METHODOLOGY



# Polygonal contact model revisited: notes on usage and improved implementation

Gerhard Hippmann<sup>1</sup>

Received: 29 November 2022 / Accepted: 28 February 2023 © The Author(s) 2023

## Abstract

The polygonal contact model is an established contact algorithm for multibody dynamics based on polygonal surfaces. Some guidance for its practical use is presented. In particular, assignment and discretization of the surface meshes are discussed with regard to correct results and optimal efficiency. Moreover, notes on stiffness and damping parameters are given.

An improved implementation of the polygonal contact model is introduced. The new algorithm goes without quite complicated intersection construction steps. As a consequence, it is less complex, more robust and usually more efficient than the classic implementation.

Application examples of varied complexity demonstrate both notes on usage and improved implementation of the polygonal contact model.

Keywords Contact mechanics · Computer graphics · Collision detection

# 1 Introduction

The polygonal contact model (PCM) is an established contact algorithm for multibody dynamics that allows to evaluate compliant contact between complexly shaped bodies [1]. In contrast to standard methods based on point contacts [2], PCM implements an areal approach that can handle multi-point contacts as well as conforming contacts with arbitrary contact patch shape. Since PCM uses polygonal surfaces (3D meshes), which can be exported easily from CAD software, it is quite straightforward to use in technical applications.

PCM has been implemented in several academic and commercial multibody tools, partly with substantial extensions like surface height correction, flexible body support and self contact [3–5]. Its areal approach and polygonal discretisation have been taken over in other contact models [6–8]. However, this publication takes the original version as a starting point. So the reader is referred to [1] or [9] to get acquainted with the basic terms and concepts used in what follows.

Section 2 gives notes on the usage of PCM. In Sect. 3 a new improved contact detection algorithm is presented. Section 4 shows some application examples. In the end, Sect. 5 concludes this article.

G. Hippmann gerhard.hippmann@dlr.de

<sup>&</sup>lt;sup>1</sup> Institute of System Dynamics and Control, German Aerospace Center, Münchener Str. 20, 82234 Weßling, Oberpfaffenhofen, Germany



**Fig. 1** Sectional view of contact elements (red arrows) of a large penetration sphere-sphere contact with penetration line start points (dots at base triangles) and intersection points (arrowheads at target surface). (Colour figure online)

# 2 Notes on usage

Over the past few years, PCM has been utilised in many research and industrial applications [10–14]. This section summarises some useful experience and gives modelling hints for optimal robustness and efficiency of the method.

## 2.1 Base/target assignment and discretisation

The contact detection of PCM uses a *base/target*<sup>1</sup> approach, i.e. one of the contact surfaces is defined as base and the other one as target. Based on this assignment, the PCM contact evaluation encompasses the following steps:

- 1. All triangles of the base surface for which the geometric barycentres (centroids) are located on the inner side of the target surface are processed as *contact elements*.
- 2. At each contact element a *penetration line* is constructed. It starts at the triangle barycentre and points along its normal direction to the inside of the base surface.
- 3. The intersection points of the penetration lines and the target surface are computed.
- 4. The distances between the penetration line start points (at the base triangle barycentres) and the intersection points are used as penetrations of the contact elements (Fig. 1).
- 5. A spring-damper force law is applied along the penetration lines of the contact elements.
- 6. A regularised friction force law is applied perpendicular to the penetration lines of the contact elements.
- 7. The total contact force is calculated as a sum of all contact element forces.

As a consequence of this procedure, the polygonal discretisation of the base surface defines the scanning resolution of the contact patch. Just like every discretisation, it should be chosen with care for an optimal trade-off between accuracy and efficiency of the contact force evaluation. Thus, surfaces with large polygons, which may e.g. occur in case of planar regions, are not suitable as a base surface because of too coarse contact patch discretisation. The problem might be solved by remeshing the surface accordingly (i.e. using more polygons than required for visualisation) or by interchanging the base/target assignment.

