GEM4D DOCUMENTATION

BEM Stress Analysis

Theory and Verification

PRODUCT	GEM4D
DOCUMENT	BEM Stress Analysis — Theory and Verification
VERSION	1.9.1.5
DATE	9 November 2026

basrock.net

Contents

1. The boundary element formulation 4
1.1 The indirect boundary element method 4
1.2 Fictitious-force (FF) elements 4
1.3 Kernels 4
1.4 Displacement-discontinuity (DD) elements 5
1.5 Combined FF and DD models 5
1.6 In-situ stress 5
1.7 What a solution holds 6
2. Field-point evaluation and classification 6
2.1 Evaluating a point 6
2.2 Classification 6
2.3 Interior values 7
2.4 The wall probe 7
3. How GEM4D prepares meshes 7
3.1 FF / DD classification 7
3.2 FF meshes 8
3.3 Watertightness 8
3.4 Mixed normals and seams 8
3.5 DD surfaces 9
3.6 Thin closed solids: the mid-surface 9
3.7 Glued and stepped blocks 10
3.8 Several surfaces and several steps 10
3.9 Refusals 10
3.10 The model quality check 11
4. How the mesh repair tools work 12
4.1 Remesh for BEM analysis: the triangle budget 12
4.2 Balloon wrap for BEM 13
4.3 Refine for BEM (keep shape) 14
5. Solver paths and memory 15
5.1 Direct and iterative solves 15
5.2 The H-matrix 15
5.3 The cache budget 16
5.4 The near-field threshold 16
5.5 What is logged 17
5.6 Energy evaluations 17
5.7 The GPU 17
6. Energy formulations 20
6.1 Stored strain-energy density 20
6.2 Released energy on FF boundaries 20
6.3 Released energy on DD elements 20

6.4 Path independence 21
6.5 Area normalisations 21
7. Excess shear stress and Kirsch tangential stress 21
7.1 Plane-specific ESS 21
7.2 Coulomb ESS 21
7.3 Kirsch tangential stress 22
8. Accuracy and validation 22
8.1 Thin bodies and element size 22
8.2 Triangle shape 22
8.3 Coarsening a thin body 23
8.4 Gaps and steps between DD pieces 23
8.5 Released energy: the pressurised sphere 24
8.6 DD elements: the penny crack and curved reefs 24
8.7 Element size on a mine model 24
8.8 The balloon wrap: checks and limits 25
8.9 Where to trust a stress: distance over element length 25
9. Result changes from BEM library updates 26
9.1 BasRock.BEM 1.3.0.0 to 1.7.0.0 (October 2026) 26
9.2 BasRock.BEM 1.2.1.0 (October 2026) 27
9.3 BasRock.BEM 1.2.0.0 (October 2026) 27
Appendix A — Worked example: checking `BEM_Tang_MPa` by hand 28
A.1 The example row 28
A.2 Step 1 — the tunnel axis 29
A.3 Step 2 — the two cross-section axes 29
A.4 Step 3 — project the stress tensor onto the cross-section 29
A.5 Step 4 — the in-plane stress extremes 30
A.6 Step 5 — the Kirsch tangential stress 30
A.7 The same thing in a spreadsheet 30
A.8 Notes on the fine print 30
Appendix B — Worked example: checking a released-energy row by hand 31
B.1 The example 31
B.2 Step 1 — the traction that acted on the face 32
B.3 Step 2 — how far the wall moved 32
B.4 Step 3 — the release density 32
B.5 Step 4 — the element's contribution 32
B.6 The check 32
B.7 Checking a whole column in a spreadsheet 33
Appendix C — Worked example: checking a DD element by hand 33
C.1 The case 33
C.2 Step 1 — the traction that acted on the reef before mining 33
C.3 Step 2 — how far the reef closed 34
C.4 Step 3 — the release density 34

C.5 Step 4 — the element's contribution, and the reef total 34
C.6 The check 34
C.7 On a dipping reef 35
Appendix D — Measured principal stresses: the physics checks and a worked example 35
D.1 Balance with gravity 35
D.2 The half-space 35
D.3 A worked example 35
Document history ..... **Error! Bookmark not defined.**

This document is the companion to the *GEM4D — BEM Stress Analysis: User Guide*. The User Guide tells you how to build a model and read its results. This document explains how the module works inside, how it was checked against known solutions, and how three of its results can be reproduced by hand. It is written for engineers who want to know what is behind a number before they rely on it.

> **Scope and engineering responsibility.** The same scope applies as in the User Guide: the BEM module is a linear-elastic tool, and its results are only as good as the geometry, rock properties and in-situ stress supplied. Most accuracy figures in this document were measured on test geometries with known answers, the rest on real mine models against a finer mesh or another tool; either way they show how the method behaves, not the accuracy of any particular site model.

1. The boundary element formulation

GEM4D's stress solver is the BasRock BEM library. This chapter describes the formulation it implements. The library has been audited independently; its kernels are validated against the Lamé sphere, the Kirsch cylinder, the Boussinesq half-space, Sneddon's penny-shaped crack and plane-strain crack solutions.

1.1 The indirect boundary element method

The rock is a linear-elastic, homogeneous, isotropic solid with Young's modulus E and Poisson's ratio v. The excavations are holes in it. Only their surfaces are discretised: each triangle of a mesh is one **element**. The rock around them is represented exactly by an elasticity solution — the **kernel** — so there is no outer boundary and no volume mesh.

GEM4D uses the **indirect** method. Each element carries an unknown, fictitious source of constant strength. The solver chooses the source strengths so that the boundary conditions hold at one point on each element, its **collocation point**. Once the strengths are known, the stress and displacement anywhere in the rock are sums of each element's analytic influence (Chapter 2).

Every reported stress is the **total** stress: the in-situ stress plus the induced change. The boundary condition on a mined surface is that the total traction on it is zero.

1.2 Fictitious-force (FF) elements

An FF element carries a constant **force density** — three unknowns per triangle. The solver finds the force densities that make every excavation surface free of traction, which is exactly what an open void does to the stress field.

The force density sits on a sheet, so the stress jumps across the element: on the element itself, the traction is the average of its two sides, and the half-jump appears on the diagonal of the system. This is why the energy maps evaluate just inside the rock (§2.4).

1.3 Kernels

Kernel	Domain	Notes
Kelvin	Infinite, full space	Elements close to the evaluation point, and an element's influence on itself, are integrated in closed form. Exact up to the wall.
Mindlin	Half-space with a traction-free plane surface at the ground surface Z	The Kelvin part is integrated in closed form; the free-surface image term by adaptive quadrature. Points above the surface are outside the model.

The solver works in **internal coordinates**, with Z = 0 at the ground surface and depth = −Z. GEM4D converts to and from mine coordinates at every exchange.

1.4 Displacement-discontinuity (DD) elements

A DD element is a surface across which the displacement may jump. It carries three unknowns: the **normal discontinuity** D_n (opening-positive; closure is $-D_n$) and two in-plane **ride** components, along the element's strike (D_s) and dip (D_t) axes. Mining an element means requiring the total traction across it to vanish.

DD elements are **triangles**. The displacement field of each is the analytic, artefact-free triangular dislocation of Nikkhoo & Walter (2015), built from three angular dislocations. The stress is obtained from Hooke's law on a high-order finite-difference gradient of that displacement. An element's influence on itself is the limit of the two-sided average as the evaluation point approaches the element.

Each element is given its reef's strike as a hint, so that the ride components of a facetted reef share one frame. Older models carry rectangular DD elements, which still load.

1.5 Combined FF and DD models

A model with both kinds of element is solved as **one coupled system**: the excavation walls feel the reef and the reef feels the walls. DD influence columns are of a different magnitude from FF columns (of order 2μ/L against order 1), so the library scales the columns of the system before solving and rescales the unknowns afterwards. This keeps the iterative solver's accuracy uniform across both kinds.

The combined formulation uses the infinite-domain kernel only. A DD row with the Mindlin kernel is refused.

1.6 In-situ stress

Compression is positive. Depth d is measured below the ground surface Z. The solver only ever asks for the in-situ tensor at a point, so every description below reaches it the same way.

The first two descriptions have one vertical and two horizontal principal components:

- **Uniform:** σv as entered; σH = kH · σv along the trend; σh = kh · σv at right angles to it.
- **Gravitational:** σv = γ · d, with σH and σh in the same ratios at every depth. Above the ground surface the depth is taken as zero, so the stress there is zero — which is why GEM4D checks for meshes above it (*User Guide* §7.6). Locked-in horizontal stresses L_H and L_h as Examine3D and RS3 define them (σH = kH · σv + L_H, σh = kh · σv + L_h) are the *Linear* field below with vertical and horizontal axes: a_i = L_H, L_h, 0 and b_i = kH · γ, kh · γ, γ.

Measured principal stresses may have any orientation. Each row gives a magnitude σ_i and a trend T_i and plunge P_i (below horizontal; a negative plunge is the same line with P → −P and T → T + 180°). With trends read from true north, they are first turned to mine grid, T_grid = T − θg, θg being the bearing of mine-grid north east of true north. In mine axes (East, North, Up) each direction is

$u_i = (\sin T_i \cos P_i, \cos T_i \cos P_i, -\sin P_i)$.

A row whose direction is to be derived takes the normalised cross product of the other two. Rounded directions are seldom exactly perpendicular, so the matrix $M = [u_1\ u_2\ u_3]$ (its third column negated if det M < 0 — an axis has no sense) is replaced by the **nearest rotation** R, the orthogonal factor of its polar decomposition ($U \cdot V^T$ of its singular-value decomposition). It turns all three axes by about the same small angle and favours none. GEM4D notes the largest turn when it exceeds 0.5° (a warning above 3°), and refuses two entered directions more than 10° from perpendicular. The tensor at the measurement depth d_ref is then

$S_{ref} = \Sigma\ \sigma_i\ r_i \otimes r_i$, r_i the columns of R,

so its principal values are exactly the σ_i entered. Away from d_ref the field is linear in depth, S(d) = S_ref + G · (d − d_ref), with

Change with depth	G (per metre of depth)
None (constant)	0
Proportional to depth	S_ref / d_ref (needs d_ref > 0; zero stress at the surface)

and the depth is taken as zero above the ground surface, as for Gravitational.

With **Linear: a + b × depth** each row has its own straight line, $\sigma_i(d) = a_i + b_i \cdot d$ along its squared-up axis r_i, so the field is $S(d) = \Sigma\, a_i\, r_i \otimes r_i + d \cdot \Sigma\, b_i\, r_i \otimes r_i$ (reference depth 0; the magnitudes and the measurement RL are not used). Its principal values are exactly $a_i + b_i \cdot d$ at every depth and its axes never turn; the two options above are the special cases $b_i = 0$ (constant) and $a_i = 0$, $b_i = \sigma_i$ / d_ref (proportional). Two rows i, j change places in the S1 / S2 / S3 order at $d^* = (a_j - a_i) / (b_i - b_j)$, which GEM4D reports when it lies within the model.

With vertical and horizontal axes, *Proportional to depth* reproduces the Gravitational description exactly (tested to 10^{-9} MPa at three depths), and the two old descriptions build exactly the stress they built before, so models made before this description solve unchanged.

Appendix D checks these descriptions against the balance with gravity and a free ground surface, and works one measurement through.

1.7 What a solution holds

A solved model is the set of element geometries and their solved unknowns: force densities on FF elements, discontinuities on DD elements, for every mining step. Everything GEM4D reports afterwards is evaluated from them. A `.bem4d` file stores exactly this, in mine coordinates, together with the Setup, so reloading restores the solution without solving again.

2. Field-point evaluation and classification

2.1 Evaluating a point

Once the system is solved, the stress and displacement at any point in the rock follow by summing every element's influence on that point, then adding the in-situ stress. Each element is a **constant-strength** element: one force density (FF) or one displacement discontinuity (DD) over the whole triangle.

Elements close to the point are integrated analytically, so the Kelvin result is exact right up to the wall. The Mindlin kernel adds a free-surface image term, integrated by adaptive quadrature; its error falls below 1 % of the in-situ stress beyond about two element edges from a wall. With either kernel, values within about one element of a wall are less accurate than values deeper in the rock, because a constant-strength element cannot represent the stress gradient across its own width.

2.2 Classification

Every evaluated point is classified:

1. **Inside test.** A point inside a closed excavation surface — or, with the Mindlin kernel, above the ground surface — is `Interior`.
2. **Distance.** The distance to the nearest wall is calculated and reported as `DistToSurface_m`, negative inside.

3. **Near-surface band.** A point outside the excavations but within the near-surface band (Setup, in multiples of the mean element edge) is `NearSurface`; everything else is `Far`. The default band is 0 with the Kelvin kernel — its integration is exact to the wall — and 2 with Mindlin.

The inside test relies on closed surfaces. Points near holes or non-manifold edges may be classified on the wrong side (§3.3).

`NearSurface` is a flag only: the values written to tables and files are the evaluated values.

