Part 618 – Soil Properties and Qualities

Subpart B – Exhibits

618.80 Guides for Estimating Risk of Corrosion Potential for Uncoated Steel

Property	Limits		
	Low	Moderate	High
Internal free water occurrence class (or drainage class) and general texture group 1/ 2/	•Very deep internal free water occurrence (or excessively drained to well drained) coarse to medium textured soils; *or* •Deep internal free water occurrence (or moderately well drained) coarse textured soils; *or* •Moderately deep internal free water occurrence (or somewhat poorly drained) coarse textured soils	•Very deep internal free water occurrence (or well drained) moderately fine textured soils; *or* •Deep internal free water occurrence (or moderately well drained) moderately coarse and medium textured soils; *or* •Moderately deep internal free water occurrence (or somewhat poorly drained) moderately coarse textured soils; *or* •Very shallow internal free water occurrence (or very poorly drained) soils with a stable high water table	•Very deep internal free water occurrence (or well drained) fine textured or stratified soils; *or* •Deep internal free water occurrence (or moderately well drained) moderately fine and fine textured or stratified soils; *or* •Moderately deep internal free water occurrence (or somewhat poorly drained) medium to fine textured or stratified soils; *or* •Shallow or very shallow internal free water occurrence (or poorly or very poorly drained) soils with a fluctuating water table
Total acidity (cmol(+)/kg^{-1}) 3/ 4/	<10	10-25	≥25
Conductivity of saturated extract (dS/m^{-1}) 3/ 5/	<1	1-4 4-10 for saturated soils 6/	>4 >10 for saturated soils 6/
Resistivity at saturation (ohm/cm) 1/ 7/	≥5,000	2,000-5,000	<2,000

1/ Based on data in the publication "Underground Corrosion," table 99, p.167, Circular 579, U.S. Department of Commerce, National Bureau of Standards.

2/ The depth classes for internal free water occurrence are defined in table 3-5 of the Soil Survey Manual (1993). The classes relate to the wet water state in soils (i.e., free water present). The general texture groups are defined in chapter 3 of the Soil Survey Manual.

3/ Based on data in Moore and Hallmark (1987), "Soil Properties Influencing Corrosion of Steel in Texas Soils", Soil Sci. Soc. Am. J. 51:1250-1256.

4/ Total acidity is roughly equal to extractable acidity. Extractable acidity is determined by method 4B2a1a1, as outlined in Soil Survey Investigations Report No. 42, Soil Survey Laboratory Methods Manual, Version 4.0, November 2004.

5/ Electrical conductivity is measured using method 4F2, as outlined in Soil Survey Investigations Report No. 42, *Soil Survey Laboratory Methods Manual*, Version 4.0, November 2004. The relationship between resistivity of a saturated soil paste and electrical conductivity of the saturation extract is influenced by variations in the saturation percentage, salinity, and conductivity of the soil minerals. These two measurements generally correspond closely enough to place a soil in one risk of corrosion potential class.

6/ Soils that remain saturated for extended periods are excluded from the high risk of corrosion potential class unless EC values

are more than 10 dS m^{-1} (Moore and Hallmark, 1987). In the NASIS steel corrosion calculation, saturation for extended periods is defined as having very shallow internal free water occurrence for 12 months.

7/ Resistivity at saturation is roughly equivalent to resistivity of fine and medium textured soils measured at saturation (Method 4F2b2 as outlined in Soil Survey Investigations Report No. 42, Soil Survey Laboratory Methods Manual, Version 4.0, November 2004). Resistivity at saturation for coarse textured soils is generally lower than that obtained at field capacity and may cause the soil to be placed in a higher risk of corrosion potential class.

618.81 Guide for Estimating Risk of Corrosion Potential for Concrete

Property	Limits 1/		
	Low	Moderate	High
Texture and reaction	Sandy and organic soils with pH>6.5 *or* Loamy and clayey soils with pH>6.0	Sandy and organic soils with pH 5.5 to 6.5 *or* Loamy and clayey soils with pH 5.0 to 6.0	Sandy and organic soils with pH<5.5 *or* Loamy and clayey soils with pH<5.0
Na and/or Mg sulfate (ppm)	Less than 1000	1000 to 7000	More than 7000
NaCl (ppm)	Less than 2000	2000 to 10000	More than 10000

1/ Based on data in National Handbook of Conservation Practices, Standard 606, Subsurface Drain, 1980.

618.82 Crop Names and Units of Measure

Refer to the NASIS-related metadata at https://www.nrcs.usda.gov/wps/portal/nrcs/detail/soils/survey/tools/?cid=nrcs142p2_053548 . Then follow the link to the "NASIS Version 7.x" index web page. On the NASIS Version index web page, see the file named "Domains.pdf" for the most current list of crop names and crop yield units.

618.83 Classification of Soils and Soil-Aggregate Mixtures for the AASHTO System

General Classification	Granular Materials (35% or less passing No. 200)							Silt-Clay Materials (More than 35% passing No. 200)			
	A-1		A-3	A-2				A-4	A-5	A-6	A-7
Group classification	A-1-a	A-1-b		A-2-4	A-2-5	A-2-6	A-2-7				A-7-5 A-7-6
Sieve analysis, % passing											
No. 10	50 max	-	-	-	-	-	-	-	-	-	-
No. 40	30 max	50 max	51 min	-	-	-	-	-	-	-	-
No. 200	15 max	25 max	10 max	35 max	35 max	35 max	35 max	36 min	36 min	36 min	36 min
Characteristics of fraction passing No. 40											
Liquid limit	-		-	40 max	41 min	40 max	41 min	40 max	41 min	40 max	* 41 min
Plasticity index	6 max		NP	10 max	10 max	11 min	11 min	10 max	10 max	11 min	11 min
Usual types of significant constituent materials	Stone Fragments, Gravel and Sand		Fine Sand	Silty or Clayey Gravel and Sand				Silty Soils		Clayey Soils	
General rating as subgrade	Excellent to Good					Fair to Poor					

* Plasticity index of A-7-5 subgroup is equal to or less than LL minus 30. Plasticity index of A-7-6 subgroup is greater than LL minus 30.

618.84 Potential Frost Action

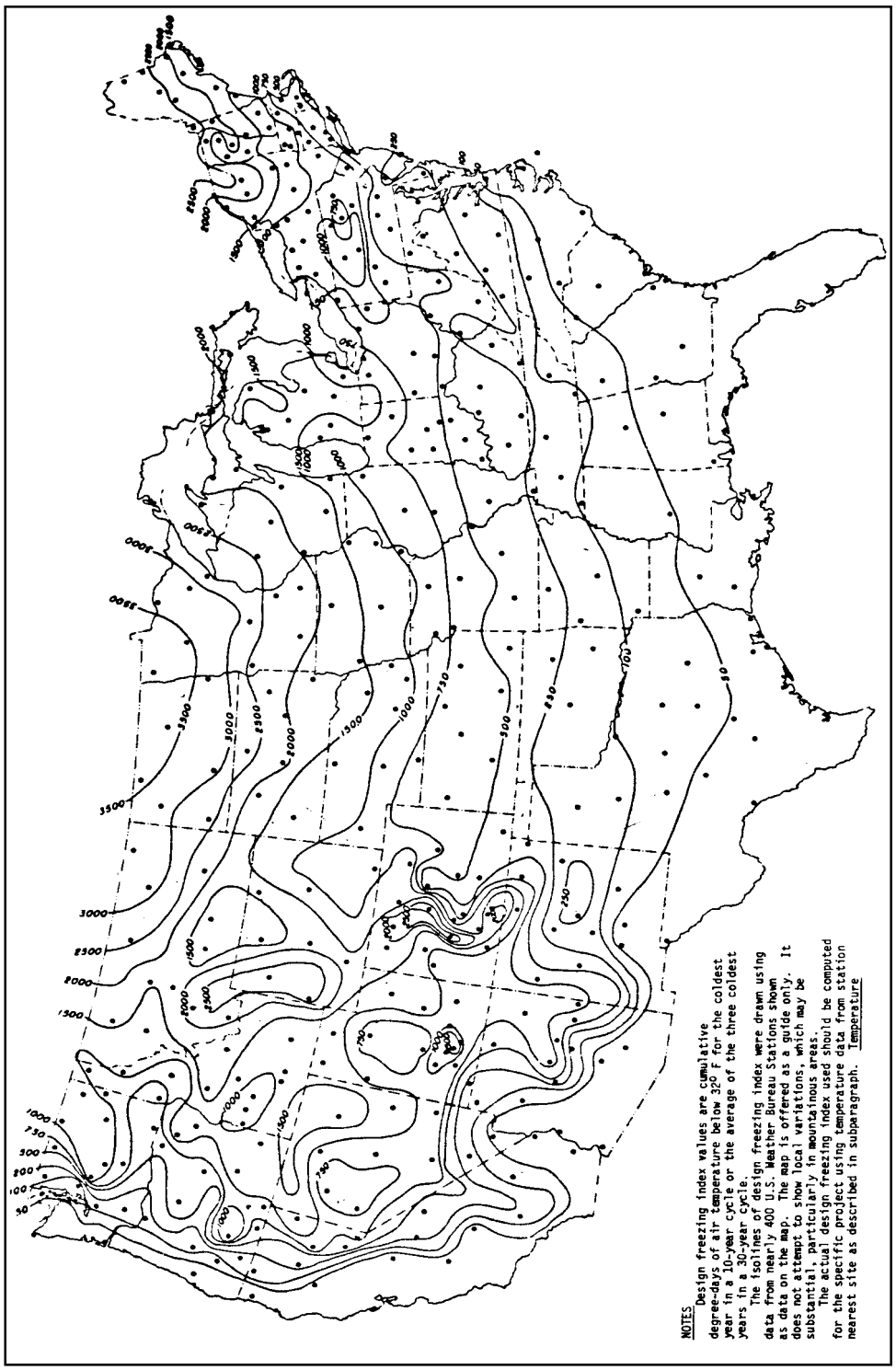
Soil moisture regime	Frost action classes 1/, 2/		
	Low	**Moderate**	**High** 3/
Aquic, Peraquic	Cindery, Fragmental, Pumiceous	Sandy, Sandy-skeletal	Coarse-loamy, Fine-loamy, Coarse-silty, Fine-silty, Loamy, Loamy-skeletal, Clayey, Clayey-skeletal, Fine, Very-fine, Organic soil material, Ashy, Ashy-pumiceous, Ashy-skeletal, Medial, Medial-pumiceous, Medial-skeletal, Hydrous-pumiceous, Hydrous-skeletal, Hydrous
Udic, Perudic, Xeric, Ustic (when irrigated), Aridic and torric (when irrigated)	Fragmental, Cindery, Sandy, Sandy-skeletal, Pumiceous	Coarse-loamy, Fine-loamy, Loamy-skeletal, Clayey, Clayey-skeletal, Fine, Very-fine, Ashy-pumiceous, Ashy-skeletal, Hydrous-skeletal, Medial-skeletal, Medial-pumiceous	Coarse-silty, Fine-silty, Ashy, Medial, Hydrous-pumiceous, Hydrous
Ustic, Aridic and torric	Fragmental, Sandy, Sandy-skeletal, Clayey, Clayey-skeletal, Fine, Very-fine, Cindery, Ashy, Ashy-pumiceous, Ashy-skeletal, Medial, Medial-skeletal, Pumiceous	Coarse-loamy, Fine-loamy, Coarse-silty, Fine-silty, Loamy, Loamy-skeletal, Medial-pumiceous, Hydrous-pumiceous, Hydrous-skeletal, Hydrous	

1/ Taxonomic family particle-size classes apply to the whole soil to the depth of frost penetration, which is not necessarily the same as the taxonomic family particle-size control section.

2/ Isomesic and warmer soil temperature regimes should have no frost action problems ("none").

3/ Organic soil materials with a mesic or colder soil temperature regime and a udic soil moisture regime (e.g., Folists) have a "high" frost action class.

618.85 Distribution of Design Freezing Index Values in the Continental United States



618.86 Estimating LL and PI from Percent and Type of Clay

The following two formulas provide estimates of liquid limit and plasticity index. These calculations are included in the NASIS database and provide default values to LL and PI.

LL = 11.60 + [1.49 × 15 bar water %] + [1.35 × org. carbon %] + [0.6 × LEP] + [0.26 × noncarbonate clay %]*

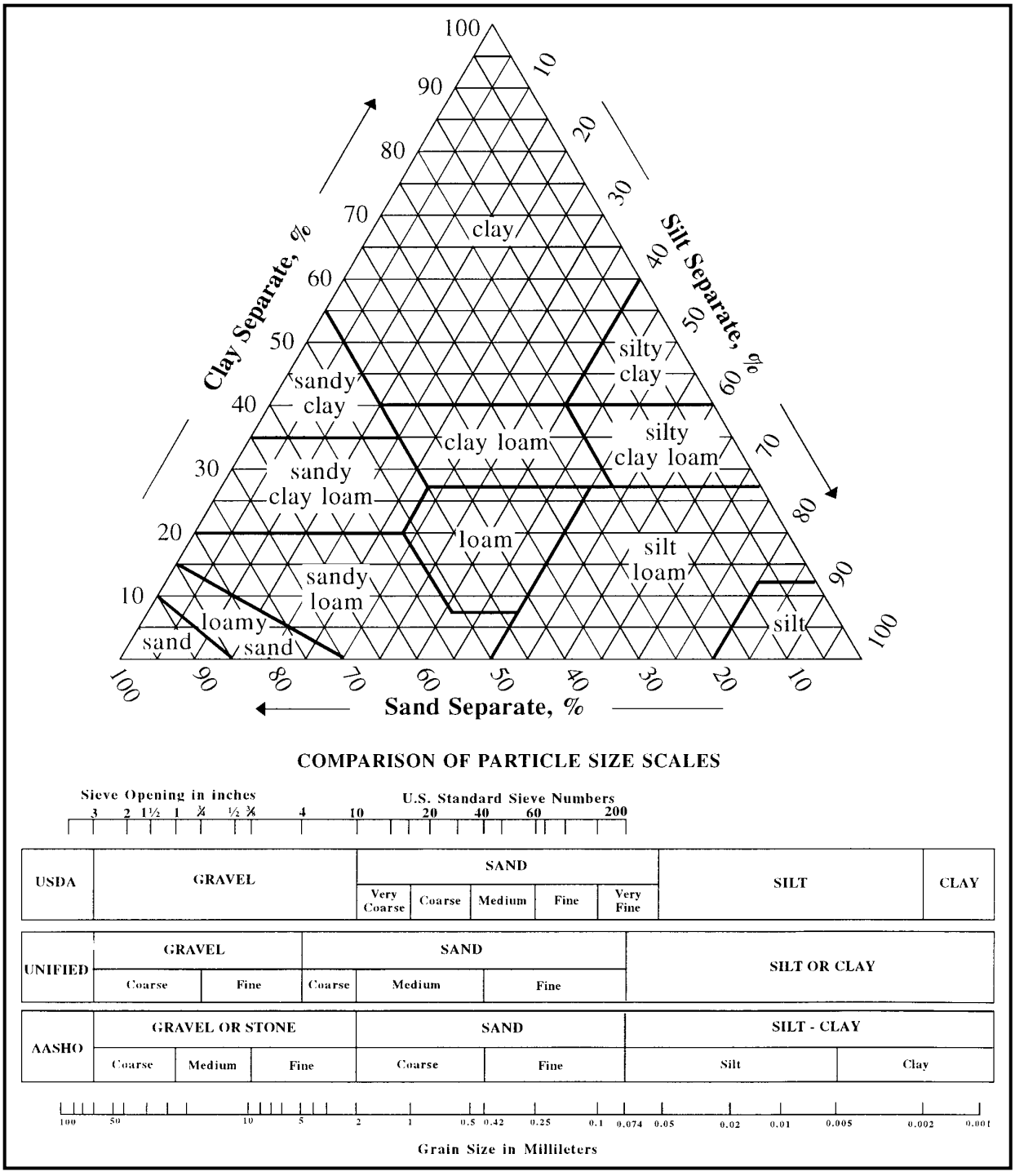
where LL is liquid limit and LEP is linear extensibility percent

PI = -1.86 + [0.69 × 15 bar water %] - [1.19 × organic carbon %] +[0.13 × LEP] +[0.47 × noncarbonate clay %]*

where PI is plasticity index and LEP is linear extensibility percent

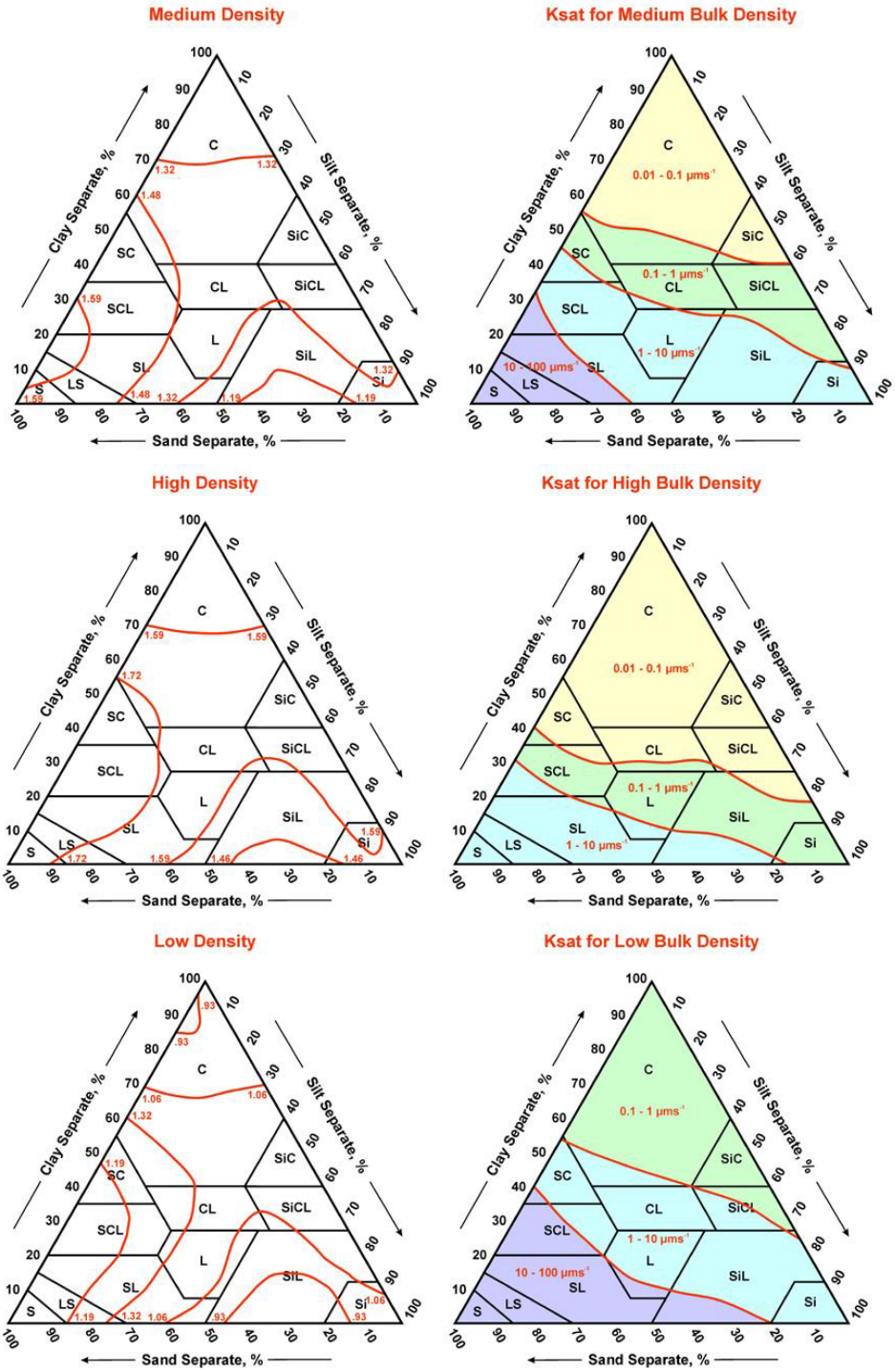
* When the calculated PI < 0.5, the PI is set to zero (nonplastic). When the calculated LL < 15 or PI < 0.5, the LL is set to zero.

618.87 Texture Triangle and Particle-Size Limits of AASHTO, USDA, and Unified Classification Systems



618.88 Guide for Estimating K_{sat} from Soil Properties

Estimate saturated hydraulic conductivity (K_{sat}) from soil texture by first selecting the bulk density class of medium, low, or high. Then use the corresponding texture triangle to select the range of saturated hydraulic conductivity in μms^{-1}. Overrides follow the texture triangles.



If overriding conditions (listed below) exist, use this table to estimate K_{sat} instead of the texture triangles. A single property statement is sufficient for an override from the texture guides.

Overriding Condition	**Saturated Hydraulic Conductivity ($\mu m\ s^{-1}$)**
All fragmental, cindery, or pumiceous.	≥ 100
Many medium or coarser vertical pores that extend through the layer.	≥ 100
Medial-pumiceous, medial-skeletal, ashy-pumiceous, ashy-skeletal, or hydrous-pumiceous material that is very friable, friable, soft, or loose.	10 – 100
When material is moderately moist or wetter, structure is moderate or strong granular, strong blocky, or prismatic smaller than very coarse; no stress surfaces or slickensides.	10 – 100
Common medium or coarser vertical pores extend through the layer.	10 – 100
Strong very coarse blocky or prismatic structure and no stress surfaces or slickensides.	1 – 10
≥ 35 percent clay that is soft, slightly hard, very friable or friable; no stress surfaces or slickensides and the clay activity is in the range of the Subactive class (i.e., CEC7/noncarbonate clay = < 0.24) after subtracting the quantity [2 × (% OC × 1.7)].	1 – 10
Few stress surfaces, few slickensides, or both.	0.1 – 1
Massive and very firm or extremely firm or weakly cemented.	0.1 – 1
Continuously moderately cemented.	0.1 – 1
Common or many stress surfaces or common or many slickensides.	0.01 - 0.1
Continuously indurated or very strongly cemented.	< 0.01

618.89 Guide to Estimating Water Movement Through Bedrock for Layers Designated as R and Cr

This table is to be used as a guide and may be adjusted to reflect local, regional, or State bedrock permeability data[1, 2]. Fracturing may increase hydraulic conductivity of consolidated rock by a factor of 10^4 to 10^6, which is dependent on the degree and interconnection of fracturing. This table assumes that materials are level bedded. Tilted beds of some materials may have rapid rates of water movement for water that goes directly to an aquifer.

Material	Water Movement $\mu m\ s^{-1}$
Sandstone	
unfractured	<10
fractured	10-100
weathered	10-100
Limestone	
unfractured	<1
fractured	<10
weathered	<10
Limestone, Karst	>100
Shales and Mudstones	
consolidated	<1
weathered	<10
Igneous and Metamorphic Rocks	
unfractured	<1
fractured	1-100
weathered	<1

[1] Freeze, R., and J. Cherry. 1979. *Groundwater*.

[2] Legget, R., and P. Karrow. 1983. *Handbook of Geology in Civil Engineering*.

618.90 Rock Fragment Modifier of Texture

Instructions for Table 1, Guide for determining rock fragment modifier of texture: First choose the row with the appropriate total rock fragments. Then read the criteria in the columns under "Gravel, cobbles, stones, and boulders," starting from the left-most column and proceeding to the right. Stop in the first column in which a criterion is met.

Table 1.—Guide for Determining Rock Fragment Modifier of Texture.

(Click here for an MS Excel spreadsheet that calculates the texture modifiers for flat and nonflat rock fragments.).

Total Rock Fragments (Vol. %)	Gravel (GR), cobbles (CB), stones (ST), and boulders (BY) (Substitute channers for gravel and flagstones for cobbles, where applicable)[1]			
	IF GR ≥ 1.5 CB + 2 ST + 2.5BY	IF CB ≥ 1.5 ST + 2 BY	IF ST ≥ 1.5 BY	IF ST < 1.5 BY
≥ 15 < 35	Gravelly	Cobbly	Stony	Bouldery
≥ 35 < 60	Very Gravelly	Very Cobbly	Very Stony	Very Bouldery
≥ 60 < 90	Extremely Gravelly	Extremely Cobbly	Extremely Stony	Extremely Bouldery
≥ 90	Gravel	Cobbles	Stones	Boulders

Example: Determine the rock fragment modifier for a soil that contains 15 percent gravel (GR), 10 percent cobbles (CB), and 3 percent stones (ST).

