//TRS 2018-10 set up for root exudation if needed

int OrganicExudate = 0; //assume no exudation unless otherwise told

try

{

OrganicExudate = organic\_exudate\_sa->GetValue(); //what kind of exudation do we have?

}

catch (...)

{

OrganicExudate = 0; //the XML config doesn't know about exudation so assume there's none

}

/\*

if (organic\_exudate\_sa != NULL) //i.e. if the XML config knows about exudation

{

OrganicExudate = organic\_exudate\_sa->GetValue(); //what kind of exudation do we have?

} //identify the exudate type or none

\*/

//TRS 2018-11 This is where we do the root exudation mods

int exudate\_test = 5;

switch (OrganicExudate) {

case 0: //no exudation

exudate\_test = 0;

break;

case 1: //lecithin

//insert soilPLevels into appropriate curve function for lecithin

//then do the relevant mod

exudate\_test = 1;

break;

case 2: //citrate

//insert soilPLevels into appropriate curve function for citrate

//then do the relevant mod

exudate\_test = 2;

exudate\_factor = 2.4; //curve function a simple hyperbola for now

break;

case 3: //malate

//insert soilPLevels into appropriate curve function for malate

//then do the relevant mod

exudate\_test = 3;

exudate\_factor = 0.4; //curve function a simple hyperbola for now

break;

default: //we don't know what this number means, whoops

//error message to log "Unrecognised organic exudate"

exudate\_test = 4;

}

for (BoxIndex box\_index = 0; box\_index < BOX\_COUNT; ++box\_index)

//This is where I need to add my exudation root tip stuff, then add the result to the totalPUptake that follows, then hopefully the rest of the readjusting that happens includes the exudation by default.

//if exudation is [on] AND it's the right time of day, then:

const int boxtipcount = 0; //need a way to get this, others use something like scoreboard->GetCharacteristicValue(TipCount\_Indices[idx], box\_index);

double tipExUptake = 0.0;

// if boxtipcount != 0 then:

//work out how much extra P they take up, using the mods provided before the per-box runthrough.

const double totalPUptake = ((delta\_amounts[0] + delta\_amounts[1] + delta\_amounts[2] + delta\_amounts[3]) \* dT) + totalHairPUptake + tipExUptake; //-µgP