

License Agreement

The Aminpro Add-in (“the Add-In”) is free. By installing and using the Add-in, you agree not to hold us responsible for errors, omissions, or usage. You also agree not to redistribute it to third parties, or otherwise make the Add-In publicly available. You agree not to resell the Add-In. We encourage the not-for-profit distribution of the Add-in, but request, for security purposes, that it is done by linking to the web page given below. This enables us to ensure that the most up-to-date version is being used, and helps reduce virus threats.

Installation Instructions

Create a folder on C:/ root directory called Aminpro. The path should be: C:/Aminpro/

Copy and paste the Aminpro Add-in.xla file into the new folder.

Excel 2003

Open Excel 2003 and Select “Tools > Add-Ins > Browse” and browse the Aminpro Add-In.xla file

Click Ok.

Excel 2007

Click the Office Button (the big round decoration in the top left of the Excel screen). This opens the Office Menu. Click the Excel Options button at the bottom of this menu. Click the Add-Ins item in the list along the left edge of the dialog to see the Add-Ins panel. Make sure that the Manage dropdown at the bottom shows Excel Add-Ins, then press the Go button. This will bring up the familiar Add-Ins dialog. Browse to the Aminpro Add-In.xls file and click OK.

Updates

The Aminpro Add-in is updated periodically. The most current version can be downloaded from www.ameltda.com/add-in.php.

Have a suggestion? Just send us a spreadsheet with an example of the calculation and we will consider it for the next version of the add-in. You will receive an acknowledgement on our website.

List of Functions

FLOTATION FUNCTIONS	4
=RECOVERY(HEAD, CON, TAIL)	4
=CMFLOTREC(K, T, RMAX, RFROTH)	4
=CMTOTREC(K, T, RMAX, RFROTH, RENT).....	4
=BTRECOVERY(K, T, RMAX)	4
=MASSREC(HEAD, CON, TAIL).....	4
=DEGENT(DE0, DE20, DE50, SIZE)	5
GRINDING AND MILLING FUNCTIONS	6
=BONDTPH(Wi, P80, F80, CFNET, POWER).....	6
=BONDP80(Wi, F80, CFNET, POWER, TPH)	6
=BONDSE(Wi, P80, F80, CFNET).....	6
= VC(DIAMETER, U)	6
= MILLRPM(DIAMETER, VPC, U)	6
= BONDWI(SREENSIZE, GRAMS_PER_REV, P80, F80)	6
CHEMISTRY & MINERALOGY FUNCTIONS	7
=CHEMFORM("MINERAL" AS STRING).....	7
=SPECGRAV(MINERAL_FORMULA)	7
=MOLWEIGHT(MINERAL_FORMULA)	7
=MINFRAC(ELEMENT, MINERAL_FORMULA).....	7
=ATOMICWEIGHT("A").....	7
PUMPING FUNCTIONS	8
= LTOS(PSM)	8
=PSM(LTOS).....	8
=PSV(PSM, SGs)	8
=PULPDENSITY(PSM, SGs)	8
=PULPPERSONL(PULPDENSITY, SOLIDSSG)	8
=FRICITIONLOSS(HAZENWILLIAMSC, Q, ID, EQLLENGTH).....	8
=CRITICALVELOCITY(PSV, SGs, D50, ID)	8
=WATERVISC(TEMP).....	9
=PULPVISCOSITY(PSM, SGs, TEMP)	9
SIZE DISTRIBUTION FUNCTIONS	10
= GGSK50(D50, M)	10
= GGSK80(D80, M)	10
= GGSP80(K, M).....	10
=GGSX(K, M, P)	10
= GGSP(K, M, X)	10
= ROSRAMPX(B, M, PERCENT)	10
= ROSRAMB(B, M, SIZE)	10
= ROSRAMP80(P80, M, SIZE).....	10
= ROSRAMP80X(P80, M, PERCENT).....	11
=SWEBREC(XMAX, X50, B, SIZE)	11
=SWEBRECX(XMAX, X50, B, PERCENT).....	11

=SWEBRECP80(x100, x80, x50, SIZE)	11
=SWEBRECP80X(x100, x80, x50)	11
=SWEBRECB(x100, 80, BX50,)	11
=TYLERSIZE(MESH_NUMBER)	12
=ASTMSIZE(SIEVE_NUMBER)	12
CLASSIFICATION FUNCTIONS	13
=WHITEN(D50, ALPHA, SIZE)	13
=PLITT(D50, M, SIZE)	13
=WHITENB(D50, ALPHA, BETA, SIZE)	13
=WHITENBX(D50, ALPHA, BETA, PERCENT)	13