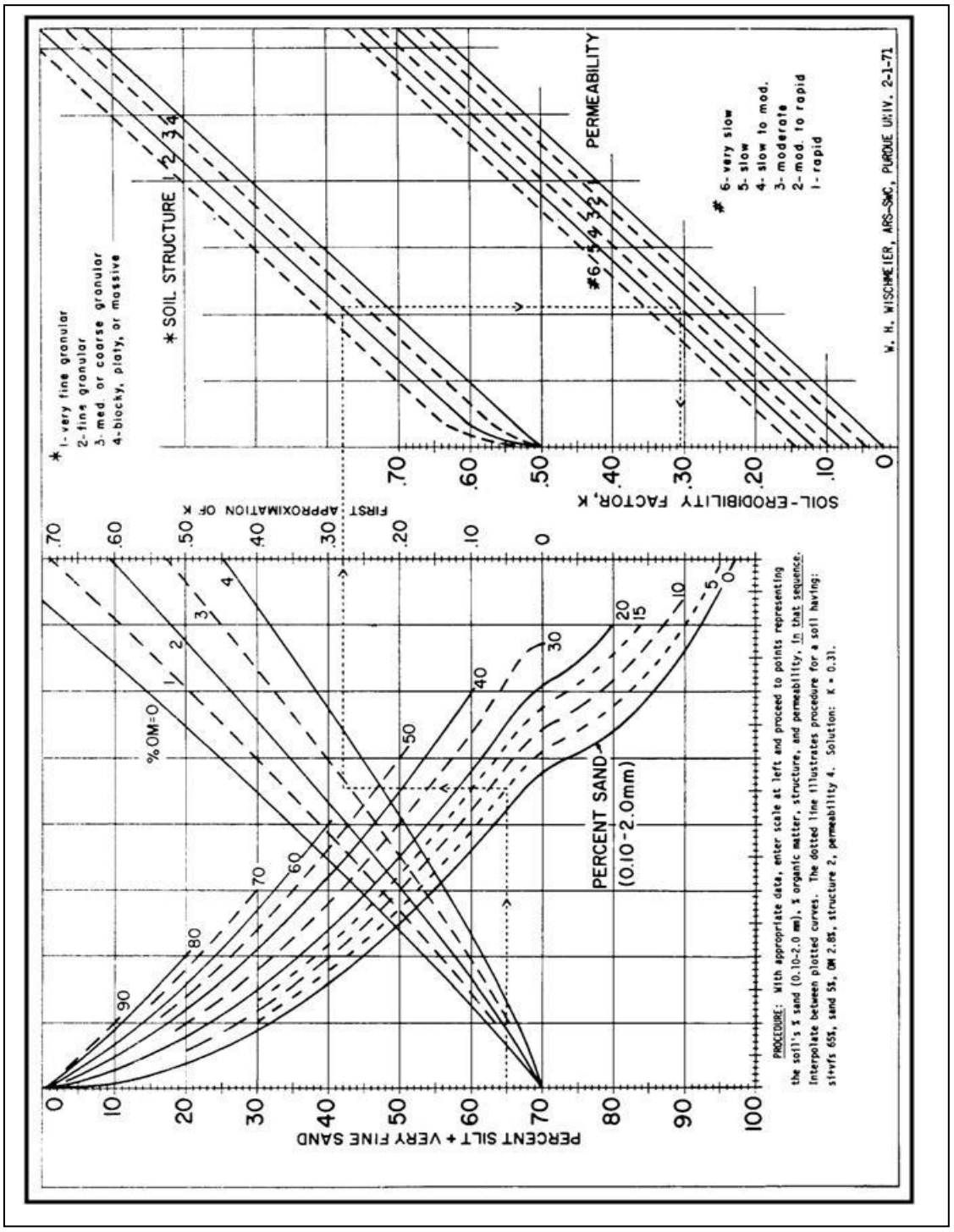
1. Since total rock fragments are 28 percent, choose the first row (≥ 15 and < 35).
2. Under "Gravel (GR), cobbles, . . .", test the criterion in the left-most column. Is 15%GR ≥ 1.5 (10% CB) + 2 (3% ST)? Answer: NO.
3. Proceed to the next column. Is 10%CB ≥ 1.5 (3% ST)? Answer: YES. STOP. The modifier is Cobbly.

[1] If both flat and nonflat rock fragments are present, the quantity in each size class is summed (e.g., gravel + channers, cobbles + flagstones). The sums are used to determine the appropriate quantity/size modifier. If the amounts of flat and nonflat rock fragments within any given size class are equal, the nonflat modifier takes precedence. For example, if there are 10 percent gravel and 10 percent channers, the modifier is gravelly.

Soils With Pararock Fragments Only.—The same basic weighting rules apply with pararock fragments as with flat and nonflat rock fragments. However, the above spreadsheet only outputs modifier terms for rock fragments. To assign the correct pararock fragment modifier to the outputted rock modifier term, simply precede the modifier with "para." For example, if the calculator outputs "very cobbly," the correct modifier is "very paracobbly."

Soils With Both Rock and Pararock Fragments.—Refer to instructions in section 618.67H(2)(vii) of this handbook.

618.91 Soil Erodibility Nomograph



618.92 Kw Value Associated With Various Fragment Contents

Fragment vol. %	Mulch factor 1/	Kf value classes of less than 2 mm soil fraction										
		.10	.15	.20	.24	.28	.32	.37	.43	.49	.55	.64
5	.90	.09	.14	.18	.22	.25	.29	.33	.39	.44	.50	.58
10	.77	.08	.12	.15	.18	.22	.25	.28	.33	.38	.42	.49
15	.68	.07	.10	.14	.16	.19	.22	.25	.29	.33	.37	.43
20	.61	.06	.09	.12	.15	.17	.20	.23	.26	.30	.37	.39
25	.54	.05	.08	.11	.13	.15	.17	.20	.23	.26	.30	.35
30	.48	.05	.07	.10	.12	.13	.15	.18	.21	.24	.26	.31
35	.43	.04	.06	.09	.10	.12	.14	.16	18	.21	.24	.28
40	.38	.04	.06	.08	.09	.11	.12	.14	.16	.19	.21	.24
45	.34	.03	.05	.07	.08	.10	.11	.13	.15	.17	.19	.22
50	.30	.03	.05	.06	.07	.08	.10	.11	.13	.15	.17	.19
55	.26	.03	.04	.05	.06	.07	.08	.09	.11	.13	.12	.14
60	.22	.02	.03	.04	.05	.06	.07	.08	.09	.11	.12	.14
65	.19	.02	.03	.04	.05	.05	.06	.07	.08	.09	.10	.12
70	.16	.02	.02	.03	.04	.04	.05	.06	.07	.08	.09	.10
75	.13	.01	.02	.03	.04	.04	.04	.04	.06	.06	.07	.08
80	.10	.01	.02	.02	.02	.03	.03	.04	.04	.05	.06	.06
85	.08	.01	.02	.02	.02	.02	.03	.03	.03	.04	.04	.05
90	.06	.01	.01	.01	.01	.02	.02	.02	.03	.03	.03	.04
95	.04	.01	.01	.01	.01	.01	.01	.02	.02	.02	.02	.03
100	.03	.01	.01	.01	.01	.01	.01	.01	.01	.02	.02	.02

1/ Mulch factor is the ratio of the soil loss from soils with the specified fragment volumes to that from soils with no fragments. The table was constructed from the zero canopy curve, figure 6, page 19 in Agriculture Handbook 537 (USDA Science and Education Administration, 1978).

618.93 General Guidelines for Assigning Soil Loss Tolerance "T"

Soil loss tolerance "T" is assigned according to properties of root and plant growth limiting subsurface soil layers. The designation of a limiting layer implies that the material above the layer has more favorable plant growth properties. As limiting or less favorable soil layers become closer to the soil surface, the relative ability of a soil to maintain its productivity through natural and managed processes decreases.

Caution should be used in comparing T factors across soils for soil quality or productivity. For examples, soils with a T factor of 5 may not be the most productive and soils with the same T factor rating may not be equally productive. For example, a soil that has a T factor of 5 and is sandy throughout is not as naturally fertile nor can it hold as much available water as a soil that has a T factor of 5 that is loamy throughout.

The criteria for assigning T factor are estimated from both of the following:

1. The severity of physical or chemical properties of subsurface layers

2. The economic feasibility of utilizing management practices to overcome limiting layers or conditions

The following general guide was used with specific soil properties and conditions to write criteria statements for programming T factors as a calculation in NASIS.

Depth to limiting layer (cm)	Soil loss tolerance in tons per acre		
	Group 1	Group 2	Group 3
0 - 25	1	1*	3
25 - 50	1	2	3
50 - 100	2	3	4
100 - 150	3	4	4
>150	5	5	5

* Some soils are assigned a soil loss tolerance of 2.

Group 1.—The limitations are significant or there are permanent layers of root limitation (nonrenewable).

Group 2.—The limitations for roots are moderate, or there is a less than permanent loss to productivity (renewable).

Group 3.—The limitations can be overcome through natural or managed processes, and the productivity level of the noneroded soil can be achieved (very renewable).

All restrictions in the NASIS "Component Restrictions Table" are considered root-limiting, either physically or chemically.

T factors are assigned based on the criteria presented below. If there is more than one limiting soil characteristic, then the soil is rated based on the most limiting soil characteristic based on the "T" criteria in the table.

"T" Criteria

12/31/2009

"T" Criteria [1/]			
Soil Characteristic	**Definition**	**Depth Limit (cm)**	**T Factor [2/]**
1. Organic	A. For Histosols, depth to the first mineral horizon with < 20% organic matter; soil is not in a lithic, limnic, hydric, or fluvaquentic subgroup and not in Sulfohemists or Sulfihemists great groups.	≤150 >150	1 2
	OR		
	B. For soils that are Histosols in a lithic, limnic, hydric, or fluvaquentic subgroup or are in Sulfohemists or Sulfihemists great groups.		1
	OR C. Except in Alaska; mineral soils that are histic intergrades (i.e., histic subgroup). If the histic epipedon has been destroyed, then ignore this rating and rate T based on other limiting features of the mineral soil.		1
2. Bedrock	A. Except in Alaska; depth to densic bedrock as identified in the NASIS "Component Restrictions Table." (renewable)	<25 25-50 50-100 100-150 >150	1 2 3 4 5

	In Alaska (nonrenewable)	<50 50-100 100-150 >150	1 2 3 5
	OR		
	B. Except in Alaska; depth to paralithic bedrock as identified in the NASIS "Component Restrictions Table." (renewable)	<25 25-50 50-100 100-150 >150	1 2 3 4 5
	In Alaska (nonrenewable)	<50 50-100 100-150 >150	1 2 3 5
	OR		
	C. Depth to lithic bedrock as identified in the NASIS "Component Restrictions Table." (nonrenewable)	<50 50-100 100-150 >150	1 2 3 5
3. Permafrost	Depth to permafrost as identified in the NASIS "Component Restrictions Table." (nonrenewable)	<50 50-100 100-150 >150	1 2 3 5
4. Cemented layers/pans	A. Depth to duripan or petroferric, petrocalcic, petrogypsic, placic, ortstein, or cemented layer (or contiguous layers) that is ≤ 7.6 cm (3 inches) thick; hardness (i.e., rupture resistance) is extremely weakly, very weakly, weakly, or moderately as identified in the NASIS "Component Restrictions Table." (very renewable)	<50 50-150 >150	3 4 5

OR		
B. Except in Alaska; depth to a duripan or petroferric, petrocalcic, petrogypsic, placic, ortstein, or cemented layer (or contiguous layers) that is ≤ 7.6 cm (3 inches) thick; hardness (i.e., rupture resistance) is strongly or very strongly as identified in the NASIS "Component Restrictions Table." (renewable)	<25	1
	25-50	2
	50-100	3
	100-150	4
	>150	5
Alaska (nonrenewable)	<50	1
	50-100	2
	100-150	3
	>150	5
OR		
C. Depth to a duripan or petroferric, petrocalcic, petrogypsic, placic, ortstein, or cemented layer that is ≤ 7.6 cm (3 inches) thick; hardness (i.e., rupture resistance) is indurated as identified in the NASIS "Component Restrictions Table." (nonrenewable)	<50	1
	50-100	2
	100-150	3
	>150	5
OR		
D. Except in Alaska; depth to a duripan or petrocalcic, petrogypsic, petroferric, placic, ortstein, or cemented layer (or contiguous layers) that is > 7.6 cm (3 inches) thick (or if thickness is not specified); hardness (i.e., rupture resistance) is extremely, very weakly, weakly, or moderately as identified in the NASIS "Component Restrictions Table." (renewable)	<25	1
	25-50	2
	50-100	3
	100-150	4
	>150	5
In Alaska (nonrenewable)	<50	1
	50-100	2
	100-150	3
	>150	5
OR		

	E. Depth to a duripan or petrocalcic, petrogypsic, petroferric, placic, ortstein, or cemented layer (or contiguous layers) that is > 7.6 cm (3 inches) thick (or if thickness is not specified); strongly or greater hardness (i.e., rupture resistance) as identified in the NASIS "Component Restrictions Table." (nonrenewable)	<50 50-100 100-150 >150	1 2 3 5
5. Fragmental	Depth to fragmental layer (i.e., consists of "in lieu of" textures of artifacts, boulders, cobbles, channers, flagstones, gravel, or stones). (nonrenewable)	<50 50-100 100-150 >150	1 2 3 5
6. Rock fragments	A. Except in Alaska; if the weighted average of rock fragments in the 0-25 cm depth is < 35% (by volume), then rate T based on depth to the first subsurface layer with ≥ 60% rock fragments that has its lower boundary extending to 150 cm or more. (renewable)	<50 50-100 100-150 >150	2 3 4 5
	In Alaska only; if the weighted average of rock fragments in the 0-12 cm depth is < 35% (by volume), then rate T based on depth to the first subsurface layer with ≥ 60% rock fragments that has its lower boundary extending to 150 cm or more. (nonrenewable)	<50 50-100 100-150 >150	1 2 3 5
	OR		
	B. Except in Alaska; if the weighted average of rock fragments in the 0-25 cm depth is < 35% (by volume), then rate T based on depth to the first subsurface layer with ≥ 60% rock fragments that has its lower boundary within 150 cm. (very renewable)	<50 50-150 >150	3 4 5
	In Alaska only; if the weighted average of rock fragments in the 0-12 cm depth is < 35% (by volume), then rate T based on depth to the first subsurface layer with ≥ 60% rock fragments that has its lower boundary within 150 cm. (nonrenewable)	<50 50-100 100-150 >150	1 2 3 5

	OR		
	C. Except in Alaska; if the weighted average of rock fragments in the 0-25 cm depth is ≤ 15% (by volume), then rate T based on depth to the first subsurface layer with ≥ 35% rock fragments that has its lower boundary extending to 150 cm or more. (very renewable)	<50 50-150 >150	3 4 5
	In Alaska only; if the weighted average of rock fragments in the 0-12 cm depth is ≤ 15% (by volume), then rate T based on depth to the first subsurface layer with ≥ 35% rock fragments that has its lower boundary extending to 150 cm or more. (nonrenewable)	<50 50-100 100-150 >150	1 2 3 5
7. Plinthite	Depth to plinthite as identified in the NASIS "Component Restrictions Table." (nonrenewable)	<50 50-100 100-150 >150	1 2 3 5
8. Fragipan and fragic soil properties	A. Depth to fragipan, as identified in the NASIS "Component Restrictions Table," that has ≥ 35% rock fragments. (renewable)	<50 50-100 100-150 >150	2 3 4 5
	OR		
	B. Depth to fragipan, as identified in the NASIS "Component Restrictions Table," that has < 35% rock fragments. (very renewable)	<50 50-150 >150	3 4 5
	OR		
	C. Soils that are in a fragic subgroup or have an "x" suffix symbol (fragipan character) in any horizon designation; rate T based on depth to the layer that has the greatest bulk density change (from a lower to higher bulk density) and has a $K_{sat} \leq 1.41$ µm/s (0.5 cm/h). (very renewable)	<50 50-150 >150	3 4 5

9. Natric	A. Depth to natric horizon, as identified in the NASIS "Component Restrictions Table." (renewable)	<50	2
		50-100	3
		100-150	4
		>150	5
	OR		
	B. For soils that have horizons that do not meet natric horizon criteria but have a subsoil with ≥ 35% clay, K_{sat} ≤ 1.41 µm/s (0.5 cm/h), and SAR ≥ 13 and have adjacent upper layers with a K_{sat} > 1.41 µm/s (0.5 cm/h) that extend to the surface (and the soils are not glossic). (renewable)	<50	2
		50-100	3
		100-150	4
		>150	5
10. Dense layers	Except in Alaska; depth to densic material, as identified in the NASIS "Component Restrictions Table." (renewable)	<50	2
		50-100	3
		100-150	4
		>150	5
	In Alaska (nonrenewable)	<50	1
		50-100	2
		100-150	3
		>150	5
11. Abrupt increase in clay	A. Depth to layer with an abrupt textural change as identified in the NASIS "Component Restrictions Table." (very renewable)	<50	3
		50-150	4
		>150	5
	OR		
	B. Depth to substratum that has ≥ 35% clay, has a K_{sat} < 1.41 µm/s (0.5 cm/h), and an adjacent upper layer with a K_{sat} > 4.23 µm/s (1.5 cm/h); or a subsoil that has a K_{sat} < 0.42 µm/s (0.15 cm/h) and an adjacent upper surface layer with a K_{sat} > 1.41 µm/s (0.5 cm/h); the clay increase between the adjacent layers must be ≥ 20%, and the surface or upper layers are greater than or equal to 25 cm thick. (very renewable)	<50	3
		50-150	4
		>150	5

12. Strongly contrasting textural stratification	Depth to layers that have strongly contrasting textural stratification, as identified in the NASIS "Component Restrictions Table." (very renewable)	<50 50-150 >150	3 4 5
13. Sandy substratum	A. Except in Alaska; depth to sandy substratum (COS, S, FS, LS, LCOS, or LFS) with a K_{sat} > 42.3 µm/s (15.2 cm/h) that extends to 150 cm or more, and the adjacent upper layers have a K_{sat} < 42.3 µm/s (15.2 cm/h) and have < (0.667*% clay + 50) percent fine or coarser sand separates in the fine-earth fraction that extends to the surface [3/]; the surface or upper layers are greater than or equal to 25 cm in thickness. (renewable)	<50 50-100 100-150 >150	2 3 4 5
	In Alaska only; depth to sandy substratum (COS, S, FS, LS, LCOS, or LFS) with a K_{sat} > 42.3 µm/s (15.2 cm/h) that extends to 150 cm or more, and the adjacent upper layers have a K_{sat} < 42.3 µm/s (15.2 cm/h) and have < (0.667*% clay + 50) percent fine or coarser sand separates in the fine-earth fraction that extends to the surface [3/]; the surface or upper layers are greater than or equal to 12 cm in thickness. (renewable)	<50 50-100 100-150 >150	2 3 4 5
	OR		
	B. Except in Alaska; depth to substratum with strongly contrasting sandy textural stratification or stratified with sandy textures of COS, S, LS, FS, LCOS, or LFS and K_{sat} > 42.3 µm/s (15.2 cm/h) that extends to 150 cm or below, and the adjacent upper layers have a K_{sat} < 42.3 µm/s (15.2 cm/h) and have < (0.667*% clay + 50) percent fine or coarser sand separates in the fine-earth fraction that extends to the surface [3/]; the surface or upper layers are greater than or equal to 25 cm in thickness. (renewable)	<50 50-100 100-150 >150	2 3 4 5

	In Alaska only; depth to substratum with strongly contrasting sandy textural stratification or stratified with sandy textures of COS, S, LS, FS, LCOS, or LFS and K_{sat} > 42.3 µm/s (15.2 cm/h) that extends to 150 cm or below, and the adjacent upper layers have a K_{sat} < 42.3 µm/s (15.2 cm/h) and have < (0.667*% clay + 50) percent fine or coarser sand separates in the fine-earth fraction that extends to the surface [3/]; the surface or upper layers are greater than or equal to 12 cm in thickness. (renewable)	<50 50-100 100-150 >150	2 3 4 5
14. High gypsum	A. Soils that have a surface layer with ≤ 15% gypsum (by weight) and have subsurface layers with ≥ 25% gypsum; rate T based on depth to the first subsurface layer with ≥ 25% gypsum. (renewable)	<50 50-100 100-150 >150	2 3 4 5
	OR		
	B. Soils that have a surface layer with > 15 and ≤ 60% gypsum (by weight) and have subsurface layers that exceed the surface layer in gypsum content by 20% or more; rate T based on depth to the first subsurface layer that exceeds the surface by 20% or more gypsum. (renewable)	<50 50-100 100-150 >150	2 3 4 5
15. High salts	Depth to salic horizon, as identified in the NASIS "Component Restrictions Table." (very renewable)	<50 50-150 >150	3 4 5
16. High sulfur	Depth to sulfuric horizon, as identified in the NASIS "Component Restrictions Table." (very renewable)	<50 50-150 >150	3 4 5
17. High carbonates	A. Soils that have a surface layer with ≤ 15% calcium carbonate ($CaCO_3$) equivalent and have subsurface layers with ≥ 40% $CaCO_3$ equivalent; rate T based on depth to the first subsurface layer with ≥ 40% $CaCO_3$ equivalent. (renewable)	<50 50-100 100-150 >150	2 3 4 5

	OR		
	B. Soils that have a surface layer with > 15 and ≤ 60% calcium carbonate ($CaCO_3$) equivalent and have subsurface layers that exceeds the surface layer in $CaCO_3$ equivalent by 25% or more; rate T based on depth to the first subsurface layer that exceeds the surface layer by 25% or more $CaCO_3$ equivalent. (renewable)	<50 50-100 100-150 >150	2 3 4 5
18. Human-manufactured materials	Depth to human-manufactured materials ('M' layers), as identified in the NASIS "Component Restrictions Table." (nonrenewable)	<50 50-100 100-150 >150	1 2 3 5
19. High aluminum	A. Depth to ≥ 60% Al saturation of the ECEC and soil pH (1:1 water) is ≤ 5.5. (very renewable)	<50 50-150 >150	3 4 5
	OR		
	B. In Pacific Islands only; for Oxisols or soils in Oxic subgroups or for Ultisols with an isohyperthermic soil temperature regime; depth to ≥ 45% Al saturation of the ECEC and soil pH (1:1 water) is ≤ 5.5. (nonrenewable)	<50 50-100 100-150 >150	1 2 3 5

1/ Subaqueous soils (i.e., soils classified in Wassents or Wassists suborders) are excluded from assignment of T factors.

2/ Severely eroded soils, as designated by a the local phase or erosion class of 3 or 4, are adjusted one class of T factor lower.

3/ Determines the line between the 30% clay and 70% sand point and the 0% clay and 50% sand point on the texture triangle (see part 618, subpart B, section 618.87). If the total of the vcos, cos, ms, and fs sand separates and total clay (of the adjacent layers above the sandy substratum) plot above this line on the texture triangle, then the uppers layers are considered different from the sandy substratum textures. Thus, the sandy substratum is recognized and the T factor is lowered.

