# *Initial value problems: multi-step methods*

In the last lecture, we considered a variety of one-step methods. We will go through a similar process with multi-step methods.

## 1 *Multi-step methods*

You will recall from the last lecture that finite difference methods for IVPs amount to locally interpolating the IVP $\dot{\boldsymbol{u}} = \boldsymbol{f}(\boldsymbol{u}, t)$. We moreover identified that many (but not all!) of these interpolation procedures were derived by integrating the IVP from $t_k$ to $t_{k+1}$:

$$\boldsymbol{u}(t_{k+1}) - \boldsymbol{u}(t_k) = \int_{t_k}^{t_{k+1}} \boldsymbol{f}(\boldsymbol{u}, t)\, dt \tag{1}$$

Remember that (1) arose from integrating the IVP from $t_k$ to $t_{k+1}$ and noting that the left-hand side $\int_{t_k}^{t_{k+1}} \dot{\boldsymbol{u}}\, dt$ could be *exactly* represented as $\boldsymbol{u}(t_{k+1}) - \boldsymbol{u}(t_k)$.

and using a local interpolation rule to approximate the righthand side term.

In this lecture, we will motivate the general definition of multi-step methods—which applies irrespective of whether the method was derived from (1)—through a class of methods called *Adams* methods, which are defined using (1).

In a few lectures, we will consider multi-step methods derived using the differential form $\dot{\boldsymbol{u}} = \boldsymbol{f}(\boldsymbol{u}, t)$ instead of the integrated variant (1). These methods will arise in the context of particularly nefarious IVPs referred to as *stiff* equations. We will define these and characterize the challenges in simulating them soon!

Notice that since the left-hand side of (1) is exactly correct, the inexactness in our numerical approximation will arise from interpolating $\boldsymbol{f}(\boldsymbol{u}, t)$ locally near the interval $t \in [t_k, t_{k+1}]$. A natural means to reduce the error in this approximation is to use a higher order approximation of $\boldsymbol{f}$ over this time interval. Whereas multi-stage one-step methods do this by creating additional points within the time interval (recall, *e.g.*, the RK4 method from the last lecture), multi-step methods utilize the broader interval $t \in [t_{k-j}, t_{k+1}]$ $(j \geq 1)$.

Note the value of utilizing the earlier time instances $t \in [t_{k-j}, t_{k-1}]$: we already have the solution at these times, so we can avoid additional functional evaluations!

Adams methods approximate $\boldsymbol{f}(\boldsymbol{u}, t)$ as a polynomial over this extended interval, and integrate this polynomial from $t_k$ to $t_{k+1}$ to evaluate the righthand side of (1). Adams methods can be divided into explicit (*Adams-Bashforth*) and implicit (*Adams-Moulton*) methods.

Adams-Bashforth methods (explicit):

Since Adams-Bashforth methods are explicit, the interval *does not* involve $t_{k+1}$.

To arrive at an $r$-step Adams-Bashforth method, we represent $\boldsymbol{f}$ as a degree $r-1$ polynomial using the interval $t \in [t_{k-r+1}, t_k]$. The result of this process is that (1) can be approximated as

$$\boldsymbol{u}_{k+1} - \boldsymbol{u}_k = \Delta t \sum_{j=k-r+1}^{k} \beta_{j-(k-r+1)} \boldsymbol{f}(\boldsymbol{u}_j, t_j) \tag{2}$$

Note that, as with one-step methods, we have replaced $\boldsymbol{u}(t_k)$ with $\boldsymbol{u}_k$ to reflect the fact that we do not have access to the true solution.
The somewhat cumbersome subscript on $\beta$ ensures that the index runs from $0, \dots, r-1$.

where the $\beta_{j-(k-r+1)}$ $(j = k-r+1, \dots, k)$ are determined by our polynomial interpolant. Let us consider some examples.

2-step (degree 1 interpolant) Adams-Bashforth method:

$$\boldsymbol{u}_{k+1} - \boldsymbol{u}_k = \frac{\Delta t}{2}\big[ -\boldsymbol{f}(\boldsymbol{u}_{k-1}, t_{k-1}) + 3\boldsymbol{f}(\boldsymbol{u}_k, t_k)\big] \tag{3}$$

3-STEP (DEGREE 2 INTERPOLANT) ADAMS-BASHFORTH METHOD:

$$\boldsymbol{u}_{k+1} - \boldsymbol{u}_k = \frac{\Delta t}{12}\big[5\boldsymbol{f}(\boldsymbol{u}_{k-2}, t_{k-2}) - 16\boldsymbol{f}(\boldsymbol{u}_{k-1}, t_{k-1}) + 23\boldsymbol{f}(\boldsymbol{u}_k, t_k)\big] \tag{5}$$

Of course, there is no need for us to stop here. We could continue to derive Adams-Bashforth methods using increasingly high-order polynomials. Extremely high-order Adams-Bashforth methods are rarely encountered in practice—indeed, Adams-Bashforth methods involving more than four steps are not often seen.