Flotation Functions

Notes

1. All recovery input values should be in decimal format (i.e. fractions). Some functions contain minor data validation checks that are not included in the descriptions

=Recovery(head, con, tail)

Calculates metal or other assay value recovery using the three-product formula

$$\text{Recovery} = \text{con} / \text{head} * (\text{head} - \text{tail}) / (\text{con} - \text{tail})$$

=CMFlotRec(K, t, Rmax, Rfroth)

Returns metal recovery for a continuous mixer, excluding entrainment

$$\text{CMFlotRec} = \text{Rc} * \text{Rfroth} / (\text{Rc} * \text{Rfroth} + 1 - \text{Rc})$$

Where:

$$\text{Rc} = (\text{K} * \text{t}) / (1 + \text{K} * \text{t}) * \text{Rmax}$$

=CMTotRec(K, t, Rmax, Rfroth, Rent)

Returns metal recovery for a continuous mixer, including entrainment

$$\text{CMTotRec} = (\text{Rflot} + ((1 - \text{Rflot}) * \text{Rent}))$$

Where:

$$\text{Rc} = (\text{K} * \text{t}) / (1 + \text{K} * \text{t}) * \text{Rmax}$$

$$\text{Rflot} = \text{Rc} * \text{Rfroth} / (\text{Rc} * \text{Rfroth} + 1 - \text{Rc})$$

=BTRecovery(K, t, Rmax)

Returns metal recovery for a batch test, excluding entrainment

$$\text{BTRecovery} = \text{Rmax} * (1 - \text{Exp}(-\text{K} * \text{t}))$$

=MassRec(head, con, tail)

Returns mass recovery from metal assays

$$\text{MassRec} = (\text{head} - \text{tail}) / (\text{con} - \text{tail})$$

=DegEnt(DE0, DE20, DE50, size)

Returns degree of entrainment for a given size from the DE0, DE50, and DE20,

Using the modified Swebrec function

$$\text{DegEnt} = fx / (1 + fx)$$

Where

$$fx = Mx ^ (\text{Log}(0.25) / \text{Log}(M80))$$

$$Mx = \text{Log}(DE0 / \text{size}) / \text{Log}(DE0 / DE50)$$

$$M80 = \text{Log}(DE0 / DE20) / \text{Log}(DE0 / DE50)$$

Grinding and Milling Functions

=BondTPH(Wi, P80, F80, CFnet, power)

Returns dry metric tonnes/hr using the Bond equation

$$\text{BondTPH} = \text{power} / (\text{CFnet} * 10 * \text{Wi} * (1 / \text{Sqr}(\text{P80}) - (1 / \text{Sqr}(\text{F80}))))$$

=BondP80(Wi, F80, CFnet, Power, TPH)

Returns the P80 of a ball mill circuit using the Bond equation

$$\text{BondP80} = 1 / ((\text{F80} ^{-0.5}) + (\text{Power} / (10 * \text{TPH} * \text{Wi} * \text{CFnet}))) ^ 2$$

=BondSE(Wi, P80, F80, CFnet)

Returns kWh/tonne using the Bond equation

$$\text{BondSE} = 10 * \text{Wi} * \text{CFnet} * (1 / \text{sqrt}(\text{F80}) - 1 / \text{sqrt}(\text{P80}))$$

= Vc(diameter, u)

Returns the critical velocity of a tumbling mill, in RPM. u specifies units (metric is default)

$$\text{Vc} = 76.64 / \text{Sqr}(\text{diameter})$$

= MillRPM(diameter, Vpc, u)

Returns the RPM of a tumbling mill, given the diameter and percent of critical speed (Vpc). U specifies units (metric is default)

$$\text{Vcr} = 76.64 / \text{Sqr}(\text{diameter}) * \text{Vpc}$$

= BondWi(screensize, grams_per_rev, P80, F80)

Returns the Bond work index, in metric tonnes, from the Bond test data

$$\text{Wi} = 44.5 / (\text{ScreenSize} ^ 0.23 * \text{grams_per_rev} ^ 0.82 * (10 / (\text{P80} ^ 0.5) - 10 / (\text{F80} ^ 0.5)))$$

$$\text{BondWi} = 2204.6/2000 * \text{Wi}$$

Chemistry & Mineralogy Functions

Notes

1. All formula inputs should be in the form of text strings (i.e. in quotations if direct input).

=ChemForm("mineral" as string)

