Parasolid V36.0

What's New in Parasolid

July 2023

Important Note

Tel: +44 (0)1223 371555
email: parasolid.support.plm@siemens.com
Web: www.parasolid.com

SIEMENS

Trademarks

Table of Contents

What's New in Parasolid V36.0

1

1.1 Introduction

This document describes functional enhancements and interface changes that have been added to Parasolid V36.0 and gives information on any other matters that are specific to this release.

1.2 Lattices

This section describes improvements that have been made to Parasolid's support for representing lattices in Parasolid V36.0. Full details on Parasolid lattices can be found in Chapter 20, "Lattice Geometry", of the Parasolid *Functional Description*.

Note: A Convergent Modelling license is required for all operations on lattices.

1.2.1 Embedding a lattice in a region

In V36.0 lattices can be embedded in a region as boundary geometry. This functionality can provide additional context and information about the construction of the associated body. For example, a lattice with a solid interior in a void region could be used as a lightweight but strong structure.

Parasolid also provides additional support in allowing you to enquire if a given lattice is embedded in a region, and if a given region has an embedded lattice.

Note: Please be aware there will be limited scope as we implement the first stages of this functionality in V36.0. Many modelling operations and other enquiries work on bodies with embedded lattices, though modelling operations will not alter a region with an embedded lattice in this release. Operations that are restricted from working on embedded lattices will return PK_ERROR_lattice_geometry. In future releases we aim to implement a full set of functions to work on bodies with embedded lattices.

