

```

void TrombaString::calculateUpdateEq()
{
    for (int l = 2; l < N - 2; ++l)
    {
        u[0][l] = A1 * u[1][l] + A2 * (u[1][l + 1] + u[1][l - 1])
                - A3 * (u[1][l + 2] + u[1][l - 2])
                + A4 * u[2][l] - A5 * (u[2][l + 1] + u[2][l - 1]);
    }
    int l = 1;
    u[0][l] = A1ss * u[1][l] + A2 * (u[1][l + 1] + u[1][l - 1])
            - A3 * (u[1][l + 2] + 2.0 * offset)
            + A4 * u[2][l] - A5 * (u[2][l + 1] + u[2][l - 1]);
    l = N - 2;
    u[0][l] = A1ss * u[1][l] + A2 * (u[1][l + 1] + u[1][l - 1])
            - A3 * (u[1][l - 2] + 2.0 * offset)
            + A4 * u[2][l] - A5 * (u[2][l + 1] + u[2][l - 1]);

    if (!bowing)
    {
        return;
    }
    else if (bowFlag)
    {
        // for using the same 'dynamic variables' during one loop
        Vb = _Vb.load();
        Fb = _Fb.load();
        sig3w = (rand.nextFloat() * 2 - 1) * sig3;
        fC = _fC.load();
        fS = _fS.load();
        bp = floor (_bowPos.load());
        alpha = _bowPos.load() - bp;
        NRbow();
        if (getBowModel() == exponential)
            excitation = Eexp * Fb * q * Global::exp1(-a * q * q);
        else if (getBowModel() == elastoPlastic)
            excitation = Eelasto * (sig0 * z
                + sig1 * zDot + sig2 * q + sig3w); /* (rho * csA);

        Global::extrapolation (u[0], bp, alpha, -excitation);
    }
}

```