Returns the chemical formula for the user-specified mineral name

=SpecGrav(mineral_formula)

Returns the specific gravity for the user-specified mineral formula (as string)

=MolWeight(mineral_formula)

Returns the molecular weight of the user-specified mineral formula (as string)

=MinFrac(element, mineral_formula)

Returns the % of an element comprising the user-specified mineral formula (as string)

=AtomicWeight("A")

Returns the atomic weight of an element (element in quotations)

Pumping Functions

= LtoS(PSm)

Returns liquids to solids ratio from the percent solids of a slurry

$$\text{LtoS} = (100 - \text{PSm}) / \text{PSm}$$

$$\text{LtoS} = (1 - \text{PSm}) / \text{PSm}$$

=PSm(LtoS)

Returns the percent solids from the L:S ratio

$$\text{PSm} = 100 / (1 + \text{LtoS})$$

=PSv(PSm, SGs)

Returns the percent solids by volume from the SG of the solids and the percent solids by mass

$$\text{PSv} = \text{PSm} / \text{SGs} * (1 / (100 - \text{PSm} + \text{PSm} / \text{SGs}))$$

=PulpDensity(PSm, SGs)

Returns the pulp density from solids SG and the percent solids by mass for water slurry

$$\text{PulpDensity} = 1 / (\text{PSm} / \text{SGs} + (1 - \text{PSm}))$$

=PulpPerSol(PulpDensity, SolidsSG)

Returns the pulp percent solids from the pulp density and solids specific gravity

$$\text{PulpPerSol} = (1 / \text{PulpDensity} - 1) / (1 / \text{SolidsSG} - 1)$$

=FrictionLoss(HazenWilliamsC, Q, ID, EqLength)

Returns friction losses in a pipe using ^ Hazen-Williams equation (units are metric)

$$\text{FrictionLoss} = \text{EqLength} * 0.002083 * (100 / \text{HazenWilliamsC}) ^ 1.85 * (\text{Q} * 1000 / 3.785 / 60) ^ 1.85 / (\text{ID} * 100 / 2.54) ^ 4.8655$$

=CriticalVelocity(PSv, SGs, D50, ID)

Returns critical settling velocity for a slurry

$$\text{CriticalVelocity} = 1.25 * (-0.0044 + \text{PSv} * 0.0034 + \text{Log}(\text{D50}) * 0.182975) * (2 * 9.81 * \text{ID} * (\text{SGs} - 1)) ^ 0.25$$

=WaterVisc(Temp)

Returns water viscosity as a function of the temperature (in degrees Kelvin)

$$A = 0.00002414$$

$$b = 247.8$$

$$C = 140$$

$$\text{WaterVisc} = A * 10 ^ (b / (\text{Temp} - C)) * 1000$$

=PulpViscosity(PSm, SGs, Temp)

Returns slurry viscosity from the percent solids and solids specific gravity for a given temperature (Kelvin)

$$Vw = \text{WaterVisc}(\text{Temp})$$

$$\text{PSVol} = \text{PSv}(\text{PSm}, \text{SGs})$$

$$\text{PulpViscosity} = Vw / (1 - \text{PSVol} ^ 0.333)$$

Size Distribution Functions

= GGSK50(D50, m)

Returns the Gates-Gaudin-Schuhman constant K from the D50 and constant m

$$\text{GGSK} = \text{D50} / (0.5)^{(1 / m)}$$

= GGSK80(D80, m)

Returns the Gates-Gaudin-Schuhman constant K from the D80 and constant m

$$\text{GGSK} = \text{D80} / (0.8)^{(1 / m)}$$

= GGSP80(K, m)

Returns the Gates-Gaudin-Schuhman P80 from the K and m constants

$$\text{GGSP80} = 0.8^{(1 / m)} * K$$

= GGSX(K, m, P)

Uses the Gates-Gaudin-Schuhman distribution to calculate the size for a given percent passing value

$$\text{GGSX} = (P^{(1 / m)}) * K$$

= GGSP(K, m, X)

Uses the Gates-Gaudin-Schuhman distribution to return the percent passing size X

$$\text{GGSPx} = (X / K)^m$$

= RosRamPX(b, m, percent)

Returns Px size (size at which x percent is finer) using Rosin-Rammler equation

$$\text{RosRamPX} = (\text{Log}(1 - \text{percent} / 100) / -b)^{(1 / m)}$$

= RosRamB(b, m, size)