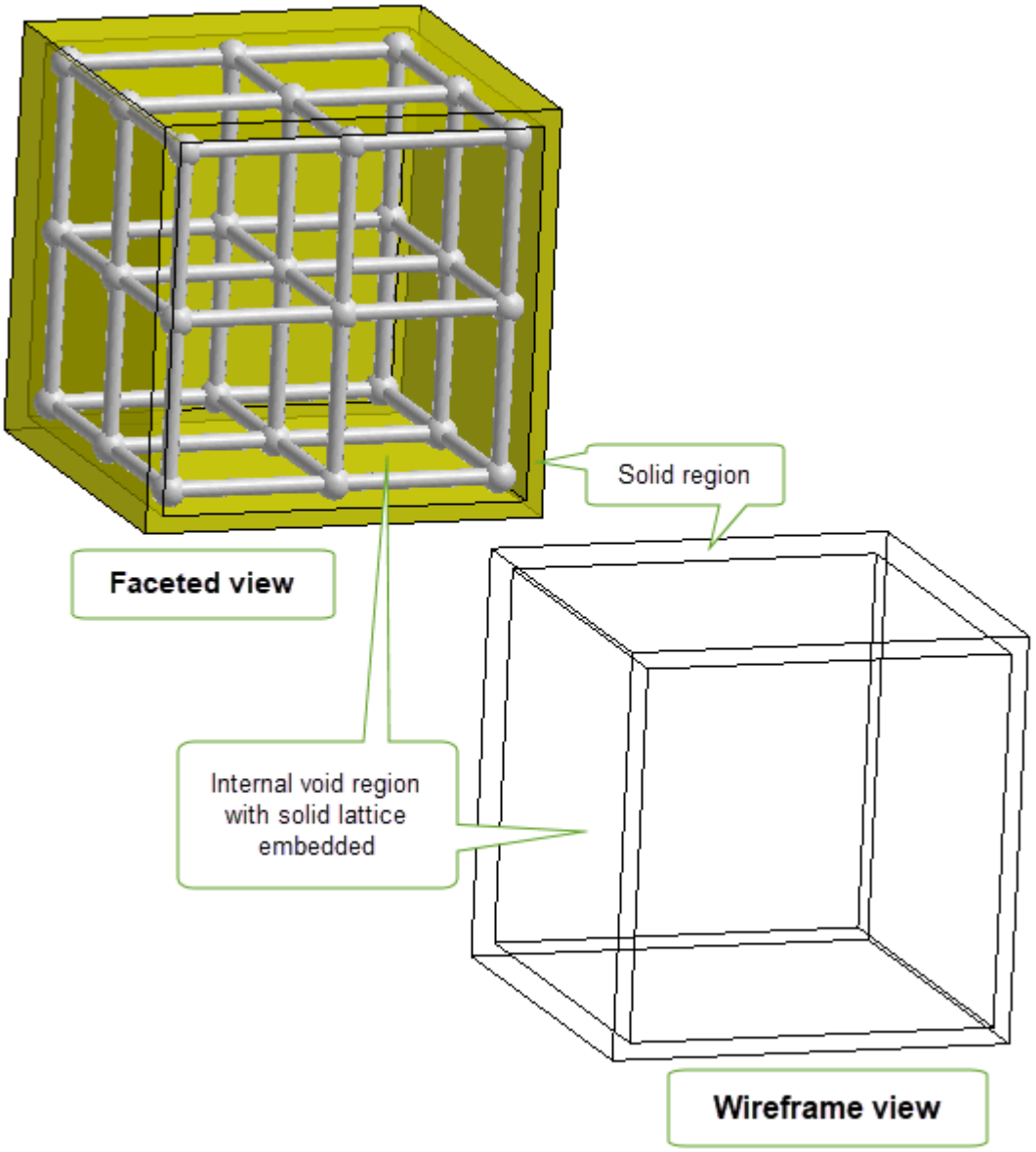

Figure 1–1 *A lattice embedded in the internal void of a hollowed cube*

For more information see Section 20.10, "Lattices in regions" in the Parasolid *Functional Description*.

1.2.1.1 Interface changes

Changed	PK_FRAME_t, PK_TOPOL_t, PK_ATTDEF_sf_2_t, PK_GROUP_create_from_entities, PK_GROUP_create_from_entities_2, PK_PART_delete_attribs, PK_TOPOL_ask_entities_by_attdef, PK_TOPOL_categorise_geom, PK_PART_find_entity_by_ident,
New	PK_FACE_ask_type, PK_FRAME_ask_owner, PK_FRAME_ask_geometry, PK_FRAME_ask_sense, PK_FRAME_ask_body, PK_BODY_ask_frames, PK_FRAME_reverse, PK_REGION_ask_lattices, PK_LATTICE_ask_regions, PK_REGION_ask_type, PK_REGION_embed_lattices, PK_REGION_remove_lattice, PK_TOPOL_find_frames, SDL/TYSA_FRAME_DENSITY

1.2.2 Using cylindrical patterns to create a patterned lattice

In V36.0, Parasolid has extended its support for patterned lattices by enabling the creation of a patterned lattice based on a cylindrical pattern rather than a rectilinear one. This is useful where you wish to pattern over periodic surfaces.

In addition, the errors returned from PK_LATTICE_make_patterned and PK_BODY_make_patterned have been refined to provide more guidance on the nature of the problems with the inputs.

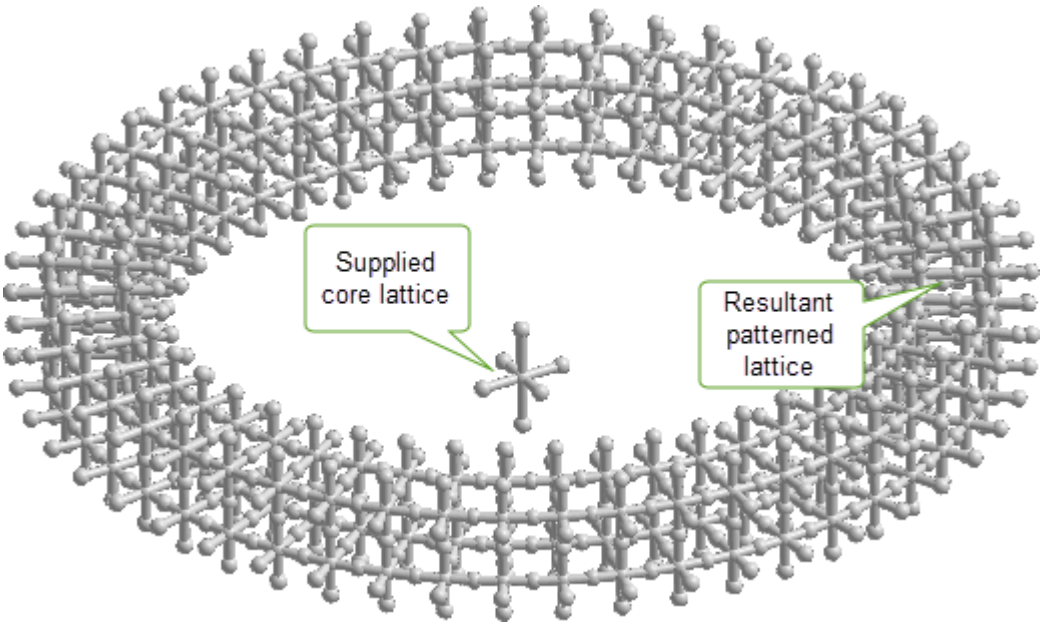

Figure 1–2 *Creating a patterned lattice using cylindrical patterns*

For more information, see Section 20.6.2.2, “Producing an axial pattern” in the Parasolid *Functional Description*.