<sup>&</sup>lt;sup>1</sup>Called *master/slave* in former publications.



Fig. 2 2D example for penetrations (top) and resulting contact element force history (bottom) in case of a) sharp edges, b) chamfered edges and c) interchanged base/target assignment

As a matter of principle, large polygons are not a functional issue for target surfaces. However, in Sect. 4.2 an example is presented, which shows that using balanced base and target discretisations can be more efficient than very large target triangles.

## 2.2 Sharp edges

PCM's discretisation of the contact patch can lead to bad results in case of sharp edges of the target surface, i.e. angles of 90° or more between adjacent polygons. The problem is sketched in Fig. 2 regarding two boxes sliding along each other.

In case a) there are sharp edges of the target surface resulting in discontinuous penetrations of the contact elements of the base surface: When the front edge of the upper box passes a contact element, its penetration suddenly jumps to a non-zero value, and at the rear edge it jumps back to zero again. This leads to unphysical behaviour because potential energy conservation of the elastic force law is violated.

There are two workarounds to address this problem: Sharp edges of the target surface might be sanitised by chamfering (case b)) or the base/target assignment might be interchanged (case c)) so that discontinuous penetrations do not occur anymore. If both of them are not an option, the original PCM is not suitable for the application.

## 2.3 Coordinate reference frame assignment

PCM contact surfaces are labelled E and F. From the modelling point of view this assignment is entirely interchangeable, i.e. results are identically except for numerical round-off errors. In particular the base/target assignment can be chosen independently of the E/F assignment.

However, the latter can have great impact on the computation effort because the reference frame of surface E is used for coordinate representation in all collision detection and contact element vector operations. As a consequence, the numerical efficiency of PCM is usually optimal if the surface with a greater number of triangles is defined as contact surface E, such that the effort for costly coordinate transformations (applied to surface F data only) is minimised. This effect is significant in the example presented in Sect. 4.3.

Note that the influence of the E/F assignment on the computation effort does not only depend on the number of surface triangles, but also on the size of the triangles and on the quality of the bounding box hierarchies. So if efficiency is important in an application, it might be worth to compare both options by test simulations.

#### 2.4 Stiffness and damping

For elastic normal contact forces, PCM implements the elastic foundation model [15]. The idea is to cover the contacting bodies with thin elastic layers which do not propagate deformation or stress in tangential direction. Contact elements are used to discretise the contact patch. Therefore, the normal direction and the area of the base polygon represent a subdivision of a combined elastic layer in which constant normal pressure is assumed. Thus, the elastic force law of a contact element k is defined by

$$F_{ck} = c_l \cdot A_k \cdot u_{nk} \tag{1}$$

with layer stiffness  $c_l$ , contact element area  $A_k$  and penetration  $u_{nk}$ . This is equivalent to a linear spring with the stiffness given by the product of  $c_l$  and  $A_k$ . According to the elastic foundation model, the layer stiffness of each body is defined by

$$c_l = \frac{K}{b} \tag{2}$$

with the elastic modulus

$$K = \frac{1 - \nu}{(1 + \nu) \cdot (1 - 2\nu)} \cdot E$$
(3)

as a material parameter and the layer thickness b.

An advantage of this approach is that common material properties (Young's modulus E and Poisson's ratio  $\nu$ ) and the easily comprehensible layer thickness b are sufficient to define the elastic contact properties. However, in most applications there is no comparatively soft layer on the body surfaces, and finding suitable values for b can be difficult.

Note that E appears in the nominator and b in the denominator of the layer stiffness, i.e. 1/b acts like a scaling factor and therefore has great influence on the contact stiffness. So, if the elastic contact properties are crucial for system dynamics, experiments may be used to tune the layer stiffness.