2.3 Interior values

By default every map writes `NaN` for interior points. With *Iso-surface treats interior as in-situ* ticked in Setup, all three maps — Grid, Octree and datagrid, single models and sequences, combined files and per-step files — write the in-situ stress instead, with the in-situ values of every derived column and zero displacement, so that an iso-surface interpolated from the points relaxes smoothly into the void. The in-situ tensor is evaluated at the point itself, so a gravitational field is back-filled with the stress for that depth. The Status column still reads `Interior`.

On a sequence each step is classified on its own geometry.

2.4 The wall probe

An FF element is a sheet of force. Exactly on the sheet, the stress is the **average** of its two sides — the traction jumps by half the force density across it. A point placed exactly on a wall therefore does not return the wall stress on the rock side.

The energy maps (**Stored on surfaces**, **Released on surfaces**) need the rock-side value. They evaluate each element at a point **1 % of the element size** ($0.01 \cdot \sqrt{A}$) into the rock, and report the result at the wall position. This keeps the evaluation on the analytic near-field integrator, so the probe converges to the rock-side limit and is meaningful at the wall despite the general near-wall caution of §2.1. The probe removes the averaging, not the element-size error: wall values settle more slowly under refinement than values inside the rock (§8.7).

3. How GEM4D prepares meshes

This chapter describes what **Add FF and DD objects** does to a mesh between the mesh table and the solver (*User Guide*, Chapter 7). It follows the build in order: the FF / DD classification when the mesh table opens, the FF ingest and its checks, the preparation of DD reefs, the refusals, and the quality check of the whole model after the build. The tools that repair meshes before they are added are described in Chapter 4.

3.1 FF / DD classification

When the mesh table opens, every mesh is classified, whatever its size. The verdict is cached until the mesh changes.

Open or closed. A mesh is **open** when any of these holds:

- its open edges exceed the noise threshold of §3.3;
- it encloses no volume and has any open edge at all;
- its open edges are more than 5 % of all its edges (this catches the rim of a small, folded open surface, which the absolute threshold would miss).

Holes are not a rim. Open edges are grouped into loops. If the mesh encloses volume and no body's largest loop spans a quarter of that body, the loops are holes and cracks in a closed shell. The mesh is then classified as the closed body it is, and the tooltip counts the holes. An open surface's rim is its whole outline, so it stays open.

Thin or volumetric. A closed body is **thin** when its thickness is at most 15 % of its mid extent **and** at most 5 m. The ceiling exists because the plate test alone is scale-free: bulk stopes pass it. As-mined snapshots of sublevel-shrinkage and open-stoping mines, 11–24 m thick, pass it, and so does an open pit with a sublevel-shrinkage block (characteristic thickness 118 m). A DD plane stands in for an orebody only while the mining width is small compared with the distance at which results are wanted. That is South-African-style narrow tabular mining, about 1 to 3 m wide. 5 m leaves room above 3 m, because the measured thickness 2|V|/A reads a little under the width: 1.92 m on a 100 × 100 × 2 m slab.

Every closed body is suggested FF, a thin one included. Whether DD suits a thin solid depends on the distance at which results are wanted, which the mesh cannot tell, and the two wrong choices do not cost the same. FF on a narrow reef gives correct results with more elements. DD on a bulk stope gives wrong results near the stope, with nothing to show it. A thin solid set to DD is accepted without a question, and its measured thickness becomes the width.

Measured	Shape	Suggested Type
Open	Surface	DD (width to be entered)
Closed, thin	Thin solid *w* m	FF; DD accepted without a question, with the measured thickness as width
Closed, no volume	Sheets	DD
Closed, volumetric	Solid	FF
Thin and volumetric bodies in one mesh	Mixed	FF — split the mesh

3.2 FF meshes

Each ticked FF mesh goes through five stages:

1. **Triangulation.** Quads and triangle strips become triangles; lines and points are dropped.
2. **Welding.** Vertices that nearly coincide are merged, restoring the shared edges that DXF round-trips often break.
3. **Orientation.** Each connected, manifold region is walked from its own seed so that all its triangles face the same way. The solver's sign convention is then applied internally, so the user never has to fix the winding.
4. **Degenerate triangles.** Triangles of zero or near-zero area — below about 5×10^{-15} m² , where the element's normal cannot be formed — are skipped and counted.
5. **Elements.** Every remaining triangle becomes one constant-strength fictitious-force element.

3.3 Watertightness

GEM4D counts the **open** edges (used by one triangle) and the **non-manifold** edges (used by three or more) of the combined surface. A few such edges at floating-point-noise scale have no effect on the solver, so they are only logged. The threshold is the larger of 10 edges and 1/2 000 of the estimated edge count. Above it, the completion summary reports *"Not watertight"*: field points near the defects may then be classified on the wrong side of a wall.

3.4 Mixed normals and seams

The orientation walk in §3.2 repairs a flipped face anywhere on a manifold region. It cannot cross a **non-manifold seam** — an internal wall or a T-junction where three or more triangles share an edge — because it cannot tell which sheet is the excavation wall. Sheets on either side of a seam therefore keep the winding the file gave them.

GEM4D measures the winding of the source file before any repair. A mesh is flagged with **mixed normals** when both of these hold:

- some edges are traversed the same way by both their triangles (one of the pair is wound backwards), and
- the mesh has non-manifold seams above the noise threshold of §3.3.

A flipped face on a seam-free mesh is repaired and only noted (*"re-oriented automatically"*).

A wrongly facing element carries the wrong boundary traction, so results near it are wrong. It does not, on its own, stop the solve converging: a test model with about 2 500 reversed elements converged normally and was silently wrong. Meshes with mixed normals do, however, usually also carry coincident sheets and slivers, which do stop it. The mesh this check was built on ran 2 000 GMRES iterations without converging; after **Remesh for BEM analysis** it converged in 59.

3.5 DD surfaces

A DD row's own triangles become **triangular displacement-discontinuity elements**. There is no reference plane, lattice or limit on how the surface folds.

1. **Duplicates.** Coincident triangles — the second sheet of a two-sided export — are removed.
2. **Weld.** The surface is welded into one mesh. A triangle that would be the third on an edge is kept as a loose element.
3. **Refinement.** If the longest edge exceeds 1.5 × the DD element size, the surface is refined: edges are split, then several quality passes collapse short edges and smooth the triangles, projected back onto the source. The outline is kept as a polyline and its corners are pinned. The remesh runs one connected piece at a time and is single-threaded, so identical inputs give identical elements. A finer surface is used exactly as it is; above 200 000 triangles that is noted.
4. **Slivers.** A triangle whose altitude is below 1 % of the element size is dropped and counted. Such a triangle's collocation point sits against a neighbour's edge and makes the system singular; the slit it leaves is negligible.

3.6 Thin closed solids: the mid-surface

A thin closed solid is reduced to its **mid-surface**, and each element takes the roof-to-floor gap measured under it as its width.

The roof and floor are found locally, with no preferred direction:

1. Every face looks **inward** from its centroid and three further points.
2. If it meets an **opposing** face — within 30° of anti-parallel — within three widths, at a consistent distance, it is a **sheet** face. A side wall looks along the plate, meets faces at varying distances, and is not a sheet.
3. Sheet faces join into patches across gentle creases. The patches are then sorted into roof and floor through their pairings.
4. Each roof vertex moves onto the mid-plane of the roof faces around it. At a crease this lands on the exact corner of the two mid-planes.

So a thin solid folds as freely as a surface: a fold through 90° keeps both limbs, each on its own mid-plane.

Further rules handle awkward geometry:

- **Inside-out regions.** A vertical extrusion of an overhanging plane puts the offset copy through the original, so roof and floor swap sides there. These faces find their partner outward instead, and the mid-surface runs between them at the small width they actually have.
- **Rim-wall pairs.** The side walls of a narrow ledge or raise can face each other within three widths and pass as sheets. A patch that meets a sheet across a sharp crease, with a crossing

distance more than 1.5 times the sheet's, is demoted to a wall — but only when the patch it pairs with qualifies too.

- **The rim.** The mid-surface's rim is simplified at the scale of the thickness (twice the width, at most half the element size). Where the rim crosses a steep limb, the roof and floor rims sit a thickness or more apart in plan, and the mid-surface rim would otherwise step between them. An open surface's own outline is never touched.
- **Width fallback.** Where the roof-to-floor gap cannot be measured under an element, the row's Width is used and counted.

3.7 Glued and stepped blocks

A stope-shape export often comes as separate closed blocks — one per panel or sub-block — with neighbours sharing walls. GEM4D treats such a row as the **union** of its blocks:

- **Shared walls removed.** A wall two blocks share is in the file twice, often under two different triangulations. Each copy finds its twin facing it at zero distance; both are removed before the roof and floor are sought.
- **Coplanar slivers joined.** A face rejected by the crossing test but coplanar (within 5°) with an adjacent sheet face joins that sheet. This catches the sliver along a cut whose centroid ray leaves through the cut.
- **One block at a time.** Each block's sheet is refined on its own, so blocks touching at a corner cannot be run together.
- **Overlaps.** Where two blocks were drawn over the same ground, the smaller sheet's elements there are dropped and counted.
- **Gaps closed.** Sub-blocks at slightly different levels make roof and floor step at every joint, so each sub-block's mid-surface is a separate piece. Gaps narrower than **1 m** between pieces are closed: each rim vertex moves halfway towards the nearest rim of another piece, so the sheets meet. A move that would fold an element by more than 20° is halved once, then refused, and counted. A wider gap — a real pillar — stays. The 1 m acts on the sheets, which lie at mid-height.

A part of a reef where blocks interpenetrate at different orientations cannot give a clean mid-surface and comes out in fragments. The remedy is to fix the blocks in CAD, or to make one envelope with **Remesh for BEM analysis**.

§8.4 gives the measured effect of open and closed gaps on the results.

3.8 Several surfaces and several steps

- **Pillars.** A hole in a surface is a pillar: nothing is mined there.
- **Prepared once.** Each row is prepared once, and the same elements go into every step it belongs to. An element mined at step 2 is the identical triangle in step 5's model, which is what lets the released-energy bookkeeping recognise it.
- **Ground already covered.** Rows are ordered by their first step. A later row's element is skipped in a step an earlier row shares when the earlier surface passes **through** it — a cast along the element's normal, within a quarter of the element size. An element that merely sits beside the earlier surface's edge is new reef and is kept.

3.9 Refusals

The build is refused, leaving the existing model untouched, for the following. The message names the mesh, except for the last.

- a DD row with the Mindlin kernel (the combined FF + DD formulation has no free surface);
- a DD row without a width;
- a surface with no usable triangles, or with no fittable plane (all points collinear);

- a "thin solid" with no roof or no floor sheet — usually an open sheet flagged as closed;
- every DD element covered by an earlier surface in every step — nothing new to mine. A single surface that is covered everywhere is not refused: it adds no elements, and its row line in `GEM4D_diag.log` says 0 placed.

3.10 The model quality check

After the build, GEM4D checks **all the FF triangles of the model together** — of each step, on a mining sequence — as the build left them (each closed body facing outward), so it also sees two meshes drawn through or on top of each other.

Count	Test	Grade
Crossing triangles	On one weld of the whole model (corners within 10^{-6} m are one corner, whichever mesh they come from): crosses or touches a triangle with which it shares no corner (exact Möller test, coplanar contact left out), folds through a triangle with which it shares one corner, or folds flat onto the triangle across its edge — normals within about 11° of opposite **and** the two far corners on the same side of the edge, so that the two lie on top of each other, on an edge that only those two triangles use. Two triangles on the same three corners are left out (the next two counts).	Share of the model's triangles: red ≥ 10 %, yellow ≥ 2 %, green above 0
Duplicate triangles	On the same three corners as another triangle, facing the same way	Share of the area: red ≥ 25 %, yellow ≥ 5 %, green above 0
Shared walls	On the same three corners as another triangle, facing the opposite way — a wall two closed cells share, or the two faces of a zero-volume sheet	Information
Zero-volume body	Characteristic thickness 2V/A below 1 cm	Information
Solid inside a solid	The generalised winding number of each other body, |w| > 0.5, at each triangle's centroid moved 2 % of the triangle's size (√2A) against its normal, into its own body; reported above 0.5 % of the body's area. Bodies weld within a mesh, and across meshes only on open rims, so two closed solids that share a wall stay two.	Information
Sliver triangle	Longest edge ÷ height above 100 (per mesh)	Information

A flat fold counts only on an edge that just the two triangles use. On an edge that more triangles use, the two belong to two surfaces lying on each other: a wall two cells share, which each cell has divided into triangles differently, is a shared wall, not a fold.

Holes (open edges above the watertightness noise level) and an open surface built as FF are green, mixed normals (§3.4) yellow, non-manifold edges information. The summary takes the worst grade. A model with more than 1 500 bodies skips the solid-inside-solid test.