618.94 Texture Class, Texture Modifier, and Terms Used in Lieu of Texture

Texture Class		Texture Modifier 1/		Terms Used in Lieu of Texture	
Class	**Code**	**Modifier**	**Code**	**Term**	**Code**
Clay	C	Artifactual	ART	Artifacts	ART
Clay loam	CL	Very artifactual	ARTV	Bedrock	BR
Coarse sand	COS	Extremely artifactual	ARTX	Boulders	BY
Coarse sandy loam	COSL	Ashy	ASHY	Cobbles	CB
Fine sand	FS	Bouldery	BY	Coarse gypsum material	CGM
Fine sandy loam	FSL	Bouldery-artifactual	BYART	Channers	CN
Loam	L	Very bouldery	BYV	Fine gypsum material	FGM
Loamy coarse sand	LCOS	Very bouldery-artifactual	BYVART	Flagstones	FL
Loamy fine sand	LFS			Gravel	GR
Loamy sand	LS	Extremely bouldery	BYX	Highly decomposed plant material	HPM
Loamy very fine sand	LVFS	Extremely bouldery-artifactual	BYXART		
Sand	S			Material	MAT
Sandy clay	SC	Cobbly	CB	Moderately decomposed plant material	MPM
Sandy clay loam	SCL	Cobbly-artifactual	CBART		
Silt	SI	Very cobbly	CBV	Mucky peat	MPT
Silty clay	SIC	Very cobbly-artifactual	CBVART	Muck	MUCK
Silty clay loam	SICL			Paraboulders	PBY
Silt loam	SIL	Extremely cobbly	CBX	Paracobbles	PCB
Sandy loam	SL	Extremely cobbly-artifactual	CBXART	Parachanners	PCN
Very fine sand	VFS			Peat	PEAT
Very fine sandy loam	VFSL	Cemented	CEM	Paraflagstones	PFL
		Channery	CN	Paragravel	PG
		Channery-artifactual	CNART	Parastones	PST
		Very channery	CNV	Shells	SHL
		Very channery-artifactual	CNVART	Slightly decomposed plant material	SPM
		Extremely channery	CNX	Stones	ST
		Extremely channery-artifactual	CNXART	Water	W
		Coprogenous	COP		
		Diatomaceous	DIA		
		Flaggy	FL		
		Flaggy-artifactual	FLART		
		Very flaggy	FLV		
		Very flaggy-artifactual	FLVART		
		Extremely flaggy	FLX		
		Extremely flaggy-artifactual	FLXART		
		Gravelly	GR		
		Gravelly-artifactual	GRART		
		Coarse gravelly	GRC		
		Fine gravelly	GRF		
		Medium gravelly	GRM		
		Very gravelly	GRV		
		Very gravelly-artifactual	GRVART		
		Extremely gravelly	GRX		

		Extremely gravelly-artifactual	GRXART		
		Grassy	GS		
		Gypsiferous	GYP		
		Herbaceous	HB		
		Highly organic	HO		
		Hydrous	HYDR		
		Medial	MEDL		
		Mucky	MK		
		Marly	MR		
		Mossy	MS		
		Paraboulder y	PBY		
		Very paraboulder y	PBYV		
		Extremely paraboulder y	PBYX		
		Paracobbly	PCB		
		Very paracobbly	PCBV		
		Extremely paracobbly	PCBX		
		Parachannery	PCN		
		Very parachannery	PCNV		
		Extremely parachannery	PCNX		
		Permanently frozen	PF		
		Paraflaggy	PFL		
		Very paraflaggy	PFLV		
		Extremely paraflaggy	PFLX		
		Paragravelly	PGR		
		Very paragravelly	PGRV		
		Extremely paragravelly	PGRX		
		Parastony	PST		
		Very parastony	PSTV		
		Extremely parastony	PSTX		
		Peaty	PT		
		Shelly	SHF		
		Very shelly	SHFV		
		Extremely shelly	SHFX		
		Shelly-artifactual	SHFART		
		Very shelly-artifactual	SHFVART		
		Extremely shelly-artifactual	SHFEART		
		Stony	ST		
		Stony-artifactual	START		
		Very stony	STV		
		Very stony-artifactual	STVART		
		Extremely stony	STX		
		Extremely stony-artifactual	STXART		
		Woody	WD		

1/ "Texture modifiers" may apply to both "texture class" and "terms used in lieu of texture." Some apply to both, others only apply to one or the other. See part 618, subpart A, section 618.71, for more information.

618.95 Wind Erodibility Groups (WEG) and Index

WEG [1,3,4,5,7]	Properties of Soil Surface Layer	Dry Soil Aggregates More Than 0.84 mm (wt.%)	Wind Erodibility Index (I) (tons/ac/yr)
1	Very fine sand, fine sand, sand, or coarse sand[2]	1 2 3 5 7	310 250 220 180 160
2	Loamy very fine sand, loamy fine sand, loamy sand, and loamy coarse sand; very fine sandy loam and silt loam with 5 or less percent clay; and sapric soil materials (as defined in Soil Taxonomy), except Folists.	10	134
3	Very fine sandy loam (but does not meet WEG criterion 2), fine sandy loam, sandy loam, and coarse sandy loam; noncalcareous silt loam that has greater than or equal to 20 to less than 50 percent very fine sand and greater than or equal to 5 to less than 12 percent clay.	25	86
4	Clay, silty clay, noncalcareous clay loam that has more than 35 percent clay and noncalcareous silty clay loam that has more than 35 percent clay; all of these do not have sesquic, parasesquic, ferritic, ferruginous, or kaolinitic mineralogy (high iron oxide content).	25	86
4L	Calcareous[6] loam, calcareous silt loam, calcareous silt, calcareous sandy clay, calcareous sandy clay loam, calcareous clay loam, and calcareous silty clay loam.	25	86
5	Noncalcareous loam that has less than 20 percent clay, noncalcareous silt loam with greater than or equal to 5 to less than 20 percent clay (but does not meet WEG criterion 3), noncalcareous sandy clay loam, noncalcareous sandy clay, and hemic soil materials (as defined in Soil Taxonomy).	40	56
6	Noncalcareous loam and silt loam that have greater than or equal to 20 percent clay; noncalcareous clay loam and noncalcareous silty clay loam that have less than or equal to 35 percent clay; silt loam that has parasesquic, ferritic, or kaolinitic mineralogy (high iron oxide content).	45	48
7	Noncalcareous silt; noncalcareous silty clay, noncalcareous silty clay loam, and noncalcareous clay that have sesquic, parasesquic, ferritic, ferruginous, or kaolinitic mineralogy (high content of iron oxide) and are Oxisols or Ultisols; and fibric soil materials (as defined in Soil Taxonomy).	50	38
8	Soils not susceptible to wind erosion due to rock and pararock fragments at the surface and/or wetness; and Folists.	--	0

The following footnotes are applied in the order listed:

1 For all WEGs except 1 and 2 (sands and loamy sand textures), if percent rock and pararock fragments (>2mm) by volume is 15-35, reduce "I" value by one group with more favorable rating. If percent rock and pararock fragments by volume is 35-60, reduce "I" value by two favorable groups except for sands and loamy sand textures which are reduced by one group with more favorable rating. If percent rock and pararock fragments is greater than 60, use "I" value of 0 for all textures except sands and loamy sand textures which are reduced by three groups with more favorable ratings. An example of more favorable "I" rating is next lower number: "I" factor of 160 to "I" factor of 134 or "I" factor of 86 to "I" factor of 56. The index values should correspond exactly to their wind erodibility group (e.g., "I" factor of 56 = WEG 5).

2 The "I" values for WEG 1 vary from 160 for coarse sands to 310 for very fine sands. Use an "I" of 220 as an average figure.

3 All material that meets criterion 3 in the required characteristics for andic soil properties as defined in the Keys to Soil Taxonomy, 11th edition. Such material is placed in WEG 2 regardless of the texture class of the fine-earth fraction.

4 All material that meets criterion 2, but not criterion 3, in the required characteristics for andic soil properties as defined in the Keys to Soil Taxonomy, 11th edition. Such material is placed in WEG 6, regardless of the texture class of the fine-earth fraction. The only exception to this is for Cryic Spodosols which have a medial substitute class and a MAAT < 4 degrees C.; these soils are placed in WEG 2.

5 For surface layers or horizons that do not meet the required characteristics for andic soil properties but do meet Vitrandic, Vitritorrandic, Vitrixerandic, and Ustivitrandic subgroup criteria (thickness criterion excluded) move one wind erodibility group (WEG) with a less favorable rating.

6 Calcareous is a strongly or violently effervescent reaction (class) of the fine-earth fraction to cold dilute (1N) HCL; a paper "Computing the Wind Erodible Fraction of Soils" by D. W. Fryear et.al (1994) in the Journal of Soil and Water Conservation 49 (2) 183-188 raises a yet unresolved question regarding the effect of carbonates on wind erosion.

7 For mineral soils with thin "O" horizons, the WEG is based on the first mineral horizon.

618.96 Key Landforms and Their Susceptibility to Slippage

Topography	Landform or Geological Materials	Slippage Potential [A]
I. Level Terrain		
A. Not elevated	Flood plain, till plain, lakebed	3
B. Elevated		
1. Uniform tones	Terrace, lakebed,	2
2. Surface irregularities, sharp cliffs	basaltic plateau	1
3. Interbedded, porous over impervious layers	lakebed, coastal plain	1
II. Hilly Terrain		
A. Surface drainage not well integrated		
1. Disconnected drainage	Limestone	3
2. Deranged drainage, overlapping hills, associated with lakes and swamps (glaciated areas only)	Moraine	2
B. Surface drainage well integrated		
1. Parallel ridges		
a. Parallel drainage, dark tones	Basaltic hills	1
b. Trellis drainage, ridge-and-valley topography, banded hills	Downslope tilted sedimentary rock	1
c. Pinnate drainage, vertical-sided gullies	Loess	2
2. Branching ridges, hilltops at common elevation		
a. Pinnate drainage, vertical-sided gullies	Loess	2
b. Dendritic drainage		
(1) Banding on slopes	Flat-lying sed. rocks	2
(2) No banding on slopes		
(a) Moderately to highly dissected ridges, uniform slopes	Clay shale	1
(b) Low ridges associated with coastal features	Dissected coastal plains	1
(c) Winding ridges connection, conical hills, sparse vegetation	Serpentinite	1
3. Random ridges or hills		
a. Dendritic drainage		
(1) Low, rounded hills, meandering streams	Clay shale	1
(2) Winding ridges, connecting conical hills, sparse vegetation	Serpentinite	1
(3) Massive, uniform, rounded to A-shaped hills	Granite	2
(4) Bumpy topography (glaciated areas only)	Moraines	2
III. Level to Hilly Terrain		
A. Steep slopes	Talus, colluvium	1
B. Moderate to flat slopes	Fan, delta	3
C. Hummocky slopes with scarp at head	Old slide	1

A. Ratings for slippage potential:
1 = susceptible to slippage (unstable); 2 = susceptible to slippage under certain conditions (moderately unstable); 3 = not susceptible to slippage except in vulnerable locations (slightly unstable to stable)

618.97 Example Worksheets for Soil Moisture State by Month and Depth

SOIL MOISTURE STATE BY MONTH AND DEPTH

Aridic Thermic

	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
Ppt (mm)	10	10	8	4	6	2	8	10	6	4	8	8
0 SOIL DEPTH 200 cm	M D	M D	M D	M D	D	D	D	D	D	D	D	M D

Xeric Mesic

	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
Ppt (mm)	180	140	110	60	40	30	10	20	40	80	170	200
0 SOIL DEPTH 200 cm	M W	M W	M W	M	M	D M	D M	D M	M D M	M	M	 W

Ustic Mesic

	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
Ppt (mm)	10	15	50	60	80	100	70	70	70	40	25	15
0 SOIL DEPTH 200 cm	M D M	M D M	M D M	M	M	M	D M	D M	D M	M D M	M D M	M D M

Udic Mesic

	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
Ppt (mm)	50	60	80	80	100	100	110	90	70	50	80	70
0 SOIL DEPTH 200 cm	M/W	M/W	M/W	M/W	M/W	M	M	D/M	D/M	M	M/W	M/W

SOIL MOISTURE STATE BY MONTH AND DEPTH

	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
Ppt (mm)												
0 SOIL DEPTH 200 cm												

SOIL MOISTURE STATE BY MONTH AND DEPTH

	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
Ppt (mm)												
0 SOIL DEPTH 200 cm												

SOIL MOISTURE STATE BY MONTH AND DEPTH

	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
Ppt (mm)												
0 SOIL DEPTH 200 cm												

618.98 NASIS Calculation for Estimating AASHTO Group Index

Definition.—Computes the AASHTO Group Index for a horizon

Inputs.—This calculation requires the following data to be populated:

number 200 sieve
liquid limit
plasticity index

Calculation.

```
DEFINE skip_ll      ANY (aashto_class == "a-2-6" or aashto_class == "a-2-7").
DEFINE aashind      .01*(sieveno200_r-15)*(pi_r-10).

ASSIGN aashind      IF skip_ll OR (sieveno200_r<35 and pi_r>=10) THEN aashind ELSE (sieveno200_r
- 35)*(.2 + .005*(ll_r - 40)) + aashind.

ASSIGN aashind      IF pi_r==0 OR (aashind<0 AND NOT ISNULL(aashind)) THEN 0 ELSE aashind.
```

618.99 NASIS Calculation for Estimating Cation-Exchange Capacity

Definition.—If the rv pH (in water) is greater than or equal to 5.5, then CEC is calculated; if the rv pH is less than 5.5, then ECEC is calculated for a horizon.

Caution: Estimates of CEC or ECEC for soil layers with andic soil properties may be unreliable. Read the documentation to this calculation to see if it will work for your soils.

Inputs.—This calculation requires the following data to be populated:

organic matter (high, low, and rv)
pH in water (high, low, and rv)
pH in CaCl (for organic layers) (high, low, and rv)
total clay (high, low, and rv)
carbonate clay (high, low, and rv)
total silt (high, low, and rv)
$CaCO_3$ (high, low, and rv) used only for soils containing gypsum
gypsum (high, low, and rv) used only for soils containing gypsum
taxonomic family mineralogy
taxonomic order
taxonomic CEC-activity class

1) The calculation is based on Soil Taxonomy (Soil Survey Staff, 1999). The CEC or ECEC is calculated based on the family mineralogy/CEC-activity class first. If an equation does not exist for a mineralogy/CEC-activity class, then the CEC/ECEC is based on the soil order.
2) If the family mineralogy class, CEC-activity class (if appropriate), and soil order are not populated, a null is returned.
3) If there is more than one family mineralogy class populated in the "Component-Taxonomic-Family-Mineralogy Table," the first one is used.
4) If the pH in water is not populated, then pH in CaCl is used. If the pH (CaCl) is greater than or equal to 5.1, then CEC is calculated; otherwise ECEC is calculated for a horizon.
5) If any required data element (OM, pH, clay, silt) for an equation is not populated (null entry), a null is returned, except for carbonate clay.
6) If carbonate clay is null, zero carbonate clay is assumed.
7) Noncarbonate clay is calculated by subtracting percent carbonate clay from total clay (noncarbonated clay = total_clay - carbonate_clay).
8) Percent organic matter is converted to percent organic C by dividing by 1.72 (OC = OM/1.72).
9) If only low and high values are populated for a data element, an rv is calculated by taking the average of the low and high values.
10) In the calculation of CEC for isotic and amorphic mineralogies and Andisols and for isotic mineralogy for ECEC, gravimetric 15-bar water is used. An internal calculation calculates the 15-bar water using the following formula:
    15-bar water = [total_clay (1 - organic_matter / 100) 0.4 + organic_matter]
11) In the calculation of ECEC, if the mineralogy class is not parasesquic, smectitic, or isotic and is not mixed or siliceous with a CEC activity class and the soil order is Andisols, Gelisols, Aridisols, or Vertisols, then no ECEC is calculated (null is returned).

Calculation.

Estimate # -------- gypsum soils ----------

DEFINE cecr IF ISNULL(ph1) THEN 1/0

```
ELSE IF ph1 =="yes" and gyp2mm > 4 and gyp2mm <= 40 THEN EXP(0.851*ln_clayr - 0.02*caco3_r -
0.009*gyp2mm + 0.174) * (1-(gyp2mm/100))
ELSE IF ph1 =="yes" and gyp2mm > 40 THEN 10.035 - 0.093*gyp2mm

# -------- phwat >= 5.5 or phcacl2 >=5.1 and phwat<=7.0 and OC > 8 ------------------
ELSE IF ph1 =="yes"and ocr>14.5 and ph2 =="no"and
 (lieutex1 == "muck" OR lieutex1 == "hpm")
THEN ((2.12 * (ocr)) + (9.992 * (phcacl2r)) - 10.684)
ELSE IF ph1 =="yes"and ocr>14.5 and ph2 =="no"and
(lieutex1 == "mpt" OR lieutex1 == "mpm")
THEN ((2.03 * (ocr)) + (3.396 * (phcacl2r)) - 2.939)
ELSE IF ph1 =="yes"and ocr>14.5 and ph2 =="no"and
  (lieutex1 == "peat" or lieutex1 == "spm")
THEN ((1.314 * (ocr)) + 27.047)
ELSE IF ph1 =="yes"and ocr>8 and ph2 =="no"and ocr <= 14.5
THEN ((1.823 * (ocr)) + (0.398 * (nclayr)) + 15.54)

# ------------ phwat >=5.5 or phcacl2r >= 5.1 and OC >8 and phwat >7.0 ---------------
ELSE IF ph1 =="yes"and ocr>8 and ph2 =="yes"and ocr <= 14.5
THEN EXP(1.316 * (ln_ocr) + 1.063 * (ln_nclayr) - 3.211)
ELSE IF ph1 =="yes"and ocr>8 and ph2 =="yes"and ocr >14.5
THEN (4.314 * (ocr) - 26.492)

# ------------ phwat >= 5.5 or phcacl2 >= 5.1 and OC <= 8, use Mineralogy  --------------
ELSE IF (ph1 =="yes"and ocr <= 8 and taxminalogy1 == "ferruginous")
THEN (2.48 * (ocr) + 0.128 * (siltr) + 3.208)
ELSE IF (ph1 =="yes"and ocr <= 8 and taxminalogy1 == "amorphic")
THEN EXP((0.182*(ln_ocr)) + (0.817*(ln_w15barr)) + (0.736*(ln_phwatr)) - 0.608)
ELSE IF (ph1 =="yes"and ocr <= 8 and taxminalogy1 == "glassy")
THEN EXP((0.102*(ln_ocr)) + (1.219*(ln_w15barr)) - 0.005)
ELSE IF (ph1 =="yes"and ocr <= 8 and (taxminalogy1 == "carbonatic" or taxminalogy1 =="calcareous"))
THEN EXP((0.253*(ln_ocr)) + (0.828*(ln_nclayr)) + 0.321)
ELSE IF (ph1 =="yes"and ocr <= 8 and taxminalogy1 == "magnesic")
THEN (2.38*(ocr) + 0.555*(nclayr) - 0.219*(siltr) + 10.428)
ELSE IF (ph1 =="yes"and ocr <= 8 and taxminalogy1 == "parasesquic")
THEN EXP((0.13*(ln_ocr)) + (0.65*(ln_nclayr)) + (0.340*(ln_phwatr)) - 0.406)
ELSE IF (ph1 =="yes"and ocr <= 8 and taxminalogy1 == "kaolinitic")
THEN EXP((0.206*(ln_ocr)) + (0.618*(ln_nclayr)) + (0.303*(ln_siltr)) + (0.491*(ln_phwatr)) - 1.786)
ELSE IF (ph1 =="yes"and ocr <= 8 and (taxminalogy1 == "smectitic" OR taxminalogy1 ==
"montmorillonitic"))
THEN EXP((0.033*(ln_ocr)) + (0.861*(ln_nclayr)) + 0.246)
ELSE IF (ph1 =="yes"and ocr <= 8 and taxminalogy1 == "illitic")
THEN EXP((0.102*(ln_ocr)) + (0.596*(ln_nclayr)) - (1.108*(ln_phwatr)) + 2.892)
ELSE IF (ph1 =="yes"and ocr <= 8 and taxminalogy1 == "vermiculitic")
THEN (0.365*(nclayr) - 9.724*(phwatr) + 90.293)
ELSE IF (ph1 =="yes"and ocr <= 8 and taxminalogy1 == "isotic")
THEN EXP((0.163*(ln_ocr)) + (0.683*(ln_w15barr)) + (0.812*(ln_phwatr)) - 0.299)
ELSE

# ----------------- use CEC Activity Class ----------------------
IF (ph1 =="yes"and ocr <=8 and taxceactcl1 == "superactive")
THEN EXP((0.039*(ln_ocr)) + (0.901*(ln_nclayr)) + 0.131)
```

```
ELSE IF (ph1 =="yes"and  ocr <= 8 and taxceactcl1  == "active")
THEN EXP((0.015*(ln_ocr)) + (0.987*(ln_nclayr))  - 0.576)
ELSE IF (ph1 =="yes"and  ocr <= 8 and taxceactcl1  == "semiactive")
THEN EXP((0.02*(ln_ocr)) + (0.974*(ln_nclayr))  - 0.927)
ELSE IF (ph1 =="yes" and ocr <= 8 and taxceactcl1  == "subactive")
THEN EXP((0.009*(ln_ocr)) + (1.02*(ln_nclayr))  - 1.675)
ELSE

# --------------------use Taxonomic Order -------------------------
IF (ph1 =="yes"and  ocr <= 0.3 and taxorder1  == "alfisols")
THEN EXP((0.911*(ln_nclayr))  - 0.308)
ELSE IF (ph1 =="yes"and  ocr > 0.3 and ocr <= 8 and taxorder1  == "alfisols")
THEN EXP((0.158*(ln_ocr)) + (0.805*(ln_nclayr))  + 0.216)
ELSE IF (ph1 =="yes"and  ocr <= 8 and taxorder1  == "andisols")
THEN EXP((0.088*(ln_ocr)) + (0.885*(ln_w15barr)) + (0.867*(ln_phwatr))  - 0.985)
ELSE IF (ph1 =="yes"and  ocr <= 8 and taxorder1  == "aridisols")
THEN EXP((0.042*(ln_ocr)) + (0.828*(ln_nclayr))  + 0.236)
ELSE IF (ph1 =="yes"and  ocr <= 8 and taxorder1  == "entisols")
THEN EXP((0.078*(ln_ocr)) + (0.873*(ln_nclayr))  + 0.084)
ELSE IF (ph1 =="yes"and  ocr <= 8 and taxorder1  == "inceptisols")
THEN EXP((0.134*(ln_ocr)) + (0.794*(ln_nclayr))  + 0.239)
ELSE IF (ph1 =="yes"and  ocr <= 0.3 and taxorder1  == "mollisols")
THEN EXP((0.932*(ln_nclayr))  - 0.174)
ELSE IF (ph1 =="yes"and  ocr > 0.3 and ocr <= 8 and taxorder1  == "mollisols")
THEN EXP((0.113*(ln_ocr)) + (0.786*(ln_nclayr))  + 0.475)
ELSE IF (ph1 =="yes"and  ocr <= 8 and taxorder1  == "oxisols")
THEN (2.738*(ocr) + 0.103*(nclayr) + 0.123*(siltr)  - 2.531)
ELSE IF (ph1 =="yes"and  ocr <= 8 and taxorder1  == "spodosols")
THEN EXP((0.045*(ln_ocr)) + (0.798*(ln_nclayr))  + 0.029)
ELSE IF (ph1 =="yes"and  ocr <= 8 and taxorder1  == "ultisols")
THEN EXP((0.184*(ln_ocr)) + (0.57*(ln_nclayr))  + (0.365*(ln_siltr))  - 0.906)
ELSE IF (ph1 =="yes"and  ocr <= 8 and taxorder1  == "vertisols")
THEN EXP((0.059*(ln_ocr)) + (0.86*(ln_nclayr))  + 0.312)
ELSE IF (ph1 =="yes"and  ocr <= 8 and taxorder1  == "histosols")
THEN EXP((0.319*(ln_ocr)) + (0.497*(ln_nclayr))  + 1.075)  ELSE 1/0.
```

618.100 NASIS Calculation for Estimating Effective Cation-Exchange Capacity

Inputs.—See the documentation in part 618, subpart B, section 618.99, on the NASIS calculation for estimating cation-exchange capacity.

