Part 618 – Soil Properties and Qualities

Subpart B – Exhibits

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 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 non-carbonate clay fraction.
3) If clay sized carbonate is null, then non-carbonate 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 claytottall  IF ISNULL(claytotal_l) THEN 0 ELSE claytotal_l.
DEFINE claysizedcarbl  IF ISNULL(claysizedcarb_l) THEN 0 ELSE claysizedcarb_l.
DEFINE ncclayl  claytottall - 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 (claytottall * (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 claytotalr  IF ISNULL(claytotal_r) THEN (claytotal_l + claytotalh)/2 ELSE claytotal_r.
DEFINE claysizedc arbr  IF ISNULL(claysizedcarb_r) THEN (claysizedcarb_l + claysizedcarb_h)/2 ELSE claysizedcarb_r.
DEFINE ncclayr  claytotalr - claysizedcarbr.

# 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 integer 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).