# Acceleration function benchmarks

Sebastian Micluța-Câmpeanu, Mikhail Vaganov

July 15, 2020

Solving the equations of notions for an N-body problem implies solving a (large) system of differential equations. In `DifferentialEquations.jl` these are represented through ODE or SDE problems. To build the problem we need a function that describe the equations. In the case of N-body problems, this function gives the accelerations for the particles in the system.

Here we will test the performance of several acceleration functions used in N-body simulations. The systems that will be used are not necesarly realistic as we are not solving the problem, we just time how fast is an acceleration function call.

```
using BenchmarkTools, NBodySimulator
```

```
Error: ArgumentError: Package BenchmarkTools not found in current path:
- Run `import Pkg; Pkg.add("BenchmarkTools")` to install the BenchmarkTools
 package.
```

```
using NBodySimulator: gather_bodies_initial_coordinates,
gather_accelerations_for_potentials,
    gather_simultaneous_acceleration, gather_group_accelerations
using StaticArrays

const SUITE = BenchmarkGroup();
```

```
Error: UndefVarError: BenchmarkGroup not defined
```

```
function acceleration(simulation)

    (u0, v0, n) = gather_bodies_initial_coordinates(simulation)

    acceleration_functions = gather_accelerations_for_potentials(simulation)
    simultaneous_acceleration = gather_simultaneous_acceleration(simulation)

    function soode_system!(dv, v, u, p, t)
        @inbounds for i = 1:n
            a = MVector(0.0, 0.0, 0.0)
            for acceleration! in acceleration_functions
                acceleration!(a, u, v, t, i);
            end
            dv[:, i] .= a
        end
        for acceleration! in simultaneous_acceleration
            acceleration!(dv, u, v, t);
        end
    end
    return soode_system!
end
```

```
acceleration (generic function with 1 method)
```

## 0.1 Gravitational potential

```
let SUITE=SUITE
    G = 6.67e-11 # m^3/kg/s^2
    N = 200 # number of bodies/particles
    m = 1.0 # mass of each of them
    v = 10.0 # mean velocity
    L = 20.0 # size of the cell side

    bodies = generate_bodies_in_cell_nodes(N, m, v, L)
    g_parameters = GravitationalParameters(G)
    system = PotentialNBodySystem(bodies, Dict(:gravitational => g_parameters))
    tspan = (0.0, 1.0)
    simulation = NBodySimulation(system, tspan)

    f = acceleration(simulation)
    u0, v0, n = gather_bodies_initial_coordinates(simulation)
    dv = zero(v0)

    b = @benchmarkable $f(dv, $v0, $u0, $g_parameters, 0.) setup=(dv=zero($v0)) evals=1

    SUITE["gravitational"] = b
end
```

```
Error: LoadError: UndefVarError: @benchmarkable not defined
in expression starting at none:18
```

## 0.2 Coulomb potential

```
let SUITE=SUITE
    n = 200
    bodies = ChargedParticle[]
    L = 20.0
    m = 1.0
    q = 1.0
    count = 1
    dL = L / (ceil(n^(1 / 3)) + 1)
    for x = dL / 2:dL:L, y = dL / 2:dL:L, z = dL / 2:dL:L
        if count > n
            break
        end
        r = SVector(x, y, z)
        v = SVector(.0, .0, .0)
        body = ChargedParticle(r, v, m, q)
        push!(bodies, body)
        count += 1
    end

    k = 9e9
    τ = 0.01 * dL / sqrt(2 * k * q * q / (dL * m))
    t1 = 0.0
    t2 = 1000 * τ
    potential = ElectrostaticParameters(k, 0.45 * L)
    system = PotentialNBodySystem(bodies, Dict(:electrostatic => potential))
    pbc = CubicPeriodicBoundaryConditions(L)
    simulation = NBodySimulation(system, (t1, t2), pbc)

    f = acceleration(simulation)
    u0, v0, n = gather_bodies_initial_coordinates(simulation)
    dv = zero(v0)

    b = @benchmarkable $f(dv, $v0, $u0, $potential, 0.) setup=(dv=zero($v0)) evals=1

    SUITE["coulomb"] = b
end
```

```
Error: LoadError: UndefVarError: @benchmarkable not defined
in expression starting at none:34
```

## 0.3 Magnetic dipole potential

```
let SUITE=SUITE
    n = 200
    bodies = MagneticParticle[]
    L = 20.0
    m = 1.0
    count = 1
    dL = L / (ceil(n^(1 / 3)) + 1)
    for x = dL / 2:dL:L, y = dL / 2:dL:L, z = dL / 2:dL:L
        if count > n
            break
        end
        r = SVector(x, y, z)
        v = SVector(.0, .0, .0)
        mm = rand(SVector{3})
        body = MagneticParticle(r, v, m, mm)
        push!(bodies, body)
        count += 1
    end

    μ_4π = 1e-7
    t1 = 0.0  # s
    t2 = 1.0 # s
    τ = (t2 - t1) / 100

    parameters = MagnetostaticParameters(μ_4π)
    system = PotentialNBodySystem(bodies, Dict(:magnetic => parameters))
    simulation = NBodySimulation(system, (t1, t2))

    f = acceleration(simulation)
    u0, v0, n = gather_bodies_initial_coordinates(simulation)
    dv = zero(v0)

    b = @benchmarkable $f(dv, $v0, $u0, $parameters, 0.) setup=(dv=zero($v0)) evals=1

    SUITE["magnetic_dipole"] = b
end
```

```
Error: LoadError: UndefVarError: @benchmarkable not defined
in expression starting at none:33
```

