filled with 20 m bodies, each one being composed by particles spaced by 1 m. The number of bodies were determined to result in an optical depth comparable to the observed value. The self-gravity was taken into account, and all bodies were assumed to



(a) 0.17 hours



(b) 1.5 hours



(c) 3 hours

Fig. 7. The first three hours of a giant impact between a $0.83 M_{\oplus}$ impactor and $6.4 M_{\oplus}$ target, obtaining a graze & capture scenario, which generates an intact, $0.5 M_{\oplus}$ exomoon. The colour scheme shows the density in units of $\text{kg} \times \text{m}^{-3}$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

be composed by ice particles subject to the gravitational force of Chariklo and the satellite, both treated as external forces. The motion of the particles was followed in a rotating frame that moves with the satellite and periodic boundary conditions were applied in the azimuthal direction.

To validate the results obtained with miluphcuda we compared the simulation outcome with a similar setup integrated using the IAS15 algorithm, a high-order numerical integrator implemented in REBOUND (Rein and Spiegel, 2015). Preliminary results show that both runs are in agreement and the SPH code performed well, even more when we consider that N-body run required some simplification in the effects that can be added to the simulation.

3.4. Collapse of a granular column

To show the applicability of the code to model granular medium, we have simulated the three dimensional collapse of a

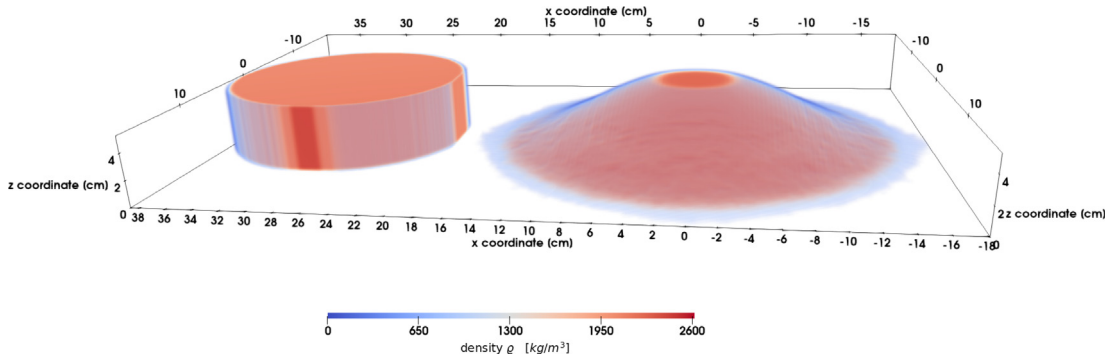


Fig. 8. Gravitational collapse of a granular column. The SPH density is shown in the volume rendered image for the initial state on the left hand side and for the final state on the right hand side. The initial ratio between radius and height was $a = 0.55$.

granular column under gravitation following a validation simulation from [Szewc \(2016\)](#), who compared simulations to experimental results by [Lube et al. \(2004\)](#). Initially, a granular column of radius R_0 and height $H_0 = aR_0$ (with $a = 0.55$ in the simulation presented here) is placed on the ground. The granular material has a density of $\rho = 2.6 \text{ g/cm}^3$, a sound speed of 100 m/s , an internal angle of friction of 30 deg and zero cohesion. Depending on the ratio a between initial height and radius, [Szewc \(2016\)](#) and [Lube et al. \(2004\)](#) find fitting values for the final runout R_∞ of the medium on the ground. The runout length determined by [Szewc \(2016\)](#) is

$$R_\infty = \begin{cases} 0.72aR_0 + R_0 & a < 1.7 \\ 1.02a^{3/5}R_0 + R_0 & a \geq 1.7 \end{cases}, \quad (57)$$

while the experimental data indicate ([Lube et al., 2004](#))

$$R_\infty = \begin{cases} 1.24aR_0 + R_0 & a < 1.7 \\ 1.6a^{1/2}R_0 + R_0 & a \geq 1.7. \end{cases} \quad (58)$$

The rheology is modelled using the Mohr–Coulomb yield strength. In order to obtain a weakly compressible material, we apply the Murnaghan EOS with $n = 7$, use a bulk modulus of $K = 10^8 \text{ Pa}$ and a shear modulus of $\mu = 10^7 \text{ Pa}$. The number of particles in the simulation is about 470 000. We find a final runout length of $R_\infty = 17.94 \text{ cm}$, which is larger than the experimental value of 16.32 cm , found by [Lube et al. \(2004\)](#), a deviation of about 6%. For the same ratio a and initial radius R_0 , the simulations of [Szewc \(2016\)](#) yield a runout length of 13.54 cm , a lower value than the experimental outcome. [Fig. 8](#) shows renderings of the initial granular column and the material at the end.

3.5. High-speed dynamics

The code was used to simulate several high-velocity impacts in the context of asteroid target fragmentation by [Buruchenko and Schäfer \(2017\)](#). Moreover, its performance was compared to commercial software. In this section we present exemplary simulations showing the capability of `miluphcuda` to handle high-speed dynamics and shock-wave physics related problems. [Piekutowski \(2001\)](#) studied the experimental outcome of hyper-velocity impacts of non-spherical projectiles on zinc plates by looking at specific features in the produced debris clouds. He documented changes in the debris clouds by shooting a zinc sphere, rod, and disc into 0.965 mm -thick zinc plates with a speed of about 5 km s^{-1} . The three different shapes of the projectile yield highly distinct debris clouds: The debris cloud produced by the spherical projectile features a hourglass shape, the rod forms a cone-like structure of plate material attached to the remaining intact material of the impacting rod, the disc projectile generates a pillar-shaped debris cloud with only little dispersion. We have simulated the impact with 212 000, 795 000, and

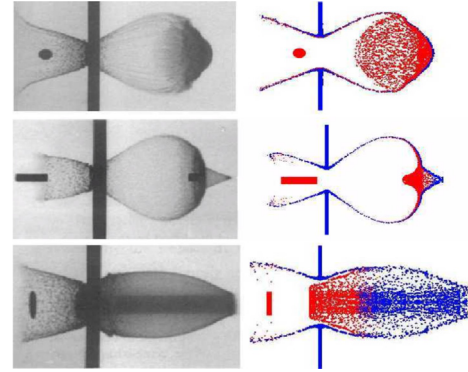


Fig. 9. Debris clouds produced in high-velocity impacts of three differently shaped projectiles into zinc plates. The left panels show the radiographs as published by [Piekutowski \(2001\)](#) and the right panels show the two dimensional simulation outcome using the SPH code. The projectile was (from top to bottom) a sphere, rod, disc.

Table 7

Parameters for the Tillotson EOS which were applied for the high-velocity impact simulations. The shear modulus μ and the von Mises yield stress Y_M are also given.