The same holds for the damping characteristics. Deduced from the stiffness force law (1), the damping force of a PCM contact element is calculated by

$$F_{dk} = d_l \cdot A_k \cdot \dot{u}_{nk},\tag{4}$$

where  $d_l$  is the area related layer damping. Similar to the layer thickness *b*, this parameter cannot be specified directly according to material and geometry data, but requires test simulations to find plausible or verified values. However, this is typical for damping and restitution coefficients in multibody dynamics. Note that there are options in PCM to avoid discontinuous and tensile contact forces caused by the damping share.

#### 3 Improved implementation

In [1] PCM's classic *intersection construction* algorithm for contact element generation is presented, which consists of the following steps:

1. All intersection lines of the surface triangles are determined by collision detection based on object oriented bounding box trees.

- 2. Intersection polygons are constructed by assembling the intersection lines.
- 3. Active areas of the surfaces are determined by searching recursively for adjacent triangles on the inner side of the intersections.
- 4. For each active triangle of the base surface, a ray-triangle intersection test between its barycentre normal and the active triangles of the target surface is performed.
- 5. For valid intersections, contact elements are generated as described in Sect. 2.1.

This algorithm works fine for most applications. However, in extensive practical use it has shown several weak points:

- The overall procedure is quite complex and includes a lot of geometric calculations, which can fail in extreme cases (e.g. vanishing intersection angles, conforming contact with small penetration/high stiffness, degenerated surface polygons). So numerical robustness and computational effort are not optimal.
- For searching active surface areas, information about adjacent triangles is required. In case of "unclean" polygonal surfaces (e.g. non-manifold edges, cracks, duplicate vertices) generation of this data would fail.
- The numerical complexity of the ray-triangle tests grows quadratically with the number of active surface triangles. Hence efficiency becomes poor in case of large contact patches with fine discretisation.
- In every contact, evaluation memory is allocated and deallocated for intersections, active areas and contact elements. This involves additional computation effort.

To overcome these disadvantages, a new algorithm for contact element generation has been developed. It is called *penetration detection* because the basic idea is to find the contact elements directly by collision detection and do without construction of intersection polygons and active surface regions.

Penetration detection is based on a minor modification of the classic collision detection. Its basic functionality of hierarchical intersection tests between object oriented bounding box trees remains entirely unchanged. The substantial difference is that the bounding box tree of the base surface is not generated with its triangles, but with its *penetration lines* as leaf elements. These are defined for every triangle as connection lines between the barycentre and a maximum penetration point which is located on the triangle normal at a user defined distance on the inner side of the surface. On the other hand, the bounding box tree of the target surface is still generated with its triangles (Fig. 3).

As a consequence, the collision detection identifies candidates for triangle-penetration line intersections instead of triangle-triangle intersections. Thus, in case of colliding bounding boxes, the finalising triangle-triangle intersection test is replaced with a ray-triangle intersection test similar to [16]. If this test yields a penetration, i.e. an intersection of a base penetration line and a target triangle, the contact element data of the base triangle is activated with the computed penetration value.

Evidently, this method allows to skip the complex intersection construction steps and therefore is in general more robust and efficient than the classic solution. Moreover, it can be easily implemented without dynamic memory allocation by storing contact element data persistently together with the base triangles. So it avoids all weak points listed above and simplifies implementation remarkably. Incidentally, the penetration detection algorithm is similar to ray tracing and thus suitable for high performance GPU implementations.

On the other hand, the maximum penetration is required as an additional user parameter, which must be chosen with care. Since contact element penetrations exceeding this limit value are not detected by the algorithm, it must be greater than the maximum contact penetration that happens in the entire simulation. PCM prints an info message about the greatest penetration after simulation so that the chosen parameter value can be checked.



Fig. 3 Penetration lines (red) and leaf bounding boxes (grey) of a separated sphere-sphere contact. (Colour figure online)



Fig. 4 2D examples for valid (red) and invalid (orange) contact penetrations. (Colour figure online)

Note that iterative solvers with step-size control can induce contact evaluations with larger penetrations than those visible in the results. As a consequence, some safety reserve (e.g. 50%) should be provided and a step-size limit (e.g. 10 ms) should be defined. Also note that using greater maximum penetration values than necessary does not affect the results, but implicates additional computation effort caused by unnecessarily large bounding boxes and handling of invalid penetrations.