## 0.4 Lennard Jones potential

```
let SUITE=SUITE
    T = 120.0 # K
    T0 = 90.0 # K
    kb = 8.3144598e-3 # kJ/(K*mol)
    ϵ = T * kb
    σ = 0.34 # nm
    ρ = 1374/1.6747# Da/nm^3
    N = 200
    m = 39.95# Da = 216 # number of bodies/particles
    L = (m*N/ρ)^(1/3)#10.229σ
    R = 0.5*L
    v_dev = sqrt(kb * T / m)
    bodies = generate_bodies_in_cell_nodes(N, m, v_dev, L)

    τ = 0.5e-3 # ps or 1e-12 s
    t1 = 0.0
    t2 = 2000τ

    lj_parameters = LennardJonesParameters(ϵ, σ, R)
    lj_system = PotentialNBodySystem(bodies, Dict(:lennard_jones => lj_parameters));

    pbc = CubicPeriodicBoundaryConditions(L)
    simulation = NBodySimulation(lj_system, (t1, t2), pbc, kb)

    f = acceleration(simulation)
    u0, v0, n = gather_bodies_initial_coordinates(simulation)
    dv = zero(v0)

    b = @benchmarkable $f(dv, $v0, $u0, $lj_parameters, 0.) setup=(dv=zero($v0)) evals=1

    SUITE["lennard_jones"] = b
end
```

```
Error: LoadError: UndefVarError: @benchmarkable not defined
in expression starting at none:29
```

## 0.5 WaterSPCFw model

```
function acceleration(simulation::NBodySimulation{<:WaterSPCFw})

    (u0, v0, n) = gather_bodies_initial_coordinates(simulation)

    (o_acelerations, h_acelerations) = gather_accelerations_for_potentials(simulation)
    group_accelerations = gather_group_accelerations(simulation)
    simultaneous_acceleration = gather_simultaneous_acceleration(simulation)

    function soode_system!(dv, v, u, p, t)
        @inbounds for i = 1:n
            a = MVector(0.0, 0.0, 0.0)
            for acceleration! in o_acelerations
                acceleration!(a, u, v, t, 3 * (i - 1) + 1);
            end
            dv[:, 3 * (i - 1) + 1]  .= a
        end
        @inbounds for i in 1:n, j in (2, 3)
            a = MVector(0.0, 0.0, 0.0)
            for acceleration! in h_acelerations
```

```
                acceleration!(a, u, v, t, 3 * (i - 1) + j);
            end
            dv[:, 3 * (i - 1) + j]   .= a
        end
        @inbounds for i = 1:n
            for acceleration! in group_accelerations
                acceleration!(dv, u, v, t, i);
            end
        end
        for acceleration! in simultaneous_acceleration
            acceleration!(dv, u, v, t);
        end
    end

    return soode_system!
end
```

```
Error: UndefVarError: WaterSPCFw not defined
```

```
let SUITE=SUITE
    T = 370 # K
    T0 = 275 # K
    kb = 8.3144598e-3 # kJ/(K*mol)
    ϵOO = 0.1554253*4.184 # kJ
    σOO = 0.3165492 # nm
    ρ = 997/1.6747# Da/nm^3
    mO = 15.999 # Da
    mH = 1.00794 # Da
    mH2O = mO+2*mH
    N = 200
    L = (mH2O*N/ρ)^(1/3)
    R = 0.9 # ~3*σOO
    Rel = 0.49*L
    v_dev = sqrt(kb * T /mH2O)
    τ = 0.5e-3 # ps
    t1 = 0τ
    t2 = 5τ # ps
    k_bond = 1059.162*4.184*1e2 # kJ/(mol*nm^2)
    k_angle = 75.90*4.184 # kJ/(mol*rad^2)
    rOH = 0.1012 # nm
    ∠HOH = 113.24*pi/180 # rad
    qH = 0.41
    qO = -0.82
    k = 138.935458 #
    bodies = generate_bodies_in_cell_nodes(N, mH2O, v_dev, L)
    jl_parameters = LennardJonesParameters(ϵOO, σOO, R)
    e_parameters = ElectrostaticParameters(k, Rel)
    spc_parameters = SPCFwParameters(rOH, ∠HOH, k_bond, k_angle)
    pbc = CubicPeriodicBoundaryConditions(L)
    water = WaterSPCFw(bodies, mH, mO, qH, qO,  jl_parameters, e_parameters,
spc_parameters);
    simulation = NBodySimulation(water, (t1, t2), pbc, kb);

    f = acceleration(simulation)
    u0, v0, n = gather_bodies_initial_coordinates(simulation)
    dv = zero(v0)

    b = @benchmarkable $f(dv, $v0, $u0, $spc_parameters, 0.) setup=(dv=zero($v0)) evals=1

    SUITE["water_spcfw"] = b
```

```
end
```

```
Error: LoadError: UndefVarError: @benchmarkable not defined
in expression starting at none:39
```

Here are the results of the benchmarks

```
r = run(SUITE)
```

```
Error: UndefVarError: SUITE not defined
```

```
minimum(r)
```

```
Error: UndefVarError: r not defined
```

and

```
memory(r)
```

```
Error: UndefVarError: memory not defined
```