ρ_0 [kg/m ³]	A_T [GPa]	B_T [GPa]	E_0 [MJ/kg]	E_{iv} [MJ/kg]	E_{cv} [MJ/kg]
7140	145	115	315	1.138	1.138
α_T	β_T	a_T	b_T	μ [GPa]	Y_M [GPa]
0.5	1.5	10	10	39	0.34

175 000 particles respectively, using the Tillotson EOS with the values as specified in [Table 7](#) in two dimensions, and a constant smoothing length of $2.5 \times$ the initial particle separation. The parameters of the three impact experiments are as follows: the diameter of the sphere is 5.76 mm and the impact speed is 4.98 km s^{-1} , the dimensions of the rod are 3.988 mm diameter and 14.148 mm length, and of the disc 13 mm are diameter and 0.795 mm thickness, with impact speeds of 4.97 km s^{-1} and 5 km s^{-1} , respectively. The radiographs of the debris clouds produced in the experiment and the result of the SPH simulation are shown in [Fig. 9](#). The simulation is capable to reproduce the resulting debris clouds accurately in shape and size.

3.6. Shocktube test problem

One standard test for a hydro code is the one dimensional shocktube test problem (see, e.g., [Monaghan and Gingold, 1983](#)). The initial values at $t = 0$ are given by

$$\mathbf{U}(x, t = 0) = \begin{cases} \mathbf{U}_l & x \leq x_0 \\ \mathbf{U}_r & x > x_0, \end{cases} \quad (59)$$

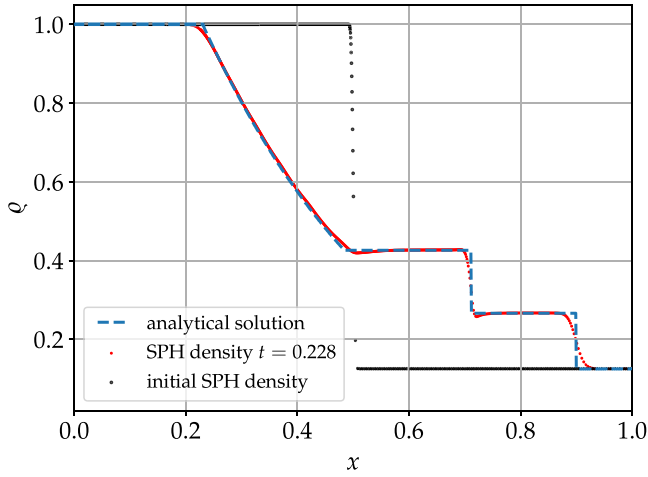


Fig. 10. Shocktube problem. The initial density distribution is indicated by the black dots. The analytical solution at time $t = 0.228$ is given by the dashed blue curve. The density after this time obtained with the SPH simulation is shown by the red dots. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

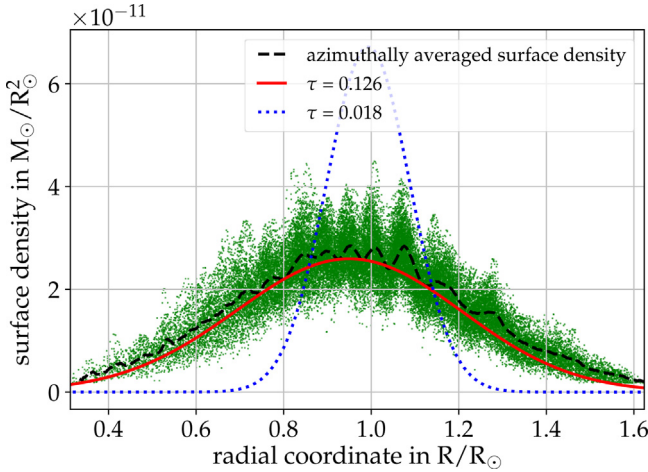


Fig. 11. Radial surface density profile of the viscous dust ring after the viscous time $\tau = 0.126$. The black dashed line denotes the azimuthally averaged surface density, the green dots the density of the particles, and the solid red line the analytical solution given by Eq. (62).

where $\mathbf{U}_{l,r} = (\varrho_{l,r}, v_{l,r}, u_{l,r})$ denotes the constant initial values for density, velocity and internal energy left and right of x_0 . The pressure is given by the ideal gas equation

$$p = (\gamma - 1)\varrho u, \quad (60)$$

with $\gamma = 7/5$ in our setup. We place particles in the interval $[-1, 2]$ with $x_0 = 0.5$, and set $\mathbf{U}_l(x, t = 0) = (1, 0, 2.5)$ and $\mathbf{U}_r = (0.125, 0, 2)$. The particles have equal masses with a separation of 4×10^{-3} in the lower density region $[-1, 0.5]$ and a ϱ_r/ϱ_l narrower separation in the higher density region $[0.5, 2]$, leading to a total number of 3376 particles. The smoothing length is 1×10^{-2} . We apply the standard Monaghan artificial viscosity with $\alpha = 1, \beta = 2$. The initial density at $t = 0$, the density at $t = 0.0228$ and the analytical solution is shown in Fig. 10. We find good agreement with the analytical solution.

3.7. Viscously spreading ring

In order to test the Navier–Stokes implementation, we simulate the two dimensional viscously spreading ring with the

parameters given in Speith and Kley (2003). The viscously spreading ring is pressure-less and orbits a central point mass on a Keplerian orbit. With the approximation of a small kinematic viscosity coefficient ν that is independent of the surface density, an analytic solution was derived by Lynden-Bell and Pringle (1974). With the initial radial profile of a ring with total mass M at $t = 0$

$$\Sigma(R) = \frac{M}{2\pi R_0} \delta(R - R_0), \quad (61)$$

the surface density evolves according to

$$\Sigma(\tau, x) = \frac{M}{\pi R_0^2} \frac{1}{\tau x^{1/4}} I_{1/4} \left(\frac{2x}{\tau} \right) \exp \left(-\frac{1+x^2}{\tau} \right), \quad (62)$$

with $x = R/R_0$, $\tau = 12\nu t/R_0^2$, and $I_{1/4}$ is the modified Bessel function to the order of $1/4$. Following Speith and Kley (2003), we simulate the viscously spreading ring with $N = 40\,000$ particles. The mass of the ring is $10^{-10} M_\odot$, the central mass is $M_c = 1 M_\odot$, and the initial radius is $R_0 = 1 R_\odot$. The kinematic viscosity coefficient is $\nu = 3 \times 10^{-8} R_\odot^2/\text{s}$. The (constant) smoothing length is set to $h = 0.0501 R_\odot$. The initial particle distribution represents the surface density given by Eq. (62) at the viscous time $\tau = 0.018$. The initial velocity for each particle is given by the Keplerian velocity around the central mass

$$v_\phi = \sqrt{\frac{GM_c}{R}}. \quad (63)$$

Fig. 11 shows the radial surface density profile of the ring after the viscous time $\tau = 0.126$. The simulation matches the analytical solution amply while slightly overestimating the surface density at the outer rim of the ring. We find also the spiral instability as described and analysed by Speith and Kley (2003). The variation of the azimuthally averaged surface density (black dashed line) around the analytical (1D-) solution emerges from this (2D-) instability.

