ADAMS–BASHFORTH–MOULTON PREDICTOR–CORRECTORS

The basic idea of the predictor-corrector method is that a first step "predicts" information about step n+1 just from information about step n. This step is called the explicit step. Then a second step will use that first estimation to get a more "correct" result. This is called the implicit step.

In an Adams-Bashforth-Moulton algorithm, the explicit step is an Adams-Bashforth method, and the implicit step is an Adams-Moulton method. Both of these methods will depend on the previous value of the function being evaluated and on its derivatives at m timesteps, where m is the order of the method. Notation (ie, indices) have been chosen for convenience when programming in C.

Adams-Bashforth

$$y_{n+1} = y_n + \Delta t \sum_{k=0}^{m-1} \alpha_k y'_{n-k}$$

order	α_0	α_1	α_2	α_3	α_4
3	23/12	-16/12	5/12	—	—
4	55/24	-59/24	37/24	-9/24	—
5	1901/720	-2774/720	2616/720	-1274/720	251/720

Adams-Moulton

$$y_{n+1} = y_n + \Delta t \sum_{k=0}^{m-1} \beta_k y'_{n-k+1}$$

$n-k+1 = n-(k-1)$

order	β_0	β_1	β_2	β_3	β_4
3	5/12	8/12	-1/12	—	—
4	9/24	19/24	-5/24	1/24	—
5	251/720	646/720	-264/720	106/720	-19/720

Before this algorithm can be used, "old" values must be found. Use a one-step method (eg Runge-Kutta) to do the first few steps.

Using ABM to integrate the equations of motion

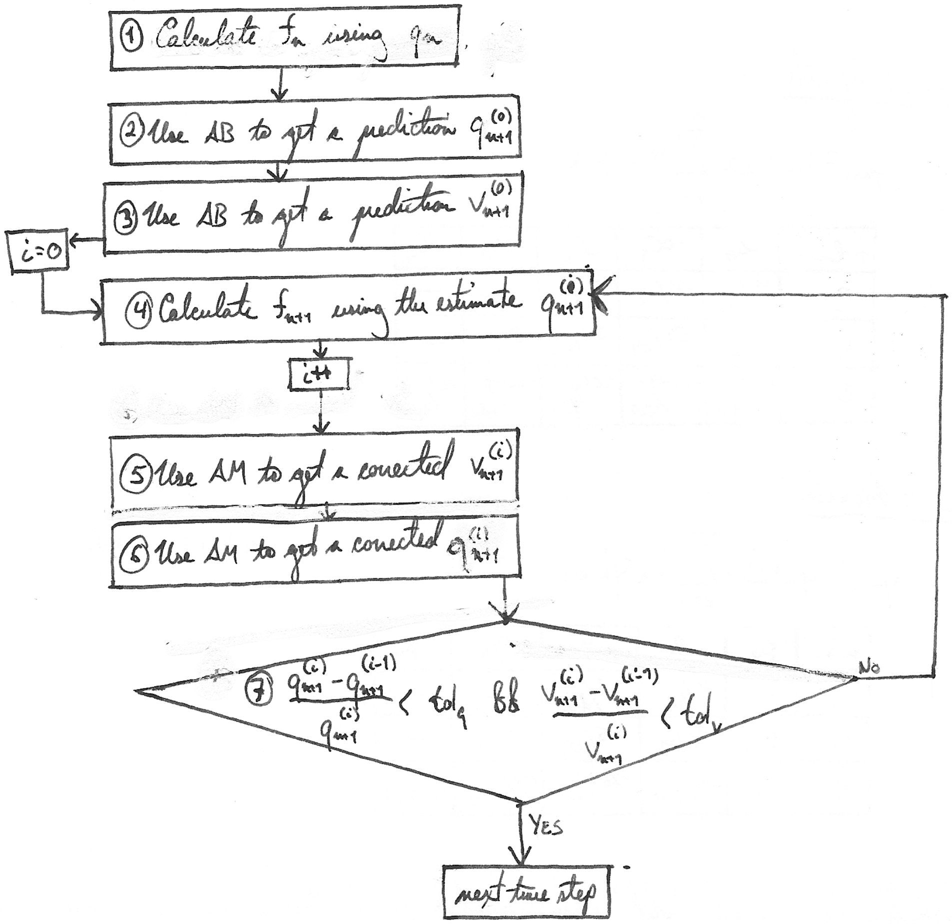
Will actually use ABM to propagate both the positions and the velocities. For the velocities, we make the assignments

$$y \mapsto \text{velocity}$$
$$y' \mapsto \text{acceleration} = \text{force}/\text{mass}$$

For the positions, the assignments are

$$y \mapsto \text{position}$$
$$y' \mapsto \text{velocity}$$

The algorithm is sketched below:



The iteration to self-consistency is optional (and probably not as effective as reducing the timestep)

NB: This algorithm uses the corrected V_{n+1} to correct q_{n+1}. This makes more sense to me, but if you didn't want to do that for some reason, just switch steps 5 and 6