Why these counts and lines. They were chosen by solving (October 2026, GEM4D 1.9.1.5): four mine models as built, each against a Remesh for BEM analysis and a Balloon wrap version of about the same size, and clean models with one defect added at a time, in doses. Each model's σ1 at about

3 000 points was compared with a clean model's, by distance from the walls. The two clean versions of a mine agreed within 0.1–2.5 % (median, 10–20 m from the walls).

What the model had	GMRES iterations (clean)	σ1 change at 10–20 m, median
A stope twice, exactly on itself	152 (10)	8.9 % (43 % within 5 m)
20 % of a stope's triangles duplicated	14 (10)	1.0 %
A design solid crossing its own survey; 11.6 % of triangles crossing	105 (18)	19.8 %
The same without the design solid; 6.3 % crossing	24 (18)	1.9 %
Stacked stope designs; 16 % crossing	130 (22)	17 %
A remesh at elements longer than the stopes are thick; 25 % crossing	no convergence, stopped at 400	—
A remesh with 23 % crossing (another mine)	170 (18)	2.5 %
Stope and development, 6.5 % crossing	191 (14)	3.9 %
A stope stored as 2 or 4 closed cells	11 (10)	0.0 %
Up to 20 zero-volume sheets; up to 37 % slivers (aspect about 500)	18–19 (18)	0.0 %
A solid wholly inside the stope's void (up to 41 % of the area)	15 (10)	0.0 %
Holes up to 5 % of the triangles; up to 3 open roofs	17–18 (18)	≤ 0.5 %

The 10 % line sits between the models that were 17–20 % off (11.6 %, 16 %) and those that were 2–4 % off (6.3–6.5 %). The model with 23 % of its triangles crossing that still came within 2.5 % took nine times the iterations; the grade calls it red. Slivers at aspect ratios far above those tested (10^3–10^7, near-collinear) remain a known cause of stalled solves on raw CAD meshes.

A triangle that is the third on an edge (an internal wall) is put in on copies of its corners, and the copies count as the corners they copy: compared as separate points, the wall would touch every triangle it meets without sharing a corner with it, and all of them would count as crossing. The check takes 0.4 s on 153 000 triangles. If it stops with an error, the summary says so under **For information** and the grade rests on the other notes; it never reads as "no defects found".

4. How the mesh repair tools work

Three tools under Toolbox 1 ▸ **Prepare meshes for BEM** rebuild meshes before they are added (*User Guide* Chapter 6). This chapter describes how each works; Chapter 8 gives what was measured on the first two (§8.3, §8.8).

4.1 Remesh for BEM analysis: the triangle budget

The triangle budget of **Remesh for BEM analysis** (*User Guide* §6.3) keeps each body's elements at no more than half its thickness — the size at which the bodies measured in §8.3 kept about 99 % of their volume — and not below the cell of the grid that body was built on. A body thinner than half a cell is a skin the grid lays round an open sheet, not a body, and is left out. When the budget cannot hold every body at that size, one size for all keeps more volume than coarsening every body in the same proportion, because the large bodies hold most of the volume: on mine 1 of §8.3 at 150 000 triangles, 81 % against 77 % (with 5 080 against 6 671 self-intersecting triangles left).

Reaching the count. The remesh is steered to within 5 % of the budget, but the repair that follows (crossing triangles cut out and refilled) takes triangles out: 49 090 → 37 738 on a crumpled scan at 50

000 (13.5 % crossing), 143 624 → 138 088 on mine 1 at 150 000, under 3 % on mines 2 and 3. A result more than 5 % off the budget is remeshed again, at most twice — first aiming at the budget plus the number the repair took out (it takes out about the same number again), then along the line through the last two rounds. The scan ends at 49 784 and mine 1 at about +3 %; each correction costs 6–10 s, and a model already within 5 % is not remeshed again.

When the count jumps. On a body thinner than the elements and folded more closely than them, the count jumps as the element length changes: a few dozen triangles at one length, tens of thousands at a slightly shorter one. Once two tries lie on either side of the budget, the next stays between them; when no try comes within 5 %, the result nearest by ratio is kept (twice and half the budget count as equally far). On test bands of plates 0.5–1 m thick the results lie within −51 % to +50 % of the budget, and the report says when the count ends more than 5 % off. On the real models the rule never acts.

4.2 Balloon wrap for BEM

Balloon wrap for BEM (*User Guide* §6.4) is GEM4D's version of the IGM Balloon Survey Filler, a Python tool by IGM Geotechnical. It keeps that tool's closing, blur and triangle-count loop. It changes how the surveys are filled and how the surface is remeshed, because tests on real files showed the need (§8.8).

Filling. Each visible mesh is welded at 1 mm and split into patches: triangles joined through edges that exactly two triangles use. For each patch, the *opening ratio* q = |Σ face vector areas| ÷ Σ face areas is 0 for a closed patch, about 0.15 for an open-topped stope or a box missing a face, and 1 for a flat sheet. A patch with q below 0.4 is filled on its own, inside where the absolute fast winding number exceeds 0.5, so a stope wound inwards counts the same as one wound outwards. The flatter patches are grouped with the sheets they touch, turned to enclose positive volume about the group's centroid, and filled together. The Python tool turned every patch about the centroid of its whole DXF layer instead; on a layer holding many stopes that turned the far ones inside out.

A stope stored as cells, with one flat wall between neighbours, needs one more step. Every edge where a wall meets the outer skin is used by three triangles, so the skin breaks into short tubes, each open at both ends. A tube's opening ratio is near 0, yet inside it the winding number stays under one half, so only a skin a voxel or two thick along its walls was filled. The patches are therefore found a second time with the flat patches (q above 0.95: the walls) left out, and a joined patch that is still volume-like (q below 0.4) is filled as one. On narrow stopes stored this way (17 layers, 159 cell skins), a single pass left 32 000 m³ of cells hollow, 3 % of the set, and the thin skins it did fill were where the remesh crossed itself.

Closing and surface. A ball of radius R is rolled in from the grid border on an exact Euclidean distance transform. Whatever it cannot reach is solid, which seals gaps narrower than 2R and fills pillars thinner than 2R. The solid is blurred (Gaussian, σ 0.7 voxel), contoured by marching cubes and remeshed to the element length with a sharp-edge (face-aligned) reprojection and uniform smoothing. The same input gives the same wrap on every run; a remesh that comes out with more than twice the triangles its area holds at the element length is reported in the log.

Repair. Self-intersections are cut out and refilled; slivers left by the fills, and triangles folded flat onto a neighbour on a fin narrower than the elements, are removed by local edge collapses and flips. Each repair keeps the best state it reaches — the fewest crossings among states that are closed, keep 90 % of the area and lose no body — because on features thinner than the elements the later, wider passes can make more crossings than they remove. With it, the narrow stopes at 1 m elements end with none of their 1.68 million triangles crossing.

After the repair (from version 1.3 of the Python tool). Bodies that enclose less than the smallest body kept are removed. The remesh and the repair can pinch a flat piece off a feature thinner than the elements: two sheets of triangles on top of each other with no volume between them, which the voxel stage's smallest-body rule never saw. On the narrow stopes at 2.8 m elements, 9 such bodies went:

596 of 151 540 triangles and 78 m³ of 938 000. They held three quarters of the crossing triangles that were left (482 before, 117 after). The other six test sets did not change by a triangle. Pairs of triangles lying on top of each other are counted for information: centroids closer than a quarter of the larger triangle's radius (centroid to farthest corner), with no corner in common. There were 6 on the narrow stopes and none on the other sets. Version 1.3 also refuses a mesh that keeps a triangle above aspect 100 after its sliver repair. GEM4D keeps slivers advisory, because up to 37 % of a model's triangles at aspect about 500 changed σ1 by 0.0 % (§3.10). It already takes triangles above aspect 30 out by local edge operations.

4.3 Refine for BEM (keep shape)

Refine for BEM (keep shape) refines models of large flat faces — Map3D blocks, CAD stopes, mining-step objects — without moving a wall. Three things about such models called for it:

- **Coarse faces.** Each triangle is one element, and a face can be tens of metres across: on a test model of five bricks, σ1 2 m from a wall was 22 % off as loaded. §8.9 measures how the error follows the element size.
- **Faces that overlap in part.** Where faces lie in one plane and overlap only partly — a small block face against a large one — the solve fails: a collocation point lying on another element gets that element's two-sided average (§2.4), so its equation is wrong, not merely coarse. Two Map3D models did not converge as loaded. Two copies of a wall with identical triangles, as two step objects hold it, cancel exactly instead.
- **Corners.** The grid of Remesh for BEM analysis and the voxels of the balloon wrap round the corners and move the walls between steps apart.

All the visible meshes are worked on together, edges first, then faces:

1. **Planes.** Corners within 5 mm are welded. The triangles are grouped by the plane they lie in; each plane is fitted by least squares to all its corners, two planes merge while one fit through both holds every corner within 5 mm, and every corner is moved onto the planes of its triangles.
2. **Faces held by several objects.** In each plane a constrained Delaunay triangulation of every triangle edge cuts the plane into cells, each covered by the same triangles. A cell's owners are the objects that cover it, with the side each faces; two triangles of one object facing opposite ways cancel, so a wall inside one object goes. Neighbouring cells with the same owners form one flat region.
3. **Edges.** A region corner that lies on another region's edge, in any plane, splits that edge, so the objects stay closed. Every edge is then divided by a rule that depends only on its two ends and the size field, so every region using it gets the same points.
4. **Faces.** Each region is meshed once in its own plane, inside its fixed edges: points spaced by the size field, smoothed, then a constrained Delaunay triangulation. Every owner gets the same triangles, wound for its side, so a wall two objects hold is identical in both, and no point leaves its face. A region that cannot be meshed keeps its own corners and is counted in the report.

The size field follows the geometry, so that the answer is good enough anywhere, not only where one looked first. The element size is

h = min(E + g · d, w_rock / r_rock, w_void / r_void, H) × s, and at least the smallest element,

with E the size at edges and corners, g the growth, d the distance to the nearest crease (an edge where a surface turns by more than 20°, or ends), H the largest element and s the scale a triangle budget sets. w_rock and w_void are the widths of rock and of opening across the face, found by a ray along each side of its plane that counts only a face within 30° of parallel — the far wall of a pillar, a drive or a slot; a face met at an angle is the next face of a corner. The two width rules never ask for less than a quarter of E, nor below the smallest element. The settings and their defaults are in *User Guide* §6.5.

As-mined snapshots. When a quarter or more of the face area lies in two objects facing the same way, the objects are read as snapshots, each holding everything mined by its step. A width ray then counts only a face that shares an object with the face it starts from, so an early step's wall does not take the later steps' walls beyond it for a pillar. Each snapshot is solved alone, so the estimate and the budget are for the largest object; for any other objects they are for all of them together.

Estimate and budget. The count is estimated from samples over every face at 2.31 / h² triangles per square metre (equilateral triangles of side h). A budget finds the scale s by bisection; a smallest element can hold the count below the budget.

Holes. Sliver holes left where corners lie a few millimetres apart across two planes are filled in every object that came in closed; an open object stays open. The same arrangement, each block taken as a closed solid, merges the overlapping blocks of a Map3D model loaded as as-mined snapshots.

The same meshes and settings give the same result on every run.

5. Solver paths and memory

This chapter explains how GEM4D chooses the numerical method and manages memory during **Run model**. The user-facing settings are in *User Guide* §5.4.

5.1 Direct and iterative solves

The FF and DD influence equations form a dense system with three unknowns per element.

- **Up to 6 700 elements: direct LU factorisation.** Exact, with no iterations. The matrix is assembled as one array of (3N)² numbers — 3.2 GB at 6 700 elements. The factorisation uses Intel MKL; without it, the same solve runs roughly 100 times slower, and GEM4D warns at the first solve of a session. The `BEM linear algebra:` line in `GEM4D_diag.log` names the provider in use.
- **Above 6 700 elements: iterative GMRES.** The matrix is never formed whole; each iteration applies it element by element (EBE) through the H-matrix of §5.2. Iteration stops when the relative residual falls below the GMRES tolerance set in Setup.

The choice is made per solve, and per step on a sequence.

5.2 The H-matrix

The iterative solver does not store the full matrix. It builds a **hierarchical matrix** (H-matrix) from an octree of the elements:

- **Near-field blocks** — pairs of nearby element groups — are kept dense.
- **Far-field blocks** — well-separated groups — are compressed by adaptive cross approximation (ACA) to a relative tolerance of 1e-4 (or the GMRES tolerance, if that is coarser).

Each GMRES iteration multiplies by this matrix. A block held in RAM costs little per iteration; a block **matrix-free** — recalculated every iteration — costs much more, and a matrix-free far-field block much more than a matrix-free near-field one. On a 67 000-element mine step (16 threads, BEM library 1.6.0.0), one matrix-vector product took 0.42 s with everything cached, 0.58 s with the near field matrix-free, and 2.4 s with only 3 GB of cache, most far-field blocks matrix-free.