1.2.2.1 Interface changes

Changed	PK_LATTICE_make_patterned_o_t, PK_BODY_make_patterned: new value pk_pattern_axial_c PK_pattern_form_t: new union member PK_pattern_axial_o_t

1.2.3 Enlarging a lattice

In Parasolid V36.0, you can enlarge a supplied lattice. The resultant enlarged lattice will be trimmed to fit inside the size box or a user supplied trimming box.

1.2.3.1 Interface changes

None, but PK_GEOM_enlarge now supports lattices.

1.2.4 Creating facet geometry from lattice geometry

Parasolid V36.0 allows you to convert an array of regions or a single input body containing lattice geometry, into a collection of non-lattice regions bounded by mesh faces. This functionality can be useful if, for example, you want to export the input body to a format that does not support lattices or you want to use other functionality that supports meshes but does not support lattice geometry.

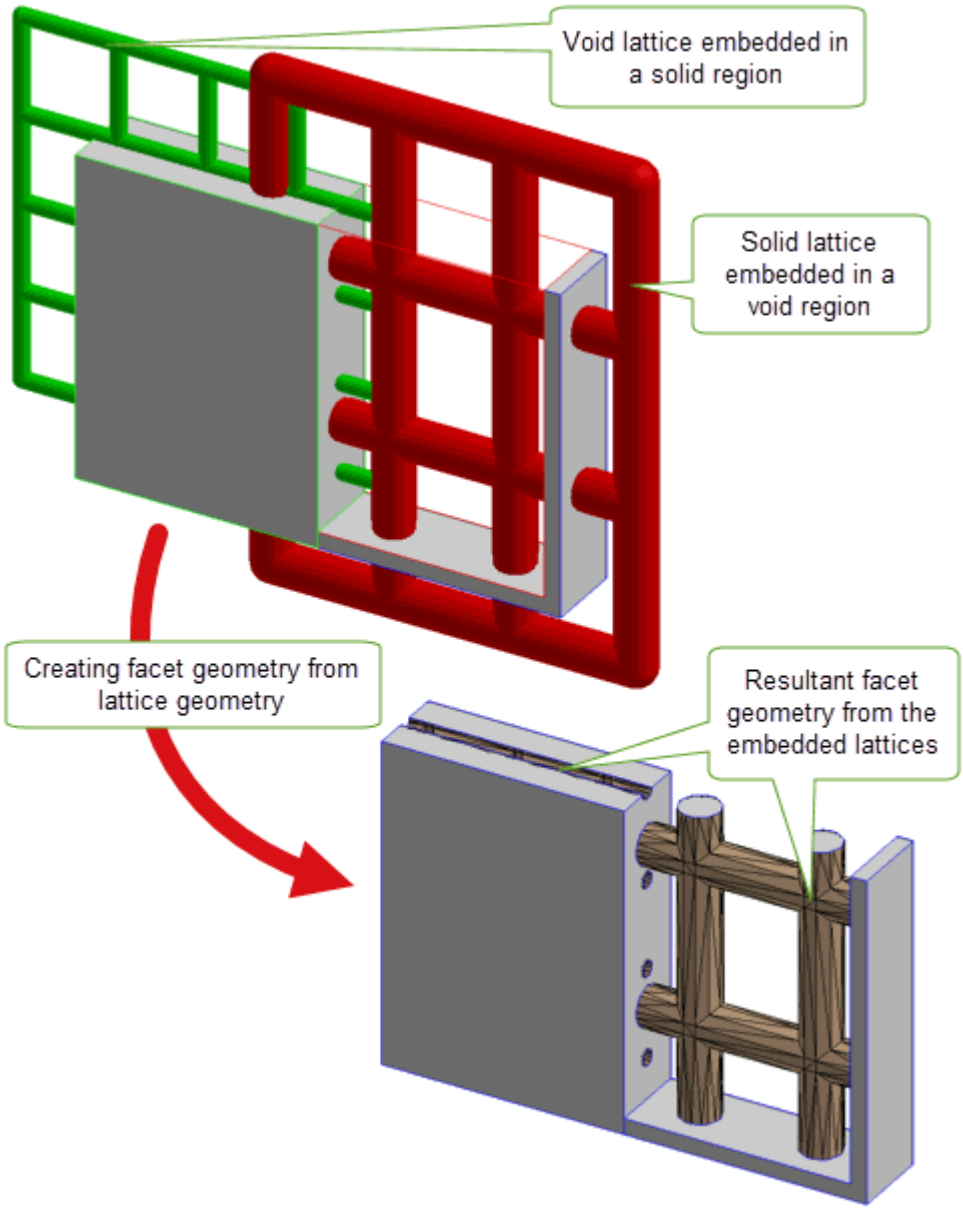

Figure 1–3 *Creating facet geometry by imprinting the frames associated with a region*

For more information, see Section 20.10.4, “Imprinting embedded lattices” in the Parasolid *Functional Description*.

