

Anhang B — Rechenkern

Quelltext zum Statik-Dossier «Hochsitz mit Netz»

VERSION 3.0 · 02.09.2026 · ZUR PRÜFUNG

BUILD ADBA937

5 DATEIEN · 936 ZEILEN

Dieser Anhang druckt den Code ab, der sämtliche Zahlen des Statik-Dossiers erzeugt. Er läuft bei jedem Build und unverändert im Browser (Statik-Explorer). Lesereihenfolge: Eingabewerte in der Datenquelle, daraus abgeleitete Masse, dann Formeln und Nachweise; die Tests halten die Goldwerte fest, gegen die jede Änderung geprüft wird. Prüfsummen (SHA-256) identifizieren jede Datei eindeutig — die Kurzform steht auch im Dokument selbst.

NR.	DATEI	INHALT	ZEILEN	SHA-256 (ANFANG)
B.1	projekt.toml	Datenquelle — alle Eingabewerte (Masse, Material, Lasten, Befestigung, Bedingungen)	344	320399cb755f79f5
B.2	src/statik/geometrie.ts	Abgeleitete Geometrie aus den Primärmassen	58	85ff6ffd1334a2b8
B.3	src/statik/rechenkern.ts	Rechenkern — Lasten, Seilmodell, Nachweise, Verbindungsmittel, Mengen	317	19e0f3cdf03da4cb
B.4	kit/src/lib/raster.ts	Rasterpositionen (Kit) — Schrauben- und Augplattenreihen	29	86083242cdb1a8f0
B.5	src/statik/rechenkern.test.ts	Tests — Goldwerte und Grenzfälle	188	307f55aba0b05eb2
Gesamt		Prüfsumme über alle Dateien	936	29ccb2a75bbf5999

Ausführen: `npm test` prüft den Rechenkern gegen die Goldwerte, `npm run build` erzeugt Dokumente, Abbildungen und PDFs neu. Alle Resultate des aktuellen Builds: `results.json`. Gesamt-Prüfsumme `29ccb2a75bbf599930124eebdc2627aba6607234b9057d7b12c5580490def614`.

B.1 projekt.toml

Datenquelle — alle Eingabewerte (Masse, Material, Lasten, Befestigung, Bedingungen) · 344 Zeilen · SHA-256
320399cb755f79f5d96baf753a14c0885ea900f10b24f8d901139c07dd77cbd7

```

1  # =====
2  # Hochsitz mit Netz – DIE Datenquelle des Projekts.
3  # Alles, was in Dokumenten, 3D-Modell, Explorer und Stückliste als Zahl oder
4  # Text erscheint, kommt von hier oder wird im Rechenkern daraus abgeleitet.
5  # Masse in cm (Geometrie), Kräfte in kN, Spannungen in N/mm², Gewichte in kg.
6  # Platzhalter in Texten: {g.luecke_cm}, {r.eta|pct}, {p.gelaender.anker_anforderung_kn}
7  # =====
8
9  [projekt]
10 kuerzel    = "hochsitz"
11 titel      = "Hochsitz mit Netz"
12 undertitel = "Kletternetz zwischen den Hochsitzen im Effinger"
13 ort        = "Effinger Bern"
14 traeger    = "YOLU"
15 kicker     = "YOLU · Effinger Bern · Jugendprojekt"
16 lead       = "Ein geknüpftes Kletternetz zwischen den beiden Hochsitzen im Effinger – gemeinsam
geplant, transparent gerechnet und im Workshop selbst gebaut. Hier ist das ganze Projekt: zum Drehen,
Nachrechnen, Prüfen und Bauen."
17 verfasst   = "Projektteam «Hochsitz mit Netz» (YOLU), rechnerische Aufbereitung mit transparentem
Rechenkern"
18 stand      = 2026-09-01
19 eckdaten   = [
20   "Workshop {p.termine.workshop|datumLang}",
21   "Rahmen {g.rahmen_x_cm} × {g.rahmen_z_cm} cm · Netz ≈ {g.netz_m2|m2}",
22   "Netzebene {g.ok_rahmen_cm} cm",
23 ]
24 fusszeile  = "Erstellt vom Projektteam (YOLU) mit Unterstützung von Claude."
25
26 [termine]
27 workshop  = 2026-09-04
28
29 [modell]
30 titel      = "3D-Modell"
31 kategorie   = "Modell"
32 beschreibung = "Drehen, zoomen, Ebenen ein- und ausblenden – bis zur einzelnen Schraube"
33
34 [explorer]
35 titel      = "Statik-Explorer"
36 kategorie   = "Werkzeug"
37 beschreibung = "Wie stark biegt sich das Netz mit drei Personen durch? Schieberegler für Personen,
Knüpf-Durchhang und Randabstand – alle Kräfte und Nachweise rechnen live mit."
38
39 # -----
40 # Dokumente: Version, Stand, Status und Änderungsverzeichnis je Dokument.
41 # Hauptversion steigt bei Lasten/Geometrie/Nachweisen/Bedingungen (neue Prüfung),
42 # Nebenversion bei redaktionellen Klarstellungen.
43 # Konvention seit 3.0: alle Dokumente tragen dieselbe Versionsnummer wie das Statik-Dossier;
44 # springt das Dossier, ziehen die anderen mit (Eintrag «an das Dossier angeglichen»).
45 # -----
46
47 [dokumente.statik-dossier]
48 titel      = "Statik-Dossier"
49 kategorie   = "Dokument"
50 beschreibung = "Das massgebende Dokument zur Prüfung: Lasten, Nachweise, Schraubenplan, Bedingungen
{B} – mit offenem Rechenkern und Prüf-Checkliste für die Fachperson."
51 version     = "3.0"
52 stand       = 2026-09-02
53 status      = "zur-pruefung"
54 aenderungen = [
55   { version = "3.0", datum = 2026-09-02, was = "Holz-Bemessungswerte korrigiert (Hinweis der
prüfenden Fachperson): Nutzlasten durch Personen zählen nach EC5 als mittlere Einwirkungsdauer, k_mod =

```

0.8 statt 0.9 (kurz), Nutzungsklasse 1/2 (im Gebäude). $f_{m,d}$ damit 14.8 statt 16.6 N/mm²; Ausnutzung des Längsbalkens 86 statt 76 % (Varianten Knüpfmass 0/5/10 cm: 86/85/84 %) – Nachweis erfüllt, aber neu im gelben Bereich (80–100 %, kleine Reserve). Auflagerpressung (7 statt 6 %) und Blattschulter (unter 5 %) bleiben unkritisch; alle übrigen Nachweise unverändert. Ampel-Schwellen (gelb ab 80 %, rot ab 100 %) neu als Daten in projekt.toml, Legende unter der Nachweistabelle; die Aussage «alle Nachweise erfüllt» gilt bis 100 % Ausnutzung, nicht nur bei Grün. Schraube S2 auf Hinweis des Schreiners weiter nach vorne: Kopf 3.2 statt 4 cm hinter der Frontkante (= 4d Randabstand, näher geht nicht); sie kreuzt die Blattfuge bei $z = 6.7$ statt 7.5 cm und tritt bei $z = 10.1$ statt 10.9 cm in die Deckplatte ein – mehr Reserve, damit sie bei 30° sicher im unteren Blatt des Längsbalkens bleibt. Länge, Neigung und Spitzenlage unverändert." },

56 { version = "2.1", datum = 2026-09-02, was = "Anhang B neu als Prüfsummen-Tabelle (Datei, Zeilen, SHA-256); der vollständige Quelltext – Datenquelle, Geometrie, Rechenkern, Raster, Tests – erscheint als separater Anhang B (eigenes PDF und Webseite /quelltext/) mit Syntaxhervorhebung und Zeilennummern. Das Dossier-PDF wird dadurch sechs Seiten kürzer. Inhaltlich unverändert." },

57 { version = "2.0", datum = 2026-09-01, was = "Neuaufbau aus einer Datenquelle (projekt.toml) mit Rechenkern in TypeScript; ersetzt Version 1.3 vom 31.08.2026. Annahmen, Lasten und Nachweise inhaltlich unverändert; Ausnutzungen neu direkt aus dem Rechenkern gerundet (Augplatte 23 statt 24 %). Gebrauchslage 1 Person neu mit symmetrischer Lastaufteilung (H längs 1.63 statt 1.53 kN) und korrekter lichter Höhe ab Knüpfebene 208 cm (186 statt 181 cm). Höhenbezug auf OK Podest vereinheitlicht (OK Deckplatte 200): Geländerseil in Feldmitte 90 statt ~95 cm über OK Rahmen; Kabinendach +12 statt +17 cm über Deckplatte (B5) – vor Ort nachmessen. Verbindungsmittel vereinfacht: zwei Schrauben je Ecke (S1 senkrecht, S2 8×180 um 30° geneigt, Kopf 4 cm hinter der Kante, damit sie die Blattfuge kreuzt und das Blatt des Längsbalkens durchläuft; Köpfe bündig, nicht versenkt) und zwei zusätzliche je Querbalken statt drei je Ecke und alle 50 cm; Ringdruck formschlüssig über die Blattschultern; Abhebekraft aus dem Kippmoment des Längsbalkens hergeleitet; neuer Nachweis Verschieben des Rahmens; geometrische Prüfung der Schraubenlage im Rechenkern. Augplatten (Deckplatten) neu verteilt: Reihen beginnen in den Ecken mit zwei Platten je Ecke (eine je Balkenseite, Randseil in zwei Mal 45° umgelenkt), Mindestabstand 30 cm am Netzrand und 60 cm am Geländer: 50 statt 67 Platten. Stahlsäule als doppelter H-Träger; Geländer-Deckenvariante mit vertikalen Randseilen an den Netzenden." },

58]

59

60 [dokumente.bauanleitung]

61 titel = "Bauanleitung Workshop"

62 kurz = "Bauanleitung"

63 kategorie = "Workshop"

64 untertitel = "Hochsitz mit Netz · Effinger · {p.termine.workshop|datumLang}"

65 beschreibung = "Rollen, Zuschnitt, Schrauben- und Augplattenplan, Ablauf mit Kontrollpunkten und Materialliste für den Bau-Workshop – inklusive der wichtigsten Regel: zuerst vor Ort messen, dann sägen."