Building the H-matrix is the other cost of an iterative solve on the processor. The build calculates four element pairs at a time with the processor's vector instructions — about eight times faster than before BEM library 1.6.0.0, with the same numbers: the 67 000-element step builds its 12.3 GB H-matrix in about 8 s and solves in 29 s. A solve on the GPU (§5.7) builds no H-matrix.

5.3 The cache budget

H-matrix cache (GB) in Setup sets how much RAM these blocks may use.

- **Auto** probes the RAM free when each step starts, after releasing the previous step's cache, and keeps a reserve for what the solve needs outside the cache (see below): 6 GB up to about 98 000 elements, and 4 GB + 22 kB per element above that (9.5 GB at 268 000 elements). The budget has a floor of 2 GB. The figure shown in the Setup dialog is only a snapshot from when it was opened, with the reserve for the model's largest step (6 GB before any objects are added). Unticking **Auto** keeps that figure as a fixed budget.
- **A positive value** pins the budget.
- **0** removes the cap.

How the budget is shared depends on the near-field threshold (§5.4). On **Auto** (the default) the far field is compressed and cached first, and the near field is kept dense, block by block, in what remains. With a threshold entered by hand, a dense near field is charged against the budget first, at its exact measured size, and the far field is cached in what remains. Blocks that do not fit are left matrix-free. The post-build status line reports the total built and the number of far-field blocks left matrix-free (`· N MF`); `GEM4D_diag.log` also gives the near-field blocks left matrix-free.

Memory outside the cache. A solve also holds the preconditioner, the lists of matrix-free blocks, the iteration vectors and, for a while, the garbage of the build. Measured on a mine model, with the iteration vectors at their 200-iteration ceiling, this came to 2.8 GB above the cache at 67 000 elements, 3.9 GB at 105 000 and 6.7 GB at 268 000 — about 18 kB per element, and about 21 kB with GEM4D's own share. A fixed 6 GB reserve left about 2 GB of RAM free at 105 000 elements but none at 268 000; the reserve that grows with the element count keeps at least about 2 GB free at every size measured. Solved through GEM4D on a 32 GB computer with 22 GB free, the 268 000-element model (a mine step meshed four times finer) took 29 minutes, and at least 3.2 GB of RAM stayed free throughout; with the fixed 6 GB reserve its budget would have been 3 GB larger, and by the same measurement no RAM would have been left free (calculated, not run).

If the free RAM cannot hold the reserve plus the 2 GB minimum cache — a very large model on a small computer, or other programs holding the memory — the step's line in `GEM4D_diag.log` says **LOW RAM** and the status after the solve ends in `· low RAM`. The solve still runs, but it may page to disk and run slowly. Close other programs, or solve a model that size on a computer with more RAM.

Single precision. The cached blocks are stored in single precision; every number is still calculated, and every iteration summed, in double precision. A budget therefore holds about twice as much of the matrix: on a 19 232-element mine model the solution moved by 3.5e-8 of its largest value (wall stresses by at most 14 Pa), and a seven-step mine model solved in 19.9 minutes instead of 25.7 with double-precision storage (BEM library 1.2.7.0).

5.4 The near-field threshold

Near-field threshold sets the element count above which the near field is matrix-free from the start. **Auto** derives it from the computer's **total** RAM: about 41 000 elements at 16 GB, 72 000 at 32 GB, 110 000 at 64 GB and 160 000 at 128 GB.

On **Auto** with a capped cache budget — the default — the threshold is not used for an iterative solve. GEM4D asks the BEM library (version 1.7.0.0 and later) to build the **far field first**: the far-field blocks are compressed and cached up to the whole budget, then the near-field blocks are kept dense, one by one in the order of the octree, for as long as the RAM the far field left can hold them, and the rest are recalculated at every iteration. Near and far together never exceed the budget. A threshold entered by hand is your choice and keeps the older rule — the near field dense below the threshold, matrix-free from it, and charged against the budget before the far field (§5.3) — and so does an uncapped budget (0), which has room for everything.

Why the far field first. Whole solves were timed — the build plus every iteration — with the near field dense and with it matrix-free, at budgets from 1 to 15 GB, on four steps of a seven-step mine model (19 000 to 105 000 elements) and on a balloon-wrapped mine model of 86 000 elements (October 2026, BEM library 1.6.0.0, on the processor). A dense near field was faster when it and the whole far field fitted in the budget (in all 9 such cases, by 9–21 %). Where they did not, the matrix-free near field was faster in 17 of 19 cases, by 4–42 %, because every byte of dense near field is taken from the far field's cache, and an uncached far-field block costs more per iteration than a matrix-free near-field block (§5.2). The share of the budget at which both fit differs from model to model — 0.12 to 0.24 on these models — and is not known until the far field has been compressed, so no fixed share chooses well (15 % cost 5 % on average against the better choice, and up to 39 %). Building the far field first needs no such guess:

- where the far field does not fit, no near-field block is kept dense — the same as a matrix-free near field;
- where everything fits, every block is kept — the same as a dense near field;
- between the two, it keeps the whole far field and as much of the near field as fits, which neither older choice could do. Timed back to back in that range, the whole solve took 3.5–6.2 % less time than the better older choice on three of the models (the 19 000- and 67 000-element steps and the 86 000-element model), and the same, within the variation between runs, on the 48 000-element step.

This matters only for a solve on the processor; a solve on the GPU builds no H-matrix (§5.7).

5.5 What is logged

Each solved step writes one line to `GEM4D_diag.log`: the element count, LU or EBE, how the near field is stored — dense, matrix-free, or dense within what the budget leaves after the far field — with its measured size, the cache budget (auto, override or uncapped) and the free RAM at the time. The solver's own phase messages are logged too, including the H-matrix actually built: its size, the near field kept dense, the far field cached, and how many far-field and near-field blocks were left matrix-free. Comparing these lines between a fast run and a slow one answers "why did the same model get slower?" — almost always, less free RAM. When the file passes 1 MB, GEM4D renames it `GEM4D_diag.old.log` (replacing an older one) and starts a new file, so the lines of a long run are never lost: read both files, and send both with a report.

5.6 Energy evaluations

The **Stored on surfaces** and **Released on surfaces** maps evaluate the stress at every boundary element **directly**, without the H-matrix. Direct evaluation is exact, needs no block RAM, and measured faster at every size tried (Chapter 9). The field maps (Grid, Octree, datagrid) also evaluate directly, streaming the points. With a GPU, these direct sums run on it (§5.7).

5.7 The GPU

Since BEM library 1.3.0.0 a computer's graphics processor (GPU) can take over the long sums of a Kelvin model, and since 1.4.0.0 the solve of half-space (Mindlin) and DD models too (*User Guide* §5.4, **Use the GPU of this computer**, ticked at every start of GEM4D when a GPU is found). The approach follows *IGM BEM Solve* by IGM Geotechnical, which solved a 107 000-element GEM4D model in under a minute on an NVIDIA graphics card in October 2026.

The solve. The GPU holds the elements, not a matrix. At every GMRES iteration it sums the influence of every element on every other afresh, in single precision, by the integration rules of the processor method: a six-point rule up to 20 element lengths away, a single point beyond (an element length is $\sqrt{A}$). Nothing is compressed — the 1e-4 tolerance of §5.2 does not apply — and nothing is cached, so the H-matrix, the cache budget and the near-field threshold (§5.2–5.4) play no part. The pairs of elements closer than **five element lengths** stay on the processor: they are integrated once,

in double precision, exactly as the processor method integrates them (in closed form within 1.2 lengths), and kept — on a mine model about 175 pairs per element, 1.3 GB at 105 000 elements and 2.6 GB at 268 000. These are the largest terms of the sum and they cancel each other: when only the pairs within 1.2 lengths stayed on the processor, the single-precision sum left the solution three times the default tolerance away from the exact one, and GMRES needed twice the iterations. The preconditioner and GMRES are those of the processor method, so a model takes the same number of iterations.

The maps and energy. At every point the GPU sums every element in single precision and hands back the element–point pairs closer than 1.2 element lengths, which the processor integrates exactly.

Half-space and DD models (BEM library 1.4.0.0). The solve works the same way. For a half-space model the GPU uses the processor method's rules, its mirror-image term included; it differentiates that term exactly, where the processor method takes finite differences. A DD element is summed as a dislocation at points over its area: nine points (six on a triangle) up to 12 times its longest side away, four (three) up to 250 times, a single point beyond — each within 1e-5 of the exact influence. The pairs within five longest sides stay on the processor, as for the FF elements. Measured on the RTX 4050 laptop: a half-space model of six stopes 85 m below the surface, 19 608 elements, 51 s on the processor, 3.3 s on the GPU and 11.8 s on the built-in graphics; six stopes over a reef of 8 000 DD elements, 18 752 elements in all, 17 s, 2.0 s and 7.4 s — with the same iterations everywhere. Against the exact operator the GPU's solutions leave a residual of 9e-7 (half-space) and 1.2e-6 (FF + DD) of the load, the processor method's 9e-5 and 3e-4 (its compression), so here too the GPU is the closer to the exact answer.

Their maps and energy (BEM library 1.5.0.0). The maps and the two energy calculations of half-space and DD models use the GPU too, by the rules of their solves. The processor keeps the element–point pairs it integrates in closed form — closer than 1.2 element lengths, for a half-space model with the mirror-image term — and, for a DD element, the points within five of its longest sides. The GPU sums the rest, a DD element as a dislocation at points over its area. Points above the ground surface stay blank, as on the processor. A half-space evaluation goes to the GPU from 200 000 point–element pairs, because its sums are costly on the processor; a DD model's from 20 million, as a Kelvin model's. Measured on the RTX 4050 laptop (built-in graphics in brackets): the half-space model above, 20 000 grid points 48 s on the processor, 0.40 s (0.71 s) on the GPU, and its 19 608 skin points of the energy calculations 60 s, 0.16 s (0.70 s); the FF + DD model, 20 000 grid points 8.5 s, 0.42 s (0.39 s), and its skin points 8.1 s, 0.53 s (0.87 s). GPU and processor agree within about 0.001 to 0.003 % of the local stress — 0.0002 MPa on a half-space model 300 m deep, the rounding of single precision — and the released energy within about 0.0001 %.

Speed. Measured on a 32 GB laptop (16 processor threads, NVIDIA RTX 4050 Laptop GPU with 6 GB, Intel UHD built-in graphics) on a seven-step mine model with BEM library 1.7.0.0 (8 November 2026), the processor method at its automatic cache budget (14–15 GB):

Elements	Processor	GPU	Iterations
19 232	4.4 s	2.1 s	20
67 012	30 s	8.7 s	37
99 702	191 s	20 s	49
104 558	211 s	23 s	52
all seven steps	13.0 min	1.6 min	293
268 048 (a step meshed four times finer; library 1.3.0.0)	29 min	76 s	37

The iteration counts are the processor's; the GPU took 295 over the seven steps. With library 1.3.0.0 the processor took 16 s, 100 s, 315 s and 292 s for the four steps above and 21.3 min for all seven, mostly because since 1.6.0.0 it builds its working data about eight times faster (§5.2).

On the built-in Intel UHD graphics the 104 558-element step took 109 s, so GEM4D also uses built-in graphics when the computer has no graphics card. The maps and energy calculations ran 4 to 9 times faster on the RTX 4050 and about 2 times on the built-in graphics.

Accuracy. The two methods solve the same equations; the GPU leaves out the compression and sums in single precision. Against an **exact** solve — double precision everywhere, no compression, GMRES to 1e-9 — at 32 000 points 0.25 to 50 m into the rock:

Step	GPU: median / largest difference	Processor method: median / largest difference
19 232 elements	0.000005 / 0.0004 MPa	0.0014 / 0.057 MPa
99 702 elements	0.000007 / 0.008 MPa	0.0023 / 0.69 MPa

The compression of the processor method is the larger error: 93 % of its points were within 0.01 MPa of the exact answer, all of the GPU's. Its largest errors were 5 to 20 m from the walls. The GPU's field calculations agree with the processor's within 0.007 MPa at those points and within 0.002 MPa at the skin points of the energy calculations.

When the GPU is not used.