There is no mystery in how we obtained the various $\beta$ coefficients for these 2-step and 3-step example cases. In the case of the 2-step method, we expressed $\boldsymbol{f}$ in terms of our old friends the Lagrange polynomials. Specifically,

$$\boldsymbol{f}(\boldsymbol{u}, t) \approx \boldsymbol{f}(\boldsymbol{u}_{k-1}, t_{k-1})L_{k-1}(t) + \boldsymbol{f}(\boldsymbol{u}_k, t_k)L_k(t) \tag{4}$$

We then integrated this from $t_k$ to $t_{k+1}$ to arrive at (3). How would you derive the 3-step method?

The rarity of high-order Adams methods is rooted in the fact that high-order polynomial interpolation in uniformly spaced points is a recipe for disaster.

ADAMS-MOULTON METHODS (IMPLICIT):

To arrive at an $r$-step Adams-Moulton method, we use the larger interval $t \in [t_{k-r+1}, t_{k+1}]$. To accommodate the extra time instance $t_{k+1}$, we represent $\boldsymbol{f}$ as a degree $r$ polynomial (not $r-1$). The result of this process is that (1) can be approximated as

$$\boldsymbol{u}_{k+1} - \boldsymbol{u}_k = \Delta t \sum_{j=k-r+1}^{k+1} \beta_{j-(k-r+1)} \boldsymbol{f}(\boldsymbol{u}_j, t_j) \tag{6}$$

where again the $\beta_{j-(k-r+1)}$ $(j = k-r+1, \ldots, k+1)$ are determined by our polynomial interpolant. Let us consider some examples.

Just as with one-step methods, we will see that the extra work incurred from these implicit methods comes with the benefit of larger stability regions.

Note that an $r$-step Adams-Moulton method involves a degree $r$ polynomial interpolant, whereas an $r$-step Adams-Bashforth method involves a degree $r-1$ polynomial interpolant

2-STEP (DEGREE 2 INTERPOLANT) ADAMS-MOULTON METHOD:

$$\boldsymbol{u}_{k+1} - \boldsymbol{u}_k = \frac{\Delta t}{12}\big[-\boldsymbol{f}(\boldsymbol{u}_{k-1}, t_{k-1}) + 8\boldsymbol{f}(\boldsymbol{u}_k, t_k) + 5\boldsymbol{f}(\boldsymbol{u}_{k+1}, t_{k+1})\big] \tag{7}$$

3-STEP (DEGREE 3 INTERPOLANT) ADAMS-MOULTON METHOD:

$$\boldsymbol{u}_{k+1} - \boldsymbol{u}_k = \frac{\Delta t}{24}\big[\boldsymbol{f}(\boldsymbol{u}_{k-2}, t_{k-2}) - 5\boldsymbol{f}(\boldsymbol{u}_{k-1}, t_{k-1}) + 19\boldsymbol{f}(\boldsymbol{u}_k, t_k) + 9\boldsymbol{f}(\boldsymbol{u}_{k+1}, t_{k+1})\big] \tag{8}$$

The Adams-Bashforth and Adams-Moulton methods form an important class of multi-step methods, but are not the only family of multi-step methods in existence. A more general expression that encompasses a wider range of multi-step methods (including the Adams-Bashforth and Adams-Moulton methods) is

*General form*: multi-step methods for IVPs

An $r$-step method is defined by

$$\sum_{j=k-r+1}^{k+1} \alpha_{j-(k-r+1)} \boldsymbol{u}_j = \Delta t \sum_{j=k-r+1}^{k+1} \beta_{j-(k-r+1)} \boldsymbol{f}(\boldsymbol{u}_j, t_j) \tag{9}$$

where $\alpha_{j-(k-r+1)}, \beta_{j-(k-r+1)} \in \mathbb{R}$ and $j = k-r+1, \ldots, k+1$.

Why do we incorporate more terms on the left-hand side than $\boldsymbol{u}_{k+1}$ and $\boldsymbol{u}_k$ when we know the left-hand side of the IVP $\dot{\boldsymbol{u}} = \boldsymbol{f}(\boldsymbol{u}, t)$ can be evaluated exactly as $\boldsymbol{u}(t_{k+1}) - \boldsymbol{u}(t_k)$? This gives us extra flexibility to cancel out additional error terms that arise from a Taylor series expansion of the various $\boldsymbol{f}(\boldsymbol{u}_j, t_j)$ terms $(j = k-r+1, \ldots, k+1)$.

Notice that the Adams-Bashforth and Adams-Moulton methods can be represented using the form (9) with $\alpha_r = 1, \alpha_{r-1} = -1, \alpha_m = 0$ $(m < r-1)$. What are the $\beta$ coefficients for the 3-step Adams-Bashforth method?

STARTING VALUES:

There remains one unexplored feature to successfully implementing a multi-step method. Consider that we are at time $t_0$ and wish to advance to $t_1$. An $r$-step method would require information of the approximate solution at times $t_{-r+1}, \ldots, t_{-1}, t_0, t_1$. Of course, we do not have access to information prior to our initial time instance $t_0$. How do we remedy this? We use a one-step method of appropriate accuracy to advance the solution to time $t_{r-1}$, and let our multi-step method take over from there.