3.8. Circumbinary disc evolution

An arbitrary number of (point-) mass objects can be added to the simulation as described in Section 2.9. We have simulated the evolution of a circumbinary disc in a two dimensional simulation. A circumbinary disc is an accretion disc orbiting a binary system that consists for example of a binary star. The numerical simulation of circumbinary discs may be quite arduous depending on the dynamical parameters of the binary system, e.g., the eccentricity (Thun et al., 2017). The gravitational force acting from the binary stars on the gas may lead to spiral shock waves in the disc and to the formation of an inner gap around the two stars eventually. The code was applied to model the accretion disc around HD104237 using the parameters as described in Dunhill et al. (2015). The two binaries have masses of $2.2 M_\odot$ and $1.4 M_\odot$, a semi-major axis of $a = 0.22 \text{ au}$ and an eccentricity of $e = 0.6$. The gas density shows the spiral shock waves exerted by violent gravitational interaction with the binary stars. The detailed analysis of our simulations of circumbinary discs are beyond of the scope of this paper and will be published elsewhere (Audiffren et al. in prep.) (see Fig. 12).

4. Conclusion

The numerical particle method SPH has proven to be an appropriate scheme when dealing with PDEs in the context of hydrodynamics and continuum mechanics. The GPU SPH code miluphcuda is now publicly available with this new release. We have presented several applications of the code from a wide range of fields, including inviscid and viscous flows, strengthless bodies

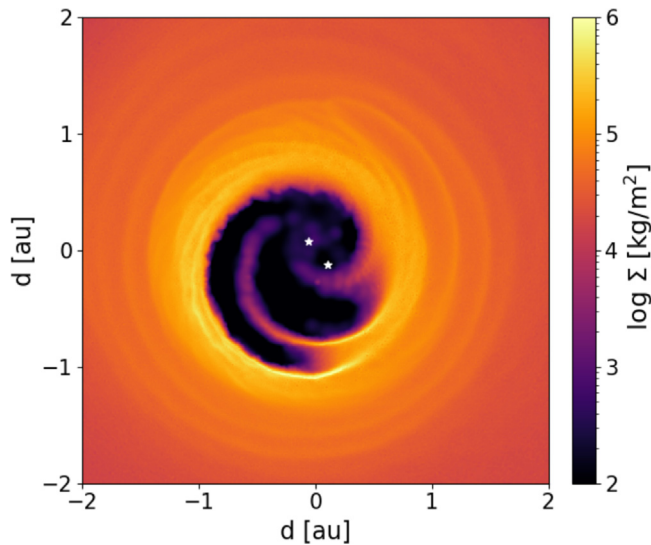


Fig. 12. The surface density distribution in the circumbinary disc around HD104237 after a simulation time of 159 binary orbits on a logarithmic scale. The two white stars indicate the current position of the primary secondary.

and solid bodies, and self-gravitating objects. The main focus of the code lies in collisional and impact studies with different materials and high and low velocities, although it has been extended to handle liquid and gaseous flows from one to three dimensions, e.g., to study circumbinary discs. Another promising application has been the combination of the SPH code with the REBOUND software package to study the long term evolution of water transfer in late stage accretion. The code runs on single Nvidia GPU, which can be plugged in normal workstations running Linux or macOS with ease, and allows for speed-ups compared to a single core CPU code (Schäfer et al., 2016) depending on the specific problem up to 40 in pure SPH simulations and up to 80 in simulations including self-gravity. Since the new modules have only been implemented in the GPU version of our SPH code, we cannot provide speed-up values for simulations including porosity. Compared to non-porous simulations, simulations including porosity demand for much lower time step sizes to obtain convergence of the crush-curve. Since the computational effort per time step remains unchanged overall, we expect comparable speed-ups for porous simulations.

CRedit authorship contribution statement

C.M. Schäfer: Project administration, Conceptualization, Methodology, Validation, Writing - original draft, Writing - review & editing. **O.J. Wandel:** Methodology, Validation, Writing - original draft. **C. Burger:** Methodology, Validation, Writing - original draft. **T.I. Maindl:** Methodology, Validation, Writing - original draft. **U. Malamud:** Writing - original draft. **S.K. Buruchenko:** Writing - original draft. **R. Sfair:** Writing - original draft. **H. Audiffren:** Writing - original draft. **E. Vavilina:** Writing - review & editing. **P.M. Winter:** Writing - original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Several plots in this publication have been made by the use of the matplotlib package by Hunter (2007) or with ParaView (Ayachit, 2015). We thank the anonymous reviewers whose comments and suggestions helped to improve and clarify this manuscript. CB and TIM appreciate support by the FWF Austrian Science Fund project S 11603-N16. CMS and CB appreciate support by the DFG German Science Foundation project 398488521. CMS wants to thank Daniel Thun, Martin Jutzi and Willy Kley for helpful discussions during the course of this project.

The authors acknowledge support by the High Performance and Cloud Computing Group at the Zentrum für Datenverarbeitung of the University of Tübingen, Germany, the state of Baden-Württemberg through bwHPC, Germany and the German Research Foundation (DFG) through grant no. INST 37/935-1 FUGG.

Appendix A. The code basics and speedup

The code is written in the C programming language and uses the Nvidia CUDA application programming interface (CUDA version ≥ 6.0). The different physical modules and the dimension can be set by compile time switches. The basic tree code is based on the implementation by Bertscher (2011). A description of the source and configuration files is shown in Table A.8. The material configuration files are parsed using libconfig (Lindner, 2015). The preferred data format for the data files is HDF5. However, if the library is missing, the simulations can also run with simple ASCII table files, one SPH particle per line. A skeleton Makefile is included in the repository and has to be modified according to the local settings and GPU architecture. The code runs on Linux and macOS with a CUDA-capable GPU. The number of threads for each CUDA kernel can be set as a compile time option in `timeintegration.h`. Note, that the best settings depend on your specific hardware and physical model. There is no distinct recommendation we can provide. Usually, a reduction of the computational time can be obtained by changing slightly the number of threads for the kernel with the maximal computational time. It is advisable to tweak the number of threads for the kernels before starting production runs. All compile time options related to the physical and numerical model are set in `parameter.h`. More options like the choice of the integrator, simulation time and number of output files, are parsed on the command line on the execution. All available command line options are explained via the help option `miluphcuda -h`.