66 version = "3.0"

67 stand = 2026-09-02

68 status = "entwurf"

69 grundlage = "Grundlage: Statik-Dossier v{p.dokumente.statik-dossier.version} (massgebend)"

70 aenderungen = [

71 { version = "3.0", datum = 2026-09-02, was = "Schraubenplan: Kopf von S2 neu 3.2 statt 4 cm hinter der Frontkante (Hinweis Schreiner); Zeichnungen und 3D-Ansichten folgen der Datenquelle. Versionsnummer an das Statik-Dossier angeglichen – ab jetzt tragen alle Dokumente dieselbe Nummer." },

72 { version = "2.1", datum = 2026-09-02, was = "Für den Workshop eigenständig lesbar: Schraubenplan, Augplattenplan und die zugehörigen Abbildungen aus dem Dossier eingebettet (dieselbe Datenquelle), Ablauf in sechs Schritte mit Kontrollpunkten gegliedert, Materialliste als Anhang. Inhaltlich unverändert." },

73 { version = "2.0", datum = 2026-09-01, was = "Neuaufbau aus der Datenquelle; ersetzt Version 1.1 vom 30.08.2026. An das Statik-Dossier angeglichen: Augplatten statt Ösen, Geländer-Deckenvariante ohne Abhängungen, Pfosten nur noch als Variante." },

74]

75

76 [dokumente.betriebskonzept]

77 titel = "Betriebs- und Wartungskonzept"

78 kurz = "Betriebskonzept"

79 kategorie = "Betrieb"

80 untertitel = "Hochsitz mit Netz · Effinger"

81 beschreibung = "Nutzungsregeln, Aushang für die Aufstiege, Inspektionsplan mit Protokoll und das Erstabnahme-Protokoll für die Freigabe."

82 version = "3.0"

83 stand = 2026-09-02

84 status = "entwurf"

85 grundlage = "Tritt in Kraft mit Freigabe des Statik-Dossiers + Erstabnahme"

86 aenderungen = [

87 { version = "3.0", datum = 2026-09-02, was = "Versionsnummer an das Statik-Dossier angeglichen – ab jetzt tragen alle Dokumente dieselbe Nummer. Inhaltlich unverändert." },

88 { version = "2.0", datum = 2026-09-01, was = "Neuaufbau aus der Datenquelle; ersetzt Version 1.0 vom 30.08.2026. Erstabnahme-Protokoll an das Statik-Dossier angeglichen (Augplatten, Deckenvariante)." },

```

89 ]
90
91 # -----
92 # Geometrie (cm). Höhenbezug: OK Podest = 0. Achsen wie im 3D-Modell:
93 # x = Breite (links → rechts), y = Höhe, z = Tiefe (Rückwand → Front).
94 # -----
95 [geometrie]
96 kasten_links_tiefe_cm = 130      # Kasten links, x von 0 bis 130
97 luecke_cm             = 328      # freie Spannweite zwischen den Kästen
98 kasten_rechts_tiefe_cm = 70      # Kasten rechts
99 kasten_tiefe_cm       = 235      # Tiefe der Kästen (Rückwand → Front) = Querbalkenlänge
100 wand_cm               = 5        # Dreischichtplatten der Kästen (Wände und Deckplatte)
101 ok_deckplatte_cm      = 200      # Oberkante Deckplatte ab Podest
102 balken_cm             = 12       # Balkenquerschnitt 12 × 12
103 raum_hoehe_cm         = 350      # Decke ab Podest
104 podest_cm             = 5        # Holzpodest, auf dem die Kästen stehen
105 lattenwand_hoehe_cm   = 104      # Lattenwand hinten, ab Deckplatte
106 netz_seillage_cm      = 4        # Knüpfebene liegt so viel unter OK Rahmen
107 netz_durchhang_modell_cm = 16     # dargestellter Durchhang im 3D-Modell (unbelastet, dekorativ)
108 rohlaenge_balken_cm   = 400      # Ausgangslänge der Balken
109
110 [geometrie.kabine]
111 anzahl                = 2
112 breite_cm             = 100
113 tiefe_cm              = 100
114 hoehe_cm              = 217
115
116 # -----
117 # Material
118 # -----
119 [holz]
120 art                   = "Fichte/Tanne"
121 sortierung            = "C24"
122 f_mk                  = 24.0      # Biegung, charakteristisch
123 f_vk                  = 4.0       # Schub
124 f_c0k                 = 21.0      # Druck parallel
125 f_c90k                = 2.5       # Druck quer
126 rho_kn_m3             = 4.2      # Eigengewicht C24 (~420 kg/m³)
127 k_mod                 = 0.8       # mittlere Einwirkungsdauer (Nutzlast durch Personen), Nutzungsklasse 1/2 (im
Gebäude); bis v2.1 fälschlich 0.9 (kurz)
128 gamma_m               = 1.3      # Vollholz
129
130 [netz]
131 seil                   = "Paracord 550"
132 durchmesser_mm         = 4
133 ea_kn                  = 8        # Dehnsteifigkeit je Seil (Literatur-/Erfahrungswert, B3 kalibrieren)
134 n_wirksam              = 10      # wirksame Seile im Lastband
135 bruchlast_kn           = 2.4
136 zulaessig_kn           = 0.6     # abgemindert für Knoten und Dauerhaftigkeit
137 masche_cm              = 10
138 eigengewicht_kg        = 15
139
140 [netz.randseil]
141 name                   = "Tendon PRO WORK 11 mm"
142 norm                   = "EN 1891 A"
143 mbl_kn                 = 33
144
145 [augplatte]
146 name                   = "Augplatte 50×40 V2A (Decksplatte)"
147 kurz                   = "Augplatte 50×40"
148 werkstoff              = "Edelstahl V2A AISI 304, geschlossener Bügel"
149 schrauben              = "4× Linsenkopfschrauben 5×50"
150 bruch_kn               = 16.2
151 sicherheitsfaktor      = 3
152 pruefbericht           = "210528-1"
153 pruefbericht_quelle    = "Drahtseile24, 31.05.2021"
154 pruef_holz             = "Kiefer"
155
156 # -----

```

```

157 # Lasten und Bemessungsszenario
158 # -----
159 [lasten]
160 erwachsene_kg    = 85
161 jugendliche_kg   = 55
162 kind_kg          = 30
163 stossfaktor_kind = 2.5    # springende Kinder
164 gamma_q          = 1.5
165 lastband_m       = 1.2    # wirksame Lastbandbreite
166 knuepfmass_cm    = 0      # Durchhang ohne Last (Knüpfmass) im Bemessungsfall
167 ungleichverteilung = 1.3  # +30 % auf die Kraft je Randbefestigung
168 rahmenanteil_kn  = 0.3    # Anteil Rahmen-Eigengewicht je Auflagerecke
169 lichte_hoehe_min_cm = 170
170 probebelastung_kg = 80
171 probebelastung_soll_cm = [20, 25]
172
173 [lasten.szenario]
174 erwachsene = 1
175 jugendliche = 1
176 kinder     = 1
177
178 [ampel]                                # Farbstufen der Nachweise (Dossier-Tabelle, Explorer), Ausnutzung  $\eta$  =
Einwirkung / Widerstand
179 gelb_ab = 0.80                        # ab hier gelb: erfüllt, aber kleine Reserve
180 rot_ab  = 1.00                        # ab hier rot: Nachweis nicht erfüllt
181
182 [gelaender]
183 variante          = "decke"          # decke | pfosten
184 holmlast_kn_m     = 0.8              # SIA 358
185 einzellast_kn     = 1.5              # angesetzter Punktstoss
186 auslenkung_m      = 0.30             # seitliche Auslenkung des weichen Seils
187 auslenkung_weich_m = 0.50
188 seil_laenge_m     = 5.16
189 anker_y_cm        = 344              # bestehende Vorhangseil-Anker nahe Decke
190 mitte_y_cm        = 302             # Seil in Feldmitte
191 anker_anforderung_kn = 10
192 schlupf_breite_cm = 50
193 schlupf_hoehe_cm  = 75
194 leiter_erste_sprosse_cm = 55
195 pfosten_hoehe_cm  = 104
196 bruestung_en1176_cm = 70
197
198 # -----
199 # Befestigung
200 # -----
201 [befestigung]
202 netzrand_abstand_cm = 30            # MINDESTabstand der Augplatten am Netzrand; Reihen beginnen in den
Ecken (zwei Platten je Ecke, eine je Balkenseite), gleichmässig verteilt
203 gelaender_abstand_cm = 60          # Mindestabstand der Augplatten am Geländer (kleine Lasten;
Feinverteilung vor Ort)
204 kastenwand_zul_nmm2 = 2.0          # knickbegrenzt, konservativ
205
206 [befestigung.ecke]
207 s1 = "VG 8×160"                    # senkrecht durchs Blatt, in die Deckplatte
208 s2 = "VG 8×180"                    # geneigt, kreuzt die Blattfuge, durchläuft das Blatt des
Längsbalkens, endet in der Deckplatte
209 neigung_grad = 30                  # Neigung von S2 in Tiefenrichtung zum Kasteninneren
210 vorbohren_mm = 5
211 abstand_min_cm = 3.2                # 4d
212 kopf_versekt_cm = 0.0              # Köpfe bündig, NICHT versenken: die Längen sind auf die 5-cm-
Deckplatte abgestimmt (S1 endet 1 cm über deren Unterseite)
213 # Lage der Schraubenköpfe im 12×12-Blatt je Ecke: [x ab kastenseitiger Balkenkante, z ab Frontkante]
in cm
214 # S2 muss so weit vorne starten, dass sie unterhalb der Blattfuge noch im Blatt des Längsbalkens (z <
12) läuft: prüft r.verbindung.
215 # Kopf S2 bei 3.2 cm = 4d: näher an die Kante geht nicht ohne den Mindestrandabstand zu verletzen
(Hinweis Schreiner: möglichst weit vorne)
216 positionen = { s1 = [4.0, 8.5], s2 = [8.0, 3.2] }
217