Calculation.

```
#---------Calculates ECEC when pH(water) < 5.5-------------------------------------------
DEFINE ececr
IF ISNULL(ph1) THEN 1/0
ELSE IF ph1 == "no" AND ocr > 8 AND taxminalogy1 == "andisols"
THEN EXP(0.938*ln_ocr - 0.029*phcacl2r - 0.054)
ELSE IF ph1 == "no" AND ocr > 8
THEN EXP(0.699*ln_ocr + 0.556*phcacl2r - 1.497)
ELSE IF ph1 =="no"AND ocr <= 8 AND desgnmaster1 == "E"
THEN EXP((0.371*ln_ocr) + (0.728*ln_nclayr) + (0.392*ln_siltr) + (0.728*ln_phwatr) - 2.145)
ELSE IF ph1 =="no"AND ocr <= 8 AND taxminalogy1 == "parasesquic"
THEN EXP(0.109*ln_ocr + 0.904*ln_clayr - 0.927*ln_phwatr - 0.083)
ELSE IF ph1 =="no"AND ocr <= 8 AND (taxminalogy1 == "smectitic" OR taxminalogy1 ==
"montmorillonitic")
THEN EXP(0.965*ln_clayr + 0.939*ln_phwatr - 1.974)
ELSE IF ph1 =="no"AND ocr <= 8 AND taxminalogy1 == "isotic"
THEN EXP(0.124*ln_ocr + 0.535*ln_w15barr + 0.405*ln_siltr - 0.455)
ELSE IF ph1 =="no"AND ocr <= 8 AND taxceactcl1 == "superactive"
THEN EXP(0.035*ln_ocr + 0.913*ln_clayr - 0.341)
ELSE IF ph1 =="no"AND ocr <= 8 AND taxceactcl1 == "active"
THEN EXP(1.15*ln_clayr - 0.115*ln_ocr - 1.725)
ELSE IF ph1 =="no"AND ocr <= 8 AND taxceactcl1 == "semiactive"
THEN EXP(1.049*ln_clayr - 0.058*ln_ocr - 1.864)
ELSE IF ph1 =="no"AND ocr <= 8 AND taxceactcl1 == "subactive"
THEN EXP(0.757*ln_clayr - 1.01*ln_phwatr + 0.214*ln_siltr - 0.465)
ELSE IF ph1 =="no"AND ocr <= 8 AND taxorder1 == "alfisols"
THEN EXP(0.019*ln_ocr + 0.834*ln_clayr + 0.325*ln_phwatr + 0.288*ln_siltr - 1.937)
ELSE IF ph1 =="no"AND ocr <= 8 AND taxorder1 == "entisols"
THEN EXP(0.387*ln_ocr + 0.818*ln_clayr - 0.343)
ELSE IF ph1 =="no"AND ocr <= 8 AND taxorder1 == "inceptisols"
THEN EXP(0.283*ln_ocr + 0.541*ln_clayr + 1.913*ln_phwatr - 2.869)
ELSE IF ph1 =="no"AND ocr <= 8 AND taxorder1 == "mollisols"
THEN EXP(0.122*ln_ocr + 0.721*ln_clayr + 0.6*ln_phwatr - 0.635)
ELSE IF ph1 =="no"AND ocr <= 8 AND taxorder1 == "oxisols"
THEN EXP(0.21*ln_ocr + 0.685*ln_clayr - 2.381*ln_phwatr + 0.355*ln_siltr + 1.169)
ELSE IF ph1 =="no"AND ocr <= 8 AND taxorder1 == "spodosols"
THEN EXP(0.309*ln_ocr + 0.526*ln_clayr + 0.25*ln_siltr - 0.535)
ELSE IF ph1 =="no"AND ocr <= 8 AND taxorder1 == "ultisols"
THEN EXP(0.555*ln_clayr + 0.481*ln_siltr - 1.204*ln_phwatr + 0.016)
ELSE IF ph1 =="no"AND ocr <= 8 AND taxorder1 == "histosols"
THEN 0.443*clayr + 2.377*ocr - 2.906
ELSE 1/0.
```

618.101 NASIS Calculation for Estimating Extractable Acidity

Definition.—Computes the extractable acidity for a horizon.

Inputs.—This calculation requires the following data to be populated:

- organic matter (high, low, and rv)
- pH in water (high, low, and rv)
- pH in $CaCl_2$ (high, low, and rv) only used for organic layers
- total clay (high, low, and rv) only used for medial textures
- CEC or ECEC (high, low, and rv)
- texture (used for identifying hydrous, medial, ashy, and organic soil layers)
- taxonomic order

Limitations.

1) The calculation is based on regression equations developed from measured data in the characterization database. There are regression equations for O horizons of Histosols, O horizons of other soil orders, hydrous textures, medial textures, ashy textures, and mineral layers for each soil order.
2) There are a set of regression equations that use CEC and another set that use ECEC as a predictor variable. IF the pH is < 5.5, then the set of equations that use ECEC is used. IF ECEC in not populated then a null is returned (regardless if CEC is populated or not).
3) If any required data element (OM, pH, CEC, or ECEC) for an equation is not populated (null entry), a null is returned.
4) Organic C is used in the equations. Percent organic matter is converted to percent organic C by dividing by 1.72 (OC = OM/1.72).

Calculation.

```
DEFINE ocr om_r/1.72.
DEFINE ocl om_l/1.72.
DEFINE och om_h/1.72.

#-----Calculate RV extractable acidity----------

DEFINE acidr IF NOT ISNULL(cec7_r) AND (ph1to1h2o_r >= 5.5 OR ph01mcacl2_r >= 5.5) THEN
IF (hzname matches "*O*" OR texture matches "*MPT*" OR texture matches "*MUCK*" OR texture
matches "*PEAT*" OR texture matches "*HPM*" OR texture matches "*MPM*" OR texture matches
"*SPM*") AND ph1to1h2o_r > 6.1 THEN 0.19*cec7_r - 11.411*ph1to1h2o_r + 78.341 ELSE IF taxord
== "histosols" AND NOT ISNULL(taxord) AND (hzname matches "*O*" OR texture matches "*MPT*"
OR texture matches "*MUCK*" OR texture matches "*PEAT*") THEN 0.289*cec7_r + 0.358*ocr -
26.390*ph01mcacl2_r + 149.662 ELSE IF (hzname matches "*O*" AND NOT ISNULL(hzname)) OR
((texture matches "*HPM*" OR texture matches "*MPM*" OR texture matches "*SPM*" OR texture
matches "*MPT*" OR texture matches "*MUCK*" OR texture matches "*PEAT*") AND NOT
ISNULL(texture)) THEN 0.470*cec7_r + 0.298*ocr - 19.702*ph01mcacl2_r + 100.585 ELSE IF texture
matches "*HYDR*" AND NOT ISNULL(texture) THEN -0.312*cec7_r + 3.726*ocr -
20.442*ph1to1h2o_r + 159.093 ELSE IF texture matches "*MEDL*" AND NOT ISNULL(texture)
THEN 0.564*cec7_r - 0.326*claytotal_r - 8.825*ph1to1h2o_r + 75.799 ELSE IF texture matches
"*ASHY*" AND NOT ISNULL(texture) THEN 0.134*cec7_r + 2.669*ocr - 1.972*ph1to1h2o_r +
14.051 ELSE IF taxord == "histosols" THEN 0.673*cec7_r - 7.659*ph1to1h2o_r + 44.466 ELSE IF
taxord == "gelisols" THEN 0.36*cec7_r - 4.301*ph1to1h2o_r + 31.87 ELSE IF taxord == "entisols"
THEN 0.148*cec7_r + 1.679*ocr - 1.791*ph1to1h2o_r + 12.254 ELSE IF taxord == "mollisols" THEN
```

```
0.112*cec7_r + 0.595*ocr - 2.745*ph1to1h2o_r + 19.964 ELSE IF taxord == "alfisols"  THEN
0.205*cec7_r + 1.113*ocr - 2.928*ph1to1h2o_r + 19.545 ELSE IF taxord == "aridisols"  THEN
0.047*cec7_r + 0.535*ocr - 0.973*ph1to1h2o_r + 7.735 ELSE IF taxord == "ultisols"  THEN
0.850*cec7_r + 0.361*ocr - 2.125*ph1to1h2o_r + 11.741 ELSE IF taxord == "inceptisols"  THEN
0.496*cec7_r + 0.698*ocr - 5.010*ph1to1h2o_r + 31.299 ELSE IF taxord == "vertisols"  THEN
0.061*cec7_r + 0.775*ocr -3.557*ph1to1h2o_r + 26.936 ELSE IF taxord == "spodosols"  THEN
1.226*cec7_r - 0.524*ocr - 3.429*ph1to1h2o_r + 20.975 ELSE IF taxord == "oxisols"  THEN
0.499*cec7_r + 1.679*ocr - 2.055*ph1to1h2o_r + 16.422 ELSE IF taxord == "andisols"  THEN
0.763*cec7_r - 4.328*ph1to1h2o_r + 28.591  ELSE 1/0 ELSE IF NOT ISNULL(ecec_r) THEN IF taxord
== "histosols"  AND NOT ISNULL(taxord)  AND (hzname matches "*O*" OR texture matches "*MPT*"
OR texture matches "*MUCK*" OR texture matches "*PEAT*") THEN 0.471*ocr -
20.556*ph01mcacl2_r + 142.732 ELSE IF (hzname matches "*O*" AND NOT ISNULL(hzname)) OR
((texture matches "*HPM*" OR texture matches "*MPM*" OR texture matches "*SPM*" OR texture
matches "*MPT*" OR texture matches "*MUCK*" OR texture matches "*PEAT*") AND NOT
ISNULL(texture))  THEN 1.03*ocr - 19.587*ph01mcacl2_r + 110.208 ELSE IF taxord == "histosols"
THEN 2.717*ocr + 9.247 ELSE IF taxord == "gelisols"  THEN 0.965*ocr - 4.503*ph1to1h2o_r + 35.377
ELSE IF taxord == "entisols"  THEN 0.147*claytotal_r + 2.7*ocr - 1.484*ph1to1h2o_r + 9.572 ELSE IF
taxord == "mollisols"  THEN 0.148*claytotal_r + 1.692*ocr - 2.411*ph1to1h2o_r + 15.606 ELSE IF
taxord == "alfisols"  THEN 0.188*claytotal_r + 2.353*ocr - 4.612*ph1to1h2o_r + 25.601 ELSE IF taxord
== "aridisols"  THEN 0.033*claytotal_r + 2.392*ocr + 2.391*ph1to1h2o_r - 9.935 ELSE IF taxord ==
"ultisols"  THEN 0.899*ecec_r + 0.111*claytotal_r + 2.438*ocr - 1.254*ph1to1h2o_r + 6.046 ELSE IF
taxord == "inceptisols"  THEN 0.429*ecec_r + 0.078*claytotal_r + 3.052*ocr - 2.053*ph1to1h2o_r +
15.165 ELSE IF taxord == "vertisols"  THEN 0.157*claytotal_r + 2.437*ocr - 2.949*ph1to1h2o_r +
15.531 ELSE IF taxord == "spodosols"  THEN 1.581*ecec_r + 3.054*ocr + 6.68 ELSE IF taxord ==
"oxisols"  THEN 0.342*ecec_r + 0.078*claytotal_r + 3.176*ocr + 3.932 ELSE IF taxord == "andisols"
THEN 0.879*ocr - 12.847*ph1to1h2o_r + 96.871 ELSE 1/0 ELSE 1/0.


ASSIGN acidr  IF acidr < 0 AND NOT ISNULL(acidr)  THEN 0 ELSE acidr.


#-----Calculate low extractable acidity----------

DEFINE acidl

IF NOT ISNULL(cec7_l)  AND (ph1to1h2o_r >= 5.5 OR ph01mcacl2_r >= 5.5) THEN IF (hzname
matches "*O*" OR texture matches "*MPT*" OR texture matches "*MUCK*" OR texture matches
"*PEAT*" OR texture matches "*HPM*" OR texture matches "*MPM*" OR texture matches "*SPM*")
AND ph1to1h2o_r > 6.1 THEN 0.19*cec7_l - 11.411*ph1to1h2o_h + 78.341 ELSE IF taxord ==
"histosols"  AND NOT ISNULL(taxord)  AND (hzname matches "*O*" OR texture matches "*MPT*" OR
texture matches "*MUCK*" OR texture matches "*PEAT*") THEN 0.289*cec7_l + 0.358*ocl -
26.390*ph01mcacl2_h + 149.662 ELSE IF (hzname matches "*O*" AND NOT ISNULL(hzname)) OR
((texture matches "*HPM*" OR texture matches "*MPM*" OR texture matches "*SPM*" OR texture
matches "*MPT*" OR texture matches "*MUCK*" OR texture matches "*PEAT*") AND NOT
ISNULL(texture))  THEN 0.470*cec7_l + 0.298*ocl - 19.702*ph01mcacl2_h + 100.585 ELSE IF texture
matches "*HYDR*" AND NOT ISNULL(texture)  THEN -0.312*cec7_h + 3.726*ocl -
20.442*ph1to1h2o_h + 159.093 ELSE IF texture matches "*MEDL*" AND NOT ISNULL(texture)
THEN 0.564*cec7_l - 0.326*claytotal_h  - 8.825*ph1to1h2o_h + 75.799 ELSE IF texture matches
"*ASHY*"  AND NOT ISNULL(texture)  THEN 0.134*cec7_l + 2.669*ocl - 1.972*ph1to1h2o_h +
14.051 ELSE IF taxord == "histosols"  THEN 0.673*cec7_l - 7.659*ph1to1h2o_h + 44.466 ELSE IF
taxord == "gelisols"  THEN 0.36*cec7_l - 4.301*ph1to1h2o_h + 31.87 ELSE IF taxord == "entisols"
THEN 0.148*cec7_l + 1.679*ocl - 1.791*ph1to1h2o_h + 12.254 ELSE IF taxord == "mollisols"  THEN
```

0.112*cec7_l + 0.595*ocl - 2.745*ph1to1h2o_h + 19.964 ELSE IF taxord == "alfisols" THEN 0.205*cec7_l + 1.113*ocl - 2.928*ph1to1h2o_h + 19.545 ELSE IF taxord == "aridisols" THEN 0.047*cec7_l + 0.535*ocl - 0.973*ph1to1h2o_h + 7.735 ELSE IF taxord == "ultisols" THEN 0.850*cec7_l + 0.361*ocl - 2.125*ph1to1h2o_h + 11.741 ELSE IF taxord == "inceptisols" THEN 0.496*cec7_l + 0.698*ocl - 5.010*ph1to1h2o_h + 31.299 ELSE IF taxord == "vertisols" THEN 0.061*cec7_l + 0.775*ocl -3.557*ph1to1h2o_h + 26.936 ELSE IF taxord == "spodosols" THEN 1.226*cec7_l - 0.524*och - 3.429*ph1to1h2o_h + 20.975 ELSE IF taxord == "oxisols" THEN 0.499*cec7_l + 1.679*ocl - 2.055*ph1to1h2o_h + 16.422 ELSE IF taxord == "andisols" THEN 0.763*cec7_l - 4.328*ph1to1h2o_h + 28.591 ELSE 1/0 ELSE IF NOT ISNULL(ecec_l) THEN IF taxord == "histosols" AND NOT ISNULL(taxord) AND (hzname matches "*O*" OR texture matches "*MPT*" OR texture matches "*MUCK*" OR texture matches "*PEAT*") THEN 0.471*ocl - 20.556*ph01mcacl2_h + 142.732 ELSE IF (hzname matches "*O*" AND NOT ISNULL(hzname)) OR ((texture matches "*HPM*" OR texture matches "*MPM*" OR texture matches "*SPM*" OR texture matches "*MPT*" OR texture matches "*MUCK*" OR texture matches "*PEAT*") AND NOT ISNULL(texture)) THEN 1.03*ocl - 19.587*ph01mcacl2_h + 110.208 ELSE IF taxord == "histosols" THEN 2.717*ocl + 9.247 ELSE IF taxord == "gelisols" THEN 0.965*ocl - 4.503*ph1to1h2o_h + 35.377 ELSE IF taxord == "entisols" THEN 0.147*claytotal_l + 2.7*ocl - 1.484*ph1to1h2o_h + 9.572 ELSE IF taxord == "mollisols" THEN 0.148*claytotal_l + 1.692*ocl - 2.411*ph1to1h2o_h + 15.606 ELSE IF taxord == "alfisols" THEN 0.188*claytotal_l + 2.353*ocl - 4.612*ph1to1h2o_h + 25.601 ELSE IF taxord == "aridisols" THEN 0.033*claytotal_l + 2.392*ocl + 2.391*ph1to1h2o_l - 9.935 ELSE IF taxord == "ultisols" THEN 0.899*ecec_l + 0.111*claytotal_l + 2.438*ocl - 1.254*ph1to1h2o_h + 6.046 ELSE IF taxord == "inceptisols" THEN 0.429*ecec_l + 0.078*claytotal_l + 3.052*ocl - 2.053*ph1to1h2o_h + 15.165 ELSE IF taxord == "vertisols" THEN 0.157*claytotal_l + 2.437*ocl - 2.949*ph1to1h2o_h + 15.531 ELSE IF taxord == "spodosols" THEN 1.581*ecec_l + 3.054*ocl + 6.68 ELSE IF taxord == "oxisols" THEN 0.342*ecec_l + 0.078*claytotal_l + 3.176*ocl + 3.932 ELSE IF taxord == "andisols" THEN 0.879*ocl - 12.847*ph1to1h2o_h + 96.871 ELSE 1/0 ELSE 1/0.

ASSIGN acidl IF acidl < 0 AND NOT ISNULL(acidl) THEN 0 ELSE acidl.

#-----Calculate high extractable acidity----------

DEFINE acidh IF NOT ISNULL(cec7_h) AND (ph1to1h2o_r >= 5.5 OR ph01mcacl2_r >= 5.5) THEN IF (hzname matches "*O*" OR texture matches "*MPT*" OR texture matches "*MUCK*" OR texture matches "*PEAT*" OR texture matches "*HPM*" OR texture matches "*MPM*" OR texture matches "*SPM*") AND ph1to1h2o_r > 6.1 THEN 0.19*cec7_h - 11.411*ph1to1h2o_l + 78.341 ELSE IF taxord == "histosols" AND NOT ISNULL(taxord) AND (hzname matches "*O*" OR texture matches "*MPT*" OR texture matches "*MUCK*" OR texture matches "*PEAT*") THEN 0.289*cec7_h + 0.358*och - 26.390*ph01mcacl2_l + 149.662 ELSE IF (hzname matches "*O*" AND NOT ISNULL(hzname)) OR ((texture matches "*HPM*" OR texture matches "*MPM*" OR texture matches "*SPM*" OR texture matches "*MPT*" OR texture matches "*MUCK*" OR texture matches "*PEAT*") AND NOT ISNULL(texture)) THEN 0.470*cec7_h + 0.298*och - 19.702*ph01mcacl2_l + 100.585 ELSE IF texture matches "*HYDR*" AND NOT ISNULL(texture) THEN -0.312*cec7_l + 3.726*och - 20.442*ph1to1h2o_l + 159.093 ELSE IF texture matches "*MEDL*" AND NOT ISNULL(texture) THEN 0.564*cec7_h - 0.326*claytotal_l - 8.825*ph1to1h2o_l + 75.799 ELSE IF texture matches "*ASHY*" AND NOT ISNULL(texture) THEN 0.134*cec7_h + 2.669*och - 1.972*ph1to1h2o_l + 14.051 ELSE IF taxord == "histosols" THEN 0.673*cec7_h - 7.659*ph1to1h2o_l + 44.466 ELSE IF taxord == "gelisols" THEN 0.36*cec7_h - 4.301*ph1to1h2o_l + 31.87 ELSE IF taxord == "entisols" THEN 0.148*cec7_h + 1.679*och - 1.791*ph1to1h2o_l + 12.254 ELSE IF taxord == "mollisols" THEN 0.112*cec7_h + 0.595*och - 2.745*ph1to1h2o_l + 19.964 ELSE IF taxord == "alfisols" THEN 0.205*cec7_h + 1.113*och - 2.928*ph1to1h2o_l + 19.545 ELSE IF taxord == "aridisols" THEN 0.047*cec7_h + 0.535*och - 0.973*ph1to1h2o_l + 7.735 ELSE IF taxord == "ultisols" THEN

```
0.850*cec7_h + 0.361*och - 2.125*ph1to1h2o_l + 11.741 ELSE IF taxord == "inceptisols" THEN
0.496*cec7_h + 0.698*och - 5.010*ph1to1h2o_l + 31.299 ELSE IF taxord == "vertisols" THEN
0.061*cec7_h + 0.775*och -3.557*ph1to1h2o_l + 26.936 ELSE IF taxord == "spodosols" THEN
1.226*cec7_h - 0.524*ocl - 3.429*ph1to1h2o_l + 20.975 ELSE IF taxord == "oxisols" THEN
0.499*cec7_h + 1.679*och - 2.055*ph1to1h2o_l + 16.422 ELSE IF taxord == "andisols" THEN
0.763*cec7_h - 4.328*ph1to1h2o_l + 28.591 ELSE 1/0 ELSE IF NOT ISNULL(ecec_h) THEN IF
taxord == "histosols" AND NOT ISNULL(taxord) AND (hzname matches "*O*" OR texture matches
"*MPT*" OR texture matches "*MUCK*" OR texture matches "*PEAT*") THEN 0.471*och -
20.556*ph01mcacl2_l + 142.732 ELSE IF (hzname matches "*O*" AND NOT ISNULL(hzname)) OR
((texture matches "*HPM*" OR texture matches "*MPM*" OR texture matches "*SPM*" OR texture
matches "*MPT*" OR texture matches "*MUCK*" OR texture matches "*PEAT*") AND NOT
ISNULL(texture)) THEN 1.03*och - 19.587*ph01mcacl2_l + 110.208 ELSE IF taxord == "histosols"
THEN 2.717*och + 9.247 ELSE IF taxord == "gelisols" THEN 0.965*och - 4.503*ph1to1h2o_l + 35.377
ELSE IF taxord == "entisols" THEN 0.147*claytotal_h + 2.7*och - 1.484*ph1to1h2o_l + 9.572 ELSE IF
taxord == "mollisols" THEN 0.148*claytotal_h + 1.692*och - 2.411*ph1to1h2o_l + 15.606 ELSE IF
taxord == "alfisols" THEN 0.188*claytotal_h + 2.353*och - 4.612*ph1to1h2o_l + 25.601 ELSE IF taxord
== "aridisols" THEN 0.033*claytotal_h + 2.392*och + 2.391*ph1to1h2o_h - 9.935 ELSE IF taxord ==
"ultisols" THEN 0.899*ecec_h + 0.111*claytotal_h + 2.438*och - 1.254*ph1to1h2o_l + 6.046 ELSE IF
taxord == "inceptisols" THEN 0.429*ecec_h + 0.078*claytotal_h + 3.052*och - 2.053*ph1to1h2o_l +
15.165 ELSE IF taxord == "vertisols" THEN 0.157*claytotal_h + 2.437*och - 2.949*ph1to1h2o_l +
15.531 ELSE IF taxord == "spodosols" THEN 1.581*ecec_h + 3.054*och + 6.68 ELSE IF taxord ==
"oxisols" THEN 0.342*ecec_h + 0.078*claytotal_h + 3.176*och + 3.932 ELSE IF taxord == "andisols"
THEN 0.879*och - 12.847*ph1to1h2o_l + 96.871 ELSE 1/0 ELSE 1/0.


ASSIGN acidh IF acidh < 0 AND NOT ISNULL(acidh) THEN 0 ELSE acidh.


# Calcareous soils have little or no acidity. A pH value of 8.2 approximates the pH of calcareous soils.
# A check to make sure that zero extractable acidity is included in the acidity range when pH is >= 8.3
ASSIGN acidr    IF ph1to1h2o_l >= 8.3 THEN 0 ELSE acidr.
ASSIGN acidl    IF acidr == 0 THEN 0 ELSE acidl.
ASSIGN acidl    IF ph1to1h2o_r >= 8.3 THEN 0 ELSE acidl.
```

618.102 NASIS Calculation for Estimating Liquid Limit and Plasticity Index

Definition.—This calculation computes the Atterberg Limits (liquid limit and plasticity index). The low, rv, and high are calculated.

The calculation works on all records (horizons) in your selected set that you have permission to edit, except as described in (7) below. For some horizons, such as bedrock or cemented layers, it may not be appropriate to calculate Atterberg limits. You may wish to tailor your selected set accordingly.

There is a companion report available to preview results of this calculation. The calculation script is imbedded in the report script. The report is designed to display your current stored LL and PI values alongside the calculated values. Viewing the results in this fashion might be useful in determining whether or not you wish to run the calculation on your selected set. The name of the Pangaea report is "UTIL - Comparison of LL and PI, stored vs calculated."

Caution: These calculations for liquid limit and plasticity index may produce poor estimates for Andisols and Spodosols.