If you intend to implement additional physics to the code, the following steps are required: Add new variables to the Particle structure in `miluph.h` and include the corresponding memory allocation in `memory_handling.cu`. For each substep of the integrator, the central function `rhs()` is called, which subsequently calls the CUDA kernels to determine all derivatives that are required by the integrator. A skeleton CUDA kernel running over all particles in the simulation is shown in listing 1.

```
1  __global__ void skeleton_function(int *interactions)
2  {
3      int i, inc;
4      inc = blockDim.x * gridDim.x;
5      /* main loop over all sph particles */
6      for (i = threadIdx.x + blockIdx.x * blockDim.x;
7           i < numParticles; i += inc) {
8
9          /* loop over all interaction partners of
10             particle i */
11             for (j = 0; j < p.noi[i]; j++) {
12                 /* index of interacting particle is
13                    stored in array interactions */
```

Table A.8

List of files in miluphcuda.

Filename	Purpose
anEOS.cu, anEOS.h	functions for initializing and running ANEOS
artificial_stress.cu, artificial_stress.h	functions for artificial stress
boundary.cu, boundary.h	special treatment of boundaries (external forces, etc.)
checks.h	consistency checks of compile time options
coupled_heun_rk4_sph_nbody.cu, coupled_heun_rk4_sph_nbody.h	coupled 2nd order Heun with 4th order RK integrator
cuda_utils.h	CUDA specific functions
damage.cu, damage.h	functions for the fragmentation and damage models
density.cu, density.h	calculate density
device_tools.cu, device_tools.h	functions to print GPU information
euler.cu, euler.h	simple Euler integrator for test purposes
gravity.cu, gravity.h	functions for the calculation of self gravity and gravitational interaction of SPH particles with point masses
internal_forces.cu, internal_forces.h	compute the change of stress tensor (regolith) and/or deviatoric stress tensor (all other solid materials), time evolution of density
io.cu, io.h	functions for input/output, ASCII and HDF5
kernel.cu, kernel.h	calculate SPH kernel values
linalg.cu, linalg.h	helping functions for matrix operations, etc.
little_helpers.cu, little_helpers.h	
Makefile	Makefile for GNU Make. Needs customization!
material.cfg	material configuration and parameters, see material_data/ subdirectory.
memory_handling.cu, memory_handling.h	functions for memory allocation and copying between host and device
miluph.cu	main
miluph.h	header file with Particle structure definition
parameter.h	file includes all important compile time options, physical model and numerical settings
pc_values.dat	minimum absolute values, which are needed for the predictor–corrector integration schemes
plasticity.cu, plasticity.h	material models (rheology): Drucker–Prager, Collins, von Mises
porosity.cu, porosity.h	porosity models
predictor_corrector.cu, predictor_corrector.h	Predictor–Corrector integrator with predictor step dt/2
predictor_corrector_euler.cu, predictor_corrector_euler.h	Predictor–Corrector integrator with predictor step dt
pressure.cu, pressure.h	calculation of the pressure according to the chosen EOS; currently there is, for solids: Tillotson EOS, Murnaghan EOS, Regolith EOS (Drucker–Prager model), ANEOS; for porous solids: Jutzi EOS (Tillotson EOS with $p - \alpha$), Jutzi–Murnaghan EOS (Murnaghan EOS with $p - \alpha$); for gas: Ideal gas EOS, Polytropic gas EOS, Isothermal gas EOS
rhs.cu, rhs.h	right-hand sides, calls the kernels that compute the time derivatives
rk2adaptive.cu, rk2adaptive.h	RK2 integrator with adaptive time step
rk4_pointmass.cu, rk4_pointmass.h	RK4 integrator for gravitating pointmasses
soundspeed.cu, soundspeed.h	calculation of the soundspeed
stress.cu, stress.h	calculation of the stress tensor
timeintegration.cu, timeintegration.h	calculation of all internal forces of the solid or fluid; calculation of artificial viscosity and artificial stress
tree.cu, tree.h	functions for the Barnes–Hut tree, interaction search
velocity.cu, velocity.h	calculate velocities
viscosity.cu, viscosity.h	Navier–Stokes equation
xsph.cu, xsph.h	functions for the XSPH algorithm

```

11         ip = interactions[i *
12         MAX_NUM_INTERACTIONS + j];
13         // now p.rho[i] contains the density of
14         particle i and
15         // p.rho[ip] the density of its
16         interaction partner...
17
18         // insert your sph calculation here
19
20     } // neighbour loop end
21 } // proceed with next particle

```

Listing 1: A skeleton CUDA kernel running over all particles and their interaction partners.

The main reason to port the code to CUDA is the expected speedup compared to the CPU-only implementation. Modern SPH codes like SWIFT are designed to run on HPC architectures like clusters with distributed memory and use the Message Passing Interface (MPI) for the process communication. However, access to HPC clusters might be limited and scarce for several reasons. Moreover, long (\sim several weeks) running simulations are

normally not possible due to queue regulations. Our code was especially designed to run on standard workstations with Nvidia GPUs to render high resolution SPH simulations possible without the access to special dedicated cluster hardware. Running on single GPUs only, the resolution is limited by the memory of the GPU, which is obviously much less than the distributed memory of a cluster. Hence, with current GPU hardware (12 GB memory), the maximal number of particles in a simulation with miluphcuda is 4×10^6 to 10^7 depending on the activated modules in the code.

The speedup for a three dimensional hydro simulation including self-gravity is presented in Fig. A.13. We have compared the runtime for four different tasks: (i) neighbour search, (ii) SPH equations, (iii) self gravity, (iv) total right hand side. The last task combines (i)–(iii) and includes also the time to build the tree and calculate pressure and sound speed for all particles. Task (ii) includes the SPH equations for momentum only, since the test case was isothermal. The highest speed up is achieved in the self-gravity module, followed by the speedup in the neighbour search. These results justify our motivation to port the code to CUDA, since most of the computational time is spent in the self-gravity task for simulations including self-gravity, and in the neighbour-searching task for simulations without self-gravity.

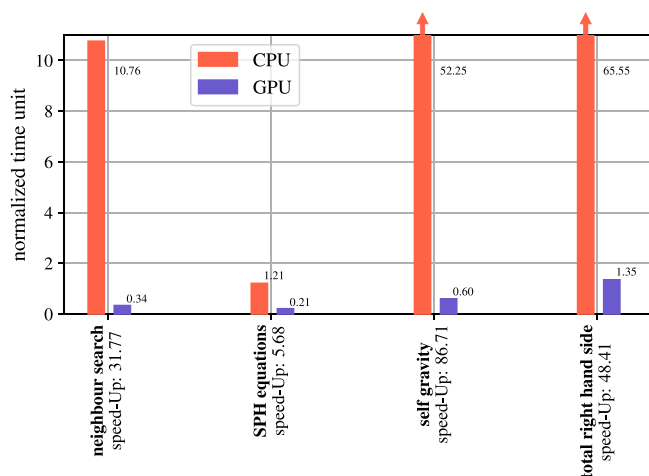


Fig. A.13. Speedup obtained with the CUDA port of the code compared to the CPU-only code (Schäfer et al., 2016). The CPU is an intel i7 clocked at 3.4 GHz. The GPU is a primitive (first release in 2013) standard gaming consumer graphics card Nvidia GTX Titan with a base clock of 837 MHz and 2688 CUDA cores. Shown are the normalized time units spent for different tasks.