1.2.5 Returning lattice geometry that exists in a given partition

In V36.0, Parasolid can return whether a partition contains lattice geometry. You can optionally return parts that contain lattice geometry or orphan lattices.

1.2.5.1 Interface changes

New	PK_PARTITION_has_lattices

1.2.6 Functional areas that support lattices

In Parasolid V36.0 bodies containing lattice geometry can be transformed if the transform is rigid or uniform scale.

1.2.6.1 Interface changes

None, but changes also affect PK_BODY_enlarge and PK_BODY_transform_2

1.3 Cellular body support

This section describes improvements that have been made to Parasolid's cellular body support functionality for Parasolid V36.0. Further details can be found in Volume 3: Basic Concepts.

1.3.1 Enquiring if a general body is a cellular body

A cellular body is a manifold or general body that has one or more solid regions potentially separated by partition faces. As of V36.0, Parasolid now allows you to query if a given body is a cellular body.

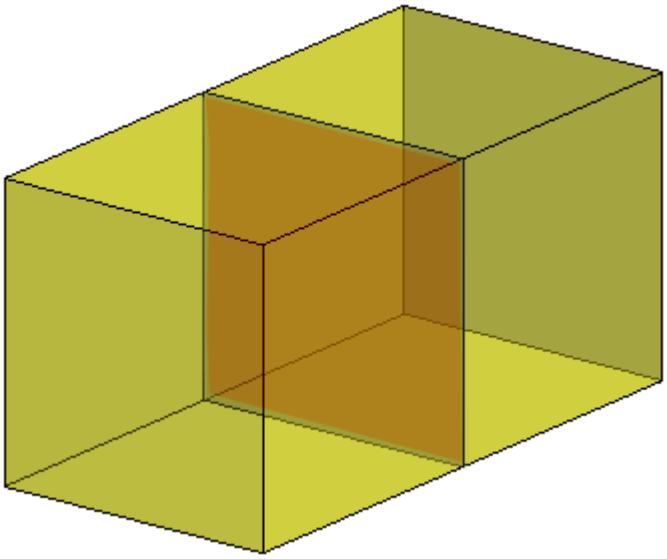

Figure 1–4 *Cellular body*

For more information, Section 15.7.1, “Cellular bodies” in the Parasolid *Functional Description*.

1.3.1.1 Interface changes

New	PK_BODY_is_cellular

1.3.2 Treating cellular bodies as manifold bodies

Parasolid has enhanced its support of cellular bodies by allowing your application to choose to treat cellular bodies either as general bodies or as manifold bodies during a Parasolid session.

When treating a cellular body as manifold, all geometric and many topological enquiries will work and return results that are consistent with the manifold view of the model. If your application supplies an entity that is not consistent with this manifold view, PK_ERROR_hidden_by_guise will be returned.

For more information on cellular guises, see Section 15.10, “Treating cellular bodies as manifold bodies” in the Parasolid *Functional Description*.

Note: Please be aware there will be limited scope as we implement the first stages of this functionality in V36.0. Modelling operations and other enquiries, such as region and shell may not be called when cellular bodies are being treated as manifold. Operations that are restricted from working when a manifold guise is set will return PK_ERROR_cellular_body or PK_ERROR_guise_not_supported.

1.3.2.1 Interface changes

New	PK_SESSION_ask_cellular_guise, PK_SESSION_set_cellular_guise

Changes span across many of the Parasolid APIs, too numerous to mention.