Inputs.—This calculation requires that the following data must be populated:

organic matter percent (l,rv,h)
linear extensibility percent (l,rv,h)
clay total separate (l,rv,h)
clay sized carbonate (l,rv,h)

Guidelines For Implementing Equations for LL and PI In NASIS

1) Values for LL and PI (low, high, and rv) are computed.
2) The calculations are based on the noncarbonate clay fraction.
3) If clay sized carbonate is null, then noncarbonate clay = total clay.
4) The water 15bar (volumetric) values from the database are not used. Instead water 15-bar value is estimated on a gravimetric basis using total clay and organic matter values.
5) If low and/or high values for LEP, clay sized carbonate, or OM are null, set to zero and proceed with estimate (reduced accuracy is < 1.5%).
6) If rv values for these input variables are null, compute as the average of low and high values (L + H/2) and proceed with the calculation.
7) If OM > 25% or total clay is null, then LL and PI are not calculated.
8) The PI is estimated first, then LL.
9) If PI equals 0, LL rv and low values are set to 0 and the LL high value is set to 14.
10) If LL is < 15, then LL rv and low values are set to 0 and LL high value is set to 14.
11) Computed values for LL and PI are converted to nearest whole number.

Calculation.

```
# Use zero if inputs are null (l).
DEFINE oml          IF ISNULL(om_l) THEN 0 ELSE om_l.
DEFINE lepl         IF ISNULL(lep_l) THEN 0 ELSE lep_l.
DEFINE claytotall          IF ISNULL(claytotal_l) THEN 0 ELSE claytotal_l.
DEFINE claysizedcarbl      IF ISNULL(claysizedcarb_l) THEN 0 ELSE claysizedcarb_l.
DEFINE ncclayl          claytotall - claysizedcarbl.
```

```
# Calculate the 15 bar water content (low) on a gravimetric basis.
# Assume ratio of 1500KPa to Clay percent is 0.4
DEFINE F INITIAL 0.4.
DEFINE wfifteenbarl (claytotall * (1 - oml/100) * F + oml).

# Calculate the low assuming all inputs are in range.
DEFINE pi_l    -1.86 + 0.69*wfifteenbarl - 0.69*oml + 0.13*lepl + 0.47*ncclayl.
DEFINE ll_l    11.6 + 1.49*wfifteenbarl + 0.78*oml + 0.6*lepl + 0.26*ncclayl.


# Use zero if inputs are null (h).
DEFINE omh         IF ISNULL(om_h) THEN 0 ELSE om_h.
DEFINE leph        IF ISNULL(lep_h) THEN 0 ELSE lep_h.
DEFINE claytotalh    IF ISNULL(claytotal_h) THEN 0 ELSE claytotal_h.
DEFINE claysizedcarbh    IF ISNULL(claysizedcarb_h) THEN 0 ELSE claysizedcarb_h.
DEFINE ncclayh         claytotalh - claysizedcarbh.

# Calculate the 15 bar water content (high) on a gravimetric basis.
# Assume ratio of 1500KPa to Clay percent is 0.4
# DEFINE F INITIAL 0.4 was done above.
DEFINE wfifteenbarh (claytotalh * (1 - omh/100) * F + omh).

# Calculate the high assuming all inputs are in range.
DEFINE pi_h    -1.86 + 0.69*wfifteenbarh - 0.69*omh + 0.13*leph + 0.47*ncclayh.
DEFINE ll_h    11.6 + 1.49*wfifteenbarh + 0.78*omh + 0.6*leph + 0.26*ncclayh.


# Use (low + high)/2 if inputs are null (rv).
DEFINE om    IF ISNULL(om_r) THEN (oml + omh)/2 ELSE om_r.
DEFINE lep    IF ISNULL(lep_r) THEN (lepl + leph)/2 ELSE lep_r.
DEFINE claytotal    IF ISNULL(claytotal_r) THEN (claytotall + claytotalh)/2 ELSE claytotal_r.
DEFINE claysizedcarb    IF ISNULL(claysizedcarb_r) THEN (claysizedcarbl + claysizedcarbh)/2 ELSE
claysizedcarb_r.
DEFINE ncclay         claytotal - claysizedcarb.

# Calculate the 15 bar water content (rv) on a gravimetric basis.
# Assume ratio of 1500KPa to Clay percent is 0.4
# DEFINE F INITIAL 0.4 was done above.
DEFINE wfifteenbar (claytotal * (1 - om/100) * F + om).

# Calculate the rv assuming all inputs are in range.
DEFINE pi_r    -1.86 + 0.69*wfifteenbar - 0.69*om + 0.13*lep + 0.47*ncclay.
DEFINE ll_r    11.6 + 1.49*wfifteenbar + 0.78*om + 0.6*lep + 0.26*ncclay.


# Check for inputs out of range and set results to null.

ASSIGN pi_r    IF ISNULL(claytotal_r) OR om > 25 OR ncclay < 0 THEN 1/0 ELSE pi_r.
ASSIGN ll_r    IF ISNULL(claytotal_r) OR om > 25 OR ncclay < 0 THEN 1/0 ELSE ll_r.
```

```
ASSIGN pi_l    IF ISNULL(claytotal_l) OR oml > 25 OR ncclayl < 0 THEN 1/0 ELSE pi_l.
ASSIGN ll_l    IF ISNULL(claytotal_l) OR oml > 25 OR ncclayl < 0 THEN 1/0 ELSE ll_l.
ASSIGN pi_h    IF ISNULL(claytotal_h) OR omh > 25 OR ncclayh < 0 THEN 1/0 ELSE pi_h.
ASSIGN ll_h    IF ISNULL(claytotal_h) OR omh > 25 OR ncclayh < 0 THEN 1/0 ELSE ll_h.

# If calculated PI is negative, set both PI and LL to zero.

ASSIGN pi_r    IF NOT ISNULL(pi_r) AND pi_r < 0 THEN 0 ELSE pi_r.
ASSIGN ll_r    IF ISNULL(pi_r) THEN 1/0 ELSE IF pi_r < 0.5 OR (NOT ISNULL(ll_r) AND ll_r < 15)
THEN 0 ELSE ll_r.
ASSIGN pi_l    IF NOT ISNULL(pi_l) AND pi_l < 0 THEN 0 ELSE pi_l.
ASSIGN ll_l    IF ISNULL(pi_l) THEN 1/0 ELSE IF pi_l < 0.5 OR (NOT ISNULL(ll_l) AND ll_l < 15)
THEN 0 ELSE ll_l.
ASSIGN pi_h    IF NOT ISNULL(pi_h) AND pi_h < 0 THEN 0 ELSE pi_h.
ASSIGN ll_h    IF ISNULL(pi_h) THEN 1/0 ELSE IF pi_h < 0.5 OR (NOT ISNULL(ll_h) AND ll_h <
15) THEN 14 ELSE ll_h.

#Set results to interger values.

ASSIGN pi_r    ROUND(pi_r).
ASSIGN ll_r    ROUND(ll_r).
ASSIGN pi_l    ROUND(pi_l).
ASSIGN ll_l    ROUND(ll_l).
ASSIGN pi_h    ROUND(pi_h).
ASSIGN ll_h    ROUND(ll_h).
```

618.103 NASIS Calculation for Estimating Particle Size

Definition.—This calculation computes representative values for the sand fractions, total sand, and total silt. The following rules apply:

1) The results will be blank if needed data are not entered. Total clay and texture are always required, and particle size class is required for textures CL, L, SCL, SICL, and SIL.
2) When a horizon has multiple textures, the one marked rv is used or the first texture is used if there is no rv. No results are calculated for stratified textures at this time.
3) If total sand (rv) has been entered, the sand fractions will be adjusted so their sum equals the specified total. If you want to calculate a new sand total, you must erase the old one before running the calculation.

Inputs.—This calculation requires that the following data must be populated:

texture
clay total separate (l,rv,h)
taxonomic particle-size class

Calculation.

```
ASSIGN texcl IF ISNULL(texcl) OR stratextsflag==1
        THEN "null" ELSE CODENAME(texcl).

DEFINE sandclass
     IF (texcl=="sl" or texcl=="cosl" or texcl=="fsl" or texcl=="vfsl") THEN
       IF ISNULL(sandtotal_r) THEN 1 ELSE
       IF sandtotal_r > 60 THEN 1 ELSE
       IF sandtotal_r >= 53 THEN 2 ELSE 3
     ELSE IF (texcl=="cl" or texcl=="l" or texcl=="scl" or texcl=="sicl" or
       texcl=="sil") THEN family_sandclass
     ELSE 0.

DEFINE paramid_by_tex LOOKUP(1, texcl==texture and
            (sandcode==0 or sandcode==sandclass),
            paramid).
DEFINE claypct_by_tex LOOKUP(1, texcl==texture and
            (sandcode==0 or sandcode==sandclass),
            claypct).
DEFINE claydiff_by_tex ABS(claypct_by_tex - claytotal_r).
DEFINE closest_clay ARRAYMIN(claydiff_by_tex).
DEFINE select_row ARRAYMIN(LOOKUP(closest_clay, claydiff_by_tex,
            paramid_by_tex)).

# Get the equation number and coefficients from the selected parameter row.

DEFINE eqn lookup(select_row, paramid, equation).
DEFINE p1 lookup(select_row, paramid, param1).
DEFINE p2 lookup(select_row, paramid, param2).
DEFINE p3 lookup(select_row, paramid, param3).
DEFINE p4 lookup(select_row, paramid, param4).
DEFINE p5 lookup(select_row, paramid, param5).

# Compute all the distributions. We compute all 5 equations first then
# pick the right result, because this language doesn't have conditional
```

```
# execution.
# Start by computing some things that are used more than once.

DEFINE diamclay LOOKUP("clay", psclass, psdiam). # Upper clay diameter.
DEFINE cr2    POW(2, 1/3).            # Cube root of 2.
DEFINE crdiam POW(psdiam, 1/3).       # Cube root of psclass diameter.
DEFINE crdiamclay POW(diamclay, 1/3). # Cube root of clay diam.
DEFINE sqr2   SQRT(2).                # Square root of 2.
DEFINE sqrdiam SQRT(psdiam).          # Square root of psclass diameter.

DEFINE eq1tmp POW(1 + p4*POW(cr2-crdiam, p3), p2).
DEFINE eq2tmp EXP(p3 * POW(cr2-crdiam, p2)).
DEFINE eq3tmp EXP(1/POW(p4*cr2,p3) - 1/POW(p4*crdiam,p3)).
DEFINE eq4tmp p3*(POW(crdiam,2) - POW(cr2,2)) + p4*(psdiam-2) +
          p5*(POW(crdiam,4) - POW(cr2,4)).
DEFINE eq5tmp p3*(1/psdiam - 1/2) + p4*(sqrdiam - sqr2).

# Next adjust the parameters to make the clay come out the same as the input.

DEFINE tmp  LOOKUP("clay", psclass, eq1tmp).
DEFINE eq1p1 (tmp * claytotal_r - 100) / (tmp - 1).

ASSIGN tmp LOOKUP("clay", psclass, eq2tmp).
DEFINE eq2p1 (tmp * claytotal_r - 100) / (tmp - 1).

ASSIGN tmp LOOKUP("clay", psclass, eq3tmp).
DEFINE eq3p2 (claytotal_r - 100 * tmp) / (1 - tmp).

ASSIGN tmp  LOOKUP("clay", psclass, eq4tmp).
DEFINE eq4p2 (claytotal_r - 100 - tmp) / (crdiamclay - cr2).

ASSIGN tmp LOOKUP("clay", psclass, eq5tmp).
DEFINE eq5p2 (claytotal_r - 100 - tmp) / (diamclay - 2).

# Compute the five equations for all particle size classes.

DEFINE eq1 eq1p1 + (100-eq1p1) / eq1tmp.
DEFINE eq2 eq2p1 + (100-eq2p1) / eq2tmp.
DEFINE eq3 eq3p2 + (100-eq3p2) * eq3tmp.
DEFINE eq4 100 + eq4p2*(crdiam - cr2) + eq4tmp.
DEFINE eq5 100 + eq5p2*(psdiam - 2) + eq5tmp.

# Select the right equation. The variable psd will have 7 vaules, one
# for each particle size class. The value for each class is picked out
# of the array with a LOOKUP.

DEFINE psd IF eqn==1 THEN eq1 ELSE
        IF eqn==2 THEN eq2 ELSE
        IF eqn==3 THEN eq3 ELSE
        IF eqn==4 THEN eq4 ELSE
        IF eqn==5 THEN eq5 ELSE
        eq5/0.          # sets psd to 7 nulls when texcl is null
```

```
# Pick out the cumulative percents then compute the individual fractions.

DEFINE clay LOOKUP("clay", psclass, psd).
DEFINE silt LOOKUP("silt", psclass, psd).
DEFINE vfs LOOKUP("vfs", psclass, psd).
DEFINE fs   LOOKUP("fs", psclass, psd).
DEFINE ms LOOKUP("ms", psclass, psd).
DEFINE cs LOOKUP("cos", psclass, psd).
DEFINE vcs  LOOKUP("vcos", psclass, psd).

ASSIGN vcs  vcs - cs.
ASSIGN cs   cs - ms.
ASSIGN ms   ms - fs.
ASSIGN fs   fs - vfs.
ASSIGN vfs  vfs - silt.
ASSIGN silt silt - clay.

DEFINE sand vfs + fs + ms + cs + vcs.

# Find an adjustment factor for the sand fractions.
# If total sand was given, adjust each sand fraction by the ratio needed to
# make the sum equal to the given total.
# If total sand was not given, verify that the sand and silt are within the
# texture class limits and if not adjust them by the appropriate ratio.

DEFINE sand_diff IF ISNULL (sandtotal_r) THEN
          IF (texcl=="cos"  or texcl=="s"  or texcl=="fs"  or texcl=="vfs")
            and ((clay + silt) > (15 - .5*clay))
          THEN (clay + silt) - (15 - .5*clay) ELSE
          IF (texcl=="lcos"  or texcl=="ls"  or texcl=="lfs"  or texcl=="lvfs")
            and (clay + silt) > (30 - clay)
          THEN (clay + silt) - (30 - clay)    ELSE
          IF texcl=="sil"  and silt < 50
          THEN silt - 50                  ELSE
          IF texcl=="sicl"  and sand > 20
          THEN 20 - sand                    ELSE
          IF texcl=="sc"  and sand < 45
          THEN 45 - sand                    ELSE
          IF texcl=="sic"  and silt < 40
          THEN silt - 40                  ELSE 0
        ELSE
          sandtotal_r - sand.
DEFINE adj (sand + sand_diff) / sand.

# Adjust the sands and silt by the adjustment factor.
# Round to one decimal place before computing total sand to avoid roundoff error.

ASSIGN vfs ROUND(vfs * adj, 1).
ASSIGN fs    ROUND(fs  * adj, 1).
ASSIGN ms    ROUND(ms  * adj, 1).
ASSIGN cs    ROUND(cs  * adj, 1).
```

```
ASSIGN vcs ROUND(vcs * adj, 1).
ASSIGN sand  vfs + fs + ms + cs + vcs.

# The rounding may result in a sum that does not equal the target sandtotal,
# so another adjustment has to be made.
# This time, apply it to the first non-zero fraction.

ASSIGN adj   IF NOT ISNULL(sandtotal_r)  THEN sandtotal_r - sand ELSE 0.

ASSIGN vcs IF vfs==0 AND fs==0 AND ms==0 AND cs==0 THEN vcs + adj ELSE vcs.
ASSIGN cs    IF vfs==0 AND fs==0 AND ms==0 AND cs>0 THEN cs + adj ELSE cs.
ASSIGN ms    IF vfs==0 AND fs==0 AND ms>0 THEN ms + adj ELSE ms.
ASSIGN fs    IF vfs==0 AND fs>0 THEN fs + adj ELSE fs.
ASSIGN vfs IF vfs > 0 THEN vfs + adj ELSE vfs.
ASSIGN sand  vfs + fs + ms + cs + vcs.

ASSIGN silt  100 - sand - clay.

# When vcos is < 0.
ASSIGN cs    IF vcs < 0 AND vfs+fs+ms+cs > sand THEN cs - ((vfs+fs+ms+cs)  - sand) ELSE cs.
ASSIGN vcs   IF vcs < 0 THEN 0 ELSE vcs.

# Store the results as RV values for the horizon.

SET sandtotal_r from sand,
   sandvc_r from vcs,
   sandco_r from cs,
   sandmed_r from ms,
   sandfine_r from fs,
   sandvf_r from vfs,
   silttotal_r  from silt.
```

618.104 NASIS Calculation for Estimating Rock Fragments and Percent Passing Sieves

Definition.—This calculation computes the percent soil material (< 3 inch basis) passing the #4 (4.7 mm), #10 (2.0 mm), #40 (0.42 mm), and #200 sieves (0.074 mm) and the percent rock fragments 3 to 10 inches and > 10 inches (whole soil basis).

1. Percent passing sieves are on a < 3-inch basis and rock fragments are a whole-soil basis.
2. The calculation of percent passing sieves and rock fragments excludes pararock fragments, wood, and noncemented fragments. Pararock fragments are defined by fragment hardness of extremely weakly, very weakly, weakly, or moderately cemented.
3. If fragment hardness is not populated, "indurated" is assumed.
4. If fragment kind is not populated, a fragment density of 2.65 g cm-3 is assumed.
5. Fragment density is assigned based on the fragment kind (table 2). If an average density for each fragment kind is not available, a default density of 2.65 g cm-3 is used.
6. If only low and high values are populated for fragment volume, fragment size, total sand, total silt, total clay, or sand separates, then rv's are generated from the high and low values (takes the average).
7. The low and high values must be populated for fragment size, otherwise the calculation will produce incorrect or no results.
8. Low and high values for percent passing #40 and #200 sieves are based on the average low and high values for the particle-size separates.
9. If low and high values are not populated for total sand, total silt, or total clay or for the sand separates, then low and high values for the #40 and #200 sieves are generated from the low and high values of total clay. If high and low values for total clay are also null, then nulls are returned.
10. Low and high values for percent passing #4 and #10 sieves and the rock fragments are based on the low and high fragment volumes (in "Horizon Table," if populated). If low and high fragment volumes (in "Horizon Table") are not populated, then total low and high fragment volumes in the "Horizon Fragment Table" are used. If low and high fragment volumes in the "Horizon Fragment Table" are null, then nulls are returned.
11. **Caution:** If percent passing sieves are populated and only clay is populated (l, rv, h) in the particle-size separates, the calculation will wipe out the calculated values and put in null values. If there is not enough data to run the calculation, nulls are returned.
12. For stratified textures, if data is populated, the calculation proceeds as normal. If the particle-size separates are not populated, then the #40 and #200 sieves are not calculated (nulls are returned).
13. If the organic matter content > 35%, then percent passing sieves is not calculated and only rock fragments (3 to 10 in and >10 in) are calculated.
14. **Caution:** If 1/3-bar bulk density rv is not populated, null values for all sieves and rock fragments are returned.
15. The calculation rounds all sieve values to the nearest whole number.

Limitations

1) The pararock fragments are not included in the calculation because they can be crushed to < 2 mm. It is assumed that the pararock fragments, when crushed, will reflect the existing particle-size distribution. If there are pararock fragments that when crushed produce a different particle-size distribution, the calculation will over- or under-estimate the percent passing the #40 and #200 sieves.
2) When actually measuring percent passing sieves, the organic matter is not removed. The calculation of percent passing sieves calculates using organic matter free particle-size fractions. The calculation does not take into account the distribution of organic matter particles.

3) The fragment densities applied here are average values from the literature and may not represent the true density of fragments in your area. Fragment densities can be highly variable from location to location for a fragment kind.
4) If the total fragment volumes in the "Component Horizon Table" are not populated, then the low and high calculated values are based on the "Horizon Fragment Table" volumes and texture ranges in the NASIS database; these values may not reflect the actual percent passing sieves and rock fragment ranges. It is assumed that the total of the lows and the total of the highs in the "Horizon Fragment Table" equal the total high and low fragment volumes.
5) If the low and high calculated values are based on the total fragment volumes in the "Component Horizon Table" because actual fragment kind distributions (e.g., rock vs. pararock) that make up the fragment volume totals are not known, percent passing sieves and rock fragment may not reflect actual ranges. It computes representative values for the sand fractions, total sand and total silt. The following rules apply:
    a. The results will be blank if needed data are not entered. Total clay and texture are always required, and particle size class is required for textures CL, L, SCL, SICL, and SIL.
    b. When a horizon has multiple textures the one marked rv is used; the first texture is used if there is no rv. No results are calculated for stratified textures at this time.
    c. If total sand (rv) has been entered, the sand fractions will be adjusted so their sum equals the specified total. If you want to calculate a new sand total you must erase the old one before running the calculation.

