

diff.diff

```
diff --git a/FEM/fem_ele_std.cpp b/FEM/fem_ele_std.cpp
index dbfbec5..552f259 100644
--- a/FEM/fem_ele_std.cpp
+++ b/FEM/fem_ele_std.cpp
@@ -10050,6 +10050,8 @@ void CFiniteElementStd::CalcSatution()
// In case the node is on the material interface
if(eS > 1.0)
eS = 1.0;
+ if(eS < MediaProp->capillary_pressure_values[1]) //MW: limit to non-negative 0saturation
+ eS = MediaProp->capillary_pressure_values[1];
//
pcs->SetNodeValue (nodes[i], idx_S, eS);
}
diff --git a/FEM/rf_mmp_new.cpp b/FEM/rf_mmp_new.cpp
index 6c5f5ce..1fa6c3c 100644
--- a/FEM/rf_mmp_new.cpp
+++ b/FEM/rf_mmp_new.cpp
@@ -1564,8 +1564,8 @@ std::ios::pos_type CMediumProperties::Read(std::ifstream* mmp_file)
}
break;
case 10: // unconfined 3D GW. 5.3.07 JOD
- in >> capillary_pressure_values[0]; // Pb
in >> capillary_pressure_values[1]; // Slr
+ in >> capillary_pressure_values[0]; // Pb
break;
default:
ScreenMessage("Error in MMPRead: no valid capillary pressure model.\n");
@@ -2460,15 +2460,19 @@ double CMediumProperties::PermeabilitySaturationFunction(const d
break;
//
case 10: // for unconfined 3D GW 5.3.07 JOD
+ //MW correct function. did only get constants before
+ /*
b = capillary_pressure_values[0];
slr = capillary_pressure_values[1];
- if(s1 > 0) {
+ if(s1 > 0 && s1 < 1) {
kr = max(0., 1 - (s1 / b));
kr = pow(kr, 2* (1 - kr)); //
kr = slr + (1 - slr) * kr;
}
else
kr = 1;
+ */
```

```

+ kr = sl;
break;
case 33: // FUNCTION: LINEAR OR POWER --> NON-WETTING krg = (b*(1-Se))^m
slr = 1.0 - maximum_saturation[phase]; // slr = 1.0 - sgm
@@ -5113,14 +5117,15 @@ double CMediumProperties::SaturationCapillaryPressureFunction(co
sl = MRange(slr+DBLEPSILON,sl,slm-DBLEPSILON);
break;
case 10: // unconfined 3D GW. 5.3.07 JOD
- /*pb = capillary_pressure_values[0];
+ //MW: remove comment to provide variables, not constants to PermeabilitySaturationFunc
+ pb = capillary_pressure_values[0];
slr = capillary_pressure_values[1];
if(pc > 0) {
sl = max(0., 1 - (pc / pb));
- sl = pow(sl, 2* (1 - sl)); //
+ sl = pow(sl, 2* (1 - sl));
sl = slr + (1 - slr) * sl;
}
- else*/
+ else
sl = 1;
break;
}

```