1.3.3 Functional areas that support cellular bodies

Parasolid V36.0 has improved its support for cellular bodies in the following areas:

- Simple boolean operations are allowed on cellular bodies with a manifold guise
- You can create outlines from faces of cellular bodies selected by the guise
- Improved support for locally manifold blending on cellular bodies
- Parasolid functions associated with attributes and groups operate appropriately with cellular bodies
- Support has been added for enlarging general bodies including cellular bodies

1.3.3.1 Interface changes

Changed	PK_BODY_enlarge, PK_BODY_make_curves_outline, PK_BODY_intersect_bodies, PK_BODY_make_persp_outline, PK_BODY_make_spun_outline, PK_BODY_subtract_bodies, PK_BODY_unite_bodies, PK_CURVE_find_surfs_common,PK_FACE_make_blend, PK_FACE_make_3_face_blend, PK_BODY_fix_blends, PK_SURF_find_curves_common
New	PK_BODY_is_cellular

1.4 Distant and multi-scale modelling support

This section describes the improvements that have been made to Parasolid's support for models of different scales in Parasolid V36.0.

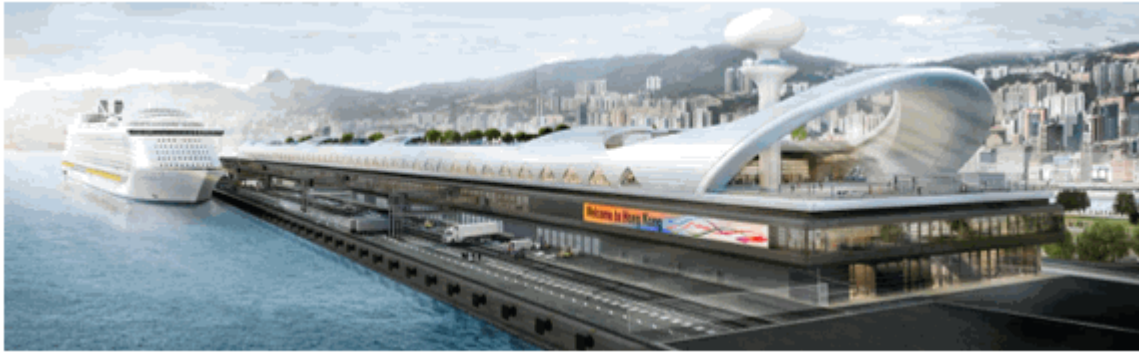

Figure 1–5 *Real world example of a multi-scale scenario*

1.4.1 Documentation support for working with models of different scales

Parasolid supports models whose parts are of different scales (enquiry functions such as rendering, faceting, boxing and measurement). It also allows geometry and parts created at a larger scale to be scaled and trimmed so they can interact with entities at a smaller scale. In V36.0 a new chapter that provides information on what distant and multi-scale modelling are, and how Parasolid uses these distinct but complementary mechanisms to enable your application to work with large scale models.

For more information see, Chapter 23, “Distant and Multi-scale Modelling” in the Parasolid *Functional Description*.

1.4.2 Supporting models created at different scales

Parasolid V35.0 provided initial support for models whose parts are of different scales (enquiry functions such as rendering, faceting, boxing and measurement). It also allows geometry and parts created at a larger scale to be scaled and trimmed so they can interact with entities at a smaller scale.

In V36.0 support for collections of parts of different scales has been added to mass properties and hidden line rendering.

For more information, see Section 28.1.3, “Summary of options” in the Parasolid *Functional Description*.

1.4.2.1 Interface changes

Changed	PK_TOPOL_eval_mass_props_o_t new options: `parts_with_scales`, `scale_factors`

1.4.3 Scaling the output results of multi-scale modelling

Options to scale the output of operations involving models of different scales have also been added to mass properties and hidden line rendering.

For more information, see Section 28.1.3, “Summary of options” in the Parasolid *Functional Description*.

1.4.3.1 Interface changes

Changed	PK_TOPOL_eval_mass_props_o_t new option: `output_scale`

1.5 Faceting and rendering

This section describes improvements that have been made to Parasolid’s faceting and rendering functionality for Parasolid V36.0. Full details on Parasolid’s faceting and rendering functionality are given in Volume 15: Graphics Support, of the Parasolid *Functional Description*

1.5.1 Rendering lattice geometry embedded in a body

In V36.0 Parasolid allows you to visualise lattices embedded in a body by outputting a collection of display primitives representing the lattice geometry along with a bounding set of facets.