Inputs.—This calculation requires the following data to be populated:

fragment volume total (high, low) in "Horizon Table"
fragment volume (high, low, and rv) in "Horizon Fragment Table"
fragment kind in "Horizon Fragment Table"
fragment size (high, low, and rv) in "Horizon Fragment Table"
fragment hardness in "Horizon Fragment Table"
total sand (high, low, and rv)
total clay (high, low, and rv)
total silt (high, low, and rv)
very fine sand (high, low, and rv)
fine sand (high, low, and rv)
medium sand (high, low, and rv)
coarse sand (high, low, and rv)
very coarse sand (high, low, and rv)
one-third bar bulk density (rv)
organic matter (rv)

Calculation

```
DEFINE curvenum_l 0.56559. #run curve fitting routine
DEFINE curvenum_h 0.56559. #run curve fitting routine
DEFINE curvenum_r  0.56559. #run curve fitting routine

DEFINE densityrock
IF fragkind2 == "`a`a lava" THEN 2.00
ELSE IF fragkind2 == "amphibolite" THEN 2.99
ELSE IF fragkind2 == "andesite" THEN 2.65
ELSE IF fragkind2 == "anorthosite" THEN 2.73
ELSE IF fragkind2 == "basalt" THEN 2.69
ELSE IF fragkind2 == "calcrete (caliche)" THEN 1.44
ELSE IF fragkind2 == "chalk" THEN 2.35
ELSE IF fragkind2 == "charcoal" THEN 0.45
```

```
ELSE IF fragkind2 == "chert" THEN 2.76
ELSE IF fragkind2 == "cinders" THEN 1.45
ELSE IF fragkind2 == "coal" THEN 1.6
ELSE IF fragkind2 == "dacite" THEN 1.67
ELSE IF fragkind2 == "diabase" THEN 2.92
ELSE IF fragkind2 == "diorite" THEN 2.83
ELSE IF fragkind2 == "dolomite (dolostone)" THEN 2.79
ELSE IF fragkind2 == "gabbro" THEN 2.99
ELSE IF fragkind2 == "gibbsite concretions" THEN 2.35
ELSE IF fragkind2 == "gneiss" THEN 2.79
ELSE IF fragkind2 == "granite" THEN 2.66
ELSE IF fragkind2 == "granodiorite" THEN 2.72
ELSE IF fragkind2 == "granulite" THEN 2.91
ELSE IF fragkind2 == "graywacke" THEN 2.69
ELSE IF fragkind2 == "gypsum, rock" THEN 2.55
ELSE IF fragkind2 == "ironstone concretions" THEN 2.93
ELSE IF fragkind2 == "limestone, unspecified" THEN 2.61
ELSE IF fragkind2 == "marble" THEN 2.74
ELSE IF fragkind2 == "monzonite" THEN 2.8
ELSE IF fragkind2 == "obsidian" THEN 2.37
ELSE IF fragkind2 == "orthoquartzite" THEN 2.41
ELSE IF fragkind2 == "peridotite" THEN 3.22
ELSE IF fragkind2 == "petroferric fragments" THEN 2.93
ELSE IF fragkind2 == "phyllite" THEN 2.74
ELSE IF fragkind2 == "pumice" THEN 0.98
ELSE IF fragkind2 == "pyroxenite" THEN 3.28
ELSE IF fragkind2 == "quartz-diorite" THEN 2.79
ELSE IF fragkind2 == "quartzite" THEN 2.7
ELSE IF fragkind2 == "rhyolite" THEN 2.51
ELSE IF fragkind2 == "sandstone, calcareous" THEN 2.03
ELSE IF fragkind2 == "sandstone, unspecified" THEN 2.29
ELSE IF fragkind2 == "schist, mica" THEN 2.76
ELSE IF fragkind2 == "schist, unspecified" THEN 2.84
ELSE IF fragkind2 == "serpentinite" THEN 2.63
ELSE IF fragkind2 == "shale, calcareous" THEN 2.67
ELSE IF fragkind2 == "shale, clayey" THEN 2.78
ELSE IF fragkind2 == "shale, unspecified" THEN 2.6
ELSE IF fragkind2 == "slate" THEN 2.81
ELSE IF fragkind2 == "soapstone" THEN 2.7
ELSE IF fragkind2 == "syenite" THEN 2.74
ELSE IF fragkind2 == "tonalite" THEN 2.67
ELSE IF fragkind2 == "trachyte" THEN 2.57
ELSE IF fragkind2 == "tuff, unspecified" THEN 1.84
ELSE IF fragkind2 == "wood" THEN 0.6
ELSE 2.65.

#-----------------------------------------------------------------------
# Start of percent passing sieves and rock fragments calculation for RV.
#-----------------------------------------------------------------------

# Compute total volume percent of rock fragments, minus pararocks,
# on a whole soil basis.
```

```
DEFINE fragvols    IF (fraghard2=="strongly" or fraghard2=="very strongly" or

fraghard2=="indurated" or ISNULL(fraghard2)) AND fragkind2 != "wood"

THEN fragvlr ELSE 0.
DEFINE fragvolr    ARRAYSUM(fragvols).

# Compute percent volume of rock fragments that are < 75mm for each row
# in the rock fragment table.

DEFINE rockfrag_row_r    IF (75 >= fragsize_l and 75 <= fragsize_r)
THEN ((75-fragsize_l)/(fragsize_r-fragsize_l)/2*fragvlr)
ELSE IF (75 > fragsize_r and 75 <= fragsize_h)
THEN (((75-fragsize_r)/(fragsize_h-fragsize_r)/2)+0.5)*fragvlr
ELSE IF 75 > fragsize_h THEN fragvlr ELSE 0.

# Compute total volume percent of rock fragments that are < 75mm on a
# whole soil basis, minus pararocks.

DEFINE totalRFvol_s    IF (fraghard2=="strongly" or fraghard2=="very strongly" or
fraghard2=="indurated" or ISNULL(fraghard2)) AND fragkind2 != "wood"
THEN rockfrag_row_r ELSE 0.

DEFINE totalRFvol_r    ARRAYSUM(totalRFvol_s).

# Compute volume percent of rock fragments that are < 75mm on a < 75mm basis.

DEFINE vol75mm_r    totalRFvol_r/(1-(fragvolr-totalRFvol_r)/100).

# Compute volume percent of rock fragments that are < 75mm
# to a weight percent for each row in the rock fragment table.

DEFINE wtRF_row_r    IF totalRFvol_r == 0 THEN 0 ELSE
(rockfrag_row_r/totalRFvol_r*vol75mm_r)*densityrock/(((vol75mm_r/100)*densityrock)+((1-
vol75mm_r/100)*dbthirdbar_r)).

# Compute volume percent of rock fragments that are < 5mm to a weight percent for each row in the rock
fragment table.

DEFINE wtRF5mm_row_r    IF 5>=fragsize_h THEN wtRF_row_r
ELSE IF (5>=fragsize_l and 5<=fragsize_r)
THEN (5-fragsize_l)/(fragsize_r-fragsize_l)/2*wtRF_row_r
ELSE IF (5>fragsize_r and 5<=fragsize_h)
THEN (((5-fragsize_r)/(fragsize_h-fragsize_r)/2)+0.5)*wtRF_row_r
ELSE 0.


# Compute weight percent of rock fragments of the whole soil for each row.

DEFINE wtRFwhole_row_r    fragvlr*densityrock/(((fragvlr/100)*densityrock)+((1-
fragvlr/100)*dbthirdbar_r)).
```

```
# Compute weight percent of rock fragments > 75mm of the whole soil for each row.

DEFINE wtRFwhole75mm_row_s  IF (75>=fragsize_l and 75<=fragsize_r)
THEN ((1-((75-fragsize_l)/(fragsize_r-fragsize_l)/2))*wtRFwhole_row_r)
ELSE IF (75>fragsize_r  and 75<=fragsize_h)
THEN ((1-(((75-fragsize_r)/(fragsize_h-fragsize_r)/2)+0.5))*wtRFwhole_row_r)
ELSE IF 75>fragsize_h  THEN 0 ELSE wtRFwhole_row_r.

# Compute the percent weight of rock fragments > 75mm of whole soil, minus pararocks.

DEFINE wtRFwhole75mm_row_2  IF ISNULL(fragvlr)  and ISNULL(fragvol_l)  and
ISNULL(fragvol_h)
THEN 0 ELSE IF (fraghard2=="strongly"  or fraghard2=="very  strongly"  or fraghard2=="indurated"  or
ISNULL(fraghard2))  AND  fragkind2  != "wood"
THEN wtRFwhole75mm_row_s  ELSE 0.

DEFINE wtRFwhole75mm_r      ARRAYSUM(wtRFwhole75mm_row_2).

# Compute weight percent of rock fragments >250mm of whole soil for each row.

DEFINE wtRFwhole250mm_row_s  IF (250>=fragsize_l  and 250<=fragsize_r)
THEN ((1-((250-fragsize_l)/(fragsize_r-fragsize_l)/2))*wtRFwhole_row_r)
ELSE IF (250>fragsize_r  and 250<=fragsize_h)
THEN ((1-(((250-fragsize_r)/(fragsize_h-fragsize_r)/2)+0.5))*wtRFwhole_row_r)
ELSE IF fragsize_l>=250  THEN wtRFwhole_row_r  ELSE 0.

# Compute the total weight percent of rock fragments >250mm of whole soil, minus pararocks.

DEFINE wtRFwhole250mm_row_2  IF ISNULL(fragvlr)  and ISNULL(fragvol_l)  and
ISNULL(fragvol_h)
THEN 0 ELSE IF (fraghard2=="strongly"  or fraghard2=="very  strongly"  or fraghard2=="indurated"  or
ISNULL(fraghard2))  AND  fragkind2  != "wood"
THEN wtRFwhole250mm_row_s  ELSE 0.

DEFINE rockfrag_250r         ARRAYSUM(wtRFwhole250mm_row_2).

# Compute total weight percent of rock fragments 75 to 250mm, minus pararocks.

DEFINE rockfrag_75r      wtRFwhole75mm_r  - rockfrag_250r.

# Compute percent passing #10 sieve, minus pararocks.

DEFINE sieve_10s      IF (fraghard2=="strongly"  or fraghard2=="very  strongly"  or
fraghard2=="indurated"  or ISNULL(fraghard2))  AND  fragkind2  != "wood"  THEN wtRF_row_r ELSE 0.

DEFINE sieve_10r      IF ISNULL(ARRAYSUM(fragvlr))
THEN 100 ELSE 100-ARRAYSUM(sieve_10s).

# Compute percent passing #4 sieve, minus pararocks.
```

```
DEFINE sieve_4s     IF (fraghard2=="strongly"  or fraghard2=="very  strongly" or fraghard2=="indurated"
or ISNULL(fraghard2))  AND fragkind2  != "wood"THEN wtRF5mm_row_r ELSE 0.

DEFINE sieve_4r     IF ISNULL(ARRAYSUM(fragvlr))  THEN 100 ELSE
ARRAYSUM(sieve_4s)+sieve_10r.

# Compute percent passing #40 sieve, minus pararocks.

DEFINE sieve_40r     IF ISNULL(ARRAYSUM(fragvlr))  THEN 100 - (vcos_r + cos_r + ms_r*0.2515)
ELSE sieve_10r/100*((0.7485*ms_r)+fs_r+vfs_r+siltr+clayr).

# Compute percent passing #200 sieve, minus pararocks.

DEFINE sieve_200r      IF ISNULL(ARRAYSUM(fragvlr))
THEN IF vfs_r < 15 THEN (vfs_r*0.56559 + siltr + clayr)
ELSE (vfs_r*curvenum_r + siltr + clayr)
ELSE IF vfs_r < 15 THEN (vfs_r*0.56559 + siltr + clayr)*sieve_10r/100
ELSE (vfs_r*curvenum_r + siltr + clayr)*sieve_10r/100.

ASSIGN sieve_10r      IF om_r > 35 OR ISNULL(clayr)  OR ISNULL(dbthirdbar_r)  THEN 1/0 ELSE
sieve_10r.

ASSIGN sieve_4r     IF om_r > 35 OR ISNULL(clayr)  OR ISNULL(dbthirdbar_r)  THEN 1/0 ELSE
sieve_4r.

ASSIGN sieve_40r IF om_r > 35 OR (stratextsflag == 1 AND (ISNULL(clayr) OR ISNULL(siltr) OR
ISNULL(vcos_r) OR ISNULL(cos_r) OR ISNULL(ms_r))) OR ISNULL(dbthirdbar_r) THEN 1/0 ELSE
sieve_40r.

ASSIGN sieve_200r     IF om_r > 35 OR (stratextsflag  == 1 AND (ISNULL(clayr) OR ISNULL(siltr)
OR ISNULL(vfs_r))) OR ISNULL(dbthirdbar_r)  THEN 1/0 ELSE sieve_200r.

ASSIGN rockfrag_250r     IF om_r > 35 AND ISNULL(fragvolr)  THEN 0 ELSE IF (ISNULL(clayr)
AND stratextsflag  != 1) OR ISNULL(dbthirdbar_r)  THEN 1/0 ELSE rockfrag_250r.

ASSIGN rockfrag_75r      IF om_r > 35 AND ISNULL(fragvolr)  THEN 0 ELSE IF (ISNULL(clayr)
AND stratextsflag  != 1) OR ISNULL(dbthirdbar_r)  THEN 1/0 ELSE rockfrag_75r.

#-------------------------------------------------------------------------
# Start of percent passing sieves and rock fragments calculation  of low values.
#-------------------------------------------------------------------------

# Compute total volume percent of rock fragments, minus pararocks,
# on a whole soil basis.

DEFINE fragvolls IF (fraghard2=="strongly"  or fraghard2=="very  strongly" or fraghard2=="indurated"
or ISNULL(fraghard2))  AND fragkind2  != "wood"  THEN fragvol_l ELSE 0.
DEFINE fragvoll     ARRAYSUM(fragvolls).

# Uses the low fragment volume total from the Horizon Table.
DEFINE sumlowfrags      ARRAYSUM(fragvol_l).
```

```
ASSIGN fragvoll      IF (fragvoll / sumlowfrags) < 1 AND fragvoll != 0 AND NOT
ISNULL(sumlowfrags) AND sumlowfrags > 0 THEN (fragvoll / sumlowfrags) * fragvoltot_l ELSE IF
ISNULL(fragvoltot_l) THEN fragvoll ELSE fragvoltot_l.

# Compute volume percent of rock fragments that are < 75mm for each row
# in the rock fragment table.

DEFINE rockfrag_row_l     IF (75 >= fragsize_l and 75 <= fragsize_r)
THEN ((75-fragsize_l)/(fragsize_r-fragsize_l)/2*fragvol_l)
ELSE IF (75 > fragsize_r and 75 <= fragsize_h)
THEN (((75-fragsize_r)/(fragsize_h-fragsize_r)/2)+0.5)*fragvol_l
ELSE IF 75 > fragsize_h THEN fragvol_l ELSE 0.

# Compute total volume percent of rock fragments that are < 75mm on a
# whole soil basis, minus pararocks.

DEFINE totalRFvol_ls      IF (fraghard2=="strongly" or fraghard2=="very strongly" or
fraghard2=="indurated" or ISNULL(fraghard2)) AND fragkind2 != "wood" THEN rockfrag_row_l
ELSE 0.

DEFINE totalRFvol_l     ARRAYSUM(totalRFvol_ls).

# Compute volume percent of rock fragments that are < 75mm on a < 75mm basis.

DEFINE vol75mm_l    totalRFvol_l/(1-(fragvoll-totalRFvol_l)/100).

# Compute volume percent of rock fragments that are < 75mm
# to a weight percent for each row in the rock fragment table.

DEFINE wtRF_row_l     IF totalRFvol_l == 0 THEN 0 ELSE (rockfrag_row_l/totalRFvol_l*vol75mm_l)
*  densityrock/(((vol75mm_l/100)*densityrock)+((1-vol75mm_l/100)*dbthirdbar_r)).

# Compute volume percent of rock fragments that are < 5mm
# to a weight percent for each row in the rock fragment table.

DEFINE wtRF5mm_row_l      IF 5>=fragsize_h THEN wtRF_row_l
ELSE IF (5>=fragsize_l and 5<=fragsize_r)
THEN (5-fragsize_l)/(fragsize_r-fragsize_l)/2*wtRF_row_l
ELSE IF (5>fragsize_r and 5<=fragsize_h)
THEN (((5-fragsize_r)/(fragsize_h-fragsize_r)/2)+0.5)*wtRF_row_l
ELSE 0.

# Compute weight percent of rock fragments of whole soil for each row.

DEFINE wtRFwhole_row_l      fragvol_l*densityrock/(((fragvol_l/100)*densityrock)+((1-fragvol_l/100)
* dbthirdbar_r)).

# Compute weight percent of rock fragments > 75mm of whole soil, minus pararocks.

DEFINE wtRFwhole75mm_row_ls       IF (75>=fragsize_l and 75<=fragsize_r)
THEN  ((1-((75-fragsize_l)/(fragsize_r-fragsize_l)/2))*wtRFwhole_row_l)
```

```
ELSE IF (75>fragsize_r and 75<=fragsize_h)
THEN ((1-(((75-fragsize_r)/(fragsize_h-fragsize_r)/2)+0.5))*wtRFwhole_row_l)
ELSE IF 75>fragsize_h THEN 0 ELSE wtRFwhole_row_l.

DEFINE wtRFwhole75mm_row_l IF ISNULL(fragvlr) and ISNULL(fragvol_l) and ISNULL(fragvol_h)
THEN 0 ELSE IF (fraghard2=="strongly" OR fraghard2=="very strongly" or fraghard2=="indurated" or
ISNULL(fraghard2)) AND fragkind2 != "wood" THEN wtRFwhole75mm_row_ls ELSE 0.

DEFINE wtRFwhole75mm_l ARRAYSUM(wtRFwhole75mm_row_l).

# Compute weight percent of rock fragments > 250mm of whole soil, minus pararocks.

DEFINE wtRFwhole250mm_row_j IF (250>=fragsize_l and 250<=fragsize_r) THEN ((1-((250-
fragsize_l)/(fragsize_r-fragsize_l)/2))*wtRFwhole_row_l) ELSE IF (250>fragsize_r and
250<=fragsize_h) THEN ((1-(((250-fragsize_r)/(fragsize_h-fragsize_r)/2)+0.5))*wtRFwhole_row_l)
ELSE IF fragsize_l>=250 THEN wtRFwhole_row_l ELSE 0.

DEFINE wtRFwhole250mm_row_l IF ISNULL(fragvlr) and ISNULL(fragvol_l) and
ISNULL(fragvol_h)
THEN 0 ELSE IF (fraghard2=="strongly" or fraghard2=="very strongly" or fraghard2=="indurated" or
ISNULL(fraghard2)) AND fragkind2 != "wood" THEN wtRFwhole250mm_row_j ELSE 0.

DEFINE rockfrag_250l ARRAYSUM(wtRFwhole250mm_row_l).

# Compute weight percent of rock fragments that are 75 to 250mm, minus pararocks.

DEFINE rockfrag_75l wtRFwhole75mm_l - rockfrag_250l.

#-------------------------------------------------------------------------------
# Start of percent passing sieves and rock fragments calculation for high values.
#-------------------------------------------------------------------------------
# Compute total volume percent of rock fragments, minus pararocks,
# on a whole soil basis.

DEFINE fragvolhs IF (fraghard2=="strongly" or fraghard2=="very strongly" or fraghard2=="indurated"
or ISNULL(fraghard2)) AND fragkind2 != "wood" THEN fragvol_h ELSE 0.

DEFINE fragvolh ARRAYSUM(fragvolhs).
ASSIGN fragvolh IF fragvolh > 90 THEN 90 ELSE fragvolh. #Assumes there is at least 10% soil (<
2mm) at the high fragment condition.

# Uses the high fragment volume total from the Horizon Table.
DEFINE sumhighfrags ARRAYSUM(fragvol_h).
ASSIGN fragvolh IF (fragvolh / sumhighfrags) < 1 AND fragvolh < fragvoltot_h AND fragvolh != 0
AND NOT ISNULL(sumhighfrags) AND sumhighfrags > 0 THEN (fragvolh / sumhighfrags) *
fragvoltot_h ELSE IF ISNULL(fragvoltot_h) THEN fragvolh ELSE fragvoltot_h.

# Compute volume percent of rock fragments that are < 75mm for each row
# in the fragment table.

DEFINE rockfrag_row_h IF (75 >= fragsize_l and 75 <= fragsize_r) THEN ((75-
fragsize_l)/(fragsize_r-fragsize_l)/2*fragvol_h) ELSE IF (75 > fragsize_r and 75 <= fragsize_h) THEN
```

(((75-fragsize_r)/(fragsize_h-fragsize_r)/2)+0.5)*fragvol_h ELSE IF 75 > fragsize_h THEN fragvol_h ELSE 0.

Compute total volume % of rock fragments that are < 75mm on a
whole soil basis, minus pararocks.

DEFINE totalRFvol_hs IF (fraghard2=="strongly" or fraghard2=="very strongly" or fraghard2 == "indurated" or ISNULL(fraghard2)) AND fragkind2 != "wood" THEN rockfrag_row_h ELSE 0.

DEFINE totalRFvol_h ARRAYSUM(totalRFvol_hs).

Compute volume percent of rock fragments that are < 75mm on a < 75mm basis.

DEFINE vol75mm_h totalRFvol_h/(1-(fragvolh-totalRFvol_h)/100).

Compute volume percent of rock fragments that are < 75mm
to a weight percent for each row in the fragment table.

DEFINE wtRF_row_h IF totalRFvol_h == 0 THEN 0 ELSE (rockfrag_row_h/totalRFvol_h *vol75mm_h)*densityrock/ (((vol75mm_h/100)*densityrock)+((1-vol75mm_h/100)*dbthirdbar_r)).

Compute volume percent of rock fragments that are < 5mm to a weight percent for each row in the fragment table.

DEFINE wtRF5mm_row_h IF 5>=fragsize_h THEN wtRF_row_h ELSE IF (5>=fragsize_l and 5<=fragsize_r) THEN (5-fragsize_l)/(fragsize_r-fragsize_l)/2*wtRF_row_h ELSE IF (5>fragsize_r and 5<=fragsize_h) THEN (((5-fragsize_r)/(fragsize_h-fragsize_r)/2)+0.5)*wtRF_row_h ELSE 0.

Compute weight percent of rock fragments of whole soil for each row.

DEFINE wtRFwhole_row_h fragvol_h*densityrock/(((fragvol_h/100)*densityrock)+ ((1-fragvol_h/100)*dbthirdbar_r)).

Compute weight percent of rock fragments > 75mm of whole soil, minus pararocks.

DEFINE wtRFwhole75mm_row_k IF (75>=fragsize_l and 75<=fragsize_r) THEN ((1-((75-fragsize_l)/(fragsize_r-fragsize_l)/2))*wtRFwhole_row_h) ELSE IF (75>fragsize_r and 75<=fragsize_h) THEN ((1-(((75-fragsize_r)/(fragsize_h-fragsize_r)/2)+0.5))*wtRFwhole_row_h)
ELSE IF 75>fragsize_h THEN 0 ELSE wtRFwhole_row_h.

DEFINE wtRFwhole75mm_row_h IF ISNULL(fragvlr) and ISNULL(fragvol_l) and ISNULL(fragvol_h)
THEN 0 ELSE IF (fraghard2=="strongly" or fraghard2=="very strongly" or fraghard2=="indurated" or ISNULL(fraghard2)) AND fragkind2 != "wood" THEN wtRFwhole75mm_row_k ELSE 0.

DEFINE wtRFwhole75mm_h ARRAYSUM(wtRFwhole75mm_row_h).

Compute weight percent of rock fragments > 250mm of the whole soil, minus pararocks.

DEFINE wtRFwhole250mm_row_k IF (250>=fragsize_l and 250<=fragsize_r) THEN ((1-((250-fragsize_l)/(fragsize_r-fragsize_l)/2))*wtRFwhole_row_h) ELSE IF (250>fragsize_r and

250<=fragsize_h) THEN ((1-(((250-fragsize_r)/(fragsize_h-fragsize_r)/2)+0.5))*wtRFwhole_row_h) ELSE IF fragsize_l>=250 THEN wtRFwhole_row_h ELSE 0.

DEFINE wtRFwhole250mm_row_h IF ISNULL(fragvlr) and ISNULL(fragvol_l) and ISNULL(fragvol_h)
THEN 0 ELSE IF (fraghard2=="strongly" or fraghard2=="very strongly" or fraghard2=="indurated" or ISNULL(fraghard2)) AND fragkind2 != "wood" THEN wtRFwhole250mm_row_k ELSE 0.

DEFINE rockfrag_250h ARRAYSUM(wtRFwhole250mm_row_h).

Compute weight percent of rock fragments that are 75 to 250mm, minus pararocks.

DEFINE rockfrag_75h wtRFwhole75mm_h-rockfrag_250h.

Compute percent passing #10 sieve, minus pararocks.

DEFINE sieve_10hs IF (fraghard2=="strongly" or fraghard2=="very strongly" or fraghard2=="indurated" or ISNULL(fraghard2)) AND fragkind2 != "wood" THEN wtRF_row_l ELSE 0.

DEFINE sieve_10h IF ISNULL(ARRAYSUM(fragvlr)) THEN 100 ELSE 100-RRAYSUM(sieve_10hs).

Compute percent passing #4 sieve (minus pararocks).

DEFINE sieve_4hs IF (fraghard2=="strongly" or fraghard2=="very strongly" or fraghard2=="indurated" or ISNULL(fraghard2)) AND fragkind2 != "wood" THEN wtRF5mm_row_l ELSE 0.

DEFINE sieve_4h IF ISNULL(ARRAYSUM(fragvlr)) THEN 100 ELSE ARRAYSUM(sieve_4hs)+sieve_10h.

Compute percent passing #40 sieve (minus pararocks).

DEFINE sieve_40h IF ISNULL(ARRAYSUM(fragvlr)) AND (ISNULL(ms_h) OR ISNULL(fs_h) OR ISNULL(vfs_h) OR ISNULL(silth)) THEN (((0.7485*ms_r)+fs_r+vfs_r+siltr+clayr)+(clayh-clayr)) ELSE IF NOT ISNULL(ARRAYSUM(fragvlr)) and NOT ISNULL(ARRAYSUM(fragvol_l)) and NOT ISNULL(ARRAYSUM(fragvol_h)) AND (ISNULL(ms_h) OR ISNULL(fs_h) OR ISNULL(vfs_h) OR ISNULL(silth)) THEN sieve_10h/100*(((0.7485*ms_r)+fs_r+vfs_r+siltr+clayr)+(clayh-clayr)) ELSE ISNULL(ARRAYSUM(fragvlr)) THEN (((0.7485*ms_r)+fs_r+vfs_r+siltr+clayr)+(((ms_h-ms_r)+(fs_h-fs_r)+(vfs_h-vfs_r)+(silth-siltr)+(clayh-clayr))/5)) ELSE sieve_10h/100 *(((0.7485*ms_r) +fs_r+vfs_r+siltr+clayr) + (((ms_h-ms_r)+(fs_h-fs_r)+(vfs_h-vfs_r)+(silth-siltr)+(clayh-clayr))/5)).

Compute percent passing #200 sieve (minus pararocks).

