

```

0001 void sample(float dl, float ul, float pl, float cl,
0002             float dr, float ur, float pr, float cr,
0003             const float pm, const float um,
0004             float &d, float &u, float &p)
0005 {
0006     float c, cml, cmr, pml, pmr, shl, shr, sl, sr, stl, str;
0007
0008     if (0.0 <= um)
0009     {
0010         if (pm <= pl)
0011         {
0012             shl = ul - cl;
0013
0014             if (0.0 <= shl)
0015             {
0016                 < d, u, p = dl, ul, pl >
0017             }
0018             else
0019             {
0020                 cml = cl * pow(pm / pl, G1);
0021                 stl = um - cml;
0022
0023                 if (0.0 > stl)
0024                 {
0025                     d = dl * pow(pm / pl, 1.0 / GAMA);
0026                     u = um;
0027                     p = pm;
0028                 }
0029                 else
0030                 {
0031                     u = G5 * (cl + G7 * ul);
0032                     c = G5 * (cl + G7 * ul);
0033                     d = dl * pow(c / cl, G4);
0034                     p = pl * pow(c / cl, G3);
0035                 }
0036             }
0037         }
0038     }
0039     else
0040     {
0041         pml = pm / pl;
0042         sl = ul - cl * sqrt(G2 * pml + G1);
0043
0044         if (0.0 <= sl)
0045         {
0046             < d, u, p = dl, ul, pl >
0047         }
0048         else
0049         {
0050             d = dl * (pml + G6) / (pml * G6 + 1.0);
0051             u = um;
0052             p = pm;
0053         }
0054     }
0055 }
0056
0057 else
0058 {
0059     if (pm > pr)
0060     {
0061         pmr = pm / pr;
0062         sr = ur + cr * sqrt(G2 * pmr + G1);
0063
0064         if (0.0 >= sr)
0065         {
0066             < d, u, p = dr, ur, pr >
0067         }
0068         else
0069         {
0070             d = dr * (pmr + G6) / (pmr * G6 + 1.0);
0071             u = um;
0072             p = pm;
0073         }
0074     }
0075     else
0076     {
0077         shr = ur + cr;
0078
0079         if (0.0 >= shr)
0080         {
0081             < d, u, p = dr, ur, pr >
0082         }
0083         else
0084         {
0085             cmr = cr * pow(pm / pr, G1);
0086             str = um + cmr;
0087
0088             if (0.0 <= str)
0089             {
0090                 d = dr * pow(pm / pr, 1.0 / GAMA);
0091                 u = um;
0092                 p = pm;
0093             }
0094             else
0095             {
0096                 u = G5 * (-cr + G7 * ur);
0097                 c = G5 * (cr - G7 * ur);
0098                 d = dr * pow(c / cr, G4);
0099                 p = pr * pow(c / cr, G3);
0100             }
0101         }
0102     }
0103 }
0104 }
0105 }
0106 }
0107 }
0108 }
0109 }
0110 }

```