```

```

218 [befestigung.querbalken]
219 schraube = "VG 8×160"
220 anzahl = 2 # zusätzliche Schrauben je Querbalken, gleichmässig zwischen den
Ecken
221
222 [befestigung.widerstand] # Richtwerte EC5, C24, vorgebohrt – von der Fachperson zu
bestätigen
223 s1_axial_kn = 1.9 # S1 axial: 8 mm, ~3.5 cm Gewinde in der 5-cm-Dreischichtplatte
(f_ax,k ≈ 10 N/mm²) – massgebend fürs Abheben
224 schraube_abscheren_kn = 2.0 # VG 8 mm, Abscheren in der Deckplatte (Verschieben des
Rahmens)
225 gelaender_anteil_rahmen = 0.5 # Anteil der Geländer-Linienlast, der über die untere
Plattenreihe in den Rahmen geht (konservativ)
226
227 # -----
228 # Quelltext-Anhang (Anhang B des Statik-Dossiers). Im Dossier steht nur die Prüfsummen-
229 # Tabelle; der vollständige Code erscheint auf /quelltext/ und als separates PDF.
230 # Pfade relativ zum Projektordner; Reihenfolge = Lesereihenfolge.
231 # -----
232 [quelltext]
233 dokument = "statik-dossier"
234 anhang = "B"
235 titel = "Rechenkern"
236 dateien = [
237   { pfad = "projekt.toml", titel = "Datenquelle – alle Eingabewerte (Masse,
Material, Lasten, Befestigung, Bedingungen)" },
238   { pfad = "src/statik/geometrie.ts", titel = "Abgeleitete Geometrie aus den Primärmassen" },
239   { pfad = "src/statik/rechenkern.ts", titel = "Rechenkern – Lasten, Seilmodell, Nachweise,
Verbindungsmittel, Mengen" },
240   { pfad = "../kit/src/lib/raster.ts", titel = "Rasterpositionen (Kit) – Schrauben- und
Augplattenreihen" },
241   { pfad = "src/statik/rechenkern.test.ts", titel = "Tests – Goldwerte und Grenzfälle" },
242 ]
243
244 # -----
245 # Bedingungen (Kapitel 9). Nummerierung stabil halten, nie stillschweigend ändern.
246 # -----
247 [[bedingungen]]
248 nr = "B1"
249 text = "Deckenanker: Tragfähigkeit der bestehenden Vorhangseil-Verankerung (geschweisste
Säulenklammer, Gewinde im Wandträger) für ≥ {p.gelaender.anker_anforderung_kn} kN je Ende verifizieren
(Probezug). Kein Bohren in den brandschutzummantelten Stahlträger ohne Freigabe von Eigentümer und
Brandschutz-Fachfirma. Fällt B1 negativ aus → Pfostenvariante (Kap. 8)"
250 verantwortung = "Projektteam + Fachperson"
251
252 [[bedingungen]]
253 nr = "B2"
254 text = "Anker Rückwand oben für das Geländer links (Verbundanker im Mauerwerk «nicht extrem stabil» –
Probezug); Eck-Augbolzen an Säule/Wand als durchgesteckte Maschinenschrauben ausführen"
255 verantwortung = "Projektteam"
256
257 [[bedingungen]]
258 nr = "B3"
259 text = "Probebelastung nach dem Knüpfen (1 Person mittig, Durchhang messen, Soll
{p.lasten.probebelastung_soll_cm.0}-{p.lasten.probebelastung_soll_cm.1} cm); Abweichung > 30 % →
Rückmeldung an Rechenkern"
260 verantwortung = "Projektteam + Fachperson"
261
262 [[bedingungen]]
263 nr = "B4"
264 text = "Fallschutz Kletterwand: keine losen Matten (Stolperfallen). Stattdessen: Tritt-Griffe so
setzen, dass die Füße max. ~1 m über Boden stehen; optional flache Fallschutzplatten (30–45 mm,
angeschrägte Ränder, kritische Fallhöhe ≥ 1 m) oder tiefe Polster-Sitzlandschaft als Möbel im Fallraum"
265 verantwortung = "Projektteam"
266
267 [[bedingungen]]
268 nr = "B5"
269 text = "Seitlicher Rand links durch Telefonkabinen (Dach +{g.kabine_ueberstand_cm} cm über
Deckplatte): als Absturzschutz-Ersatz durch Fachperson beurteilen; Kabinenposition sichern/markieren"

```

```

270 verantwortung = "Fachperson"
271
272 [[bedingungen]]
273 nr = "B6"
274 text = "Betriebskonzept: Nutzungsregeln, max. Personen, Inspektionsplan mit Protokoll,
verantwortliche Person"
275 verantwortung = "Projektteam"
276
277 [[bedingungen]]
278 nr = "B7"
279 text = "Holzqualität der vorhandenen Balken vor Einbau sichten (grosse Äste/Risse aussortieren;
Annahme {p.holz.sortierung})"
280 verantwortung = "Zimmerleute im Team"
281
282 # -----
283 # Material- und Einkaufsliste (Anhang A). menge_aus verweist auf r.mengen.<key>.
284 # -----
285 [[material]]
286 pos = "VG-Schraube 8x160, Torx, vorbohren Ø 5"
287 menge_aus = "schrauben_8x160"
288 bemerkung = "{r.mengen.ecken}× Ecken (S1) + {r.mengen.querbalken_schrauben}× Querbalken; Reserve
dazu"
289 gruppe = "befestigung"
290
291 [[material]]
292 pos = "VG-Schraube 8x180, Torx"
293 menge_aus = "schrauben_8x180"
294 bemerkung = "{r.mengen.ecken}× Ecken (S2, {p.befestigung.ecke.neigung_grad}° geneigt); Reserve dazu"
295 gruppe = "befestigung"
296
297 [[material]]
298 pos = "Augplatte 50x40 V2A (Decksplatte) inkl. 4× Linsenkopf 5x50"
299 menge_aus = "augplatten"
300 bemerkung = "Netzrand {r.mengen.augplatten_netzrand} (zwei je Ecke, Abstand
{r.mengen.rand_abstand_x_cm|z1} längs / {r.mengen.rand_abstand_z_cm|z1} quer) + Geländer
{r.mengen.augplatten_gelaender} ({r.mengen.gelaender_abstand_front_cm|z0} cm); Reserve dazu"
301 gruppe = "befestigung"
302
303 [[material]]
304 pos = "Wantenspanner M12 (Auge-Auge)"
305 menge = 2
306 bemerkung = "Nachspannen der Geländerseile"
307 gruppe = "befestigung"
308
309 [[material]]
310 pos = "Augbolzen M10 durchgesteckt / Verbundanker"
311 menge = 3
312 bemerkung = "Ecklösungen Säule/Wand, Anker Rückwand oben (B2)"
313 gruppe = "befestigung"
314
315 [[material]]
316 pos = "Augbolzen M12 durchgesteckt, grosse U-Scheiben"
317 menge = 4
318 bemerkung = "Seilenden an Pfosten (nur Pfostenvariante)"
319 gruppe = "variante"
320
321 [[material]]
322 pos = "Trägerklammer/Balkenklemme (Säule H-Profil)"
323 menge = 2
324 bemerkung = "Pfosten links an Stahlsäule, ohne Bohren (nur Pfostenvariante)"
325 gruppe = "variante"
326
327 [[material]]
328 pos = "Verbundanker M10 mit Siebhülse"
329 menge = 2
330 bemerkung = "Wand rechts (oder Verschraubung in Kastenwand)"
331 gruppe = "befestigung"
332
333 [[material]]