- Models of 6 700 elements or fewer solve on the processor (the direct method).
- **Precision floor.** Summed in single precision, the GPU's product carries a rounding error near 1e-7 of the load on a mine model, and GMRES cannot converge below it. GEM4D's finest tolerance is therefore **Standard** (1e-6); a setup or model saved at the former **Fine** (1e-8) opens as Standard, and `GEM4D_diag.log` says so. On openings much thinner than their elements the floor can exceed even Standard, because the sums over the two faces nearly cancel: the GPU solve then gives up after a few failed checks, and the step is solved again on the processor. The maps and energy do not iterate and use the GPU at every tolerance.
- **First-use check.** The first time a GPU is used in a session, it sums a small fixed model and its results are compared with the processor's, each part of the sums on its own — near and far elements, FF and DD sources, each integration rule of a DD element — so a fault in any one part shows at its full size. A correct GPU agrees to about 0.0001 %; one whose Kelvin sums are off by more than 0.01 % is not used again in that session. The half-space and DD sums are checked the first time each is needed, those of the solve apart from those of the maps and energy; a failed check sends only those solves, or only those maps and energy, to the processor, and everything else keeps the GPU.
- **Fallbacks.** Without a GPU, when it fails that check, when it cannot take the model, or when the computer has too little RAM for the element pairs kept on the processor, GEM4D uses the processor method. A solve whose GPU fails starts again on the processor. A map or energy calculation whose GPU fails partway through is calculated again on the processor from the start — on a mining sequence, from the start of the step it was on — so a map's progress drops back. A solve, map or energy calculation that was allowed the GPU but ran on the processor ends with a warning naming the reason, once per session for each kind of reason (*User Guide* §21.2). A computer without a GPU, the box cleared, or a map too small for the GPU is not a fallback and gives no warning.
- **The log.** The `BEM GPU:` line in `GEM4D_diag.log`, written at the first solve of a session, names the GPU found. The step's line before the solve names only what the step will try; after each step solved iteratively a line ends in *"solved: far field on the GPU (…)"*, or in *"solved: far field on the*

processor (H-matrix)" and the reason. A `GPU not used (…)`, `GPU failed (…)` or `GPU solve did not reach the tolerance (…)` line marks the moment it fell back.

- **Repeatability.** The same model on the same GPU gives the same result every time; another GPU, or the processor method, gives results that differ by the amounts above.

Which GPUs. Any with an OpenCL driver — NVIDIA, AMD and Intel all install one with their graphics drivers, so nothing is installed with GEM4D. A graphics card is chosen before built-in graphics. Tested on NVIDIA and Intel; AMD has not been tested.

6. Energy formulations

6.1 Stored strain-energy density

At any point, the elastic strain-energy density follows from the principal stresses and the Setup E and ν:

$$u = [\sigma 1^2 + \sigma 2^2 + \sigma 3^2 - 2\nu(\sigma 1\sigma 2 + \sigma 2\sigma 3 + \sigma 3\sigma 1)] / 2E$$

With σ in MPa and E in GPa, u comes out directly in kJ/m³. The expression is quadratic in stress, so the sign convention does not affect it.

The skin map evaluates u at the wall probe of §2.4; the SED checkbox on the Map dialogs evaluates the same u at the user's field points. Both use the same function and the same E and ν.

6.2 Released energy on FF boundaries

When mining step *k* creates new excavation surface Σ_k, the energy released is

$$\Delta W_k = \tfrac{1}{2} \int \text{ over } \Sigma_k \text{ of } t^{(k-1)} \cdot \Delta u^{(k)} \, dS$$

where $t^{(k-1)}$ is the traction that acted across that surface in the previous equilibrium state (the in-situ field for step 1), and $\Delta u^{(k)} = u^{(k)} - u^{(k-1)}$ is how far the wall moved between the two states. The traction on the new face falls from $t^{(k-1)}$ to zero while the face moves by Δu, hence the ½. Surfaces mined at earlier steps are already free of traction and contribute nothing, so the integral runs over the new surface only.

Discretised, each element contributes $w_e = \tfrac{1}{2} t^{(k-1)} \cdot \Delta u_e$ in kJ/m², and $\Delta W = \Sigma\, w_e A_e$, reported in MJ. With stress in MPa, displacement in mm and area in m², **1 MPa × 1 mm = 1 kJ/m²**, so no conversion is needed.

The traction is evaluated at the wall probe of §2.4. On the wall itself the two-sided average would not vanish on a face that is already free.

As-mined bookkeeping. In As-mined mode a step's shape is not all new surface:

- an element whose probe lies inside the previous step's void is **excluded** (`ExcludedInterior`): that ground was already mined, and no traction acted there;
- an element whose previous traction was already at least half relieved is **kept** but counted (`NearZeroTraction`): it is old wall re-covered by the new shape, and contributes almost nothing.

The excavated volume increment ΔV_k comes from the closed-body volumes of consecutive steps.

6.3 Released energy on DD elements

On DD elements the release is read directly from the solved discontinuities:

$$\Delta W_k = -\tfrac{1}{2} \Sigma \text{ over } \Pi_k \text{ of } t^{(k-1)} \cdot \Delta D^{(k)} \cdot A_e$$

over the elements Π_k mined at step *k*, with $t^{(k-1)}$ the traction across the reef before the step and ΔD the change in the solved discontinuity.

The leading minus is not a fudge. The solver's normal discontinuity D_n is opening-positive, so a mined seam under compression solves negative — it closes. The minus turns that closure into a positive release. For the same reason the `Closure_mm` column is reported positive for a seam that closes.

Elements mined at earlier steps are free of traction and contribute nothing. They are recognised by geometric matching — the same centre, area and normal in both steps — and counted as `CarriedOver`. This is why each DD row is prepared once and the identical elements enter every step (§3.8).

The touchdown count compares each element's closure with its own mining width; the mined volume is Σ area × width.

6.4 Path independence

In linear elasticity the total energy released depends only on the final geometry. Summed over any staging, $\Sigma\Delta W_k$ equals the single-step W of the final excavation.

- On FF models GEM4D reproduces this to about **0.4 %** on the validation models.
- On DD elements it is **exact to round-off** (0.000 % on the validation sequence), because the discontinuity is a solved unknown read straight off the element, with no probe or field evaluation in the increment.

6.5 Area normalisations

- **ΔW/ΔV** is well defined in both authoring modes.
- **ΔW/A** uses the new FF surface area. It is reported in Additive mode only, because in As-mined mode part of a step's mesh may re-cover existing wall.
- **ERR on DD** is ΔW over the area mined, measured in the reef plane (`ERR_MJm2`) and in plan (`ERR_plan_MJm2`). The plan area is the reef-plane area times cos(dip), so the plan figure grows as 1/cos(dip).

7. Excess shear stress and Kirsch tangential stress

7.1 Plane-specific ESS

On a plane with unit normal **n**, from the total, compression-positive stress tensor **S**:

- normal stress $\sigma N = \mathbf{n}^T \mathbf{S}\, \mathbf{n}$
- shear stress $\tau = \sqrt{(|\mathbf{S}\,\mathbf{n}|^2 - \sigma N^2)}$
- $\mathrm{ESS} = \tau - c - \tan\varphi \cdot \sigma N$

The slip direction is the direction of the shear traction on the plane; GEM4D reports it as a downward plunge (`BEM_SlipPl_deg`) and azimuth (`BEM_SlipAz_deg`).

The tensile clamp. The friction term c + tan φ · σN is a resistance, physical only while the plane is clamped (σN > 0). With the clamp on, it becomes c + tan φ · max(σN, 0). Because σN enters ESS with a minus sign, a negative σN in the unclamped form *raises* ESS.

7.2 Coulomb ESS

The largest ESS over all plane orientations at a point is a function of σ1 and σ3 only (Ryder, 1988, Eq. 13):

$$\mathrm{ESS}_{\mathrm{Coulomb}} = \sec\varphi \cdot (\sigma1 - \sigma3)/2 - \tan\varphi \cdot (\sigma1 + \sigma3)/2 - c$$

It is reached on the Mohr-Coulomb **critical plane**, whose normal lies in the σ1–σ3 plane at 45° + φ/2 from σ1 (the plane itself lies at 45° − φ/2 to σ1). There are two such planes, mirror images about σ1

and equally likely to slip; GEM4D draws both. Coulomb ESS has no tensile clamp. On the critical plane $\sigma N = (\sigma 1 + \sigma 3)/2 - (\sigma 1 - \sigma 3)/2 \cdot \sin\varphi$, which is compressive unless σ3 is tensile by more than $\sigma 1(1 - \sin\varphi)/(1 + \sin\varphi)$ — a third of σ1 at φ = 30°. In such a zone the formula, like the unclamped plane-specific ESS, counts the tension as driving stress. By construction, Coulomb ESS is never less than the plane-specific ESS at the same point.

7.3 Kirsch tangential stress

For a row with plunge and trend (Dip, DipDir) read as a drive's axis:

1. The axis **a** is built from the plunge and trend.
2. Two unit vectors $\mathbf{e_1}$, $\mathbf{e_2}$ perpendicular to **a** span the drive's cross-section.
3. The stress tensor is projected onto the section: the 2 × 2 in-plane tensor has components $e_i^T S e_j$.
4. Its principal values are **SigMax** ≥ **SigMin**.
5. The Kirsch maximum tangential stress on the wall of a circular opening is **Tang = 3 · SigMax − SigMin**.

SigMax and SigMin always lie between the global σ1 and σ3. Appendix A works one row through by hand, including the construction of the section axes.

8. Accuracy and validation

This chapter collects the measurements behind the guidance in the *User Guide*. Most were made on test geometries with known answers. Where a real mine model was used (§3.10, §8.2, §8.3, §8.7, §8.8, §8.9), the reference is a finer mesh or another tool, not an exact answer. Either way the figures show how the method behaves, not the accuracy of any particular site model. §8.9 gives the measurements behind the D/L column of the maps.

8.1 Thin bodies and element size

Measured on disc slabs and narrow stopes, with elements from twice a body's thickness down to a third of it, released energy, skin energy and wall stresses **converge as the mesh is refined** — but they have converged only once the elements are **no larger than the thickness**. Coarser than that, readings wander. On a 0.2 m-thick disc 20 m across, released energy read −5 %, +0.1 % and +2.7 % against the penny-crack value at elements 2.5, 1.7 and 1.25 times the thickness.

On the finest meshes tested, thin FF slabs read slightly **above** the zero-thickness (penny-crack) value, and by less as the slab gets thinner. On a 20 m-diameter disc: about +3 % at 1 m thick (11 880 elements), and −0.1 / +0.6 / +1.3 % at 0.5 m thick on 5 040 / 20 160 / 45 360 elements.

The penny value is a reference, not a floor. The exact released energy of a thin void can lie slightly *below* it: 0.7 % below for a spheroidal void 1 m thick at its centre and 20 m across. A reading a little under the penny value is not by itself an error.

8.2 Triangle shape

The integration decides how close a point is to an element from the element's **area**. A long, thin triangle is therefore treated as farther away than its tips really are, and the six-point rule used just outside the near band can be badly wrong for it. Against the exact integral, at 1.2 element sizes from an element's centre and in the worst direction, the single-element error is 2 % for an equilateral triangle, 5 % for a right isosceles one, 82 % for a right triangle with legs 1 : 2, and over 100 % at 4 : 1. In 95 directions out of 100 it stays below 1.3 %, 3 %, 11 % and 30–40 %. Shape is measured here as longest edge ÷ height, as in the *User Guide*'s 3 : 1 rule: the right isosceles triangle is 2 : 1 and the triangle with legs 1 : 2 is 2.5 : 1, inside that rule.

On a whole model the effect is much smaller, because every point sums many elements and the errors scatter in sign. On a 19 232-element mine model with 39 % of its triangles more elongated than 3 : 1, integrating every triangle accurately changed σ1 by a median of 0.3 MPa at the walls and 0.1 MPa one element into the rock — a twentieth to an eighth of what halving the element size changed (§8.7). The tail is longer: at 6.8 % of the wall points the change exceeded 5 MPa (99th percentile 20 MPa, largest 62 MPa). Cleaning the mesh (edge flips, collapsed needles, split caps) took the share of triangles beyond 4 : 1 from 25 % to 12.5 % and roughly halved that integration error at the walls: median 0.31 → 0.17 MPa, 90th percentile 3.3 → 1.7 MPa, share above 1 MPa 25 → 15 %. It changed neither the solver's iteration count (20 and 21) nor the difference from the refined mesh (median 6.3 MPa before and after).

Extreme slivers (longest edge ÷ height above 100) can still stop a solve: one CAD stope export triangulated with long needle facets throughout gave an unreliable solution. Remesh such a mesh (*User Guide* §6.3).

8.3 Coarsening a thin body

Remesh and smooth at an edge length larger than a body's thickness cannot keep the body's rim. On a 2.3 m-thick stope, a 4.5 m target removed half the volume and a 2.3 m target removed an eighth. **Remesh for BEM analysis**, which samples a thin body at about a third of its thickness, kept the volume within 1.5 % on the same stope.

Element size against thickness on whole bodies. The closed bodies of three mines, as **Remesh for BEM analysis** builds them (bodies over 10 m³), were each remeshed at an edge of a fixed multiple of their thickness 2V/A and repaired as the balloon wrap repairs. Volume kept:

Edge ÷ thickness	Mine 1, narrow stopes (108 bodies)	Mine 2 (4 bodies)	Mine 3 (14 bodies)
0.5	98.9 % (worst body 94 %)	100 %	99.3 %
0.75	93.8 %	100 %	93.3 %
1	78.3 %	92.6 %	82.7 %
1.5	33 %	74 %	46 %

This is why the triangle budget of **Remesh for BEM analysis** keeps every body's elements at no more than half its thickness (§4.1). Through the real tool (volume against the surface before the budget remesh): mine 3 at 150 000 triangles, 100 % and no self-intersections; mine 2 at 86 000, 99 %; mine 1 after **Remove internal surfaces**, 85 % at 150 000 and 99 % at 700 000 (it needs about 580 000 for every body at half its thickness).

