```
void sample(float dl, float ul, float pl, float cl,
            float dr, float ur, float pr, float cr,
            const float pm, const float um,
            float &d, float &u, float &p)
{
    float c, cml, cmr, pml, pmr, shl, shr, sl, sr, stl, str;

    if (0.0 <= um)
    {
        if (pm <= pl)
        {
            shl = ul - cl;

            if (0.0 <= shl)
            {
                < d, u, p = dl, ul, pl >
            }
            else
            {
                cml = cl * pow(pm / pl, G1);
                stl = um - cml;

                if (0.0 > stl)
                {
                    d = dl * pow(pm / pl, 1.0 / GAMA);
                    u = um;
                    p = pm;
                }
                else
                {
                    u = G5 * (cl + G7 * ul);
                    c = G5 * (cl + G7 * ul);
                    d = dl * pow(c / cl, G4);
                    p = pl * pow(c / cl, G3);
                }
            }
        }
        else
        {
            pml = pm / pl;
            sl = ul - cl * sqrt(G2 * pml + G1);

            if (0.0 <= sl)
            {
                < d, u, p = dl, ul, pl >
            }
            else
            {
                d = dl * (pml + G6) / (pml * G6 + 1.0);
                u = um;
                p = pm;
            }
        }
    }
    else
    {
        if (pm > pr)
        {
            pmr = pm / pr;
            sr = ur + cr * sqrt(G2 * pmr + G1);

            if (0.0 >= sr)
            {
                < d, u, p = dr, ur, pr >
            }
            else
            {
                d = dr * (pmr + G6) / (pmr * G6 + 1.0);
                u = um;
                p = pm;
            }
        }
        else
        {
            shr = ur + cr;

            if (0.0 >= shr)
            {
                < d, u, p = dr, ur, pr >
            }
            else
            {
                cmr = cr * pow(pm / pr, G1);
                str = um + cmr;

                if (0.0 <= str)
                {
                    d = dr * pow(pm / pr, 1.0 / GAMA);
                    u = um;
                    p = pm;
                }
                else
                {
                    u = G5 * (-cr + G7 * ur);
                    c = G5 * (cr - G7 * ur);
                    d = dr * pow(c / cr, G4);
                    p = pr * pow(c / cr, G3);
                }
            }
        }
    }
}
```