```

```
334 pos = "Holzbohrer Ø 5, Torx-Bits T40, Zollstock, Wasserwaage"
335 einheit = ""
336 bemerkung = "Werkzeug Workshop"
337 gruppe = "werkzeug"
338
339 [[material]]
340 pos = "Fallschutzplatten 30-45 mm, angeschrägt (optional, B4)"
341 menge = 2.5
342 einheit = "m²"
343 bemerkung = "flach begehbar statt loser Matten; Alternative: Polster-Sitzlandschaft"
344 gruppe = "optional"
```


B.2 src/statik/geometrie.ts

Abgeleitete Geometrie aus den Primärmassen · 58 Zeilen · SHA-256

85ff6ffdd1334a2b8bd31bcbd7e7a83a5a88cc896a33f0dff976608c79480428c

```

1  /**
2   * Abgeleitete Geometrie – alles, was aus den Primärmassen in projekt.toml folgt.
3   * Wird von Rechenkern, 3D-Szene, Zeichnungen und Dokumenten gemeinsam benutzt.
4   * Koordinaten: x Breite (0 = linke Aussenkante Kasten links), y Höhe (0 = OK Podest), z Tiefe (0 =
Rückwand).
5   */
6   import type { Hochsitz } from '../schema';
7
8   export function ableiten(p: Hochsitz) {
9     const G = p.geometrie;
10    const b = G.balken_cm;
11    const x_l0 = 0;
12    const x_l1 = G.kasten_links_tiefe_cm;
13    const x_r0 = x_l1 + G.luecke_cm;
14    const x_r1 = x_r0 + G.kasten_rechts_tiefe_cm;
15    const rahmen_x0 = x_l1 - b;
16    const rahmen_x1 = x_r0 + b;
17    const ok_rahmen_cm = G.ok_deckplatte_cm + b;
18    const netzfeld_x_cm = G.luecke_cm;
19    const netzfeld_z_cm = G.kasten_tiefe_cm - 2 * b;
20    return {
21      b,
22      x_l0, x_l1, x_r0, x_r1, rahmen_x0, rahmen_x1,
23      breite_total_cm: x_r1,
24      luecke_cm: G.luecke_cm,
25      tiefe_cm: G.kasten_tiefe_cm,
26      wand_cm: G.wand_cm,
27      rahmen_x_cm: G.luecke_cm + 2 * b,
28      rahmen_z_cm: G.kasten_tiefe_cm,
29      netzfeld_x_cm,
30      netzfeld_z_cm,
31      netz_m2: (netzfeld_x_cm * netzfeld_z_cm) / 1e4,
32      ok_deckplatte_cm: G.ok_deckplatte_cm,
33      ok_rahmen_cm,
34      knuepfebene_cm: ok_rahmen_cm - G.netz_seillage_cm,
35      kopffreiheit_cm: G.raum_hoehe_cm - ok_rahmen_cm,
36      raum_hoehe_cm: G.raum_hoehe_cm,
37      kastenwand_hoehe_cm: G.ok_deckplatte_cm - G.wand_cm,
38      lattenwand_ok_cm: G.ok_deckplatte_cm + G.lattenwand_hoehe_cm,
39      auflage_cm: b,
40      blatt_cm: b / 2,
41      // Kabinen stehen neben dem Podest auf dem Rohboden (podest_cm unter OK Podest); Dachhöhe über OK
Podest = hoehe - podest
42      kabine_ok_cm: G.kabine.hoehe_cm - G.podest_cm,
43      kabine_ueberstand_cm: G.kabine.hoehe_cm - G.podest_cm - G.ok_deckplatte_cm,
44      balken_rest_laengs_cm: G.rohlaenge_balken_cm - (G.luecke_cm + 2 * b),
45      balken_rest_quer_cm: G.rohlaenge_balken_cm - G.kasten_tiefe_cm,
46      verlaengerung_links_cm: rahmen_x0,
47      verlaengerung_rechts_cm: x_r1 - rahmen_x1,
48      // Querschnittswerte (mm)
49      W_mm3: (b * b * b) / 6 * 1e3,
50      A_mm2: b * b * 1e2,
51      auflager_linie_mm2: b * G.wand_cm * 1e2,
52      blatt_schulter_mm2: b * (b / 2) * 1e2,
53      gelaender_seil_mitte_ueber_rahmen_cm: p.gelaender.mitte_y_cm - ok_rahmen_cm,
54      restspalt_decke_cm: G.raum_hoehe_cm - p.gelaender.mitte_y_cm,
55      lichte_hoehe_unter_seil_cm: p.gelaender.mitte_y_cm,
56    };
57  }
58  export type Geometrie = ReturnType<typeof ableiten>;

```

B.3 src/statik/rechenkern.ts

Rechenkern — Lasten, Seilmodell, Nachweise, Verbindungsmittel, Mengen · 317 Zeilen · SHA-256

19e0f3cdf03da4cb97e03cd988a388e207342b269a7e37e946ff3f69ecc23587

```

1  /**
2   * Rechenkern Statik «Hochsitz mit Netz» – die einzige Implementierung.
3   * Läuft im Build (Dossier-Tabellen, Nachweise, Stückliste) und im Browser (Statik-Explorer)
4   * und wird im Dossier-Anhang B wörtlich abgedruckt.
5   *
6   * Basis: SN EN 1995-1-1 (EC5) / SIA 260/261/265-Logik, SIA 358 für das Gelände.
7   * Einheiten: Längen m (Geometrie kommt in cm und wird umgerechnet), Kräfte kN, Spannungen N/mm².
8   */
9   import type { Hochsitz } from '../schema';
10  import type { Geometrie } from '../geometrie';
11  import { laengeAusBezeichnung, rasterMitEcken } from '@yolu/bauprojekt-kit/lib/raster';
12
13  export interface Szenario {
14    erwachsene: number;
15    jugendliche: number;
16    kinder: number;
17    /** Knüpfmass = Durchhang ohne Last, cm */
18    knuepfmass_cm: number;
19    /** MINDESTabstand der Augplatten am Netzrand, cm (Reihen beginnen in den Ecken, gleichmässig) */
20    randabstand_cm: number;
21  }
22
23  /** Bemessungswerte des Holzes nach EC5:  $f_d = f_k \cdot k_{mod} / \gamma_M$  */
24  export function holzwerte(p: Hochsitz) {
25    const k = p.holz.k_mod / p.holz.gamma_m;
26    return { f_md: p.holz.f_mk * k, f_vd: p.holz.f_vk * k, f_c0d: p.holz.f_c0k * k, f_c90d:
p.holz.f_c90k * k };
27  }
28
29  /** Effektiver Plattenabstand bei Mindestabstand 'min' auf der Länge 'L' (Platten in beiden Ecken).
*/
30  export function abstandEffektiv(L: number, min: number): number {
31    const pos = rasterMitEcken(L, min);
32    return pos.length > 1 ? pos[1] - pos[0] : L;
33  }
34
35  /**
36   * Gekoppeltes Seilmodell mit Dehnung: beide Seilrichtungen (quer über Lx, längs über Lz) teilen sich
37   * EINE Mittendurchbiegung f. Je Richtung: Seillänge  $s(f) = 2 \cdot \sqrt{((L/2)^2 + f^2)}$ , Dehnkraft
38   *  $T = EA \cdot (s/s_0 - 1)$ , Horizontalzug  $H = T/\sqrt{1 + (2f/L)^2}$ , getragene Last  $P = 4 \cdot H \cdot f/L$ .
39   * Gesucht ist f mit  $P_{quer}(f) + P_{längs}(f) = P$  (Bisektion).
40   */
41  export function seilmodell(P: number, f0: number, Lx: number, Lz: number, EA: number, beta = 0.5) {
42    const EAx = EA * 2 * beta;
43    const EAz = EA * 2 * (1 - beta);
44    const s0z = 2 * Math.hypot(Lz / 2, f0);
45    const s0x = 2 * Math.hypot(Lx / 2, f0);
46    const stuetz = (f: number) => {
47      const sz = 2 * Math.hypot(Lz / 2, f);
48      const sx = 2 * Math.hypot(Lx / 2, f);
49      const Tz = Math.max(0, sz / s0z - 1) * EAz;
50      const Tx = Math.max(0, sx / s0x - 1) * EAx;
51      const Hz = Tz / Math.hypot(1, (2 * f) / Lz);
52      const Hx = Tx / Math.hypot(1, (2 * f) / Lx);
53      return { Pz: (4 * Hz * f) / Lz, Px: (4 * Hx * f) / Lx, Hz, Hx, Tz, Tx };
54    };
55    if (P <= 0) return { f: f0, ...stuetz(f0), Pz: 0, Px: 0, Hz: 0, Hx: 0, Tz: 0, Tx: 0 };
56    let lo = f0 + 1e-4;
57    let hi = 1.6;
58    let f = hi;
59    for (let i = 0; i < 80; i++) {