8.4 Gaps and steps between DD pieces

On a 100 × 50 m test stope 2 m wide (DD, Kelvin, 5 m elements):

Join between two halves of the reef	Closure and released energy, against the continuous stope
Open 0.3 m gap in the plane of the reef	5 % low — the gap acts as a rib
Open 0.3 m step	4 % high — the two offset sheet edges reinforce each other
Gap or step closed (§3.7)	Within 0.3 %

This is why GEM4D closes gaps narrower than 1 m between the pieces of a mid-surface.

8.5 Released energy: the pressurised sphere

A spherical cavity under hydrostatic in-situ stress p releases $W = \pi p^2 a^3/(2G)$ on excavation (Jaeger, Cook & Zimmerman). GEM4D's faceted spheres converge to it from above: **+3.36 %** at 320 elements, **+2.48 %** at 1 280 and **+1.44 %** at 5 120. Appendix B works one element of this case by hand.

8.6 DD elements: the penny crack and curved reefs

Element size. Against Sneddon's penny-shaped crack, on a circular reef triangulated exactly to its outline:

Element edge	Elements	Error in crack volume (≈ released energy)
span / 12	216	+8.6 %
span / 20	600	+5.4 %

The error falls monotonically with refinement and is always positive: constant-strength elements are slightly too compliant at the mined edge. Appendix C works this case by hand.

Curved and folded reefs. A reef folded onto a cylinder of radius R converges to the flat answer as $(a/R)^2$, with a measured order of 1.99. A thin curved slab modelled as FF and as DD agrees to 2 %.

FF and DD on the same stope. A 62 × 30 m planar stope 2.3 m wide, modelled as a thin FF solid and as a DD surface, gives the same released energy to within a few percent.

Path independence. See §6.4: about 0.4 % on FF models, exact to round-off on DD elements.

8.7 Element size on a mine model

This is the refine-and-compare check of *User Guide* §13.6, run on a real mine model. The model — 19 232 elements, median element size √A 7.2 m (the area of an equilateral triangle with 11 m edges), nine elements in ten within one and a half element sizes of a sharp edge (a bend of more than 20° between neighbouring triangles) — was solved as built and with every triangle split into four (half the edge length, 76 928 elements), at GEM4D's default solver settings. The refined mesh, the reference, was integrated with the accurate rule for elongated triangles of §8.2, which GEM4D does not use. (With GEM4D's own integration on both meshes the wall row is the same to within one percentage point.) Results were compared at the same points: each element's wall probe (§2.4), and points 0.5, 1 and 2 element sizes into the rock. A relative change is the change ÷ the refined value, with a σ1 below 1 MPa counted as 1 MPa.

Point	σ1 change, median	σ1 relative change, median / 90th percentile	SED relative change, median / 90th percentile
Wall probe	6.4 MPa	18 % / 72 %	37 % / 172 %
0.5 element in	1.9 MPa	5.5 % / 49 %	11 % / 120 %
1 element in	1.1 MPa	3.3 % / 25 %	7 % / 72 %
2 elements in	0.6 MPa	2.1 % / 8.8 %	4 % / 23 %

Released energy changed by 0.54 % with the same integration on both meshes (0.39 % against the reference's accurate integration).

The changes gather at sharp edges: at the wall the median was 20 % within one and a half element sizes of an edge, and 7 % farther away. The coarse wall values read **high**: σ1 was above the refined value on 86 % of the elements, and a typical element's wall SED read about 35 % high (the median of the per-element changes). Across all wall elements the median SED was 10.5 against 6.2 kJ/m³, the area-weighted mean 11.2 against 7.3 kJ/m³ and the largest 843 against 651 kJ/m³. The ranking survived: the rank correlation between the two meshes was 0.98, for wall σ1 and wall SED alike, and 85 % of the elements in the top tenth of wall SED were the same. The refined mesh is better, not exact, so these figures measure the change, not the full error. Halving the element size made the solve about seven times as long (18 s to just over 2 min, BEM library 1.2.6.0).

8.8 The balloon wrap: checks and limits

Balloon wrap for BEM (§4.2) was checked against the IGM Python tool on the same inputs and scored by one script (*h* = voxel size):

Case	Python tool	GEM4D
Its own example, 8 000 triangles	0.03 % of survey points > 1 m outside; 23 585 m³	0.03 %; 23 611 m³
Two open-topped stopes 300 m apart on one layer	upper stope lost (11 141 of 24 000 m³)	both kept (23 977 m³)
CMS stope scan + 14 development levels, 2 m elements	15.7 % > 1 m outside	0.4 %
Stacked stope designs on one layer, 150 000 triangles	31 % missing, 25 % extra	0.8 % missing, 1.0 % extra

Missing and extra volume are measured against GEM4D's own clean surface of the same stopes (Remove internal surfaces, then Remesh for BEM). On the first and third cases, the GEM4D wraps solve in 17 and 15 GMRES iterations, against 16 and 14 for the Python tool's.

Two limits showed. Elements longer than a stope is narrow lose it: narrow stopes on 17 layers lost 17 % of the clean surface's volume at 2.8 m elements, with about 120 of the 150 000 triangles still crossing, and their volume matched it within 1 % at 1 m. The voxel sets the finest detail: on those narrow stopes at *h* = 1 m, shifting the grid by half a voxel changed the filled volume by 5 %.

8.9 Where to trust a stress: distance over element length

The **D/L** column of the maps (*User Guide* §13.6) gives at each point its distance D to the nearest element of that step's model, divided by that element's longest edge L. FF triangles and DD elements count alike; a rectangular DD element of an older model counts as two triangles, with its longer side as L. D/L is geometry only, so points inside an excavation carry it too.

A constant-strength element cannot follow the stress gradient across its own width (§2.1), so the error of a stress near a wall depends on how far the point is from the elements against how large they are. This was measured on two models, each solved as loaded and refined at several sizes, with σ1 compared with a much finer mesh of the same model:

Model	Meshes checked	Reference
A test model of five bricks (72 triangles as loaded)	As loaded; **Refine for BEM (keep shape)** at 2, 1 and 0.5 m at edges; uniform meshes of 2 m and 1 m	Refined at 0.25 m
A Map3D model of a stoping mine, its last as-mined snapshot (2 028 elements as loaded)	As loaded; refined at 4, 2 and 1 m at edges	Refined at 0.5 m (194 382 elements)

The points were placed over each model's outer walls in proportion to area and moved 0.25 to 50 m into the rock along the wall's normal: 2 400 round the bricks and 1 798 round the stope model, each read on every mesh — 21 592 readings, under a uniform in-situ stress. The error is |σ1 − σ1 of the reference| ÷ σ1 of the reference, sorted by the D/L of the mesh checked:

D/L	σ1 error: typical (median)	9 readings in 10 within	Readings within 2 %
below 0.25	5.3 %	49 %	27 %
0.25–0.5	1.8 %	6.2 %	54 %
0.5–1	0.8 %	2.7 %	84 %
1–2	0.35 %	1.3 %	96 %
2–4	0.17 %	0.5 %	99.5 %
4 and more	0.04 %	0.2 %	99.9 %

From D/L 1 on — one element length off the walls — 96 % of the readings were within 2 %. This is why a map's completion message counts the points in the rock below D/L 1.

Why the longest edge. With √A, the square root of the element's area, in place of the longest edge, the two models disagreed: at D/L 1–2 the median error was 0.75 % on both, but nine readings in ten were within 1.9 % on the bricks and only within 5.1 % on the stope model, whose long thin triangles as loaded read better than they are. With the longest edge the two agree closely at the same D/L, as loaded or refined: medians 0.32 % and 0.45 %, nine in ten within 1.0 % and 2.0 %.

Both models are flat-faced blocks under a uniform in-situ stress, and the reference is a finer mesh, not an exact answer. Read the table as a guide to where a model needs refining, not as the error at any one point.

9. Result changes from BEM library updates

GEM4D's solver is the BasRock BEM library. This chapter says which library updates changed results, so you can decide which earlier models to re-run; `BEM_Manuals_Changelog.md` keeps the measurements behind each row. Results from version 1.2.0.0 and earlier are superseded on every FF model, and those from before 1.2.0.0 on several more paths (§9.2, §9.3). The later versions moved results by no more than the error of the processor method's compression, and the GPU's results are the closer to the exact answer (§9.1).

9.1 BasRock.BEM 1.3.0.0 to 1.7.0.0 (October 2026)

No earlier result is superseded. Newest first:

Library version	What moves
1.7.0.0 — far field cached first (§5.4)	Which blocks are cached, as between two runs with different free RAM: a processor solve moves within the compression tolerance (on a seven-step mine model at most 6.4e-5 of the largest value per element, the released energy of the whole sequence by 2e-7).
1.6.0.0 — H-matrix build with vector instructions (§5.2)	Nothing: every stored number of the H-matrix is identical, so a processor solve gives the same answer.
1.5.0.0 — GPU maps and energy of half-space and DD models (§5.7)	GPU and processor agree within 0.001–0.003 % of the local stress, and within about 0.0001 % of the released energy.

Library version	What moves
1.4.0.0, 1.4.1.0 — GPU solves of half-space and DD models (§5.7)	New GPU solves move toward the exact answer by the processor method's compression error (a residual of 9e-5 to 3e-4 of the load, against about 1e-6 on the GPU). Kelvin results are unchanged.
1.3.0.0 — the GPU (§5.7)	New GPU solves and maps of Kelvin models move toward the exact answer: on the seven-step mine model of §5.7 by a median of 0.002 MPa and at most 1.2 MPa (the error of the processor method's compression).

To repeat a result the processor gave before the GPU took over its kind of model, clear **Use the GPU of this computer** in Setup (it stays clear until GEM4D closes). A result solved at the former **Fine** tolerance is not repeated exactly, because a setup saved at Fine opens as Standard (§5.7).

9.2 BasRock.BEM 1.2.1.0 (October 2026)

Version 1.2.1.0 of the BEM library is a follow-up to 1.2.0.0 (§9.3). **Results from 1.2.0.0 and earlier are superseded on every fictitious-force (FF) model**; DD-only results are unchanged.

What changed	What moved
Far-field integration tightened — the library integrated an element with a single point from too short a distance, which put the opposite face of a thin body meshed finer than its thickness into that band	Every FF result: by about 1 % or less on ordinary, bulky stopes; by several % on thin ones — on a 62 × 30 × 2.3 m planar stope meshed at a third of its thickness, released energy +4 % and the hangingwall-centre σ1 −7 %; skin energy on the broad faces of thin slabs by 10–30 % (a few kJ/m³). Thin slabs no longer read low (*User Guide* Chapter 6).
Stored and Released energy maps corrected element by element — the compressed (H-matrix) field evaluation these two buttons used could put a large error on single elements of a large thin model while the totals stayed within 0.5 %	Per-element Stored-energy and Released-energy maps — on a 45 360-element thin disc one skin element read 829 MPa where the exact value is 0.04 MPa, and 5 % of the elements were more than 1 MPa off. GEM4D evaluates both energy maps **directly** (the exact method), which was also the faster one on every model tested
Iterative-solver compression tightened	Models above 6 700 elements move slightly toward the exact answer — skin σ1 by up to about 1 MPa on large thin bodies, about 0.1 MPa on stopes.

Re-run models whose energy maps, thin-stope results or iterative solves came from 1.2.0.0 or earlier.

9.3 BasRock.BEM 1.2.0.0 (October 2026)

Version 1.2.0.0 of the BEM library was released after an independent audit that found and fixed errors in the library's numerical integration and matrix compression. **BEM results computed with earlier versions are superseded — re-run the models** wherever they used one of these paths:

Model or output	What moved
Any model with flat-faced excavations (box stopes, drives)	Wall and skin values (Stored energy, Kirsch, near-wall stresses) by about 5–9 %; stresses more than ~0.25 m from the walls only within discretisation error
Mindlin (half-space) models	Near-wall stresses and skin energy — on a test box 400 m deep the mean skin energy moved 13 % and stresses near the walls by up to ~7 MPa; shallow excavations also change near the ground surface

Model or output	What moved
Models above 6 700 elements (iterative solver)	Element strengths by up to ~7 %; DD reef closures were 56–78 % too low
Stored energy and Released energy on models of 1 024 elements or more	Released energy was 18–26 % too low — worse on thin stopes, where it could turn negative
A Mindlin model whose roof lies exactly on the ground surface	Released energy is now reported (it came back as *NaN*); the roof elements, which have no rock above them, are left out

Two improvements come with it: models with several separated excavations solve far faster on the iterative path (four staggered stopes: 337 s → 1.6 s), and direct solves of about 5 500–6 700 elements, which failed with an out-of-memory error, now complete.