Returns cumulative percent passing size x from Rosin-Rammler b and m constants

$$\text{RosRamB} = (1 - \text{Exp}(-b * (\text{size}^m))) * 100$$

= RosRamP80(P80, m, size)

Returns cumulative percent passing size x from P80 and Rosin Rammler m constant

$$\text{RosRamP80} = (1 - \text{Exp}(\text{Log}(0.2) / (\text{P80} \wedge m) * \text{size} \wedge m)) * 100$$

= RosRamP80X(P80, m, percent)

Returns size at which the user-specified percent passing occurs, using P80 and m values

$$\text{RosRamP80X} = (\text{P80} \wedge m * \text{Log}(1 - \text{percent} / 100) / \text{Log}(0.2)) \wedge (1 / m)$$

=Swebrec(xmax, x50, b, size)

Returns cumulative percent passing size x from Swebrec function

$$\text{Swebrec} = 100 / (1 + (\text{Log}(x_{\text{max}} / \text{size}) / \text{Log}(x_{\text{max}} / x_{50})) \wedge b)$$

=SwebrecX(xmax, x50, b, percent)

Returns size X for the given percent passing using the Swebrec function

$$\text{Swebrecx} = x_{\text{max}} / \text{Exp}(\text{Log}(x_{\text{max}} / X_{50}) * (1 / \text{percent} - 1) \wedge (1 / b))$$

=SwebrecP80(x100, x80, x50, size)

Uses the Swebrec function to returns the cumulative percent passing “size” given the X100, X80, and X50 of a size distribution

$$M_x = \text{Log}(X_{100} / \text{size}) / \text{Log}(X_{100} / X_{50})$$

$$M_{80} = \text{Log}(X_{100} / X_{80}) / \text{Log}(X_{100} / X_{50})$$

$$\text{SwebrecP80} = 1 / (1 + M_x \wedge (\text{Log}(0.25) / \text{Log}(M_{80}))) * 100$$

=SwebrecP80X(x100, x80, x50)

Uses the Swebrec function to return the size at which the user-specified percent passing occurs, using the X100, X80, and X50 of a size distribution

$$M_{80} = \text{Log}(X_{100} / X_{80}) / \text{Log}(X_{100} / X_{50})$$

$$b = \text{Log}(0.25) / \text{Log}(M_{80})$$

$$\text{SwebrecP80X} = X_{100} / \text{Exp}(\text{Log}(X_{100} / X_{50}) * (1 / \text{percent} - 1) \wedge (1 / b))$$

=SwebrecB(x100, 80, bx50,)

Returns the Swebrec b parameter from the X100, X50, and X80 parameters.

$$M_{80} = \text{Log}(X_{100} / X_{80}) / \text{Log}(X_{100} / X_{50})$$

$$\text{SwebrecB} = \text{Log}(0.25) / \text{Log}(M_{80})$$

=TylerSize(mesh_number)

Returns the Tyler sieve size in microns for a given mesh number

=ASTMSize(sieve_number)

Returns the ASTM E11:87 sieve size in microns for the given sieve number

Classification Functions

=Whiten(D50, alpha, size)

Returns cumulative percent passing size x from Whiten function

$$\text{Whiten} = (\text{Exp}(\alpha * \text{size} / \text{d50}) - 1) / (\text{Exp}(\alpha * \text{size} / \text{d50}) + \text{Exp}(\alpha) - 2) * 100$$

=Plitt(d50, m, size)

Returns corrected recovery to U/F using Plitt function

$$\text{Plitt} = (1 - \text{Exp}(-\text{Log}(2) * (\text{size} / \text{d50}) ^ m)) * 100$$

=WhitenB(d50, alpha, beta, size)

Returns the cumulative percent passing (or recovery) using the modified Whiten function

$$\text{WhitenB} = (\text{Exp}(\alpha * \text{size} / \text{d50}) - \text{beta}) / (\text{Exp}(\alpha * \text{size} / \text{d50}) + \text{Exp}(\alpha) - (2 * \text{beta})) * 100$$

=WhitenBX(d50, alpha, beta, percent)

Returns the size at which a user-specified percent passing or recovery occurs, modified Whiten.

$$\text{WhitenBX} = (\text{d50} / \alpha) * \text{Log}((\text{percent} * -\text{Exp}(\alpha) + (\text{percent} * 2 * \text{beta}) - \text{beta}) / (\text{percent} - 1))$$