```

```

60     f = (lo + hi) / 2;
61     const r = stuetz(f);
62     if (r.Pz + r.Px < P) lo = f;
63     else hi = f;
64 }
65 return { f, ...stuetz(f) };
66 }
67
68 /** Ein Lastszenario vollständig durchrechnen. */
69 export function bemessung(p: Hochsitz, g: Geometrie, sz: Szenario) {
70     const L = p.lasten;
71     const Lx = g.netzfeld_x_cm / 100; // quer: Spannweite der Lücke, zieht an den Querbalken
72     const Lz = g.netzfeld_z_cm / 100; // längs: zwischen den Längsbalken, zieht an den Längsbalken
73     const EA = p.netz.ea_kn * p.netz.n_wirksam;
74     const w = L.lastband_m;
75     const hw = holzwerte(p);
76
77     // Lasten: Personen mittig konzentriert, Kinder springend mit Stossfaktor φ
78     const Qk = (L.erwachsene_kg * sz.erwachsene + L.jugendliche_kg * sz.jugendliche) / 100 + (L.kind_kg
/ 100) * sz.kinder * L.stossfaktor_kind;
79     const Pd = L.gamma_q * Qk;
80     const kg = L.erwachsene_kg * sz.erwachsene + L.jugendliche_kg * sz.jugendliche + L.kind_kg *
sz.kinder;
81
82     const s = seilmodell(Pd, sz.knuepfmass_cm / 100, Lx, Lz, EA);
83     const Hz = s.Hz; // Horizontalzug auf je einen Längsbalken (gesamt)
84     const Hx = s.Hx; // Horizontalzug auf je einen Querbalken (voll aufliegend)
85
86     // Längsbalken: zweiachsige Biegung – Horizontalzug als Bandlast mittig, Vertikalanteil als
Punktlast
87     const Mh = Hz * (Lx / 4 - w / 8);
88     const Vz = s.Pz / 2;
89     const Mv = (Vz * Lx) / 4;
90     const sigma = ((Mh + Mv) * 1e6) / g.W_mm3;
91     const eta = sigma / hw.f_md;
92
93     // Ecken / Auflager: der Ringdruck C geht formschlüssig über die Blattschulter (12 × 6 cm)
94     const C = Hz / 2; // Ringdruck in die Überblattungs-Schulter je Ecke (= Schub am Auflager)
95     const V = Hz / 2;
96     const Rv = Vz + L.rahmenanteil_kn; // Auflagerkraft vertikal je Ecke
97     const sigma_c90 = (Rv * 1e3) / g.A_mm2;
98     const sigma_wand = (Rv * 1e3) / g.auflager_linie_mm2;
99     const sigma_schulter = (C * 1e3) / g.blatt_schulter_mm2;
100
101     // Randbefestigung (Augplatten): effektiver Abstand bei Mindestabstand sz.randabstand_cm auf der
längeren Seite
102     const rand_abstand_eff_cm = Math.max(abstandEffektiv(g.netzfeld_x_cm, sz.randabstand_cm),
abstandEffektiv(g.netzfeld_z_cm, sz.randabstand_cm));
103     const Fkn = (Math.max(Hz, Hx) / w) * (rand_abstand_eff_cm / 100) * L.ungleichverteilung;
104     const Tstrang = Math.max(s.Tz, s.Tx) / p.netz.n_wirksam;
105     const f_zul_platte = p.augplatte.bruch_kn / p.augplatte.sicherheitsfaktor;
106
107     return {
108         sz, kg, Qk, Pd, f_cm: s.f * 100, Hz, Hx, Hz_line: Hz / w, Hx_line: Hx / w, Pz: s.Pz, Px: s.Px,
109         Mh, Mv, Vz, sigma, eta, C, V, Rv, sigma_c90, sigma_wand, sigma_schulter,
110         rand_abstand_eff_cm, Fkn, Tstrang, f_zul_platte, eta_platte: Fkn / f_zul_platte, eta_strang:
Tstrang / p.netz.zulaessig_kn,
111         licht_cm: g.knuepfebene_cm - s.f * 100,
112         ...hw,
113     };
114 }
115 export type Bemessung = ReturnType<typeof bemessung>;
116
117 /** Geländerseil, Deckenvariante: frei hängendes Seil, seitliche Auslenkung unter Holmlast (SIA 358).
*/
118 export function gelaender(p: Hochsitz) {
119     const G = p.gelaender;
120     const qd = G.holmlast_kn_m * p.lasten.gamma_q;
121     return {

```

```

122     qd,
123     T_punkt: (G.einzellast_kn * G.seil_laenge_m) / (4 * G.auslenkung_m),
124     T_linie: (qd * G.seil_laenge_m ** 2) / (8 * G.auslenkung_m),
125     T_linie_weich: (qd * G.seil_laenge_m ** 2) / (8 * G.auslenkung_weich_m),
126     /** horizontaler Stoss, der konservativ über die untere Plattenreihe in den Rahmen geht */
127     F_rahmen: p.befestigung.widerstand.gelaender_anteil_rahmen * qd * G.seil_laenge_m,
128 };
129 }
130
131 /**
132  * Verbindungsmittel an der Ecke – Geometrie nachgerechnet, damit Text, Zeichnung und 3D nicht
133  * auseinanderlaufen.
134  * Koordinaten wie im Schraubenplan: x ab kastenseitiger Kante des Querbalkens, z ab
135  * Front-/Rückkante, y ab OK Podest.
136  * Das Blatt (Überblattung) ist b x b gross; Blattfuge auf OK Rahmen - b/2. Unter der Fuge liegt das
137  * Blatt des
138  * Längsbalkens nur für z < b, dahinter (z > b) der untere Teil des Querbalkens (Schulter bei z = b).
139  * S1: senkrecht durchs Blatt in die Deckplatte (Fugensicherung, Abheben, Dübel gegen Verschieben).
140  * S2: um 'neigung_grad' in Tiefenrichtung zum Kasteninneren geneigt – sie muss die Blattfuge
141  * kreuzen, danach bis zur
142  * Deckplatte im Blatt des Längsbalkens bleiben (z < b) und in der Deckplatte enden, damit im Kasten
143  * keine Spitze austritt.
144  */
145 export function verbindung(p: Hochsitz, g: Geometrie) {
146     const E = p.befestigung.ecke;
147     const b = g.b;
148     const d_cm = 0.8; // Schraubendurchmesser 8 mm
149     const s1_laenge_cm = laengeAusBezeichnung(E.s1, 16);
150     const s2_laenge_cm = laengeAusBezeichnung(E.s2, 18);
151     const neig = (E.neigung_grad * Math.PI) / 180;
152     const kopf_y = g.ok_rahmen_cm - E.kopf_versenkt_cm;
153     const fuge_y = g.ok_rahmen_cm - b / 2;
154     const ok_deck = g.ok_deckplatte_cm;
155     const uk_deck = ok_deck - g.wand_cm;
156     // S1 senkrecht
157     // Schraubenlänge gilt ab Kopfoberkante (Senkkopf); Kopf sitzt auf kopf_y
158     const s1_spitze_y_cm = kopf_y - s1_laenge_cm;
159     const s1_in_deck_cm = ok_deck - s1_spitze_y_cm;
160     // S2 geneigt: z-Lage an Fuge, Deckplatte und Spitze
161     const frei = s2_laenge_cm;
162     const zBei = (y: number) => E.positionen.s2[1] + (kopf_y - y) * Math.tan(neig);
163     const s2_fuge_z_cm = zBei(fuge_y);
164     const s2_deck_z_cm = zBei(ok_deck);
165     const s2_spitze_y_cm = kopf_y - frei * Math.cos(neig);
166     const s2_spitze_z_cm = zBei(s2_spitze_y_cm);
167     const s2_in_quer_cm = (kopf_y - fuge_y) / Math.cos(neig);
168     const s2_in_laengs_cm = (fuge_y - ok_deck) / Math.cos(neig);
169     const s2_in_deck_cm = (ok_deck - s2_spitze_y_cm) / Math.cos(neig);
170     const rand = 4 * d_cm; // Mindestabstand Kopf zu Kanten 4d
171     const [s1x, s1z] = E.positionen.s1;
172     const [s2x, s2z] = E.positionen.s2;
173     const imBlatt = (x: number, z: number) => x >= rand && x <= b - rand && z >= rand && z <= b - rand;
174     return {
175         s1_laenge_cm, s2_laenge_cm, kopf_y, fuge_y, ok_deck, uk_deck,
176         s1_spitze_y_cm, s1_in_deck_cm, s1_in_deckplatte: s1_spitze_y_cm > uk_deck + 0.5 && s1_spitze_y_cm
177         < ok_deck,
178         s2_fuge_z_cm, s2_deck_z_cm, s2_spitze_y_cm, s2_spitze_z_cm, s2_in_quer_cm, s2_in_laengs_cm,
179         s2_in_deck_cm,
180         /** kreuzt die Fuge im Blatt und bleibt bis zur Deckplatte im Blatt des Längsbalkens */
181         s2_im_blatt: s2_fuge_z_cm > rand && s2_deck_z_cm < b - 0.5,
182         s2_in_deckplatte: s2_spitze_y_cm > uk_deck + 0.5 && s2_spitze_y_cm < ok_deck,
183         s2_reserve_cm: s2_spitze_y_cm - uk_deck,
184         kopfabstand_cm: Math.hypot(s1x - s2x, s1z - s2z),
185         randabstaende_ok: imBlatt(s1x, s1z) && imBlatt(s2x, s2z) && Math.hypot(s1x - s2x, s1z - s2z) >=
186         rand,
187         /** EC5-Mindestabstände für axial beanspruchte Schrauben (Tab. 8.6: a1,CG = 10d) sind im 12-cm-
188         Blatt nicht einhaltbar – Fachperson */
189         ec5_endabstand_10d_cm: 10 * d_cm,
190         ec5_endabstand_ok: Math.min(s1z, s2z, s1x, s2x) >= 10 * d_cm,
191     };
192 }