Appendix A — Worked example: checking `BEM_Tang_MPa` by hand

This appendix walks one fictitious minode row through the whole tunnel-section Kirsch calculation, so you can reproduce `BEM_SigMax_MPa`, `BEM_SigMin_MPa` and `BEM_Tang_MPa` with a calculator or a spreadsheet and satisfy yourself that the numbers are what they claim to be. Nothing here is a separate implementation — it is the same arithmetic GEM4D runs, written out.

Before you start: tick **both** *Kirsch tangential* **and** *Full stress tensor (Sxx…Sxz)* in the *Map to datagrid* options (*User Guide* Chapter 14). Without the tensor columns the row shows you the answer but not the six numbers it was computed from, and there is nothing to check.

A.1 The example row

A minode in a gently declining development drive heading south-east, after *Map to datagrid*:

Column	Value	Where it comes from
`Easting` / `Northing` / `RL`	4250.0 / 7180.0 / −620.0	Your loaded minode CSV — the point the tensor was evaluated at. Not used in the arithmetic below.
`Dip`	2.5°	Your loaded minode CSV — the **tunnel axis** plunge (positive = drive falling; a rising incline is negative).
`DipDir`	130.0°	Your loaded minode CSV — the tunnel axis trend, degrees clockwise from North.
`BEM_Sxx_MPa`	32.40	Solved total stress (in-situ + induced) at that point.
`BEM_Syy_MPa`	21.70	″
`BEM_Szz_MPa`	28.10	″
`BEM_Sxy_MPa`	−4.60	″
`BEM_Syz_MPa`	3.20	″
`BEM_Sxz_MPa`	7.90	″
`BEM_SigMax_MPa`	**33.47**	← reproduced in §A.5
`BEM_SigMin_MPa`	**16.58**	← reproduced in §A.5
`BEM_Tang_MPa`	**83.83**	← reproduced in §A.6

Axes are mine coordinates: **x = East, y = North, z = up (RL)**, compression positive (*User Guide* §3.5). The stress tensor **S** is symmetric:

```
      | Sxx  Sxy  Sxz |       | 32.4  -4.6   7.9 |
S  =  | Sxy  Syy  Syz |   =   | -4.6  21.7   3.2 |   MPa
      | Sxz  Syz  Szz |       |  7.9   3.2  28.1 |
```

One note on decimals before you start. GEM4D works in full double precision and rounds only when writing a cell. Intermediates below are printed to 4–6 decimals, so re-typing a printed intermediate into the next step can land you one unit off in the last decimal shown. That drift never reaches the two decimals the grid displays: the checks that matter (§A.5, §A.6) come out exact.

A.2 Step 1 — the tunnel axis

The CSV gives the drive's plunge p and trend t as a **line**, so the first move is to build the unit vector **a** pointing along the tunnel. It heads toward the trend azimuth and tilts down by the plunge:

```
a = ( cos p · sin t ,  cos p · cos t ,  -sin p )
```

Note this is *not* the pole-of-a-plane formula (that one puts cos in the z slot). Kirsch is solved in the **cross-section of the drive** — the plane perpendicular to **a** — so the tunnel axis itself is the plane normal.

With p = 2.5° and t = 130° (cos p = 0.999048, sin p = 0.043619, sin t = 0.766044, cos t = −0.642788):

```
ax = 0.999048 × 0.766044  =  0.765315        (East)
ay = 0.999048 × -0.642788 = -0.642176        (North)
az = -0.043619                               (Up — negative, the drive is falling)
```

Sanity checks: |a| = 1; the horizontal part of **a** bears atan2(ax, ay) = atan2(0.765315, −0.642176) = 130° — the trend, as it should (that argument order gives an azimuth clockwise from North); and atan2(−az, √(ax² + ay²)) = 2.5° — the plunge. The **sign** of **a** does not matter to anything that follows (flipping it swaps the two in-plane axes but leaves the answer unchanged), which is also why a rising drive recorded as Dip = −2.5° toward 310° gives the identical result.

A.3 Step 2 — the two cross-section axes

The Kirsch relation is two-dimensional, so the tensor has to be projected onto the cross-section. That needs two perpendicular unit vectors lying *in* that section. GEM4D builds them with cross products, and for any drive flatter than about 64° plunge they work out to the two directions you would sketch on a section anyway — **horizontally across the drive** (wall to wall) and **roof to floor**:

```
u  (across the drive) = ( cos t ,           -sin t ,           0      )
v  (roof to floor)    = ( -sin p · sin t ,  -sin p · cos t ,  -cos p )
```

For this row:

```
  u = ( -0.642788 , -0.766044 ,  0        )   horizontal, bearing 220°  (trend ± 90° —
square across the drive)
  v = ( -0.033414 ,  0.028038 , -0.999048 )   plunging 87.5° toward 310°  (straight down
the section: vertical, tipped back by the 2.5° grade)
```

u, **v** and **a** are mutually perpendicular unit vectors — check any pair with a dot product and you get 0.

A.4 Step 3 — project the stress tensor onto the cross-section

Squeeze the 3×3 tensor down to the 2×2 stress state seen in the drive's cross-section. Multiply **S** into each axis first:

```
S·u :
  x:  32.4 × -0.642788  +  -4.6 × -0.766044  +   7.9 × 0         = -17.3025
  y:  -4.6 × -0.642788  +  21.7 × -0.766044  +   3.2 × 0         = -13.6663
  z:   7.9 × -0.642788  +   3.2 × -0.766044  +  28.1 × 0         =  -7.5294

S·v :
  x:  32.4 × -0.033414  +  -4.6 ×  0.028038  +   7.9 × -0.999048 =  -9.1041
  y:  -4.6 × -0.033414  +  21.7 ×  0.028038  +   3.2 × -0.999048 =  -2.4348
```

```
    z:   7.9 × -0.033414  +   3.2 ×  0.028038  +  28.1 × -0.999048 = -28.2475
```

then dot each result back onto an axis:

```
  σ_uu = u · (S·u) = -0.642788 × -17.3025  +  -0.766044 × -13.6663  +  0 × -7.5294
= 21.5909 MPa
  σ_vv = v · (S·v) = -0.033414 ×  -9.1041  +   0.028038 ×  -2.4348  +  -0.999048 × -28.2475
= 28.4566 MPa
  σ_uv = u · (S·v) = -0.642788 ×  -9.1041  +  -0.766044 ×  -2.4348  +  0 × -28.2475
=  7.7172 MPa
```

Read physically: **21.59 MPa horizontally across the drive, 28.46 MPa roof-to-floor, with 7.72 MPa of shear between them.**

A.5 Step 4 — the in-plane stress extremes

σ_uu, σ_vv and σ_uv are a plain 2-D stress state, so SigMax and SigMin come straight off **Mohr's circle** — centre = the mean, radius = r:

```
  mean = (σ_uu + σ_vv) / 2 = (21.5909 + 28.4566) / 2 =  25.0237
  half = (σ_uu - σ_vv) / 2 = (21.5909 - 28.4566) / 2 =  -3.4328
  r    = sqrt( half² + σ_uv² ) = sqrt( (-3.4328)² + (7.7172)² ) = sqrt(71.3392) = 8.4463

  SigMax = mean + r = 25.0237 + 8.4463 = 33.4700  →  BEM_SigMax_MPa = 33.47  ✓
  SigMin = mean - r = 25.0237 - 8.4463 = 16.5775  →  BEM_SigMin_MPa = 16.58  ✓
```

Both grid columns are now reproduced. Note SigMax ≥ SigMin by construction, and both lie *in the cross-section* — they are **not** the same thing as `BEM_S1_MPa` / `BEM_S3_MPa`, the full 3-D principals, which take no notice of the drive direction. They are always bracketed by them: for this tensor the global principals are σ1 = 38.63, σ2 = 27.10, σ3 = 16.47, and indeed 38.63 ≥ 33.47 ≥ 16.58 ≥ 16.47. Only a drive exactly perpendicular to σ1 gets SigMax all the way up to σ1 — which is precisely why a drive square to σ1 reports a higher `BEM_Tang_MPa` than one along it.

A.6 Step 5 — the Kirsch tangential stress

```
  BEM_Tang_MPa = 3 × SigMax - SigMin = 3 × 33.4700 - 16.5775 = 83.8324  →  83.83  ✓
```

This last step is deliberately checkable straight off the grid, with no intermediate working at all: **3 × 33.47 − 16.58 = 83.83**. If those three columns ever fail that arithmetic on a row, something is genuinely wrong — please report it.

A.7 The same thing in a spreadsheet

To check a whole column rather than one row, the projections in §A.4 collapse to three formulas (with the tunnel dip in `A2`, trend in `B2`, Sxx…Sxz in `C2:H2`):

```
  ux = COS(RADIANS(B2))                vx = -SIN(RADIANS(A2)) * SIN(RADIANS(B2))
  uy = -SIN(RADIANS(B2))               vy = -SIN(RADIANS(A2)) * COS(RADIANS(B2))
  uz = 0                               vz = -COS(RADIANS(A2))

  σ_uu = Sxx·ux² + Syy·uy² + Szz·uz² + 2·(Sxy·ux·uy + Syz·uy·uz + Sxz·ux·uz)
  σ_vv = Sxx·vx² + Syy·vy² + Szz·vz² + 2·(Sxy·vx·vy + Syz·vy·vz + Sxz·vx·vz)
  σ_uv = Sxx·ux·vx + Syy·uy·vy + Szz·uz·vz + Sxy·(ux·vy + uy·vx) + Syz·(uy·vz + uz·vy) +
Sxz·(ux·vz + uz·vx)
```

then Mohr's circle exactly as in §A.5.

A.8 Notes on the fine print

- **The choice of in-plane axes does not affect the answer.** Across-drive and roof-to-floor are just a convenient pair. Any two perpendicular axes lying in the cross-section give different σ_uu, σ_vv and σ_uv, but the *same* Mohr's circle — the centre and radius are invariant under rotation within the plane, so SigMax, SigMin and hence `BEM_Tang_MPa` are unchanged. This is why the

recipe above is valid at every plunge even though GEM4D internally switches its reference axis for drives steeper than about 64° plunge (shafts and steep declines, where the across-drive vector would degenerate and a different pair is used instead).

- **`NaN` rows.** If Dip/DipDir or any tensor component is missing or non-numeric, all three columns are written as `NaN` together — you will never see a stress extreme sitting next to a `NaN` tangential.
- **Interior rows.** With *Iso-surface treats interior as in-situ* ticked in Setup, an `Interior` row's tensor columns hold the **in-situ** tensor, and the Kirsch columns are computed from that same in-situ tensor — so the row still checks out internally. See *User Guide* Chapter 13.
- **What the number means** is a separate question from whether it is computed correctly. `BEM_Tang_MPa` is only meaningful if the row's Dip/DipDir genuinely describes the tunnel direction at that point (*User Guide* Chapter 14 — minode data), and it inherits every assumption in *User Guide* Chapter 20 — linear elastic, dry, constant-stress elements near the wall — plus the Kirsch idealisation itself: a circular opening, long compared with its diameter, evaluated at the boundary.

Appendix B — Worked example: checking a released-energy row by hand

This appendix works **one element** of the released-energy validation case end to end — traction, wall movement, ½·t·Δu, and the area weight — so you can reproduce a `w_kJm2` row with a calculator and satisfy yourself the map means what *User Guide* Chapter 17 says it does. As in Appendix A, nothing here is a separate implementation; it is the same arithmetic GEM4D runs, written out.

The case is the one the implementation is validated against: a **spherical cavity in an infinite medium under hydrostatic compression**, which has a closed-form answer to check the total against.

B.1 The example

Quantity	Value	Where it comes from
Cavity radius **a**	5.0 m	The geometry
Hydrostatic in-situ stress **p**	30.0 MPa	Setup → In-Situ Stress (σv = σH = σh = 30)
Young's modulus **E**	70 GPa	Setup → Rock Properties
Poisson's ratio **ν**	0.25	″
Shear modulus **G** = E / (2(1+ν))	28 GPa = 28 000 MPa	Derived
The element	a facet whose centroid sits on the **+x** axis of the cavity	Any facet works; this one has the tidiest normal
Element area $\mathbf{A_e}$	0.2443 m²	From the mesh (≈ 312.7 m² of faceted surface over 1 280 elements)

Axes are mine coordinates, **compression positive** (*User Guide* §3.5), and element normals point **into the excavation** (the library's convention; GEM4D turns every mesh that way itself, §3.2). Rock occupies |x| > a, so at the point **x = (+5, 0, 0)** the normal into the void is

```
n̂ = (-1, 0, 0)
```

B.2 Step 1 — the traction that acted on the face

This is step 1 of the sequence, so the previous state is the **in-situ** field. With compression-positive hydrostatic stress the tensor is **σ = p·I**, and the traction is the ordinary **t = σ · n̂**:

```
t = 30 · | 1 0 0 |   ( -1 )     ( -30 )
         | 0 1 0 | · (  0 )  =  (   0 )  MPa
         | 0 0 1 |   (  0 )     (   0 )
```

So **t = (−30, 0, 0) MPa** — 30 MPa pointing along −x, i.e. *toward* the cavity from this face. (Sanity check on the convention: the material that used to fill the cavity was pushing the wall **outward**, in +x. The compression-positive tensor gives the equal-and-opposite bookkeeping vector, which is what pairs with the convergence below to give a positive product. *User Guide* Chapter 17.)