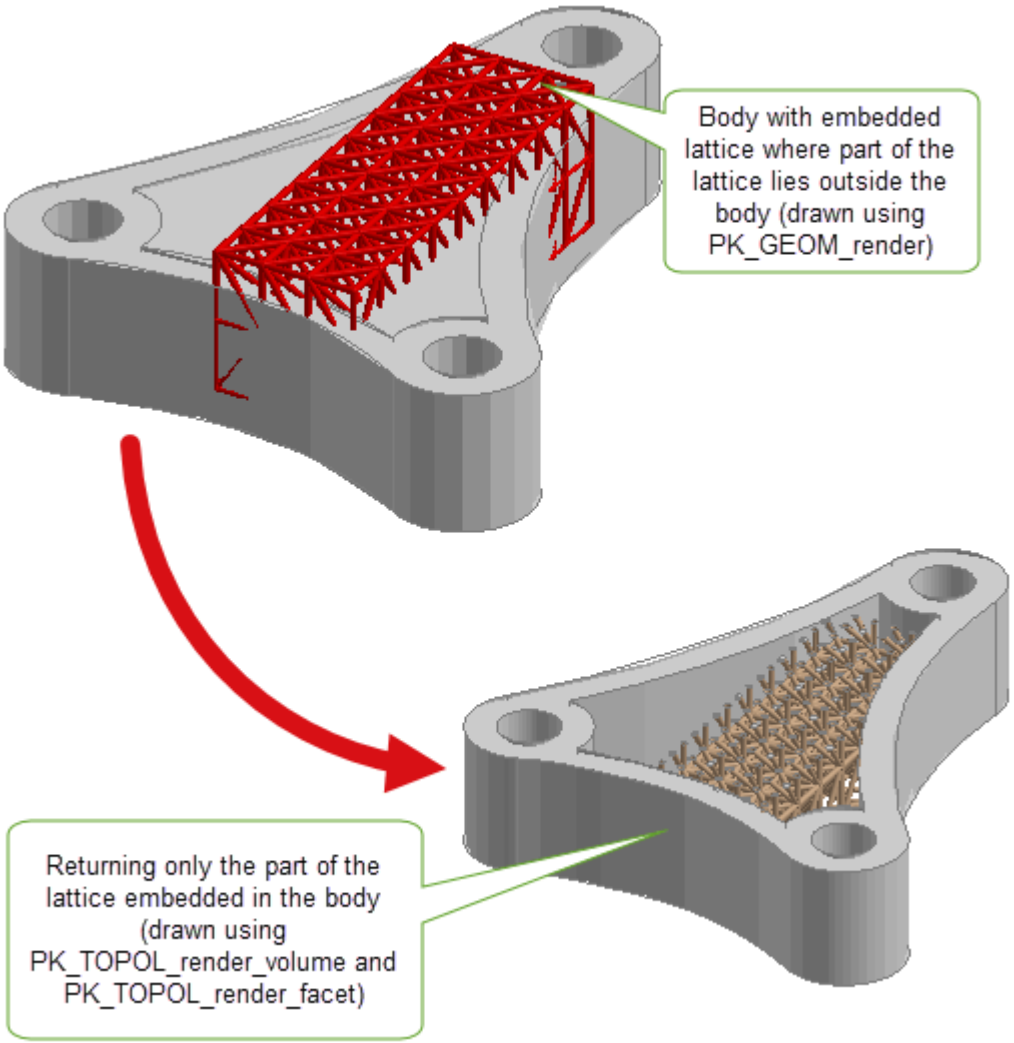

Figure 1–6 *Rendering only the part of the lattice embedded in the body*

For more information, see Section 20.10.4, "Imprinting embedded lattices" in the Parasolid *Functional Description*.

1.5.1.1 Interface changes

New	PK_TOPOL_render_volume

1.6 Modelling support

This section describes improvements that have been made to Parasolid's modelling support functionality for Parasolid V36.0. Further details can be found in Volume 3: Basic Concepts.

1.6.1 Improved support when enlarging bodies

In V36.0 you can now enlarge general bodies (including cellular) as well as wireframe bodies and minimal bodies.

1.6.1.1 Interface changes

None, but changes effect PK_BODY_enlarge.

1.7 Model interrogation

This section describes improvements that have been made to Parasolid's model interrogation functionality for Parasolid V36.0. Full details on Parasolid's interrogation functionality are given in Volume 4: Model Interrogation, of the Parasolid *Functional Description*.

1.7.1 Improvements when calculating distances between topology and arrays of vectors

In V36.0, you can now calculate distances between an array of vectors and a single topology. This can offer significant performance improvements when working with large numbers of vector positions. Previously, you could only supply a single geometry with multiple vectors.

For more information, see Section 29.2.1, "Calculating the distance between transformed entity arrays", in the Parasolid *Functional Description*.

1.7.1.1 Interface changes

None, but changes effect PK_ENTITY_range_vector.

1.8 Blending

This section describes improvements that have been made to Parasolid's blending functionality for Parasolid V36.0. Full details on Parasolid's blending functionality are given in Volume 11: Blending, of the Parasolid *Functional Description*.