The latter deals with formal intersections between penetration lines and target triangles that do not represent contact deformation. As sketched in Fig. 4, there are two cases of invalid penetrations: In case a) the target triangle is hit from the outside of the surface, and in case b) the intersection point is located at the rear side of the target body.

Storing the contact element data together with the base triangles allows to sort out invalid penetrations as follows: When the first intersection is detected, the contact element penetration is stored and its status is changed from *inactive* to *active* (if the dot product of the base and target triangle normals is negative) or *invalid* (positive dot product, case a)). Further intersections are ignored if their penetration is greater than the contact element penetration (case b)). Otherwise they are treated in the same way as the first intersection and status and penetration of the contact element are updated.



Fig. 5 Start and end position of the torus and sphere simulation

# 4 Application examples

The penetration detection algorithm has been added as a new option in the C implementation of PCM. Moreover, it has been implemented in the modern programming language Julia [17, 18] as a force element of *Modia3D* [19], the multibody extension of the *Modia* open source modelling language for multi-domain physical systems [20, 21]. Due to the quite low complexity of the penetration detection approach, the Julia implementation is compact and coherent.

In the following subsections, application examples are presented to validate the robustness and efficiency of the method and its implementation. All simulations have been computed on a Linux laptop with Intel Core i7-9850H CPU. Animations are available on the PCM website [22].

## 4.1 Torus and sphere

The first example is a bouncing ball scenario with a torus as ground geometry. A sphere  $(R = 1 \text{ m}, m = 10 \text{ kg}, J = 0.5 \text{ kgm}^2)$  with six degrees of freedom starts at 1.2 m vertical and 0.1 m lateral displacement with respect to the centre of the inertia-fixed torus (R = 1 m, r = 0.3 m). The torus surface is described by 4,096 triangles and the sphere by an icosahedron with 500 triangles. A PCM contact with the torus as base and the sphere as target is defined  $(E = 10^6 \text{ Pa}, \nu = 0.4, b = 0.01 \text{ m}, u_{max} = 0.03 \text{ m}, d_l = 10^4 \text{ Ns/m}^3, \mu = 0.3).$ 

In the simulation, the ball bounces onto the torus in varying motion phases before it comes to rest at the centre after about 5 s. In the end, there is an annular contact patch that could not be handled by contact point based methods (Fig. 5).

In both contact element generation implementations, intersection construction and penetration detection, the penetrations are calculated with the same ray-triangle intersection algorithm and the contact element forces are summed up with increasing triangle index. As a consequence, they yield bit-identical results when executed on the same platform. This is a clear advantage for verification of results and performance.

The costly part of PCM contact force evaluations is the detection of active contact elements. In Fig. 6 the number of collision and intersection tests is plotted. Obviously, penetra-



Table 1 Torus and sphere computation effort (CPU-time/RHS-evaluation [us])

MBS tool/PCM language	Intersection construction	Penetration detection
Commercial solver/C	888	739
Modia3D/C	1,106	887
Modia3D/Julia	-	1,047

tion detection requires less bounding box collision tests because the penetration line bounding boxes are smaller than the triangle bounding boxes. As a consequence, the number of intersection tests is also lower and the overall performance is better.

For computation effort analysis, the mean time of equations of motion evaluations is regarded (Table 1). Thus, the penetration detection algorithm is about 18% more efficient than intersection construction. Using the C implementation, the evaluation time of Modia3D is about 25% larger than with a commercial MBS tool, i.e. the reference solver is somewhat more efficient than Modia3D. Incidentally, the Julia implementation is about 18% less efficient than the C implementation. This is considerable for a convenient scripting-like programming language.

The results of the torus-sphere example are typical for all the two body contact scenarios simulated for verification of the new implementations.

