

Midpoint

```
void simulation::UpdateMidpoint(double h){
    particle *part;

    // Calculate the acceleration now by looping through the forces
    // dividing by the mass and handling fixed particles.
    ClearAccelerations();
    for (forces.MoveFirst(); !forces.AtEnd(); forces.MoveNext()){
        forces.GetCurrent()->CalculateForce();
    }
    DivideByMass();
    HandleFixed();

    // Then, store the positions, velocities and accelerations obtained
    // in the temporary arrays above.
    int i;
    for (particles.MoveFirst(), i = 0; !particles.AtEnd(); particles.MoveNext(), i++)
    {
        part = particles.GetCurrent();
        p[0][i] = part->position;
        v[0][i] = part->velocity;
        a[0][i] = part->acceleration;
    }

    // Then, use these to calculate a new position and velocity at the midpoint
    // and then calculate the accelerations there.
    for (particles.MoveFirst(); !particles.AtEnd(); particles.MoveNext())
    {
        part = particles.GetCurrent();
        part->position += part->velocity * h/2;
        part->velocity += part->acceleration * h/2;
    }

    // Use the velocity and acceleration there to update the original p's and v's
    ClearAccelerations();
    for (forces.MoveFirst(); !forces.AtEnd(); forces.MoveNext())
        forces.GetCurrent()->CalculateForce();
    DivideByMass();
    HandleFixed();

    for (particles.MoveFirst(), i = 0; !particles.AtEnd(); particles.MoveNext(), i++)
    {
        part = particles.GetCurrent();
        part->position = p[0][i] + part->velocity * h;
        part->velocity = v[0][i] + part->acceleration * h;
    }

    HandleGround();
}
```