DEFINE sieve_200h IF ISNULL(ARRAYSUM(fragvlr)) AND (ISNULL(vfs_h) OR ISNULL(silth)) THEN (((vfs_r*0.56559)+siltr+clayr)+(clayh-clayr)) ELSE IF NOT ISNULL(ARRAYSUM(fragvlr)) and NOT ISNULL(ARRAYSUM(fragvol_l)) and NOT ISNULL(ARRAYSUM(fragvol_h)) AND (ISNULL(vfs_h) OR ISNULL(silth)) THEN sieve_10h/100*(((vfs_r*0.56559)+siltr+clayr)+(clayh-clayr))
ELSE ISNULL(ARRAYSUM(fragvlr)) THEN IF (vfs_r) < 15 THEN (((vfs_r*0.56559)+ siltr+clayr)+(((vfs_h-vfs_r)+(silth-siltr)+(clayh-clayr))/3)) ELSE

(((vfs_r*curvenum_h)+siltr+clayr)+(((vfs_h-vfs_r)+(silth-siltr)+(clayh-clayr))/3)) ELSE IF vfs_r < 15 THEN sieve_10h/100*(((vfs_r*0.56559)+siltr+clayr)+(((vfs_h-vfs_r)+(silth-siltr)+(clayh-clayr))/3)) ELSE sieve_10h/100*(((vfs_r*curvenum_h)+siltr+clayr)+(((vfs_h-vfs_r)+(silth-siltr)+(clayh-clayr))/3)).

ASSIGN sieve_10h IF om_r > 35 OR ISNULL(clayr) OR ISNULL(dbthirdbar_r) THEN 1/0 ELSE sieve_10h.

ASSIGN sieve_4h IF om_r > 35 OR ISNULL(clayr) OR ISNULL(dbthirdbar_r) THEN 1/0 ELSE sieve_4h.

ASSIGN sieve_40h IF om_r > 35 OR (stratextsflag == 1 AND (ISNULL(clayr) OR ISNULL(siltr) OR ISNULL(vcos_r) OR ISNULL(cos_r) OR ISNULL(ms_r))) OR ISNULL(dbthirdbar_r) THEN 1/0 ELSE sieve_40h.

ASSIGN sieve_200h IF om_r > 35 OR (stratextsflag == 1 AND (ISNULL(clayr) OR ISNULL(siltr) OR ISNULL(vfs_r))) OR ISNULL(dbthirdbar_r) THEN 1/0 ELSE sieve_200h.

ASSIGN rockfrag_250h IF om_r > 35 AND ISNULL(ARRAYSUM(fragvolh)) THEN 0 ELSE IF (ISNULL(clayr) AND stratextsflag != 1) OR ISNULL(dbthirdbar_r) THEN 1/0 ELSE rockfrag_250h.

ASSIGN rockfrag_75h IF om_r > 35 AND ISNULL(ARRAYSUM(fragvolh)) THEN 0 ELSE IF (ISNULL(clayr) AND stratextsflag != 1) OR ISNULL(dbthirdbar_r) THEN 1/0 ELSE rockfrag_75h.

#---
Rest of calculation for low values.
#---

Compute percent passing #10 sieve (minus pararocks).

DEFINE sieve_10ls IF (fraghard2=="strongly" or fraghard2=="very strongly" or fraghard2=="indurated" or ISNULL(fraghard2)) AND fragkind2 != "wood" THEN wtRF_row_h ELSE 0.

DEFINE sieve_10l IF ISNULL(ARRAYSUM(fragvlr)) THEN 100 ELSE 100-ARRAYSUM(sieve_10ls).

Compute percent passing #4 sieve (minus pararocks).

DEFINE sieve_4ls IF (fraghard2=="strongly" or fraghard2=="very strongly" or fraghard2=="indurated" or ISNULL(fraghard2)) AND fragkind2 != "wood" THEN wtRF5mm_row_h ELSE 0.

DEFINE sieve_4l IF ISNULL(ARRAYSUM(fragvlr)) THEN 100 ELSE ARRAYSUM (sieve_4ls) + sieve_10l.

Compute percent passing #40 sieve (minus pararocks).

DEFINE sieve_40l IF ISNULL(ARRAYSUM(fragvlr)) AND (ISNULL(ms_l) OR ISNULL(fs_l) OR ISNULL(vfs_l) OR ISNULL(siltl)) THEN (((0.7485*ms_r)+fs_r+vfs_r+siltr+clayr)-(clayr-clayl)) ELSE IF NOT ISNULL(ARRAYSUM(fragvlr)) and NOT ISNULL(ARRAYSUM(fragvol_l)) and NOT ISNULL(ARRAYSUM(fragvol_h)) AND (ISNULL(ms_l) OR ISNULL(fs_l) OR ISNULL(vfs_l) OR ISNULL(siltl)) THEN sieve_10l/100*(((0.7485*ms_r)+fs_r+vfs_r+siltr+clayr)-(clayr-clayl)) ELSE ISNULL(ARRAYSUM(fragvlr)) THEN (((0.7485*ms_r)+fs_r+vfs_r+siltr+clayr)-(((ms_r-ms_l)+(fs_r-fs_l)+(vfs_r-vfs_l)+(siltr-siltl)+(clayr-clayl))/5)) ELSE sieve_10l/100

```
*(((0.7485*ms_r)+fs_r+vfs_r+siltr+clayr)  - (((ms_r-ms_l)+(fs_r-fs_l)+(vfs_r-vfs_l)+(siltr-siltl)+(clayr-clayl))/5)).

# Compute percent passing #200 sieve (minus pararocks).

DEFINE sieve_200l      IF ISNULL(ARRAYSUM(fragvlr))  AND (ISNULL(vfs_l) OR ISNULL(siltl)) THEN (((vfs_r*0.56559)+siltr+clayr)-(clayr-clayl))  ELSE IF NOT ISNULL(ARRAYSUM(fragvlr))  and NOT ISNULL(ARRAYSUM(fragvol_l))  and NOT ISNULL(ARRAYSUM(fragvol_h))  AND (ISNULL(vfs_l)  OR ISNULL(siltl))  THEN sieve_10l/100*(((vfs_r*0.56559)+siltr+clayr)-(clayr-clayl))  ELSE ISNULL(ARRAYSUM(fragvlr))  THEN IF (vfs_r) < 15 THEN (((vfs_r*0.56559)+siltr+clayr)-(((vfs_r-vfs_l)+(siltr-siltl)+(clayr-clayl))/3))  ELSE (((vfs_r*curvenum_l)+siltr+clayr)-(((vfs_r-vfs_l)+(siltr-siltl)+(clayr-clayl))/3))  ELSE IF vfs_r < 15 THEN sieve_10l/100*(((vfs_r*0.56559)+siltr+clayr)-(((vfs_r-vfs_l)+(siltr-siltl)+(clayr-clayl))/3))  ELSE sieve_10l/100*(((vfs_r*curvenum_l)+siltr+clayr)-(((vfs_r-vfs_l)+(siltr-siltl)+(clayr-clayl))/3)).

ASSIGN sieve_10l      IF om_r > 35 OR ISNULL(clayr)  OR ISNULL(dbthirdbar_r)  THEN 1/0 ELSE sieve_10l.

ASSIGN sieve_4l       IF om_r > 35 OR ISNULL(clayr)  OR ISNULL(dbthirdbar_r)  THEN 1/0 ELSE sieve_4l.
ASSIGN sieve_40l IF  om_r  >  35  OR  (stratextsflag  ==  1  AND  (ISNULL(clayr)  OR  ISNULL(siltr)  OR ISNULL(vcos_r)  OR  ISNULL(cos_r)  OR  ISNULL(ms_r)))  OR  ISNULL(dbthirdbar_r)  THEN  1/0  ELSE sieve_40l.

ASSIGN sieve_200l      IF om_r > 35 OR (stratextsflag  ==  1 AND  (ISNULL(clayr)  OR  ISNULL(siltr) OR ISNULL(vfs_r)))  OR  ISNULL(dbthirdbar_r)  THEN  1/0  ELSE sieve_200l.

ASSIGN rockfrag_250l      IF om_r > 35 AND ISNULL(ARRAYSUM(fragvoll))  THEN 0 ELSE IF (ISNULL(clayr)  AND stratextsflag  != 1) OR ISNULL(dbthirdbar_r)  THEN 1/0 ELSE rockfrag_250l.

ASSIGN rockfrag_75l      IF om_r > 35 AND ISNULL(ARRAYSUM(fragvoll))  THEN 0 ELSE IF (ISNULL(clayr)  AND stratextsflag  != 1) OR ISNULL(dbthirdbar_r)  THEN 1/0 ELSE rockfrag_75l.

#-------------------------------------------
# Checks for high values that are lower than the low values and vise versa for rock fragments.
# This occurs when the fragvol_l and fragvol_h  are not different  from fragvol_r.

ASSIGN rockfrag_250l      IF rockfrag_250l > rockfrag_250h  THEN rockfrag_250h ELSE IF rockfrag_250l > rockfrag_250r  THEN rockfrag_250r  ELSE rockfrag_250l.

ASSIGN rockfrag_250h      IF rockfrag_250h < rockfrag_250l  THEN rockfrag_250l ELSE IF rockfrag_250h < rockfrag_250r  THEN rockfrag_250r  ELSE rockfrag_250h.

ASSIGN rockfrag_75l      IF rockfrag_75l > rockfrag_75h  THEN rockfrag_75h  ELSE IF rockfrag_75l > rockfrag_75r  THEN rockfrag_75r  ELSE rockfrag_75l.

ASSIGN rockfrag_75h      IF rockfrag_75h < rockfrag_75l  THEN rockfrag_75l ELSE IF rockfrag_75h < rockfrag_75r  THEN rockfrag_75r  ELSE rockfrag_75h.

#-------------------------------------------
# Checks that low values are not < 0 and high  values are not > 100,

# and rounds to whole numbers.

ASSIGN sieve_10l      IF NOT ISNULL(sieve_10l)  AND  sieve_10l < 0 THEN 0 ELSE
```

```
ROUND(sieve_10l,0).
ASSIGN sieve_4l       IF NOT ISNULL(sieve_4l)  AND sieve_4l < 0 THEN 0 ELSE
ROUND(sieve_4l,0).
ASSIGN sieve_40l      IF NOT ISNULL(sieve_40l)  AND sieve_40l < 0 THEN 0 ELSE
ROUND(sieve_40l,0).
ASSIGN sieve_200l     IF NOT ISNULL(sieve_200l)  AND sieve_200l < 0 THEN 0 ELSE
ROUND(sieve_200l,0).
ASSIGN rockfrag_250l IF NOT ISNULL(rockfrag_250l)  AND rockfrag_250l < 0 THEN 0 ELSE
rockfrag_250l.
ASSIGN rockfrag_75l      IF NOT ISNULL(rockfrag_75l)  AND rockfrag_75l < 0 THEN 0 ELSE
rockfrag_75l.

ASSIGN sieve_10r    ROUND(sieve_10r,0).
ASSIGN sieve_4r     ROUND(sieve_4r,0).
ASSIGN sieve_40r    ROUND(sieve_40r,0).
ASSIGN sieve_200r   ROUND(sieve_200r,0).

ASSIGN sieve_10h     IF NOT ISNULL(sieve_10h)  AND sieve_10h > 100 THEN 100 ELSE
ROUND(sieve_10h,0).
ASSIGN sieve_4h      IF NOT ISNULL(sieve_4h)  AND sieve_4h > 100 THEN 100 ELSE
ROUND(sieve_4h,0).
ASSIGN sieve_40h     IF NOT ISNULL(sieve_40h)  AND sieve_40h > 100 THEN 100 ELSE
ROUND(sieve_40h,0).
ASSIGN sieve_200h    IF NOT ISNULL(sieve_200h)  AND sieve_200h > 100 THEN 100 ELSE
ROUND(sieve_200h,0).
ASSIGN rockfrag_250h IF NOT ISNULL(rockfrag_250h)  AND rockfrag_250h > 100 THEN 100 ELSE
rockfrag_250h.
ASSIGN rockfrag_75h     IF NOT ISNULL(rockfrag_75h)  AND rockfrag_75h > 100 THEN 100 ELSE
rockfrag_75h.

#-------------------------------------------------
# Rounding errors (rounds to 1 instead of zero, when < 0.5)

ASSIGN rockfrag_250h     IF NOT ISNULL(rockfrag_250h)  AND rockfrag_250h > 0.05 and
rockfrag_250h < 1 THEN 1 ELSE rockfrag_250h.
ASSIGN rockfrag_75h      IF NOT ISNULL(rockfrag_75h)  AND rockfrag_75h > 0.05 and rockfrag_75h
< 1 THEN 1 ELSE rockfrag_75h.

#-------------------------------------------------
assign sieve_10l     sieve_10l > sieve_4l ? sieve_4l : sieve_10l.
assign sieve_40l     sieve_40l > sieve_10l ? sieve_10l : sieve_40l.
assign sieve_200l    sieve_200l > sieve_40l ? sieve_40l : sieve_200l.

assign sieve_10h     sieve_10h > sieve_4h ? sieve_4h : sieve_10h.
assign sieve_40h     sieve_40h > sieve_10h ? sieve_10h : sieve_40h.
assign sieve_200h    sieve_200h > sieve_40h ? sieve_40h : sieve_200h.

assign sieve_10r     sieve_10r > sieve_4r ? sieve_4r : sieve_10r.
assign sieve_40r     sieve_40r > sieve_10r ? sieve_10r : sieve_40r.


assign sieve_200r    sieve_200r > sieve_40r ? sieve_40r : sieve_200r.
```

618.105 NASIS Calculation for Estimating Water Content Data

Definition.—This calculation computes the low, representative, and high values for water_one-tenth_bar (0.1 bar H_20), water_one-third_bar (0.33 bar H_2O), water_15_bar (15 bar H_2O), water_satiated (Satiated H_2O), and bulk_density_oven_dry (Db oven dry).

Inputs.—This calculation requires the following data to be populated:

organic_matter_percent (OM) l,rv,h
rock_frag_greater_than_10_in (Rock >10) l,rv,h
rock_frag_3_to_10_in (Rock 3-10) l,rv,h
sieve_number_10 (#10) l,rv,h
bulk_density_one_third_bar (Db 0.33 bar H2O) l,rv,h or
bulk_density_one_tenth_bar (Db 0.1 bar H2O) l,rv,h
clay_total_separate (Total Clay) l,rv,h
linear_extensibility_percent (LEP) l,rv,h
texture_class (Texture) or
texture_modifier_and_class (Tex Mod & Class)*

Limitations.—This calculation computes water contents for organic and mineral layers.

- If no entry is found for rock elements, it is assumed to be zero.
- Missing data in other elements may result in no output.
- Calculation uses texture_class if populated, if not use texture_modifier_and class; however, calculation does not work if texture_modifier_and_class contain SR or modifiers.
- Calculation uses the texture group marked as rv for each horizon and the first texture sequence number within that texture group. If no texture group is marked rv or no sequence number is used, one texture will be selected at random.

