```
001 void sample(float dl, float ul, float pl, float cl,
002              float dr, float ur, float pr, float cr,
003              const float pm, const float um,
004              float &d, float &u, float &p)
005 {
006     float c, cml, cmr, pml, pmr, shl, shr, sl, sr, stl, str;
007 
008     if (0.0 <= um)
009     {
010         if (pm <= pl)
011         {
012             shl = ul - cl;
013 
014             if (0.0 <= shl)
015             {
                    < d, u, p = dl, ul, pl >
019             }
020             else
021             {
022                 cml = cl * pow(pm / pl, G1);
023                 stl = um - cml;
024 
025                 if (0.0 > stl)
026                 {
027                     d = dl * pow(pm / pl, 1.0 / GAMA);
028                     u = um;
029                     p = pm;
030                 }
031                 else
032                 {
                        < high-density code, low prob >
037                 }
038             }
039         }
040         else
041         {
042             pml = pm / pl;
043             sl = ul - cl * sqrt(G2 * pml + G1);
044 
045             if (0.0 <= sl)
046             {
                    < d, u, p = dl, ul, pl >
050             }
051             else
052             {
053                 d = dl * (pml + G6) / (pml * G6 + 1.0);
054                 u = um;
055                 p = pm;
056             }
057         }
058     }
059     else
060     {
            < symmetrical branch >
109     }
110 }
```