Appendix B. Supplementary data

Supplementary material related to this article can be found online at [doi:10.1016/j.ascom.2020.100410](https://doi.org/10.1016/j.ascom.2020.100410).

References

- Akinci, N., Ihmsen, M., Akinci, G., Solenthaler, B., Teschner, M., 2012. Versatile rigid-fluid coupling for incompressible sph. *ACM Trans. Graph.* 31, 62:1–62:8. [doi:10.1145/2185520.2185558](https://doi.org/10.1145/2185520.2185558).
- Amicarella, A., Kocak, B., Sibilla, S., Grabe, J., 2017. A 3d smoothed particle hydrodynamics model for erosional dam-break floods. *Int. J. Comput. Fluid Dyn.* 31, 413–434. [doi:10.1080/10618562.2017.1422731](https://doi.org/10.1080/10618562.2017.1422731).
- Ayachit, U., 2015. *The ParaView Guide: A Parallel Visualization Application*. Kitware, Inc., USA.
- Barnes, J., Hut, P., 1986. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature* 324, 446–449. [doi:10.1038/324446a0](https://doi.org/10.1038/324446a0).
- Barr, A.C., 2016. Formation of exomoons: a solar system perspective. *Astron. Rev.* 12, 24–52. [doi:10.1080/21672857.2017.1279469](https://doi.org/10.1080/21672857.2017.1279469).
- Barr, A.C., Bruck Syal, M., 2017. Formation of massive rocky exomoons by giant impact. *Mon. Not. R. Astron. Soc.* 466, 4868–4874. [doi:10.1093/mnras/stx078](https://doi.org/10.1093/mnras/stx078).
- Benz, W., 1990. Smooth particle hydrodynamics - a review. In: Buchler, J.R. (Ed.), *Numerical Modelling of Nonlinear Stellar Pulsations Problems and Prospects*. p. 269.
- Benz, W., Asphaug, E., 1994. Impact simulations with fracture. I. Method and tests. *Icarus* 107, 98–116. [doi:10.1006/icar.1994.1009](https://doi.org/10.1006/icar.1994.1009).
- Benz, W., Asphaug, E., 1999. Catastrophic disruptions revisited. *Icarus* 142, 5–20. [doi:10.1006/icar.1999.6204](https://doi.org/10.1006/icar.1999.6204).
- Braga-Ribas, F., Sicardy, B., Ortiz, J.L., Snodgrass, C., Roques, F., Vieira-Martins, R., Camargo, J.I.B., Assafin, M., Duffard, R., Jehin, E., Pollock, J., Leiva, R., Emilio, M., Machado, D.I., Colazo, C., Lellouch, E., Skottfelt, J., Gillon, M., Ligier, N., Maquet, L., Benedetti-Rossi, G., Gomes, A.R., Kervella, P., Monteiro, H., Sfair, R., Moutamid, M.E., Tancredi, G., Spagnotto, J., Maury, A., Morales, N., Gil-Hutton, R., Roland, S., Ceretta, A., Gu, S.H., Wang, X.B., Harpsoe, K., Rabus, M., Manfroid, J., Opitom, C., Vanz, L., Mehret, L., Lorenzini, L., Schneiter, E.M., Melia, R., Lecacheux, J., Colas, F., Vachier, F., Widemann, T., Almenares, L., Sandness, R.G., Char, F., Perez, V., Lemos, P., Martinez, N., Jørgensen, U.G., Dominik, M., Roig, F., Reichart, D.E., Lacluyze, A.P., Haislip, J.B., Ivarsen, K.M., Moore, J.P., Frank, N.R., Lambas, D.G., 2014. A ring system detected around the Centaur (10199) Chariklo. *Nature* 508, 72–75. [doi:10.1038/nature13155](https://doi.org/10.1038/nature13155).
- Bui, H.H., Fukagawa, R., Sako, K., Ohno, S., 2008. Lagrangian Meshfree particles method (sph) for large deformation and failure flows of geomaterial using elastic-plastic soil constitutive model. *Int. J. Numer. Anal. Methods Geomech.* 32, 1537–1570. [doi:10.1002/nag.688](https://doi.org/10.1002/nag.688).
- Burger, C., Maindl, T.I., Schäfer, C.M., 2018. Transfer, loss and physical processing of water in hit-and-run collisions of planetary embryos. *Celestial Mech. Dynam. Astronom.* 130, [doi:10.1007/s10569-017-9795-3](https://doi.org/10.1007/s10569-017-9795-3).
- Burger, C., Schäfer, C.M., 2017. Applicability and limits of simple hydrodynamic scaling for collisions of water-rich bodies in different mass regimes. In: *Proceedings of the First Greek-Austrian Workshop on Extrasolar Planetary Systems*. pp. 63–81, [arXiv:1704.06075](https://arxiv.org/abs/1704.06075).
- Burtscher, M., 2011. *GPU Computing Gems Emerald Edition*.
- Buruchenko, S.K., Schäfer, T.I., 2017. Smooth particle hydrodynamics gpu-acceleration tool for asteroid fragmentation simulation. In: *14th Hypervelocity Impact Symposium 2017, HVIS2017, 24–28 April 2017, Canterbury, Kent, UK*. *Procedia Eng.* 204, 59–66. [doi:10.1016/j.proeng.2017.09.726](https://doi.org/10.1016/j.proeng.2017.09.726).
- Butcher, J.C., 1987. *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods*. Wiley-Interscience, New York, NY, USA.
- Carroll, M., Holt, A.C., 1972. Suggested Modification of the $P-\alpha$ Model for Porous Materials.
- Cercos-Pita, J., 2015. Aquagpusph, a new free 3d sph solver accelerated with opencl. *Comput. Phys. Comm.* 192, 295–312. [doi:10.1016/j.cpc.2015.01.026](https://doi.org/10.1016/j.cpc.2015.01.026).
- Collins, G.S., Melosh, H.J., Ivanov, B.A., 2004. Modeling damage and deformation in impact simulations. *Meteorit. Planet. Sci.* 39, 217–231. [doi:10.1111/j.1945-5100.2004.tb00337.x](https://doi.org/10.1111/j.1945-5100.2004.tb00337.x).
- Collins, G., Melosh, H.J., Wünnemann, K., 2011. Improvements to the epsilon-alpha compaction model for simulating impacts into high-porosity solar system objects. *Int. J. Impact Eng. - INT J IMPACT ENG* 38, 434–439. [doi:10.1016/j.ijimpeng.2010.10.013](https://doi.org/10.1016/j.ijimpeng.2010.10.013).
- Crespo, A., Domínguez, J., Rogers, B., Gómez-Gesteira, M., Longshaw, S., Canelas, R., Vacondio, R., Barreiro, A., García-Feal, O., 2015. Dualsphysics: Open-source parallel cfd solver based on smoothed particle hydrodynamics (sph). *Comput. Phys. Comm.* 187, 204–216. [doi:10.1016/j.cpc.2014.10.004](https://doi.org/10.1016/j.cpc.2014.10.004).
- Dunhill, A.C., Cuadra, J., Dougados, C., 2015. Precession and accretion in circumbinary discs: the case of HD 104237. *Mon. Not. R. Astron. Soc.* 448, 3545–3554. [doi:10.1093/mnras/stv284](https://doi.org/10.1093/mnras/stv284).
- Flebbe, O., Muenzel, S., Herold, H., Riffert, H., Ruder, H., 1994. Smoothed particle hydrodynamics: Physical viscosity and the simulation of accretion disks. *Astrophys. J.* 431 (754), [doi:10.1086/174526](https://doi.org/10.1086/174526).
- Genda, H., Iizuka, T., Sasaki, T., Ueno, Y., Ikoma, M., 2017. Ejection of iron-bearing giant fragment fragments and the dynamical and geochemical influence of the fragment re-accretion. *Earth Planet. Sci. Lett.* 470, 87–95. [doi:10.1016/j.epsl.2017.04.035](https://doi.org/10.1016/j.epsl.2017.04.035).
- Genda, H., Kokubo, E., Ida, S., 2011. Giant impacts and terrestrial planet formation. In: *Lunar and Planetary Science Conference*, p. 2090.
- Geretshauser, R.J., Speith, R., Güttler, M., Blum, J., 2010. Numerical simulations of highly porous dust aggregates in the low-velocity collision regime*. *Astron. Astrophys.* 513.
- Gingold, R.A., Monaghan, J.J., 1977. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Mon. Not. R. Astron. Soc.* 181, 375–389. [doi:10.1093/mnras/181.3.375](https://doi.org/10.1093/mnras/181.3.375).
- Grishin, E., Malamud, U., Perets, H.B., Wand el, O., Schäfer, C.M., 2020. The wide-binary origin of (2014) MU₆₉-like Kuiper belt contact binaries. *Nature* 580, 463–466. [doi:10.1038/s41586-020-2194-z](https://doi.org/10.1038/s41586-020-2194-z).
- Güttler, C., Krause, M., Geretshauser, R.J., Speith, R., Blum, J., 2009. The physics of protoplanetary dust agglomerates. IV. Toward a dynamical collision model. *Astrophys. J.* 701, 130–141. [doi:10.1088/0004-637X/701/1/130](https://doi.org/10.1088/0004-637X/701/1/130).
- Haghighipour, N., Maindl, T.I., Schäfer, C., Speith, R., Dvorak, R., 2016. Triggering sublimation-driven activity of main belt comets. *Astrophys. J.* 830, 22. [doi:10.3847/0004-637X/830/1/22](https://doi.org/10.3847/0004-637X/830/1/22).
- Haghighipour, N., Maindl, T.I., Schäfer, C.M., Wandel, O.J., 2018. Triggering the activation of main-belt comets: The effect of porosity. *Astrophys. J.* 855, 60. [doi:10.3847/1538-4357/aaa7f3](https://doi.org/10.3847/1538-4357/aaa7f3).
- Héroult, A., Bilotta, G., Dalrymple, R.A., 2010. Sph on gpu with cuda. *J. Hydraul. Res.* 48, 74–79. [doi:10.1080/00221686.2010.9641247](https://doi.org/10.1080/00221686.2010.9641247).
- Hernquist, L., Katz, N., 1989. TREESPH - A Unification of SPH with the hierarchical tree method. *Astrophys. J. Suppl.* 70, 419–446. [doi:10.1086/191344](https://doi.org/10.1086/191344).
- Herrmann, W., 1969. Constitutive equation for the dynamic compaction of ductile porous materials. *J. Appl. Phys.* 40.
- Hunter, J.D., 2007. Matplotlib: A 2d graphics environment. *Comput. Sci. Eng.* 9, 90–95.
- Ihmsen, M., Orthmann, J., Solenthaler, B., Kolb, A., Teschner, M., 2014. SPH Fluids in computer graphics. In: *Lefebvre, S., Spagnuolo, M. (Eds.), Eurographics 2014 - State of the Art Reports*. The Eurographics Association, [doi:10.2312/egst.20141034](https://doi.org/10.2312/egst.20141034).
- Jutzi, M., 2015. Sph calculations of asteroid disruptions: The role of pressure dependent failure models. In: *VIII Workshop on Catastrophic Disruption in the Solar System*. *Planet. Space Sci.* 107, 3–9. [doi:10.1016/j.pss.2014.09.012](https://doi.org/10.1016/j.pss.2014.09.012).
- Jutzi, M., Michel, P., Hiraoka, K., Nakamura, A.M., Benz, W., 2009. Numerical simulations of impacts involving porous bodies: II. comparison with laboratory experiments. *Icarus* 201.
- Kegerreis, J.A., Eke, V.R., Gonnet, P., Korycansky, D.G., Massey, R.J., Schaller, M., Teodoro, L.F.A., 2019. Planetary giant impacts: convergence of high-resolution simulations using efficient spherical initial conditions and SWIFT. *Mon. Not. R. Astron. Soc.* 487, 5029–5040. [doi:10.1093/mnras/stz1606](https://doi.org/10.1093/mnras/stz1606).