```

```

182   };
183 }
184
185 /**
186  * Abheben des Längsbalken-Endes: Der Netzzug wirkt auf Plattenhöhe (b/2 über der Deckplatte) und
will den Balken
187  * um seine innere Unterkante kippen. Stabilisierend wirkt das Eigengewicht; der Rest muss an den
zwei Enden
188  * durch S1 (und den aufliegenden Querbalken) gehalten werden. Rückgabe: Haltekraft je Ecke in kN.
189  */
190 export function abheben(p: Hochsitz, g: Geometrie, Hz: number) {
191   const bm = g.b / 100;
192   const G_balken = p.holz.rho_kn_m3 * bm * bm * (g.rahmen_x_cm / 100);
193   const M_kipp = Hz * (bm / 2);
194   const M_stab = G_balken * (bm / 2);
195   return { G_balken, M_kipp, M_stab, F_ecke: Math.max(0, (M_kipp - M_stab) / bm / 2) };
196 }
197
198 /** Mengen für Stückliste, 3D-Modell und Zeichnungen – aus denselben Rasterfunktionen. */
199 export function mengen(p: Hochsitz, g: Geometrie) {
200   const B = p.befestigung;
201   const ecken = 4;
202   // Querbalken: `anzahl` zusätzliche Schrauben je Balken, gleichmässig zwischen den Ecken
203   const nQ = B.querbalken.anzahl;
204   const querbalkenPositionen = Array.from({ length: nQ }, (_, i) => +((g.tiefe_cm * (i + 1)) / (nQ +
191)).toFixed(1));
205   // Augplatten Netzrand: alle vier Reihen beginnen in den Ecken → zwei Platten je Ecke (eine je
Balkenseite),
206   // das Randseil wird dort in zwei Mal 45° umgelenkt und die Umlenkkraft auf zwei Platten verteilt
207   const randXalle = rasterMitEcken(g.netzfeld_x_cm, B.netzrand_abstand_cm);
208   const randZalle = rasterMitEcken(g.netzfeld_z_cm, B.netzrand_abstand_cm);
209   const randZ = randZalle;
210   // Geländer: Frontreihe mit beiden Enden (Säule, Wand); linke Reihe ohne das vordere Ende
(Eckplatte der Frontreihe)
211   const gelaenderFront = rasterMitEcken(g.breite_total_cm, B.gelaender_abstand_cm);
212   const gelaenderLinks = rasterMitEcken(g.tiefe_cm, B.gelaender_abstand_cm).slice(0, -1);
213   const augplatten_netzrand = 2 * randXalle.length + 2 * randZ.length;
214   const augplatten_gelaender = gelaenderFront.length + gelaenderLinks.length;
215   const augplatten = augplatten_netzrand + augplatten_gelaender;
216   return {
217     ecken,
218     querbalken_positionen_cm: querbalkenPositionen,
219     querbalken_schrauben: 2 * nQ,
220     schrauben_8x160: ecken + 2 * nQ,
221     schrauben_8x180: ecken,
222     schrauben_deckplatte: ecken + 2 * nQ, // senkrecht in die Deckplatte (S1 + Querbalken) – gegen
Verschieben
223     rand_x_cm: randXalle,
224     rand_z_cm: randZ,
225     rand_abstand_x_cm: randXalle[1] - randXalle[0],
226     rand_abstand_z_cm: randZalle[1] - randZalle[0],
227     gelaender_front_cm: gelaenderFront,
228     gelaender_links_cm: gelaenderLinks,
229     gelaender_abstand_front_cm: gelaenderFront[1] - gelaenderFront[0],
230     gelaender_abstand_links_cm: gelaenderLinks.length > 1 ? gelaenderLinks[1] - gelaenderLinks[0] :
g.tiefe_cm,
231     augplatten_netzrand,
232     augplatten_gelaender,
233     augplatten,
234     augplatten_je_ecke: 2,
235   };
236 }
237
238 export interface Nachweis {
239   id: string;
240   titel: string;
241   einwirkung: string;
242   widerstand: string;
243   eta: number | null;

```

```

244   status: 'ok' | 'warn' | 'crit' | 'bedingung';
245   hinweis?: string;
246 }
247
248 const z = (v: number, d = 2) => (v + 1e-9).toFixed(d); // Epsilon gegen Binär-Abrundung (3.2249999... →
3.23)
249 /** Ampel eines Nachweises aus der Ausnutzung; Schwellen aus projekt.toml [ampel] (auch im Explorer).
*/
250 export const ampelStatus = (eta: number, a: Hochsitz['ampel']): 'ok' | 'warn' | 'crit' =>
251   eta < a.gelb_ab ? 'ok' : eta <= a.rot_ab ? 'warn' : 'crit';
252
253 /** Alle Resultate für Dossier, Startseite und Stückliste: Standardszenario, Varianten, Nachweise. */
254 export function rechne(p: Hochsitz, g: Geometrie) {
255   const L = p.lasten;
256   const std: Szenario = { ...L.szenario, knuepfmass_cm: L.knuepfmass_cm, randabstand_cm:
p.befestigung.netzrand_abstand_cm };
257   const b = bemessung(p, g, std);
258   const varianten = [0, 5, 10].map((f0) => bemessung(p, g, { ...std, knuepfmass_cm: f0 }));
259   // Gebrauchslage: 1 Person, charakteristisch (γ = 1), ohne Stossfaktor
260   const gebrauch = (() => {
261     const s = seilmodell(L.erwachsene_kg / 100, 0, g.netzfeld_x_cm / 100, g.netzfeld_z_cm / 100,
p.netz.ea_kn * p.netz.n_wirksam);
262     return { P: L.erwachsene_kg / 100, f_cm: s.f * 100, Hz: s.Hz, licht_cm: g.knuepfebene_cm - s.f *
100 };
263   })();
264   const gl = gelaender(p);
265   const m = mengen(p, g);
266   const vb = verbindung(p, g);
267   const ab = abheben(p, g, b.Hz);
268   const W = p.befestigung.widerstand;
269   const R_verschieben = m.schrauben_deckplatte * W.schraube_abscheren_kn;
270   const ampel = (eta: number) => ampelStatus(eta, p.ampel);
271
272   const nachweise: Nachweis[] = [
273     { id: 'biegung', titel: 'Längsbalken, Biegung zweiachsig',
274       einwirkung: `σ = ${z(b.sigma, 1)} N/mm² (M<sub>h</sub> ${z(b.Mh)} + M<sub>v</sub> ${z(b.Mv)}
kNm)`,
275       widerstand: `f<sub>md</sub> = ${z(b.f_md, 1)} N/mm²`, eta: b.eta, status: ampel(b.eta) },
276     { id: 'auflager', titel: 'Auflagerpressung (${g.b}×${g.b} cm)`,
277       einwirkung: `σ<sub>c,90</sub> = ${z(b.sigma_c90)} N/mm²`, widerstand: `f<sub>c,90</sub> =
${z(b.f_c90d, 1)} N/mm²`,
278       eta: b.sigma_c90 / b.f_c90d, status: ampel(b.sigma_c90 / b.f_c90d) },
279     { id: 'kastenwand', titel: 'Kastenwand ${g.wand_cm} cm unter Auflagerrecke`,
280       einwirkung: `${z(b.sigma_wand)} N/mm² (R<sub>v</sub> ≈ ${z(b.Rv, 1)} kN, Linienlager ${g.b} ×
${g.wand_cm} cm)`,
281       widerstand: `~${z(p.befestigung.kastenwand_zul_nmm2, 0)} N/mm² (knickbegrenzt, H
${g.kastenwand_hoehe_cm} cm)`,
282       eta: b.sigma_wand / p.befestigung.kastenwand_zul_nmm2, status: ampel(b.sigma_wand /
p.befestigung.kastenwand_zul_nmm2) },
283     { id: 'schulter', titel: 'Ringdruck / Schub an der Blattschulter (formschlüssig)',
284       einwirkung: `C = V = ${z(b.C)} kN je Ecke auf ${g.blatt_schulter_mm2 / 100} cm² →
${z(b.sigma_schulter)} N/mm²`,
285       widerstand: `f<sub>c,0</sub> = ${z(b.f_c0d, 1)} N/mm²; S2 (${p.befestigung.ecke.s2},
${p.befestigung.ecke.neigung_grad}°) als Redundanz`,
286       eta: b.sigma_schulter / b.f_c0d, status: 'ok' },
287     { id: 'verschieben', titel: 'Rahmen auf Deckplatte, Verschieben (Geländer-Stoss über untere
Plattenreihe)',
288       einwirkung: `F ≈ ${z(gl.F_rahmen, 1)} kN (${z(W.gelaender_anteil_rahmen * 100, 0)} % der Holm-
Linienlast ${z(gl.qd, 1)} kN/m × ${z(p.gelaender.seil_laenge_m)} m)`,
289       widerstand: `${m.schrauben_deckplatte} Schrauben senkrecht in die Deckplatte, Abscheren
R<sub>d</sub> ≈ ${m.schrauben_deckplatte} × ${z(W.schraube_abscheren_kn, 1)} = ${z(R_verschieben, 0)} kN,
plus Reibung`,
290       eta: gl.F_rahmen / R_verschieben, status: ampel(gl.F_rahmen / R_verschieben) },
291     { id: 'abheben', titel: 'Abheben / Kippen des Längsbalken-Endes',
292       einwirkung: `F = ${z(ab.F_ecke)} kN je Ecke (Netzzug ${z(b.Hz)} kN auf Plattenhöhe ${g.b / 2}
cm, Kippen um die innere Unterkante, abzüglich Eigengewicht ${z(ab.G_balken)} kN)`,
293       widerstand: `S1 axial in der ${g.wand_cm}-cm-Deckplatte (${z(vb.s1_in_deck_cm, 1)} cm Gewinde):
R<sub>d</sub> ≈ ${z(W.s1_axial_kn, 1)} kN, dazu Auflast des Querbalkens`,
294       eta: ab.F_ecke / W.s1_axial_kn, status: ampel(ab.F_ecke / W.s1_axial_kn) },