#### 4.2 Bouncing bubbles

The bouncing bubbles scenario (Fig. 7) has already been presented in [1] and has proven its worth for implementation verification since then. Again, simulation yields bit-identical results for intersection construction, penetration detection and also for arbitrary mixes of both. However, in contrast with all other application examples, penetration detection showed lower efficiency than intersection construction. So this behaviour was investigated further.

The basic idea of collision detection is to avoid relatively costly (triangle-triangle or raytriangle) intersection tests as far as possible. This is achieved by prepending hierarchical bounding box collision tests with low computation effort. The bouncing bubbles model defies this concept to a large extend because the container consists of only four large triangles



Fig. 8 Container surface discretisations: 0 to 5 midpoint subdivisions

face subdivisions				
Subdivisions	Triangles	Intersection construction	Penetration detection	
0	4	979	1,156	
1	16	774	870	
2	64	626	585	
3	256	551	436	
4	1,024	594	329	
5	4,096	706	398	
6	16,384	998	463	

Table 2Bouncing bubbles computation effort (CPU-time/RHS-evaluation  $[\mu s]$ ) dependent on container surface subdivisions

used as target surfaces. As a consequence, there are only four large leaf bounding boxes, which enclose all or very many triangles of the bubble surfaces most of the time.

In this situation, penetration detection implicates more computation effort, because it directly invokes a huge number of ray-triangle tests, whereas intersection construction performs somewhat cheaper triangle-triangle tests.

This effect has been investigated by a series of test simulations with varied discretisation of the container surface. For this, the triangles have been subdivided into four triangles by new vertices at the midpoints of their edges (Fig. 8). This scheme has been applied six times resulting in container surfaces with 16 to 16,384 triangles.

Since numerical round-off errors (in normal direction and intersection point calculations) lead to result differences that increase over simulation time, only two bounces are regarded and the simulation is stopped after 1.2 s. Table 2 compares the computation efforts for both contact element generation methods dependent on the container surface discretisation.

As expected, efficiency is improved by using finer container surface discretisations because collision detection works better. At 64 triangles and more, penetration detection be-



comes more efficient than intersection construction. After an optimum at 256 resp. 1,024 triangles, computation effort increases again because the total number of bounding box and intersection tests becomes predominant. This effect is more significant for the intersection construction algorithm because of the quadratically growing number of active triangle combinations.

The plot in Fig. 9 shows the number of penetration detection ray-triangle tests dependent on the number of container surface subdivisions. Obviously, these tests mainly determine the computation effort.

So, on the whole, penetration detection is also clearly more efficient than intersection construction for the bouncing bubbles scenario if balanced surface discretisations are used.

#### 4.3 DLR Scout rover

The third example scenario is a simulation of the DLR Scout rover [23] (Fig. 10). The Scout is a modular rover that has been developed by DLR's Institute of System Dynamics and Control for extraterrestrial cave exploration missions. It uses six "rimless wheels" that combine the advantages of wheeled and legged robots. Simulation of the Scout is markedly useful because it allows to study the functionality and efficiency of the rimless wheel propulsion system for varying terrain, payload, gravity, wheel rotation speed and phase and others.

The Scout multibody model has been implemented in Modia3D and a commercial MBS tool according to [24]. PCM is used for the 18 contact pairs between feet and terrain. The foot surface has been created by triangulation of a chamfered cylinder segment FreeCAD [25] model with 570 triangles (Fig. 11). The non-convex terrain surface formed by 32,769 triangles has been generated by a fractal terrain perturbation algorithm [26] implemented in MeshLab [27].

The PCM force elements define the feet as base and the terrain as target surfaces ( $E = 5 \cdot 10^6$  Pa,  $\nu = 0.4$ , b = 0.01 m,  $u_{max} = 0.01$  m,  $d_l = 2 \cdot 10^4$  Ns/m<sup>3</sup>,  $\mu = 0.7$ ). The wheels rotate with a constant angular velocity of 2 rad/s. In the simulation the Scout travels about 9 m in 30 s. The evaluation of the contacts between the feet and the complexly shaped terrain works robustly and efficiently (commercial MBS solver with 4 CPU cores: realtime factor 14.4).