- Kipping, D.M., Bakos, Buchhave, L., Nesvorný, D., Schmitt, A., 2012. The hunt for exomoons with kepler (hek). i. description of a new observational project. *Astrophys. J.* 750, 115. doi:10.1088/0004-637X/750/2/115.
- Korzani, M.G., Galindo-Torres, S.A., Scheuermann, A., Williams, D.J., 2017. Parametric study on smoothed particle hydrodynamics for accurate determination of drag coefficient for a circular cylinder. *Water Sci. Eng.* 10, 143–153. doi:10.1016/j.wse.2017.06.001.
- Lange, M.A., Ahrens, T.J., Boslough, M.B., 1984. Impact cratering and spall failure of gabbro. *Icarus* 58, 383–395. doi:10.1016/0019-1035(84)90084-8.
- Leinhardt, Z.M., Dobinson, J., Carter, P.J., Lines, S., 2015. Numerically predicted indirect signatures of terrestrial planet formation. *Astrophys. J.* 806 (23), doi:10.1088/0004-637X/806/1/23.
- Leinhardt, Z.M., Stewart, S.T., 2012. Collisions between gravity-dominated bodies. I. Outcome regimes and scaling laws. *Astrophys. J.* 745, 79. doi:10.1088/0004-637X/745/1/79.
- Libersky, L.D., Petschek, A.G., 1991. Smooth particle hydrodynamics with strength of materials. In: Trease, H.E., Fritts, M.F., Crowley, W.P. (Eds.), *Advances in the Free-Lagrange Method Including Contributions on Adaptive Gridding and the Smooth Particle Hydrodynamics Method*. pp. 248–257. doi:10.1007/3-540-54960-9_58.
- Lindner, M., 2015. Libconfig@ONLINE. URL <http://www.hyperrealm.com/libconfig>.
- Liu, S.F., Hori, Y., Lin, D.N.C., Asphaug, E., 2015. Giant impact: An efficient mechanism for the devolatilization of super-earths. *Astrophys. J.* 812, 164. doi:10.1088/0004-637X/812/2/164.
- Liu, M.B., Liu, G.R., 2010. Smoothed particle hydrodynamics (sph): an overview and recent developments. *Arch. Comput. Methods Eng.* 17, 25–76. doi:10.1007/s11831-010-9040-7.
- Lube, G., Huppert, H.E., Sparks, R.S.J., Hallworth, M.A., 2004. Axisymmetric collapses of granular columns. *J. Fluid Mech.* 508, 175–199. doi:10.1017/S0022112004009036.
- Lucy, L.B., 1977. A numerical approach to the testing of the fission hypothesis. *Astron. J.* 82, 1013–1024. doi:10.1086/112164.
- Lynden-Bell, D., Pringle, J.E., 1974. The evolution of viscous discs and the origin of the nebular variables. *Mon. Not. R. Astron. Soc.* 168, 603–637. doi:10.1093/mnras/168.3.603.
- Malamud, U., Perets, H.B., Schäfer, C., Burger, C., 2018. Moonfalls: collisions between the earth and its past moons. *Mon. Not. R. Astron. Soc.* 479, 1711–1721. doi:10.1093/mnras/sty1667.
- Malamud, U., Perets, H.B., Schäfer, C., Burger, C., 2020. Collisional formation of massive exomoons of superterrestrial exoplanets. *Mon. Not. R. Astron. Soc.* 492, 5089–5101. doi:10.1093/mnras/staa211.
- Marcus, R.A., Sasselov, D., Hernquist, L., Stewart, S.T., 2010a. Minimum radii of super-earths: Constraints from giant impacts. *Astrophys. J. Lett.* 712, L73–L76. doi:10.1088/2041-8205/712/1/L73.
- Marcus, R.A., Sasselov, D., Stewart, S.T., Hernquist, L., 2010b. Water/icy super-earths: Giant impacts and maximum water content. *Astrophys. J. Lett.* 719, L45–L49. doi:10.1088/2041-8205/719/1/L45.
- McInnes, L., Healy, J., Melville, J., 2018a. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv e-prints* arXiv:1802.03426.
- McInnes, L., Healy, J., Saul, N., Grossberger, L., 2018b. Umap: Uniform manifold approximation and projection. *J. Open Source Softw.* 3 (861).
- Melosh, H., 1996. *Impact Cratering: A Geologic Process*. In: *Oxford monographs on geology and geophysics*, Oxford University Press.
- Melosh, H.J., 2007. A hydrocode equation of state for sio₂. *Meteorit. Planet. Sci.* 42, 2079–2098. doi:10.1111/j.1945-5100.2007.tb01009.x.
- Mocz, P., Succi, S., 2015. Numerical solution of the nonlinear Schrödinger equation using smoothed-particle hydrodynamics. *Phys. Rev. E* 91, 053304. doi:10.1103/PhysRevE.91.053304.
- Monaghan, J.J., 1992. Smoothed particle hydrodynamics. *Annu. Rev. Astron. Astrophys.* 30, 543–574. doi:10.1146/annurev.aa.30.090192.002551.
- Monaghan, J.J., 2005. Smoothed particle hydrodynamics. *Rep. Progr. Phys.* 68, 1703–1759. doi:10.1088/0034-4885/68/8/r01.
- Monaghan, J., 2012. Smoothed particle hydrodynamics and its diverse applications. *Annu. Rev. Fluid Mech.* 44, 323–346. doi:10.1146/annurev-fluid-120710-101220.
- Monaghan, J., Gingold, R., 1983. Shock simulation by the particle method sph. *J. Comput. Phys.* 52, 374–389. doi:10.1016/0021-9991(83)90036-0.
- Murnaghan, F.D., 1937. Finite deformations of an elastic solid. *Amer. J. Math.* 59, 235–260.
- Nakamura, A.M., Michel, P., Setoh, M., 2007. Weibull Parameters of yakuno basalt targets used in documented high-velocity impact experiments. *J. Geophys. Res. (Planets)* 112, E02001. doi:10.1029/2006JE002757.