```

```

295     { id: 'randbefestigung', titel: `Netz-Randbefestigung (${p.augplatte.kurz}, e =
${z(b.rand_abstand_eff_cm, 1)} cm)`,
296       einwirkung: `F = ${z(b.Fkn)} kN je Punkt`,
297       widerstand: `zulässig ${z(b.f_zul_platte, 1)} kN = Bruch ${z(p.augplatte.bruch_kn, 1)} kN /
${p.augplatte.sicherheitsfaktor} (Prüfbericht ${p.augplatte.pruefbericht})`,
298       eta: b.eta_platte, status: ampel(b.eta_platte) },
299     { id: 'strang', titel: `${p.netz.seil}-Strang im Lastband`,
300       einwirkung: `T = ${z(b.Tstrang)} kN`, widerstand: `zulässig ≈ ${z(p.netz.zulaessig_kn, 1)} kN
(inkl. Knoten)`,
301       eta: b.eta_strang, status: ampel(b.eta_strang) },
302     { id: 'deckenanker', titel: 'Geländer-Deckenanker (Hängeseil)',
303       einwirkung: `Punktstoss ≈ ${z(gl.T_punkt, 1)} kN · Linienlast konservativ ≈ ${z(gl.T_linie, 0)}
kN`,
304       widerstand: `bestehende Vorhangseil-Anker – Anforderung ≥ ${p.gelaender.anker_anforderung_kn}
kN je Ende`,
305       eta: null, status: 'bedingung', hinweis: 'B1' },
306   ];
307
308   return {
309     std: b, varianten, gebrauch, gelaender: gl, mengen: m, verbindung: vb, abheben: ab, nachweise,
310     // erfüllt = Ausnutzung ≤ rot_ab (grün oder gelb); gelb heisst «erfüllt mit kleiner Reserve»,
nicht «nicht erfüllt»
311     alleErfuehlt: nachweise.every((n) => n.status !== 'crit') && vb.s2_in_deckplatte &&
vb.s1_in_deckplatte && vb.s2_im_blaett && vb.randabstaende_ok,
312     max_personen: L.szenario.erwachsene + L.szenario.jugendliche + L.szenario.kinder,
313     holz: holzwerte(p),
314     EA: p.netz.ea_kn * p.netz.n_wirksam,
315   };
316 }
317 export type Resultate = ReturnType<typeof rechne>;

```

B.4 kit/src/lib/raster.ts

Rasterpositionen (Kit) — Schrauben- und Augplattenreihen · 29 Zeilen · SHA-256

86083242cdb1a8f0899e4f24d035a0b76766259d3624577bff3f11ad3c1f97fa

```

1  /**
2   * Gleichmässige Positionen entlang einer Länge: Sollabstand `abstand` bleibt exakt erhalten,
3   * die Gruppe wird zentriert, an beiden Enden bleibt mindestens `rand` frei.
4   * Wird für Schraubenreihen, Augplatten-Raster und Stücklisten genutzt, damit 3D-Modell,
5   * Zeichnung und Mengen dieselben Punkte haben («alle 50 cm» heisst dann auch 50 cm).
6   */
7  export function raster(laenge: number, abstand: number, rand: number): number[] {
8      const nutz = laenge - 2 * rand;
9      if (nutz <= 0 || abstand <= 0) return [laenge / 2];
10     const n = Math.floor(nutz / abstand + 1e-9) + 1;
11     const start = (laenge - (n - 1) * abstand) / 2;
12     return Array.from({ length: n }, (_, i) => +(start + i * abstand).toFixed(2));
13 }
14
15 /**
16  * Positionen, die in BEIDEN Ecken beginnen und den Mindestabstand einhalten: n = [L/min] + 1 Punkte,
17  * gleichmässig verteilt (effektiver Abstand  $L/(n-1) \geq \text{min}$ ). Für Augplatten-Reihen «ganz in den Ecken
18  * beginnend».
19  */
20 export function rasterMitEcken(laenge: number, minAbstand: number): number[] {
21     if (laenge <= 0) return [0];
22     const n = Math.max(2, Math.floor(laenge / minAbstand + 1e-9) + 1);
23     return Array.from({ length: n }, (_, i) => +((i * laenge) / (n - 1)).toFixed(2));
24 }
25
26 /** Schraubenlänge aus der Bezeichnung «VG 8×180» in cm (Fallback, wenn nicht lesbar). */
27 export function laengeAusBezeichnung(bezeichnung: string, fallback: number): number {
28     const m = /[×]\s*(\d+)/.exec(bezeichnung);
29     return m ? Number(m[1]) / 10 : fallback;
30 }

```


B.5 src/statik/rechenkern.test.ts

Tests — Goldwerte und Grenzfälle · 188 Zeilen · SHA-256

307f55aba0b05eb20bd241ec050dd6e54feea25d79831d92d85e229ecfa85cf0

```

1  /**
2   * Goldene Werte aus dem geprüften Dossier v1.3 (31.08.2026) und Grenzfälle; Holz-Ausnutzungen seit
v3.0 mit k_mod = 0.8.
3   * Ändert sich der Rechenkern absichtlich, müssen diese Werte bewusst angepasst werden.
4   */
5   import { describe, expect, it } from 'vitest';
6   import { readFileSync } from 'node:fs';
7   import { parseProjekt } from '@yolu/bauprojekt-kit/daten';
8   import { rasterMitEcken } from '@yolu/bauprojekt-kit/lib/raster';
9   import { HochsitzSchema } from '../schema';
10  import { ableiten } from './geometrie';
11  import { bemessung, rechne, seilmodell } from './rechenkern';
12
13  const p = parseProjekt(readFileSync(new URL('../..projekt.toml', import.meta.url), 'utf8'),
HochsitzSchema);
14  const g = ableiten(p);
15  const r = rechne(p, g);
16
17  describe('Geometrie', () => {
18    it('leitet die bekannten Masse ab', () => {
19      expect(g.rahmen_x_cm).toBe(352);
20      expect(g.netzfeld_z_cm).toBe(211);
21      expect(g.ok_rahmen_cm).toBe(212);
22      expect(g.knuepfebene_cm).toBe(208);
23      expect(g.kopffreiheit_cm).toBe(138);
24      expect(g.netz_m2).toBeCloseTo(6.92, 2);
25      expect(g.W_mm3).toBeCloseTo(288e3, 0);
26      expect(g.kabine_ok_cm).toBe(212); // Kabinen 217 auf dem Rohboden, 5 cm unter OK Podest
27      expect(g.kabine_ueberstand_cm).toBe(12); // v1.3 nannte +17 – abgeleitet aus dem alten 3D-Modell
mit OK Deckplatte 195
28    });
29  });
30
31  describe('Bemessungsszenario 1E+1J+1K, Kind springend (Dossier v1.3, Kap. 3-5)', () => {
32    const b = r.std;
33    it('Lasten', () => {
34      expect(b.Qk).toBeCloseTo(2.15, 2);
35      expect(b.Pd).toBeCloseTo(3.225, 2);
36    });
37    it('Seilmodell', () => {
38      expect(Math.round(b.f_cm)).toBe(34);
39      expect(b.Hz).toBeCloseTo(3.89, 2);
40      expect(b.Hx).toBeCloseTo(1.68, 2);
41      expect(b.Tstrang).toBeCloseTo(0.41, 2);
42      expect(Math.round(b.licht_cm)).toBe(174);
43    });
44    it('Längsbalken', () => {
45      expect(b.Mh).toBeCloseTo(2.61, 2);
46      expect(b.Mv).toBeCloseTo(1.03, 2);
47      expect(b.sigma).toBeCloseTo(12.7, 1);
48      expect(Math.round(b.eta * 100)).toBe(86); // v3.0: k_mod 0.8 (mittel) statt 0.9 → f_m,d 14.8; bis
v2.1 76 %
49    });
50    it('Auflager und Rand', () => {
51      expect(b.C).toBeCloseTo(1.95, 2);
52      expect(b.Rv).toBeCloseTo(1.56, 2);
53      expect(b.sigma_c90).toBeCloseTo(0.11, 2);
54      expect(b.sigma_wand).toBeCloseTo(0.26, 2);
55      // Augplatten ab den Ecken mit Mindestabstand 30 → effektiv 32.8 cm längs; v1.3 rechnete mit fix
30 cm (1.27 kN, 24 %)
56      expect(b.rand_abstand_eff_cm).toBeCloseTo(32.8, 1);