1.8.1 Creating constant-width edge blends

In V36.0, you can set constant-width edge blends on a single chain of edges.

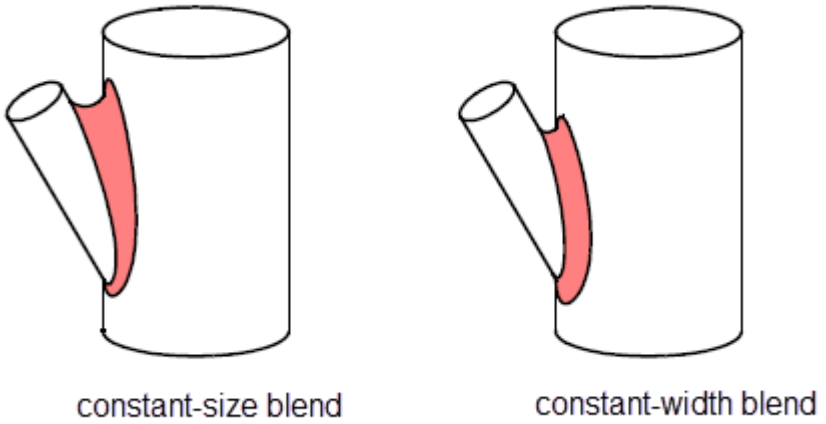

Figure 1–7 *Example showing the difference between a constant-size blend and a constant-width blend*

For more information, see Section 76.2.4, "Options for constant-width blends", in the Parasolid *Functional Description*.

1.8.1.1 Interface changes

Changed	PK_EDGE_set_blend_chain: new values PK_blend_size_width_c, PK_blend_size_ratio_c

1.9 Application support

This section describes improvements that have been made to Parasolid's application support functionality for Parasolid V36.0. Full details on Parasolid's support for application development are given in Volume 2 Application Development Essentials and Volume 14 Application Support of the Parasolid *Functional Description*.

1.9.1 Receiving a partition as locked to the calling thread

In Parasolid V36.0, you can choose to receive a partition as locked to the calling thread, ensuring it is private to that thread and can benefit from the performance of locally exclusive Parasolid calls. Previously, your application would have to make multiple calls to achieve this behaviour.

For more information, see Section 101.5.2, "Locking a received partition" in the Parasolid *Functional Description*.

1.9.1.1 Interface changes

Changed	PK_PARTITION_receive: new option `receive_locked`

1.10 New system attributes

This section lists the new system attributes that have been added to Parasolid V36.0.

- SDL/TYSA_FRAME_DENSITY

1.11 Functionality announced in previous releases and now available

The functionality below was announced in previous versions of Parasolid but was blocked at the time of release. The following have now been unblocked.

- PK_TRANSF_enlarge

1.12 Changes to the Parasolid documentation

In addition to documentation covering the changes and additions to Parasolid functionality described elsewhere in this chapter, the following changes have been made to the Parasolid documentation suite:

- A selection of edge blending images and face tapering images have been updated. For more information, see Section 77.4.4, “Internal overflows” in the Parasolid *Functional Description* and Section 67.3.2, “Curve-based and surface-based tapering” in the Parasolid *Functional Description*.

1.13 Changes to support applications

This section describes improvements that have been made to Parasolid’s support applications for Parasolid V36.0.

The following applications now ship as a Visual Studio 2022 solution and now build as 64-bit application by default:

- C++ Example Application for Windows
- Parasolid Workshop.Net (using .Net Framework version 4.8)
- C# Example Application

Note: You can also build these as a 32-bit application by making minor changes to the configuration and source code (for information see the documentation for C# and C++ Example Applications and for the Workshop.Net Source Code Overview documentation in your installation folder).

In addition, the C++ Example Application for Windows has been updated to use the following functions:

- PK_TOPOL_find_box_2
- PK_GEOM_render
- PK_FACE_pattern_2

1.14 Platforms

1.14.1 Changes for Parasolid V36.0

This section describes changes to the platforms on which Parasolid ships that are relevant to the V36.0 release. For full details of changes to platforms, please see the customer letter accompanying the Parasolid V36.0 release.