- Ortiz, J.L., Santos-Sanz, P., Sicardy, B., Benedetti-Rossi, G., Bérard, N., Duffard, R., Braga-Ribas, F., Hopp, U., Ries, C., Nascimbeni, V., Marzari, F., Granata, V., Pál, A., Kiss, C., Pribulla, T., Komžík, R., Hornoch, K., Pravec, P., Bacci, P., Maestripietri, M., Nerli, L., Mazzei, L., Bachini, M., Martinelli, F., Succi, G., Ciabattari, F., Mikuz, H., Carbognani, A., Gahrken, B., Mottola, S., Hellmich, S., Rommel, F.L., Fernández-Valenzuela, A.C., Cikota, A., Cikota, A., Lecacheux, J., Vieira-Martins, R., Camargo, J.I.B., Assafin, M., Colas, F., Behrend, R., Desmars, J., Meza, E., Alvarez-Candal, A., Beisker, W., Gomes-Junior, A.R., Morgado, B.E., Roques, F., Vachier, F., Berthier, J., Mueller, T.G., Madiedo, J.M., Unsalan, O., Sonbas, E., Karaman, N., Erece, O., Koseoglu, D.T., Ozisik, T., Kalkan, S., Guney, Y., Niaei, M.S., Satir, O., Yesilyaprak, C., Puskullu, C., Kabas, A., Demircan, O., Alikakos, J., Charmandaris, V., Leto, G., Ohlert, J., Christille, J.M., Szakáts, A.T., Varga-Verebélyi, E., Marton, G., Marciniak, A., Bartczak, P., Santana-Ros, T., Butkiewicz-Bąk, M., Dudziński, V., Gazeas, K., Tzouganatos, L., Paschalis, N., Tsamis, V., Sánchez-Lavega, A., Pérez-Hoyos, S., Hueso, R., Guirado, J.C., Peris, V., Iglesias-Marzoa, R., 2017. The size, shape, density and ring of the dwarf planet haumea from a stellar occultation. *Nature* 550, 219–223. doi:10.1038/nature24051.
- Owen, J.M., 2001. An open-source project for modeling hydrodynamics in astrophysical systems. *Comput. Sci. Eng.* 3, 54–59. doi:10.1109/5992.963428.
- Piekutowski, A.J., 2001. Debris clouds produced by the hypervelocity impact of nonspherical projectiles. *Int. J. Impact Eng.* 26, 613–624. doi:10.1016/S0734-743X(01)00122-1.
- Price, D.J., 2012. Smoothed particle hydrodynamics and magnetohydrodynamics. *J. Comput. Phys.* 231, 759–794. doi:10.1016/j.jcp.2010.12.011.
- Price, D.J., Wurster, J., Nixon, C., Tricco, T.S., Toupin, S., Pettitt, A., Chan, C., Laibe, G., Glover, S., Dobbs, C., Nealon, R., Liptai, D., Worpel, H., Bonnerot, C., Dipierro, G., Ragusa, E., Federrath, C., Iaconi, R., Reichardt, T., Forgan, D., Hutchison, M., Constantino, T., Ayliffe, B., Mentiplay, D., Hirsh, K., Lodato, G., 2017. PHANTOM: Smoothed particle hydrodynamics and magnetohydrodynamics code. *Astrophys. Source Code Libr.* arXiv:1709.002.
- Ramachandran, Prabhu, 2016. PySPH: a reproducible and high-performance framework for smoothed particle hydrodynamics, editor=sebastian benthall and scott rostrup. In: *Proceedings of the 15th Python in Science Conference*. pp. 122–129. doi:10.25080/Majora-629e541a-011.
- Rein, H., Liu, S.F., 2012. REBOUND: an open-source multi-purpose N-body code for collisional dynamics. *Astron. Astrophys.* 537, A128. doi:10.1051/0004-6361/201118085.
- Rein, Hanno, Spiegel, David S., 2015. las15: a fast, adaptive, high-order integrator for gravitational dynamics, accurate to machine precision over a billion orbits. *Mon. Not. R. Astron. Soc.* 446 (2), 1424–1437. doi:10.1093/mnras/stu2164.
- Rosswog, S., 2015. Sph methods in the modelling of compact objects. *Liv. Rev. Comput. Astrophys.* 1, 1. doi:10.1007/lrca-2015-1.
- Schäfer, C., Riecker, S., Maindl, T.I., Speith, R., Scherrer, S., Kley, W., 2016. A smooth particle hydrodynamics code to model collisions between solid, self-gravitating objects. *Astron. Astrophys.* 590, A19. doi:10.1051/0004-6361/201528060.
- Schäfer, C.M., Scherrer, S., Buchwald, R., Maindl, T.I., Speith, R., Kley, W., 2017. Numerical simulations of regolith sampling processes. *Planet. Space Sci.* 141, 35–44. doi:10.1016/j.pss.2017.04.015.
- Schäfer, C., Speith, R., Hipp, M., Kley, W., 2004. Simulations of planet-disc interactions using smoothed particle hydrodynamics. *Astron. Astrophys.* 418, 325–335. doi:10.1051/0004-6361:20034034.
- Schäfer, C., Speith, R., Kley, W., 2007. Collisions between equal-sized ice grain agglomerates. *Astron. Astrophys.* 470, 733–739. doi:10.1051/0004-6361:20077354.
- Schaller, Matthieu, Gonnet, Pedro, Draper, Peter W., Chalk, Aidan B. G., Bower, Richard G., Willis, James, Hausammann, Loïc, 2018. Swift: sph with inter-dependent fine-grained tasking. ascl:1805.020, arXiv:1805.020.
- Sironi, S.I., 2004. Conditions for collisional growth of a grain aggregate. *Icarus* 167.
- Speith, R., Kley, W., 2003. Stability of the viscously spreading ring. *Astron. Astrophys.* 399, 395–407. doi:10.1051/0004-6361:20021783.
- Springel, V., 2000. GADGET-2: A code for cosmological simulations of structure formation. *Astrophysics Source Code Library* arXiv:0003.001.
- Stellingwerf, R.F., Wingate, C.A., 1994. Impact modelling with SPH (invited paper). *Mem. Soc. Astron. Ital.* 65 (1117).
- Szewc, K., 2016. Smoothed particle hydrodynamics modeling of granular column collapse. *Granul. Matter* 19, 3. doi:10.1007/s10035-016-0684-3.
- Thun, D., Kley, W., 2018. Migration of planets in circumbinary discs. *Astron. Astrophys.* 616, A47. doi:10.1051/0004-6361/201832804.