```

```

57     expect(b.Fkn).toBeCloseTo(1.38, 2);
58     expect(Math.round(b.eta_platte * 100)).toBe(26);
59   });
60   it('Varianten Knüpfmass 0/5/10 cm', () => {
61     expect(r.varianten.map((v) => Math.round(v.f_cm))).toEqual([34, 34, 35]);
62     expect(r.varianten.map((v) => +v.Hz.toFixed(2))).toEqual([3.89, 3.86, 3.77]);
63     expect(r.varianten.map((v) => Math.round(v.eta * 100))).toEqual([86, 85, 84]); // bis v2.1 (k_mod
0.9): 76/76/74
64   });
65   it('Gebrauchslast 1 Person', () => {
66     // v1.3 rechnete diesen Fall mit  $\beta = 0.55$  (H = 1.53 kN) und einer veralteten Knüpfebene (lichte
Höhe 181 cm).
67     // Neu konsistent symmetrisch ( $\beta = 0.5$ ) und mit Knüpfebene 208 cm → dokumentiert im
Änderungsverzeichnis 2.0.
68     expect(Math.round(r.gebrauch.f_cm)).toBe(22);
69     expect(r.gebrauch.Hz).toBeCloseTo(1.63, 1);
70     expect(Math.round(r.gebrauch.licht_cm)).toBe(186);
71   });
72   it('Geländer-Deckenanker und Stoss in den Rahmen', () => {
73     expect(r.gelaender.T_punkt).toBeCloseTo(6.45, 1);
74     expect(r.gelaender.T_linie).toBeCloseTo(13.3, 1);
75     expect(r.gelaender.T_linie_weich).toBeCloseTo(8.0, 1);
76     expect(r.gelaender.F_rahmen).toBeCloseTo(3.1, 1);
77   });
78   it('alle rechnerischen Nachweise erfüllt', () => {
79     expect(r.alleErfuehlt).toBe(true);
80     const v = r.nachweise.find((n) => n.id === 'verschieben')!;
81     expect(v.eta!).toBeLessThan(0.3);
82   });
83   });
84
85   describe('Verbindungsmittel Ecke', () => {
86     it('Längen aus der Bezeichnung', () => {
87       expect(r.verbindung.s1_laenge_cm).toBe(16);
88       expect(r.verbindung.s2_laenge_cm).toBe(18);
89     });
90     it('S1 und S2 enden in der Deckplatte, nicht im Kasten', () => {
91       const v = r.verbindung;
92       expect(v.s1_in_deckplatte).toBe(true);
93       expect(v.s1_in_deck_cm).toBeCloseTo(4.0, 1);
94       expect(v.s2_in_deckplatte).toBe(true);
95       expect(v.s2_spitze_y_cm).toBeCloseTo(196.4, 1);
96       expect(v.s2_reserve_cm).toBeGreaterThan(1);
97     });
98     it('S2 kreuzt die Blattfuge und bleibt bis zur Deckplatte im Blatt des Längsbalkens', () => {
99       const v = r.verbindung;
100      expect(v.s2_fuge_z_cm).toBeCloseTo(6.7, 1); // v3.0: Kopf 3.2 statt 4 cm hinter der Kante (bis
v2.1: 7.5)
101      expect(v.s2_deck_z_cm).toBeCloseTo(10.1, 1); // bis v2.1: 10.9
102      expect(v.s2_deck_z_cm).toBeLessThan(g.b);
103      expect(v.s2_im_blaett).toBe(true);
104      expect(v.s2_in_laengs_cm).toBeCloseTo(6.9, 1);
105      expect(v.s2_in_quer_cm).toBeCloseTo(6.9, 1);
106      expect(v.s2_in_deck_cm).toBeCloseTo(4.1, 1);
107    });
108    it('die frühere Kopflage z = 8.5 hätte das Blatt des Längsbalkens verfehlt', () => {
109      const p2 = structuredClone(p);
110      p2.befestigung.ecke.positionen.s2 = [8, 8.5];
111      expect(rechne(p2, g).verbindung.s2_im_blaett).toBe(false);
112    });
113    it('Kopfabstand und Randabstände  $\geq 4d$ ', () => {
114      expect(r.verbindung.kopfabstand_cm).toBeGreaterThanOrEqual(p.befestigung.ecke.abstand_min_cm);
115      expect(r.verbindung.randabstaende_ok).toBe(true);
116    });
117    it('Abheben aus dem Kippmoment hergeleitet, S1 hält', () => {
118      expect(r.abheben.F_ecke).toBeCloseTo(0.9, 1);
119      expect(r.abheben.F_ecke / p.befestigung.widerstand.s1_axial_kn).toBeLessThan(0.6);
120    });
121    it('ein 8×200 bei 30° würde unter der Deckplatte austreten (Grund für 8×180)', () => {

```

```

122     const p2 = structuredClone(p);
123     p2.befestigung.ecke.s2 = 'VG 8x200';
124     expect(rechne(p2, g).verbindung.s2_in_deckplatte).toBe(false);
125   });
126 };
127
128 describe('Mengen', () => {
129   it('Schrauben: zwei je Ecke plus zwei je Querbalken', () => {
130     expect(r.mengen.querbalken_positionen_cm).toEqual([78.3, 156.7]);
131     expect(r.mengen.schrauben_8x160).toBe(8);
132     expect(r.mengen.schrauben_8x180).toBe(4);
133     expect(r.mengen.schrauben_deckplatte).toBe(8);
134   });
135   it('Augplatten: alle Reihen ab den Ecken (zwei Platten je Ecke), Mindestabstand eingehalten', () => {
136     expect(r.mengen.rand_x_cm[0]).toBe(0);
137     expect(r.mengen.rand_x_cm.at(-1)).toBe(g.netzfeld_x_cm);
138     expect(r.mengen.rand_x_cm).toHaveLength(11);
139     expect(r.mengen.rand_abstand_x_cm).toBeGreaterThanOrEqual(p.befestigung.netzrand_abstand_cm);
140     expect(r.mengen.rand_abstand_z_cm).toBeGreaterThanOrEqual(p.befestigung.netzrand_abstand_cm);
141     expect(r.mengen.rand_z_cm[0]).toBe(0);
142     expect(r.mengen.rand_z_cm.at(-1)).toBe(g.netzfeld_z_cm);
143     expect(r.mengen.rand_z_cm).toHaveLength(8); // Querbalken-Reihe ebenfalls mit beiden Eckplatten
144     expect(r.mengen.augplatten_netzrand).toBe(38);
145     expect(r.mengen.augplatten_gelaender).toBe(12);
146     expect(r.mengen.augplatten).toBe(50);
147   });
148   it('rasterMitEcken', () => {
149     expect(rasterMitEcken(328, 30)).toHaveLength(11);
150     expect(rasterMitEcken(328, 30)[1]).toBeCloseTo(32.8, 1);
151     expect(rasterMitEcken(60, 30)).toEqual([0, 30, 60]);
152     expect(rasterMitEcken(50, 30)).toEqual([0, 50]);
153   });
154 });
155
156 describe('Grenzfälle', () => {
157   it('ohne Last kein Zug, Durchhang = Knüpfmass', () => {
158     const s = seilmodell(0, 0.1, 3.28, 2.11, 80);
159     expect(s.Hz).toBe(0);
160     expect(s.f).toBeCloseTo(0.1);
161   });
162   it('mehr Last → mehr Durchhang, mehr Zug', () => {
163     const a = seilmodell(1, 0, 3.28, 2.11, 80);
164     const b = seilmodell(3, 0, 3.28, 2.11, 80);
165     expect(b.f).toBeGreaterThan(a.f);
166     expect(b.Hz).toBeGreaterThan(a.Hz);
167   });
168   it('steiferes Netz → weniger Durchhang', () => {
169     const weich = seilmodell(3, 0, 3.28, 2.11, 40);
170     const steif = seilmodell(3, 0, 3.28, 2.11, 160);
171     expect(steif.f).toBeLessThan(weich.f);
172   });
173   it('Gleichgewicht: getragene Last = aufgebrachte Last', () => {
174     const s = seilmodell(3.225, 0, 3.28, 2.11, 80);
175     expect(s.Pz + s.Px).toBeCloseTo(3.225, 3);
176   });
177   it('Slackline-Kontrolle:  $H \approx P \cdot L / (4f)$  je Richtung', () => {
178     const s = seilmodell(3.225, 0, 3.28, 2.11, 80);
179     expect(s.Hz).toBeCloseTo((s.Pz * 2.11) / (4 * s.f), 3);
180     expect(s.Hx).toBeCloseTo((s.Px * 3.28) / (4 * s.f), 3);
181   });
182   it('Netz unbelastet im Bemessungspfad', () => {
183     const b = bemessung(p, g, { erwachsene: 0, jugendliche: 0, kinder: 0, knuepfmass_cm: 5,
184     randabstand_cm: 30 });
185     expect(b.Pd).toBe(0);
186     expect(b.eta).toBe(0);
187     expect(b.f_cm).toBeCloseTo(5);
188   });
189 });

```

Anhang B zum Statik-Dossier «Hochsitz mit Netz» · Version 3.0 · 02.09.2026 · Zur Prüfung · Build adba937 vom 02.09.2026 ·
Erstellt vom Projektteam (YOLU) mit Unterstützung von Claude.