B.3 Step 2 — how far the wall moved

For an unsupported spherical cavity under remote pressure *p*, the Lamé thick-sphere solution with the outer radius taken to infinity gives a uniform **inward** wall displacement

```
δ = p·a / (4G) = 30 × 5 / (4 × 28 000) = 150 / 112 000 = 0.00133929 m = 1.33929 mm
```

The wall converges **into the void**, i.e. along **+n̂**, so at our element

```
Δu = δ · n̂ = 1.33929 × ( -1, 0, 0 ) = ( -1.33929, 0, 0 )  mm
```

($u^{(0)} = 0$ because GEM4D reports *induced* displacement, measured from the pre-mining state — so for step 1 the increment Δu is simply $u^{(1)}$.)

B.4 Step 3 — the release density

```
w_e = ½ ( t · Δu )
    = ½ [ (-30)(-1.33929) + (0)(0) + (0)(0) ]
    = ½ × 40.1786
    = 20.089  kJ/m²
```

Both vectors point the same way, so the product is **positive** — energy is released, not absorbed. This is the number in the `w_kJm2` column, and the unit falls out of the mixed units directly: **MPa × mm = kJ/m²** (*User Guide* Chapter 17).

B.5 Step 4 — the element's contribution

```
w_e · A_e = 20.089 × 0.2443 = 4.908 kJ
```

Every facet of a hydrostatically loaded sphere carries the same density (the geometry is symmetric), so summing over the whole surface is one multiplication:

```
W = w_e × A = 20.089 kJ/m² × 4π(5)² m²
  = 20.089 × 314.159
  = 6 311 kJ = 6.311 MJ
```

B.6 The check

The closed form for a pressurised spherical cavity (Jaeger, Cook & Zimmerman) is

```
W = π p² a³ / (2G)
  = π × 30² × 5³ / (2 × 28 000)
  = π × 900 × 125 / 56 000
  = 6.3112  MJ
```

with **p in MPa, a in m, G in MPa → the answer in MJ**. The hand-worked total above reproduces it to the digits shown, which is the point: the per-element density, the ½, the sign and the unit chain are all doing what *User Guide* Chapter 17 claims.

What GEM4D actually reports is slightly higher, because a faceted mesh is not a sphere. On the validation series the computed total converges to it from above (§8.5), so on a reasonably refined

model expect W within a percent or two of the closed form, always on the high side. If your own check lands *negative*, or wildly off, that is a real signal: check the mesh — normals, watertightness, slivers and zero-volume bodies (*User Guide* Chapter 6) — and the in-situ stress entered in Setup (*User Guide* Chapter 5).

B.7 Checking a whole column in a spreadsheet

For an arbitrary model you will not know δ analytically, but you can still audit the **relationship** on a single model, whose release is that of step 1. **Released on surfaces** loads its map (`w_kJm2`) into the BEMResult table; run **Map to datagrid** on it with **…also Ux / Uy / Uz components** ticked (*User Guide* §13.2). For step 1 the increment Δu is the row's own displacement, so for each row confirm

```
w_kJm2  =  0.5 × ( tx·Δux + ty·Δuy + tz·Δuz )      [ MPa × mm ]
```

with **t** the in-situ tensor, built from the Setup values, projected on the element's normal into the void. The points lie on the walls, so leave *Iso-surface treats interior as in-situ* unticked and skip a row whose `BEM_Status` reads `Interior` (its cells are blank). On a later step of a sequence the traction is the *previous* step's total stress at the same point, and the datagrid map of a sequence gives each step's displacement as a size only, not its components — which is why step 1 is the one that audits cleanly with nothing but the Setup dialog in front of you.

Appendix C — Worked example: checking a DD element by hand

Appendix B checks one excavation-wall element of *User Guide* Chapter 17. This appendix does the same for one **DD element** of a reef (*User Guide* Chapter 18): in-situ traction → resolved into the element's own axes → solved discontinuity → release density → area weight → whole-reef ERR against a closed form. Same arithmetic GEM4D runs, written out.

C.1 The case

The Sneddon **penny-shaped crack** — a circular reef in an infinite elastic body under a uniform field normal to it. It is the natural benchmark because a crack of zero thickness is exactly what fictitious-force elements cannot represent — they need a closed volume — and it has an exact answer.

Quantity	Value
Reef radius *a*	100 m (a 200 m span)
Field normal to the reef, *p*	60 MPa
Young's modulus *E*	70 GPa
Poisson's ratio *v*	0.25
Shear modulus *G* = *E*/2(1+*v*)	28 GPa
DD element size	10 m (span / 20, what **Suggest** fills in)

The reef is horizontal, so its local axes can be taken as ŝ = East, t̂ = North, **n̂ = up**. Take the element at the centre of the reef.

C.2 Step 1 — the traction that acted on the reef before mining

In-situ is vertical compression *p* with no shear, so the stress tensor at the element is σ_zz = 60 MPa, everything else zero. The traction on a plane with normal n̂ is **t = σ·n̂**:

```
t = σ · n̂ = ( 0, 0, 60 ) MPa      (n̂ = (0,0,1))
```

Resolved into the element's own axes (this is the step that matters on a dipping reef — here it is trivial because the axes are already aligned):

```
t_s = t·ŝ =  0 MPa
t_t = t·t̂ =  0 MPa
t_n = t·n̂ = 60 MPa      ← the reef was carrying 60 MPa of normal load
```

For step 1 this is simply the in-situ field. For a later step it is the **previous solved state's** total stress at the element's centre — the reef sees whatever the earlier mining left behind, which is exactly why an element at the abutment releases more than one in the middle of a mined-out area.

C.3 Step 2 — how far the reef closed

This is a **solved unknown**, not a field evaluation — GEM4D reads it straight off the element and reports it in the `Closure_mm` column. Sneddon's closed form for the centre of a pressurised penny crack gives the total opening

```
δ_max = 4(1−ν)·p·a / (π·G)
      = 4 × 0.75 × 60 × 100 / (π × 28 000)      [MPa, m, MPa]
      = 18 000 / 87 965
      = 0.20463 m = 204.63 mm
```

The solved reef closes slightly more: constant-strength elements are slightly too compliant at the mined edge, and at span / 20 the crack volume comes out about 5 % high (§8.6). The steps below use the closed-form value, so every number can be checked; put your row's `Closure_mm` in its place to check that row.

> **Sign.** Internally the solver stores the closure as **$D_n = -0.20463$ m**: its normal discontinuity is *opening-positive*, and the seam closed. GEM4D reports the closure **positive**. Both conventions are in play in the formula below, which is why it carries a leading minus.

C.4 Step 3 — the release density

```
w_e = −½ ( t_s·D_s + t_t·D_t + t_n·D_n )
    = −½ [ 0 + 0 + (60)(−0.20463) ]        [MPa × m]
    = −½ × (−12.278)
    = 6.139  MJ/m²  =  6 139 kJ/m²
```

Positive — energy released. The two minuses cancel: the reef was in compression (*t_n* positive) and it closed (*D_n* negative). **A negative value here means the sign or the frame was mishandled**, never a physical result for a mined seam.

Unit fact: MPa × m = MJ/m², so with closure in metres the density lands in MJ/m² directly; the `w_kJm2` column is this × 1000. With no shear on the reef, as here at step 1, a row of the panel map checks as `w_kJm2` = ½ × `tN_MPa` × `Closure_mm` (MPa × mm = kJ/m²).

C.5 Step 4 — the element's contribution, and the reef total

```
ΔW_e = w_e · A_e        ← this one element, A_e its area
```

Summed over every element mined, this is the reef's ΔW, and ΔW over the reef-plane area mined (the sum of the element areas) is its ERR. GEM4D writes them to the summary file as `dW_MJ`, `A_reef_m2` and `ERR_MJm2` (*User Guide* §18.3).

C.6 The check

Sneddon's released energy for the whole crack is

```
W = 4(1−ν)·p²·a³ / (3G)          ( = p²a³/G at ν = 0.25 )
  = (60)² × (100)³ / 28 000       [MPa², m³, MPa]
  = 3 600 × 1 000 000 / 28 000
  = 128 571 MJ = 128.57 GJ
```

and the mean ERR over the true disc area π·a² = 31 416 m² is

```
ERR = 128 571 / 31 416 = 4.093 MJ/m²
```

GEM4D's ΔW comes out high, as constant-strength elements always do at the mined edge: on a disc triangulated to its outline, by about 5.4 % at span / 20, the **Suggest** default, and 8.6 % at span / 12 (§8.6, *User Guide* §18.5). The error falls as the elements are refined. Compare ERR with 4.093 MJ/m² in the same way.

C.7 On a dipping reef

Nothing above changes except §C.2. For a reef dipping *θ* the traction resolves into three non-zero components, the two ride terms contribute real work, and the reef-plane area is the sum of the element areas while the plan area is that × cos θ. GEM4D reports **both** — `ERR_MJm2` (reef plane, the headline) and `ERR_plan_MJm2`. On this flat example the two columns are identical; on a 30° reef the plan figure is 15 % higher, and on a 70° reef it is nearly three times higher, which is why *User Guide* Chapter 18 uses the reef-plane figure and says so.

Appendix D — Measured principal stresses: the physics checks and a worked example

§1.6 describes the measured principal stresses. This appendix checks them against the balance with gravity and a free ground surface, and works one measurement through.

D.1 Balance with gravity

With z upward and compression positive, equilibrium is $\partial\sigma_{ij}/\partial x_j = b_i$ with body force b = (0, 0, −γ). For a field that varies only with depth this requires dSxz/dd = dSyz/dd = 0 and dSzz/dd = γ: no shear on horizontal planes may change with depth. The Gravitational description satisfies it at any kH, kh and trend. *Linear* with one gradient b = γ on all three rows satisfies it for any orientation of the axes, because G = γ·I has no shear. *Proportional to depth*, and *Linear* with different gradients b_i, satisfy it only when one axis is vertical: with tilted axes, G has shear components, which imply a horizontal body force of γ_h = (G_xz² + G_yz²)^½ that does not exist. Constant terms — the a_i of *Linear*, such as locked-in stresses — never upset the balance: a constant stress has no divergence. The BEM still solves — each excavation is loaded by the traction S·n on its walls — but every void then carries a small net horizontal force. GEM4D notes γ_h when it exceeds 0.05 kN/m³.

D.2 The half-space

A traction-free horizontal ground surface also requires Szz = Sxz = Syz = 0 at d = 0. Together with the balance above, a field that varies only with depth must then have a vertical principal axis. Tilted measured axes describe the stress locally — near a fault, under topography. Where the field still has a stress at the ground surface (*None (constant)*, or *Linear* with constants a_i), its shear on horizontal planes there cannot meet the Mindlin kernel's free surface: GEM4D notes it, and stresses close to the surface are approximate. *Proportional to depth*, and *Linear* with every $a_i = 0$, are zero at the surface and meet it; what they miss is the balance with gravity above, which the sideways-force note reports. (A Uniform σv with the Mindlin kernel is not traction-free at the surface either.)

D.3 A worked example

The example is checked by an independent calculation, and the same numbers pin the program's tests. Rows 45 MPa at 085/08, 30 MPa at 176/10 and 23 MPa at 318/77, mine grid, measured at RL −850 with the ground surface Z at 0 (d_ref = 850 m):

Quantity	Value
Angles between the entered directions	rows 1–2 89.59°, 1–3 89.91°, 2–3 89.69°

Quantity	Value
Turn of each axis by the squaring-up	0.210°, 0.256°, 0.160°
Squared-up axes (trend/plunge)	084.80/07.92, 176.23/10.12, 317.42/77.09
S_ref: Sxx, Syy, Szz, Sxy, Syz, Sxz (MPa)	44.435, 29.931, 23.634, 1.501, 0.937, −3.070
Principal values of S_ref	45, 30, 23 (to 10^{-9})
Proportional, at d = 1 000 m: σ1, σ2, σ3	52.941, 35.294, 27.059 MPa
… σv (= Szz), and γ·d at 27 kN/m³	27.805 MPa, 27.000 MPa (+3.0 %)
… σH, σh of the horizontal block; azimuth of σH	52.457, 35.032 MPa; 84.15°
… regime	reverse: S3 is 12.9° from vertical
Proportional: implied horizontal body force	3.8 kN/m³ (G_xz = −0.0036, G_yz = 0.0011 MPa/m)
The same rows read from true north with θg = 15° (trends 100, 191, 333)	S_ref identical to 10^{-12}
The 30 MPa row's direction derived instead	176.46/10.25 before squaring up; S_ref changes by at most 0.049 MPa