- Thun, D., Kley, W., Picogna, G., 2017. Circumbinary discs: Numerical and physical behaviour. *Astron. Astrophys.* 604, A102. doi:[10.1051/0004-6361/201730666](https://doi.org/10.1051/0004-6361/201730666).
- Tillotson, J.H., 1962. *Metallic Equations of State for Hypervelocity Impact*. Technical Report General Atomic Report GA-3216. General Dynamics. San Diego, CA.
- von Mises, R., 1913. *Mechanik der festen Körper im plastisch deformablen Zustand*. *Göttin. Nachr. Math. Phys.* 1, 582–592.
- Wadsley, J.W., Keller, B.W., Quinn, T.R., 2017. Gasoline2: a modern smoothed particle hydrodynamics code. *Mon. Not. R. Astron. Soc.* 471, 2357–2369. doi:[10.1093/mnras/stx1643](https://doi.org/10.1093/mnras/stx1643).
- Wandel, O.J., Schäfer, C.M., Maindl, T.I., 2017. Collisional fragmentation of porous objects in planetary systems. In: *Proceedings of the First Greek-Austrian Workshop on Extrasolar Planetary Systems*, vol. 22. pp. 5–242.
- Yu, M., Huang, Y., Deng, W., Cheng, H., 2018. Forecasting landslide mobility using an sph model and ring shear strength tests: a case study. *Natural Hazards Earth Syst. Sci.* 18, 3343–3353. doi:[10.5194/nhess-18-3343-2018](https://doi.org/10.5194/nhess-18-3343-2018).
- Zander, C., Hopp-Hirschler, M., Nieken, U., 2018. Mesoscopic simulation and characterization of the morphological evolution in phase separating fluid mixtures. *Comput. Mater. Sci.* 149, 267–281. doi:[10.1016/j.commatsci.2018.03.019](https://doi.org/10.1016/j.commatsci.2018.03.019).