**Fig. 10** DLR Scout rover on fractal terrain simulation







In this example the reference frame assignment discussed in Sect. 2.3 has great influence on the computational effort of PCM. The mean computation time for the evaluation of the equations of motion is 439  $\mu$ s in case of E/F = terrain/feet and 1,460  $\mu$ s for E/F = feet/terrain (commercial solver with 4 CPU cores), i.e. the computation effort is more than three times less if the reference frame of the larger terrain surface is used for internal vector representation.

# 5 Conclusions

PCM is applicable for many contact problems because it uses highly available polygon meshes for surface definition and has no restrictions regarding shape and dimension of contact patches. However, some know-how about its functioning is required to achieve optimal accuracy, robustness and efficiency. Moreover, it is not always clear how to choose the elastic layer thickness and the area-related damping factor. Section 2 gives hints and background information on these issues, which should be addressed in further development.

The new penetration detection method presented in Sect. 3 is simpler, more efficient and more robust with respect to geometric singularities and imperfect mesh topology than the classic algorithm. It conjoins contact element generation with collision detection and goes without complicated intersection construction steps. However, the physical model remains unchanged. Therefore it is applicable in the same way as the classic method.

In Sect. 4 three simulation scenarios of varied complexity are used to demonstrate the application of PCM. Torus and sphere as well as bouncing bubbles are academic examples that allow to study numerical properties and phenomena of the method. Finally the DLR Scout rover simulation shows the usage of PCM in a current research project.

Author contributions G.H. wrote the manuscript text and prepared the figures.

**Funding** Open Access funding enabled and organized by Projekt DEAL. No funding was received to assist with the preparation of this manuscript.

Data Availability Not applicable.

#### Declarations

Ethical approval Not applicable.

**Competing interests** The author declares no competing interests.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

## References

- Hippmann, G.: An algorithm for compliant contact between complexly shaped bodies. Multibody Syst. Dyn. 12, 345–362 (2004). https://doi.org/10.1007/s11044-004-2513-4
- Flores, P.: Contact mechanics for dynamical systems: a comprehensive review. Multibody Syst. Dyn. 54, 127–177 (2022). https://doi.org/10.1007/s11044-021-09803-y
- Ebrahimi, S.H., Hippmann, G., Eberhard, P.: Extension of the Polygonal Contact Model for flexible multibody systems. Int. J. Appl. Math. Mech. 1, 33–50 (2005). https://www.researchgate.net/publication/ 228524527\_Extension\_of\_the\_polygonal\_contact\_model\_for\_flexible\_multibody\_systems
- Toso, M., Rossi, V.: ESA multibody tool for launchers and spacecrafts: lesson learnt and future challenges. In: Proceedings of the 5th Joint International Conference on Multibody System Dynamics, 24-28 June 2018, Lisbon, Portugal (2018). http://imsd2018.tecnico.ulisboa.pt/Web\_Abstracts\_IMSD2018/pdf/WEB\_PAPERS/IMSD2018\_Full\_Paper\_93.pdf
- Dassault Systèmes Simulia Corp.: Simpack force element type 199 'Polygonal Contact'. SIMULIA User Assistance 2023x (2022)
- Choi, J., Ryu, H.S., Kim, C.W., Choi, J.H.: An efficient and robust contact algorithm for a compliant contact force model between bodies of complex geometry. Multibody Syst. Dyn. 23, 99 (2010). https:// doi.org/10.1007/s11044-009-9173-3
- Masterjohn, J., Guoy, D., Shepherd, J., Castro, A.: Velocity Level Approximation of Pressure Field Contact Patches (2021). arXiv:2110.04157
- Li, Z., Zeng, X., Wen, T., Zhang, Y.: Numerical comparison of contact force models in the discrete element method. Aerospace 9, 737 (2022). https://doi.org/10.3390/aerospace9110737
- Hippmann, G.: An algorithm for compliant contact between complexly shaped surfaces in multibody dynamics. In: Proceedings of the 1st ECCOMAS Multibody Dynamics Thematic Conference, 1-4 July 2003, Lisbon, Portugal (2003). http://pcm.hippmann.org/doc/eccomas03\_hippmann.pdf
- Krenn, R., Gibbesch, A., Hirzinger, G.: Contact dynamics simulation of rover locomotion. In: Proceedings of the 9th International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS), 22-26 September 2008, Nice, France (2008). https://elib.dlr.de/55783/
- Shen, L.J.: Multibody Contact Simulation of Constant Velocity Plunging Joint. IWM Institutsmitteilung 35. Institut für Maschinenwesen, Technische Universität Clausthal, Clausthal-Zellerfeld, Germany (2010). https://www.imw.tu-clausthal.de/fileadmin/Forschung/InstMitt/2010/Konstruktion\_und\_ Berechnung\_von\_Maschinenelementen/She\_10.pdf
- Hajžman, M., Polach, P., Jankovec, J.: Modelling and simulation of rigid bodies transportation by means of rotating flexible rollers. Meccanica 47, 455–468 (2012). https://doi.org/10.1007/s11012-011-9456-7