Calculation.

```
DEFINE lieutex1     CODENAME(lieutex).

DEFINE oc_r     IF ISNULL(om_r) THEN 1/0 ELSE om_r/1.72.
DEFINE oc_l     IF ISNULL(om_l) THEN 1/0 ELSE om_l/1.72.
DEFINE oc_h     IF ISNULL(om_h) THEN 1/0 ELSE om_h/1.72.

DEFINE db_r     IF ISNULL(dbthirdbar_r) THEN dbtenthbar_r ELSE dbthirdbar_r.
DEFINE db_l     IF ISNULL(dbthirdbar_l) THEN dbtenthbar_l ELSE dbthirdbar_l.
DEFINE db_h     IF ISNULL(dbthirdbar_h) THEN dbtenthbar_h ELSE dbthirdbar_h.

ASSIGN claytotal_l IF claytotal_l == 0 THEN 0.1 ELSE claytotal_l.
ASSIGN fragvol_r    IF ISNULL(fragvol_r) THEN 0 ELSE fragvol_r.
ASSIGN fragvol_l    IF ISNULL(fragvol_l) THEN 0 ELSE fragvol_l.
ASSIGN fragvol_h    IF ISNULL(fragvol_h) THEN 0 ELSE fragvol_h.

# Assume particle density of rock fragments is 2.65 g/cc
DEFINE D_p_gt_2 INITIAL 2.65.

# Try to use single texture if available.
DEFINE tex IF ISNULL(texcl) THEN texgrp ELSE UPCASE(CODENAME(texcl)).

# 1500kPa to clay ratio vaires with bulk density and texture,
```

```
# otherwise assume ratio of 1500KPa to Clay is 0.4
DEFINE F IF claytotal_r >= 40 AND db_r <= 1.60 THEN 0.65 - 0.189*db_r ELSE IF claytotal_r >= 40
AND db_r > 1.6 THEN 0.3 ELSE IF tex == "SCL" OR tex == "CL" OR tex == "SL" THEN 0.42 ELSE
IF tex == "FSL" OR tex == "COSL" OR tex == "LFS" OR tex == "LS" THEN 0.45 ELSE IF tex ==
"VFSL" OR tex == "LCOS" OR tex == "FS" THEN 0.46 ELSE IF tex == "S" THEN 0.44 ELSE IF tex
== "SC" THEN 0.36 ELSE 0.4.

DEFINE F_l IF claytotal_l >= 40 AND db_r <= 1.60 THEN 0.65 - 0.189*db_r ELSE IF claytotal_l >=
40 AND db_r > 1.6 THEN 0.3 ELSE IF tex == "SCL" OR tex == "CL" OR tex == "SL" THEN 0.42
ELSE IF tex == "FSL" OR tex == "COSL" OR tex == "LFS" OR tex == "LS" THEN 0.45 ELSE IF tex
== "VFSL" OR tex == "LCOS" OR tex == "FS" THEN 0.46 ELSE IF tex == "S" THEN 0.44 ELSE IF
tex == "SC" THEN 0.36 ELSE 0.4.

DEFINE F_h IF claytotal_h >= 40 AND db_r <= 1.60 THEN 0.65 - 0.189*db_r ELSE IF claytotal_h >=
40 AND db_r > 1.6 THEN 0.3 ELSE IF tex == "SCL" OR tex == "CL" OR tex == "SL" THEN 0.42
ELSE IF tex == "FSL" OR tex == "COSL" OR tex == "LFS" OR tex == "LS" THEN 0.45 ELSE IF tex
== "VFSL" OR tex == "LCOS" OR tex == "FS" THEN 0.46 ELSE IF tex == "S" THEN 0.44 ELSE IF
tex == "SC" THEN 0.36 ELSE 0.4.

# Assume air entrapment ratio is 0.95
DEFINE air_entrap INITIAL 0.95.

# Determine coefficients p and q for Gregson equation
DEFINE p     if tex=="CL" or tex=="L" or tex=="SICL" or tex=="SIL" then 1.415 else if tex=="COSL"
or tex=="FSL" or tex=="LVFS" or tex=="SCL" or tex=="SI" or tex=="SL" or tex=="VFS" or
tex=="VFSL" or tex=="LCOS" or tex=="LFS" or tex=="LS" then 0.343 else if tex=="S" or tex=="SG" or
tex=="G" or tex=="COS" or tex=="FS" then 0.541 else if tex=="C" or tex=="SC" or tex=="SIC" then
0.879 else 1/0.

DEFINE q     if tex=="CL" or tex=="L" or tex=="SICL" or tex=="SIL" then 0.839 else if tex=="COSL"
or tex=="FSL" or tex=="LVFS" or tex=="SCL" or tex=="SI" or tex=="SL" or tex=="VFS" or
tex=="VFSL" or tex=="LCOS" or tex=="LFS" or tex=="LS" then 1.072 else if tex=="S" or tex=="SG" or
tex=="G" or tex=="COS" or tex=="FS" then 1.469 else if tex=="C" or tex=="SC" or tex=="SIC" then
0.955 else 1/0.

# Compute particle density based on organic matter
#ASSIGN om_r if isnull(om_r) then 0 else om_r.
DEFINE Dp   100 / ((om_r / 1.4) + (100 - om_r)/2.65).

# Compute weight percent of rock fragments based on sieves
ASSIGN fraggt10_r if isnull(fraggt10_r) then 0 else fraggt10_r.
ASSIGN frag3to10_r if isnull(frag3to10_r) then 0 else frag3to10_r.
DEFINE W_gt_2      fraggt10_r + frag3to10_r + (100 - sieveno10_r) * (100 - fraggt10_r - frag3to10_r) /
100.

ASSIGN fraggt10_l if isnull(fraggt10_l) then 0 else fraggt10_l.
ASSIGN frag3to10_l if isnull(frag3to10_l) then 0 else frag3to10_l.
DEFINE W_gt_2_l    fraggt10_l + frag3to10_l + (100 - sieveno10_h) * (100 - fraggt10_l - frag3to10_l) /
100.

ASSIGN fraggt10_h if isnull(fraggt10_h) then 0 else fraggt10_h.
ASSIGN frag3to10_h  if isnull(frag3to10_h) then 0 else frag3to10_h.
```

```
DEFINE W_gt_2_h     fraggt10_h + frag3to10_h + (100 - sieveno10_l) * (100 - fraggt10_h -
frag3to10_h) / 100.

# Adjust bulk density for rock fragments
DEFINE D_b          100 / (W_gt_2/D_p_gt_2 + (100 - W_gt_2)/db_r).
DEFINE D_b_l        100 / (W_gt_2_l/D_p_gt_2 + (100 - W_gt_2_l)/db_r).
DEFINE D_b_h        100 / (W_gt_2_h/D_p_gt_2 + (100 - W_gt_2_h)/db_r).

# Compute volume percent of rock fragments
DEFINE V_gt_2       (W_gt_2 * D_b) / D_p_gt_2.
DEFINE V_gt_2_l     (W_gt_2_l * D_b_l) / D_p_gt_2.
DEFINE V_gt_2_h     (W_gt_2_h * D_b_h) / D_p_gt_2.

# ----------15 Bar Water----------

# Compute 15 bar water content uncorrected (by volume and by weight)
DEFINE theta_1500_uc    ((claytotal_r * (1 - om_r/100) * F + om_r) * db_r) / 100.
DEFINE theta_1500_uc_l    ((claytotal_l * (1 - om_l/100) * F_l + om_l) * db_r) / 100.
DEFINE theta_1500_uc_h    ((claytotal_h * (1 - om_h/100) * F_h + om_h) * db_r) / 100.

DEFINE theta_1500_uc_w    (claytotal_r * (1 - om_r/100) * F + om_r) / 100.
DEFINE theta_1500_uc_w_l (claytotal_l * (1 - om_l/100) * F_l + om_l) / 100.
DEFINE theta_1500_uc_w_h  (claytotal_h * (1 - om_h/100) * F_h + om_h) / 100.

# Compute 15 bar water content corrected for rock fragments.
# Convert to percent.
DEFINE theta_1500         theta_1500_uc * (100 - V_gt_2).
DEFINE theta_1500_l       theta_1500_uc_l * (100 - V_gt_2_h).
DEFINE theta_1500_h       theta_1500_uc_h * (100 - V_gt_2_l).

DEFINE theta_1500_w       theta_1500_uc_w * (100 - W_gt_2).
DEFINE theta_1500_w_l     theta_1500_uc_w_l * (100 - W_gt_2_h).
DEFINE theta_1500_w_h     theta_1500_uc_w_h * (100 - W_gt_2_l).

#----------Satiated Water----------

# Compute saturated water content uncorrected
DEFINE wcs_uc     air_entrap * (1 - db_r / Dp).
DEFINE wcs_uc_l air_entrap * (1 - db_h / Dp).
DEFINE wcs_uc_h   air_entrap * (1 - db_l / Dp).

# Compute saturated water content corrected for rock fragments.
# Convert to percent.
DEFINE wcs       ROUND((wcs_uc * (100 - V_gt_2)),0).
DEFINE wcs_l     ROUND((wcs_uc_l * (100 - V_gt_2_h)),0).
DEFINE wcs_h     ROUND((wcs_uc_h * (100 - V_gt_2_l)),0).

# ----------1/3 Bar Water----------

# Compute RV values
# Compute slope and intercept for the Gregson equation
# Uses volumetric water content
```

```
DEFINE ln_1500 INITIAL 7.31322.
DEFINE ln_theta logn(theta_1500_uc).
DEFINE ln_wcs   logn(wcs_uc).

DEFINE cpslope (ln_1500 - p) / (ln_theta + q).
DEFINE cpintercept ln_1500 - (cpslope * ln_theta).

# Compute field capacity uncorrected
DEFINE fc_uc_10 exp((logn(10)  - cpintercept) / cpslope).
DEFINE fc_uc_33 IF tex == "C" THEN exp(0.237*LOGN(claytotal_r)-1.26*db_r+4.162)*db_r/100
ELSE exp((logn(33)  - cpintercept) / cpslope).

# Monotonicity check: field capacity between theta_1500_uc and wcs_uc
DEFINE cpslope_adj      IF not isnull(fc_uc_10)  and (fc_uc_10 <= 1.1 * theta_1500_uc or fc_uc_10 >=
.95 * wcs_uc) then ln_1500 / (ln_theta - ln_wcs) else cpslope.

DEFINE cpintercept_adj     IF not isnull(fc_uc_10)  and (fc_uc_10 <= 1.1 * theta_1500_uc or fc_uc_10 >=
.95 * wcs_uc) then 0 - (cpslope * ln_wcs) else cpintercept.

ASSIGN fc_uc_10   exp((logn(10)  - cpintercept_adj) / cpslope_adj).

ASSIGN cpslope_adj   IF not isnull(fc_uc_33)  and (fc_uc_33 <= 1.1 * theta_1500_uc or fc_uc_33 >= .95
* wcs_uc) then ln_1500 / (ln_theta - ln_wcs) else cpslope.

ASSIGN cpintercept_adj IF not isnull(fc_uc_33)  and (fc_uc_33 <= 1.1 * theta_1500_uc or fc_uc_33 >=
.95 * wcs_uc) then 0 - (cpslope * ln_wcs) else cpintercept.

ASSIGN fc_uc_33 IF not isnull(fc_uc_33)  AND claytotal_r > 40 THEN fc_uc_33 ELSE exp((logn(33)  -
cpintercept_adj)  / cpslope_adj).

# Correct field capacity for rock fragments.
# Convert to percent.
DEFINE wtenth_r     if tex=="LCOS"  or tex=="LFS"  or tex=="LS"  or tex=="S"  or tex=="SG"  or
tex=="G"  or tex=="COS"  or tex=="FS"  THEN fc_uc_10 * (100 - V_gt_2) ELSE 1/0.
DEFINE wthird_r fc_uc_33 *(100 - V_gt_2).

# Compute low values --------------------------
# Compute slope and intercept for the Gregson equation
# Uses volumetric water content
DEFINE ln_theta_l logn(theta_1500_uc_l).
DEFINE ln_wcs_l  logn(wcs_uc_l).

DEFINE cpslope_l (ln_1500 - p) /  (ln_theta + q).
DEFINE cpintercept_l  ln_1500 - (cpslope_l * ln_theta_l).

# Compute field capacity uncorrected
DEFINE fc_uc_10_l  exp((logn(10)  - cpintercept_l)  / cpslope_l).
DEFINE fc_uc_33_l IF tex == "C" THEN exp(0.237*LOGN(claytotal_l)-1.26*db_r+4.162)*db_r/100
ELSE exp((logn(33)  - cpintercept_l)  / cpslope_l).

# Monotonicity check: field capacity between theta_1500_uc_l and wcs_uc_l
```

```
DEFINE cpslope_adj_l IF not isnull(fc_uc_10_l) and (fc_uc_10_l <= 1.1 * theta_1500_uc_l or
fc_uc_10_l >= .95 * wcs_uc_l) then ln_1500 / (ln_theta_l - ln_wcs_l) else cpslope_l.

DEFINE cpintercept_adj_l IF not isnull(fc_uc_10_l) and (fc_uc_10_l <= 1.1 * theta_1500_uc_l or
fc_uc_10_l >= .95 * wcs_uc_l) then 0 - (cpslope_l * ln_wcs_l) else cpintercept_l.

ASSIGN fc_uc_10_l  exp((logn(10) - cpintercept_adj_l) / cpslope_adj_l).

ASSIGN cpslope_adj_l IF not isnull(fc_uc_33_l) and (fc_uc_33_l <= 1.1 * theta_1500_uc_l or
fc_uc_33_l >= .95 * wcs_uc_l) then ln_1500 / (ln_theta_l - ln_wcs_l) else cpslope_l.

ASSIGN cpintercept_adj_l IF not isnull(fc_uc_33_l) and (fc_uc_33_l <= 1.1 * theta_1500_uc_l or
fc_uc_33_l >= .95 * wcs_uc_l) then 0 - (cpslope_l * ln_wcs_l) else cpintercept_l.

ASSIGN fc_uc_33_l IF not isnull(fc_uc_33_l) AND claytotal_l > 40 THEN fc_uc_33_l ELSE
exp((logn(33) - cpintercept_adj_l) / cpslope_adj_l).

# Correct field capacity for rock fragments.
# Convert to percent.
DEFINE wtenth_l if tex=="LCOS" or tex=="LFS" or tex=="LS" or tex=="S" or tex=="SG" or tex=="G"
or tex=="COS" or tex=="FS" THEN fc_uc_10_l * (100 - V_gt_2_h) ELSE 1/0.
DEFINE wthird_l fc_uc_33_l * (100 - V_gt_2_h).

# Compute high values ------------------------
# Compute slope and intercept for the Gregson equation
# Uses volumetric water content
DEFINE ln_theta_h logn(theta_1500_uc_h).
DEFINE ln_wcs_h  logn(wcs_uc_h).

DEFINE cpslope_h      (ln_1500 - p) /  (ln_theta_h + q).
DEFINE cpintercept_h  ln_1500 - (cpslope_h * ln_theta_h).

# Compute field capacity uncorrected
DEFINE fc_uc_10_h  exp((logn(10) - cpintercept_h) / cpslope_h).
DEFINE fc_uc_33_h IF tex == "C" THEN exp(0.237*LOGN(claytotal_h)-1.26*db_r+4.162)*db_r/100
ELSE exp((logn(33) - cpintercept_h) / cpslope_h).

# Monotonicity check: field capacity between theta_1500_uc and wcs_uc
DEFINE cpslope_adj_h IF not isnull(fc_uc_10_h) and (fc_uc_10_h <= 1.1 * theta_1500_uc_h or
fc_uc_10_h >= .95 * wcs_uc_h) then ln_1500 / (ln_theta_h - ln_wcs_h) else cpslope_h.

DEFINE cpintercept_adj_h IF not isnull(fc_uc_10_h) and (fc_uc_10_h <= 1.1 * theta_1500_uc_h or
fc_uc_10_h >= .95 * wcs_uc_h) then 0 - (cpslope_h * ln_wcs_h) else cpintercept_h.

ASSIGN fc_uc_10_h  exp((logn(10) - cpintercept_adj_h) / cpslope_adj_h).

ASSIGN cpslope_adj_h IF not isnull(fc_uc_33_h) and (fc_uc_33_h <= 1.1 * theta_1500_uc_h or
fc_uc_33_h >= .95 * wcs_uc_h) then ln_1500 / (ln_theta_h - ln_wcs_h) else cpslope_h.

ASSIGN cpintercept_adj_h  IF not isnull(fc_uc_33_h) and fc_uc_33_h <= 1.1 * theta_1500_uc_h or
fc_uc_33_h >= .95 * wcs_uc_h) then 0 - (cpslope_h * ln_wcs_h) else cpintercept_h.
```

```
ASSIGN fc_uc_33_h IF not isnull(fc_uc_33_h) AND claytotal_h > 40 THEN fc_uc_33_h ELSE
exp((logn(33) - cpintercept_adj_h) / cpslope_adj_h).

# Correct field capacity for rock fragments.
# Convert to percent.
DEFINE wtenth_h    if tex=="LCOS" or tex=="LFS" or tex=="LS" or tex=="S" or tex=="SG" or
tex=="G" or tex=="COS" or tex=="FS" THEN fc_uc_10_h * (100 - V_gt_2_l) ELSE 1/0.

DEFINE wthird_h     fc_uc_33_h * (100 - V_gt_2_l).


#-----------------------------------------------------------
# Additional calculations for when water contents exceed satiated water contents using the gregson
model.
# This generally occurs for compacted or dense soil layers.

ASSIGN wtenth_r      IF om_r >20 OR ((wtenth_r >= wcs) AND (NOT ISNULL(wtenth_r) OR NOT
ISNULL(theta_1500) OR NOT ISNULL(wcs))) THEN 1/0 ELSE wtenth_r.

ASSIGN wthird_r     IF om_r <= 20 AND ((wthird_r >= wcs) AND (NOT ISNULL(wthird_r) OR NOT
ISNULL(theta_1500) OR NOT ISNULL(wcs))) THEN IF claytotal_r < 40 THEN
(exp(logn(theta_1500_uc_w*100)*0.515 - 0.619*db_r + 2.696))/100 * db_r * (100 - V_gt_2) ELSE
exp(0.237*LOGN(claytotal_r)-1.26*db_r+4.162)*db_r/100 * (100 - V_gt_2) ELSE IF om_r > 20 THEN
1/0 ELSE wthird_r.

ASSIGN wthird_r      IF ((wthird_r >= wcs) AND (NOT ISNULL(wthird_r) OR NOT
ISNULL(theta_1500) OR NOT ISNULL(wcs))) THEN wcs-(0.05*wcs) ELSE wthird_r. IF om_r <= 20
AND ((theta_1500 > wcs OR theta_1500 > wthird_r) AND (NOT ISNULL(wthird_r) OR NOT
ISNULL(theta_1500) OR NOT ISNULL(wcs))) THEN wthird_r-(0.1*wthird_r) ELSE IF om_r > 20
THEN 1/0 ELSE theta_1500.

ASSIGN wtenth_l      IF om_r >20 OR ((wtenth_l >= wcs_l) AND (NOT ISNULL(wtenth_l) OR NOT
ISNULL(theta_1500_l) OR NOT ISNULL(wcs_l))) THEN 1/0 ELSE wtenth_l.

ASSIGN wthird_l     IF om_r <= 20 AND ((wthird_l >= wcs_l) AND (NOT ISNULL(wthird_l) OR NOT
ISNULL(theta_1500_l) OR NOT ISNULL(wcs_l))) THEN IF claytotal_l < 40 THEN
(exp(logn(theta_1500_uc_w_l*100)*0.515 - 0.619*db_r + 2.696))/100 * db_r * (100 - V_gt_2_h) ELSE
exp(0.237*LOGN(claytotal_l)-1.26*db_r+4.162)*db_r/100 * (100 - V_gt_2_h) ELSE IF om_r > 20
THEN 1/0 ELSE wthird_l.

ASSIGN wthird_l      IF ((wthird_l >= wcs_l) AND (NOT ISNULL(wthird_l) OR NOT
ISNULL(theta_1500_l) OR NOT ISNULL(wcs_l))) THEN wcs_l-(0.05*wcs_l) ELSE wthird_l.

ASSIGN theta_1500_l      IF om_r <= 20 AND ((theta_1500_l > wcs_l OR theta_1500_l > wthird_l)
AND (NOT ISNULL(wthird_l) OR NOT ISNULL(theta_1500_l) OR NOT ISNULL(wcs_l))) THEN
wthird_l-(0.1*wthird_l) ELSE IF om_r > 20 THEN 1/0 ELSE theta_1500_l.

ASSIGN wtenth_h     IF om_r >20 OR ((wtenth_h >= wcs_h) AND (NOT ISNULL(wtenth_h) OR NOT
ISNULL(theta_1500_h) OR NOT ISNULL(wcs_h))) THEN 1/0 ELSE wtenth_h.

ASSIGN wthird_h      IF om_r <= 20 AND ((wthird_h >= wcs_h) AND (NOT ISNULL(wthird_h) OR
NOT ISNULL(theta_1500_h) OR NOT ISNULL(wcs_h))) THEN IF claytotal_h < 40 THEN
```

(exp(logn(theta_1500_uc_w_h*100)*0.515 - 0.619*db_r + 2.696))/100 * db_r * (100 - V_gt_2_l) ELSE exp(0.237*LOGN(claytotal_h)-1.26*db_r+4.162)*db_r/100 * (100 - V_gt_2_l) ELSE IF om_r > 20 THEN 1/0 ELSE wthird_h.

ASSIGN wthird_h IF ((wthird_h >= wcs_h) AND (NOT ISNULL(wthird_h) OR NOT ISNULL(theta_1500_h) OR NOT ISNULL(wcs_h))) THEN wcs_h-(0.05*wcs_h) ELSE wthird_h.

ASSIGN theta_1500_h IF om_r <= 20 AND ((theta_1500_h > wcs_h OR theta_1500_h > wthird_h) AND (NOT ISNULL(wthird_h) OR NOT ISNULL(theta_1500_h) OR NOT ISNULL(wcs_h))) THEN wthird_h-(0.1*wthird_h) ELSE IF om_r > 20 THEN 1/0 ELSE theta_1500_h.

#----------Organic Soils, 1/3 and 15 bar Water----------

DEFINE theta_1500_org_w IF om_r <= 20 THEN 1/0 ELSE IF (lieutex1 == "mpt" OR lieutex1 == "mpm") THEN (2.019*oc_r+10.54) *0.75 ELSE IF (lieutex1 == "muck" OR lieutex1 == "hpm") THEN (1.731*oc_r+8.863) *0.75 ELSE IF (lieutex1 == "peat" OR lieutex1 == "spm") THEN (2.122*oc_r+10.539) *0.75 ELSE 1/0.

ASSIGN theta_1500 IF ISNULL(theta_1500_org_w) THEN theta_1500 ELSE theta_1500_org_w*db_r / 100 *(100 - fragvol_r).

DEFINE theta_1500_org_w_l IF om_r <= 20 THEN 1/0 ELSE IF (lieutex1 == "mpt" OR lieutex1 == "mpm") THEN (2.019*oc_l+10.54) *0.75 ELSE IF (lieutex1 == "muck" OR lieutex1 == "hpm") THEN (1.731*oc_l+8.863) *0.75 ELSE IF (lieutex1 == "peat" OR lieutex1 == "spm") THEN (2.122*oc_l+10.539) *0.75 ELSE 1/0.

ASSIGN theta_1500_l IF ISNULL(theta_1500_org_w_l) THEN theta_1500_l ELSE theta_1500_org_w_l*db_r / 100 *(100 - fragvol_h).

DEFINE theta_1500_org_w_h IF om_r <= 20 THEN 1/0 ELSE IF (lieutex1 == "mpt" OR lieutex1 == "mpm") THEN (2.019*oc_h+10.54) *0.75 ELSE IF (lieutex1 == "muck" OR lieutex1 == "hpm") THEN (1.731*oc_h+8.863) *0.75 ELSE IF (lieutex1 == "peat" OR lieutex1 == "spm") THEN (2.122*oc_h+10.539) *0.75 ELSE 1/0.

ASSIGN theta_1500_h IF ISNULL(theta_1500_org_w_h) THEN theta_1500_h ELSE theta_1500_org_w_h*db_r / 100 *(100 - fragvol_l).

DEFINE ln_theta_1500 LOGN(theta_1500_org_w).
DEFINE ln_theta_1500_l LOGN(theta_1500_org_w_l).
DEFINE ln_theta_1500_h LOGN(theta_1500_org_w_h).
DEFINE ln_db_r LOGN(db_r).
DEFINE ln_oc_r LOGN(oc_r).
DEFINE ln_oc_l LOGN(oc_l).
DEFINE ln_oc_h LOGN(oc_h).
ASSIGN wthird_r IF om_r <= 20 THEN wthird_r ELSE IF (lieutex1 == "mpt" OR lieutex1 == "mpm") THEN EXP(0.360*ln_theta_1500-1.076*ln_db_r+2.236) * db_r / 100 *(100 - fragvol_r) ELSE IF (lieutex1 == "muck" OR lieutex1 == "hpm") THEN EXP(0.142*ln_theta_1500-1.047*ln_db_r+3.340) * db_r / 100 *(100 - fragvol_r) ELSE IF (lieutex1 == "peat" OR lieutex1 == "spm") THEN EXP(0.427*ln_theta_1500-0.852*ln_db_r+2.282) * db_r / 100 *(100 - fragvol_r) ELSE 1/0.

ASSIGN wthird_l IF om_r <= 20 THEN wthird_l ELSE IF (lieutex1 == "mpt" OR lieutex1 == "mpm") THEN EXP(0.360*ln_theta_1500_l-1.076*ln_db_r+2.236) * db_r / 100 *(100 - fragvol_h) ELSE IF

```
(lieutex1 == "muck" OR lieutex1 == "hpm") THEN EXP(0.142*ln_theta_1500_l-1.047*ln_db_r+3.340)
* db_r / 100 *(100 - fragvol_h) ELSE IF (lieutex1 == "peat" OR lieutex1 == "spm") THEN
EXP(0.427*ln_theta_1500_l-0.852*ln_db_r+2.282) * db_r / 100 *(100 - fragvol_h) ELSE 1/0.

ASSIGN wthird_h IF om_r <= 20 THEN wthird_h ELSE IF (lieutex1 == "mpt" OR lieutex1 ==
"mpm") THEN EXP(0.360*ln_theta_1500_h-1.076*ln_db_r+2.236) * db_r / 100 *(100 - fragvol_l)
ELSE IF (lieutex1 == "muck" OR lieutex1 == "hpm") THEN EXP(0.142*ln_theta_1500_h-
1.047*ln_db_r+3.340) * db_r / 100 *(100 - fragvol_l) ELSE IF (lieutex1 == "peat" OR lieutex1 ==
"spm") THEN EXP(0.427*ln_theta_1500_h-0.852*ln_db_r+2.282) * db_r / 100 *(100 - fragvol_l) ELSE
1/0.

ASSIGN theta_1500 IF om_r > 13 AND om_r <= 20 AND (lieutex1 == "mpt" OR lieutex1 == "mpm"
OR lieutex1 == "muck" OR lieutex1 == "hpm" OR lieutex1 == "peat" OR lieutex1 == "spm") THEN
(EXP(0.673*ln_oc_r + 1.618)*0.75) * db_r / 100 *(100 - fragvol_l) ELSE theta_1500.

ASSIGN theta_1500_l IF om_r > 13 AND om_r <= 20 AND (lieutex1 == "mpt" OR lieutex1 ==
"mpm" OR lieutex1 == "muck" OR lieutex1 == "hpm" OR lieutex1 == "peat" OR lieutex1 == "spm")
THEN (EXP(0.673*ln_oc_l + 1.618)*0.75) * db_r / 100 *(100 - fragvol_h) ELSE theta_1500_l.

ASSIGN theta_1500_h IF om_r > 13 AND om_r <= 20 AND (lieutex1 == "mpt" OR lieutex1 ==
"mpm" OR lieutex1 == "muck" OR lieutex1 == "hpm" OR lieutex1 == "peat" OR lieutex1 == "spm")
THEN (EXP(0.673*ln_oc_h + 1.618)*0.75) * db_r / 100 *(100 - fragvol_l) ELSE theta_1500_h.

DEFINE ln_theta_1500_A LOGN(EXP(0.673*ln_oc_r + 1.618) *0.75).
DEFINE ln_theta_1500_A_l LOGN(EXP(0.673*ln_oc_l + 1.618) *0.75).
DEFINE ln_theta_1500_A_h LOGN(EXP(0.673*ln_oc_h + 1.618) *0.75).

ASSIGN wthird_r IF om_r > 13 AND om_r <= 20 AND (lieutex1 == "mpt" OR lieutex1 == "mpm"
OR lieutex1 == "muck" OR lieutex1 == "hpm" OR lieutex1 == "peat" OR lieutex1 == "spm") THEN
EXP(0.267*ln_theta_1500_A - 1.141*ln_db_r + 2.821) * db_r / 100 *(100 - fragvol_r) ELSE wthird_r.

ASSIGN wthird_l IF om_r > 13 AND om_r <= 20 AND (lieutex1 == "mpt" OR lieutex1 == "mpm"
OR lieutex1 == "muck" OR lieutex1 == "hpm" OR lieutex1 == "peat" OR lieutex1 == "spm") THEN
EXP(0.267*ln_theta_1500_A_l - 1.141*ln_db_r + 2.821) * db_r / 100 *(100 - fragvol_h) ELSE wthird_l.

ASSIGN wthird_h IF om_r > 13 AND om_r <= 20 AND (lieutex1 == "mpt" OR lieutex1 == "mpm"
OR lieutex1 == "muck" OR lieutex1 == "hpm" OR lieutex1 == "peat" OR lieutex1 == "spm") THEN
EXP(0.267*ln_theta_1500_A_h - 1.141*ln_db_r + 2.821) * db_r / 100 *(100 - fragvol_l) ELSE wthird_h.

#----------Oven Dry Bulk Denisty----------

# Compute oven dry bulk density
DEFINE bdrdr IF NOT ISNULL(lep_r) THEN (((lep_r/100) / (1 - V_gt_2/100) + 1) ** 3) * db_r ELSE
1/0.
DEFINE bdrdl IF NOT ISNULL(lep_l) THEN (((lep_l/100) / (1 - V_gt_2/100) + 1) ** 3) * db_r ELSE
1/0.
DEFINE bdrdh IF NOT ISNULL(lep_h) THEN (((lep_h/100) / (1 - V_gt_2/100) + 1) ** 3) * db_r
ELSE 1/0.

ASSIGN bdrdl IF NOT ISNULL(lep_l) AND bdrdl > db_l THEN bdrdr - (db_r - db_l) ELSE bdrdl.
ASSIGN bdrdh IF NOT ISNULL(lep_h) AND bdrdh < db_h THEN bdrdr + (db_h - db_r) ELSE bdrdh.
```

```
# Values for median bulk density differences between 1/3 bar and oven-dry divided by the total clay.
DEFINE dbdiff      IF tex == "L" OR tex == "SCL" OR tex == "SIL" OR tex == "FSL" OR tex ==
"COSL" OR tex == "SL" OR tex == "VFSL" OR tex == "SI" OR tex == "LVFS" THEN 0.004 ELSE IF
tex == "CL" OR tex == "S" OR tex == "LFS" OR tex == "FS" OR tex == "LS" THEN 0.005 ELSE IF tex
== "SICL" OR tex == "LCOS" OR tex == "COS" THEN 0.006 ELSE IF tex == "C" OR tex == "SIC"
THEN 0.007 ELSE IF tex == "SC" THEN 0.002 ELSE IF tex == "VFS" THEN 0.003 ELSE 0.004.

ASSIGN bdrdr      IF (bdrdr - db_r) > 0.75 OR bdrdr > 2.1 THEN dbdiff*claytotal_r  + db_r ELSE bdrdr.

ASSIGN bdrdl      IF ((bdrdl - db_r)*(-1)) > 0.75 OR bdrdl > 2.1 THEN dbdiff*claytotal_l  + db_r ELSE
bdrdl.

ASSIGN bdrdh      IF (bdrdh - db_r) > 0.75 OR bdrdh > 2.1 THEN dbdiff*claytotal_h  + db_r ELSE
bdrdh.

ASSIGN bdrdr IF bdrdr > 2.1 OR om_r > 20 THEN 1/0 ELSE bdrdr.
ASSIGN bdrdl IF bdrdl > 2.1 OR om_r > 20 THEN 1/0 ELSE bdrdl.
ASSIGN bdrdh     IF bdrdh > 2.1 OR om_r > 20 THEN 1/0 ELSE bdrdh.

DEFINE bdrdr2 IF bdrdr > bdrdh THEN bdrdh ELSE bdrdr.
ASSIGN bdrdh IF bdrdr > bdrdh THEN bdrdr ELSE bdrdh.
ASSIGN bdrdr   bdrdr2.

DEFINE store2 IF bdrdl > bdrdr THEN bdrdl ELSE bdrdr.
ASSIGN bdrdl IF bdrdl > bdrdr THEN bdrdr ELSE bdrdl.
ASSIGN bdrdr     store2.
```

618.106 References

A. Evans, I.S. 1979. An integrated system of terrain analysis and slope mapping. Final Report on Grant DA-ERO-591-73-G0040, Department of Geography, University of Durham, England, p 192.

B. X. Shi, A. Zhu, J. Burt, W. Choi, R. Wang, T. Pei, B. Li, and C. Qin. 2007. An experiment using a circular neighborhood to calculate slope gradient from a DEM. Photogrammetric Engineering and Remote Sensing 73:143-157.

C. X. Shi, L. Girod, R. Long, R. Dekett, J. Philippe, and T. Burke. 2012. A comparison of LiDAR-based DEMs and USGS-sourced DEMs in terrain analysis for knowledge-based digital soil mapping. Geoderma 170:217-226.

D. Zevenbergen, L.W., and C.R. Thorne. 1987. Quantitative analysis of land surface topography. Earth Surface Processes and Landforms 12:47-56.