- The default builds for Intel NT and X64 WIN platforms will be built with Visual Studio 2022.
- The default builds for the Intel Linux and ARM Linux platforms will be built on Red Hat Enterprise Linux 8, or an equivalent Linux distribution, using GCC 11.2. These builds are supported on Red Hat Linux Enterprise Linux 8.8 onwards, AlmaLinux 8.8 onwards, Rocky Linux 8.8 onwards, SUSE Enterprise Linux 15sp3 onwards, Amazon Linux 2023 onwards, and Ubuntu LTS 20.04 onwards. They should run on any 64-bit Linux for the relevant processor architecture with GCC 8 or later run-times and glibc 2.28 or later that runs on the relevant processor.
- The Intel Linux build with GCC 11.2 on CentOS 7.9 will be supported until CentOS 7.9 and Red Hat Enterprise Linux 7.9 leave support at the end of June 2024, after which it will no longer be supported.
- The ARM Android platform is now built using NDK23b and is supported on Android 9 onwards.
- This will be the last version of Parasolid that supports Windows Server 2012 R2. Support for Parasolid and related software on Windows Server 2012 R2 will end on 10th October 2023.

1.14.2 Plans for Parasolid V36.1

This section describes our current plans for Parasolid V36.0. For more details, please see the customer letter accompanying the Parasolid V36.0 release.

We intend to replace:

- The ARM MacOS and ARM iOS builds, built on macOS 11, with new builds on macOS 13, supporting macOS 12 and later and iPadOS 15 and later.
- The Intel MacOS and Intel iOS builds, built on macOS 11, with new builds done on macOS 13 and supporting macOS 12 and later and iPadOS 15 and later. These builds are intended to set the final build standard for these platforms; we do not intend to update them again before they are dropped.

1.15 New PK interface tokens

PK_ATTRIB_transfer_t

PK_ATTRIB_transfer_no_c	27190
PK_ATTRIB_transfer_move_c	27191
PK_ATTRIB_transfer_copy_c	27192

PK_blend_size_t

PK_blend_size_width_c	26313
PK_blend_size_ratio_c	26314

PK_BODY_cellular_t

PK_BODY_cellular_no_c	27120
PK_BODY_cellular_yes_c	27121

PK_BODY_patterned_t

PK_BODY_patterned_touch_c	26991

PK_boolean_report_t

PK_boolean_report_blend_failed_c	21611

PK_clip_result_t

PK_clip_result_invalid_c	26915

PK_cellular_guise_t

PK_cellular_guise_general_c	27110
PK_cellular_guise_outside_c	27111
PK_cellular_guise_solid_c	27112

PK_check_state_t

PK_BODY_state_region_1_framed_c	26776

PK_embed_frame_group_t

PK_embed_frame_group_no_c	27240
PK_embed_frame_group_add_c	27241
PK_embed_frame_group_replace_c	27242
PK_embed_frame_group_remove_c	27243

PK_FACE_material_t

PK_FACE_material_bounding_c	27150
PK_FACE_material_containing_c	27151
PK_FACE_material_none_c	27152

PK_FACE_region_t

PK_FACE_region_within_c	27160
PK_FACE_region_between_c	27161

PK_FRAME_grow_region_t

PK_FRAME_grow_region_always_c	27180
PK_FRAME_grow_region_no_c	27181

PK_FRAME_handling_t

PK_FRAME_handling_ignore_c	27230
PK_FRAME_handling_include_c	27231
PK_FRAME_handling_fail_c	27232

PK_pattern_type_t

PK_pattern_type_axial_c	26982

PK_receive_locked_t

PK_receive_locked_auto_c	27170
PK_receive_locked_always_c	27171

PK_REGION_material_t

PK_REGION_material_solid_c	27140
PK_REGION_material_part_solid_c	27141
PK_REGION_material_void_c	27142
PK_REGION_material_part_void_c	27143

PK_check_state_t

PK_REGION_state_bad_frame_c	26774
PK_REGION_state_many_frames_c	26775

PK_remove_frame_group_t

PK_remove_frame_group_remove_c	27250
PK_remove_frame_group_replace_c	27251

PK_REPORT_1_t

PK_REPORT_1_hidden_by_guise_c	23932

PK_REPORT_3_t

PK_REPORT_3_picked_frames_c	26843

1.16 New PK interface error codes

PK_ERROR_code_t

PK_ERROR_size_too_large	5289
PK_ERROR_acorn_body	5290
PK_ERROR_hidden_by_guise	5291
PK_ERROR_cant_embed_lattice	5292
PK_ERROR_bad_trim_curves	5293
PK_ERROR_bad_form	5294
PK_ERROR_bad_loop	5295
PK_ERROR_bad_loop_config	5296
PK_ERROR_core_breach	5297
PK_ERROR_no_lattice	5298
PK_ERROR_guise_not_supported	5299
PK_ERROR_no_body	5300