- Zou, R., Luo, S., Ma, W.: Simulation analysis on the coupler behaviour and its influence on the braking safety of locomotive. Veh. Syst. Dyn. 56, 1747–1767 (2018). https://doi.org/10.1080/00423114.2018. 1435893
- Zhang, Y., Li, J., Zeng, X., Wen, T., Li, Z.: High-fidelity landing simulation of small body landers: modeling and mass distribution effects on bouncing motion. Aerosp. Sci. Technol. 119, 107149 (2021). https://doi.org/10.1016/j.ast.2021.107149
- 15. Johnson, K.L.: Contact Mechanics. Cambridge University Press, Cambridge (1985)
- Möller, T., Trumbore, B.: Fast, minimum storage ray-triangle intersection. J. Graph. Tools 2, 21–28 (1997). https://doi.org/10.1080/10867651.1997.10487468
- Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: a fresh approach to numerical computing. SIAM Rev. 59, 65–98 (2017). https://julialang.org/assets/research/julia-fresh-approach-BEKS.pdf
- 18. Julia website: https://julialang.org/
- Neumayr, A., Otter, M.: Component-based 3D modeling of dynamic systems. In: Proceedings of the American Modelica Conference, 9-10 October 2018, Cambridge MA (2018). https://doi.org/10.3384/ ecp18154175
- Elmqvist, H., Otter, M., Neumayr, A., Hippmann, G.: Modia equation based modeling and domain specific algorithms. In: Proceedings of 14th Modelica Conference, 20-24 September, Linköping (2021). https://doi.org/10.3384/ecp2118173
- 21. Modia website: http://modiasim.org/
- 22. PCM website: http://pcm.hippmann.org/
- DLR Scout rover website: https://www.dlr.de/sr/en/desktopdefault.aspx/tabid-13261/23182\_read-53800/
- 24. Pignede, A., Lichtenheldt, R.: Modeling, simulation and optimization of the DLR scout rover to enable extraterrestrial cave exploration. In: Proceedings of the 6th Joint International Conference on Multibody System Dynamics (IMSD) and the 10th Asian Converence on Multibody System Dynamics (ACMD), 16-20 October 2022, New Delhi (2022).
- 25. FreeCAD website: https://www.freecadweb.org/
- 26. Ebert, D.S., Musgrave, F.K., Peachey, D., Perlin, K., Worley, S.: Texturing and Modeling: a Procedural Approach. The Morgan Kaufmann Series in Computer Graphics, San Francisco (2002)
- Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., Ranzuglia, G.: MeshLab: an opensource mesh processing tool. In: Sixth Eurographics Italian Chapter Conference, pp. 129–136 (2008). MeshLab website. https://www.meshlab.net/

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.