Math 6630: Analysis of Numerical Methods, II
Solvers for initial value problems, Part IV

See Ascher and Petzold 1998, Chapters 1-5

Akil Narayan[1]

[1] **Department of Mathematics, and Scientific Computing and Imaging (SCI) Institute**
University of Utah

$$\boldsymbol{u}'(t) = \boldsymbol{f}(t;\boldsymbol{u}), \qquad \boldsymbol{u}(0) = \boldsymbol{u}_0.$$

$$\boldsymbol{u}_n \approx \boldsymbol{u}(t_n)$$

$$\boldsymbol{u}_{n+1} \approx \boldsymbol{u}_n + \int_{t_n}^{t_{n+1}} \boldsymbol{f}(t,\boldsymbol{u}(t))\mathrm{d}t$$

We have previously discussed

- Simple schemes: forward/backward Euler, Crank-Nicolson
- Consistency and LTE
- 0-stability and scheme convergence
- absolute/A-stability and consequences
- multi-stage (Runge-Kutta) methods

Finally, we'll discuss multi-step schemes.

Preliminaries: polynomial interpolation

To begin we review some basic concepts about (univariate) polynomial interpolation:

Let $f : \mathbb{R} \to \mathbb{R}$ be a scalar function, and let $x_0, \ldots, x_n$ be any distinct points on $\mathbb{R}$.

Theorem

There is a unique polynomial $p(x)$ of degree n such that $f(x_j) = p(x_j)$ for all $j = 0, \ldots, n$.

Preliminaries: polynomial interpolation

To begin we review some basic concepts about (univariate) polynomial interpolation:

Let $f : \mathbb{R} \to \mathbb{R}$ be a scalar function, and let $x_0, \ldots, x_n$ be any distinct points on $\mathbb{R}$.

Theorem

There is a unique polynomial $p(x)$ of degree n such that $f(x_j) = p(x_j)$ for all $j = 0, \ldots, n$.

One way to construct this polynomial is via divided differences. Define

$$f[x_j] = f(x_j), \quad f[x_j, \ldots, x_{j+\ell}] = \frac{f[x_{j+1}, \ldots, x_{j+\ell}] - f[x_j, \ldots, x_{j+\ell-1}]}{x_{j+\ell} - x_j},$$

which are approximations to ℓth derivatives. Then,

$$p(x) = \sum_{\ell=0}^{n} f[x_0, \ldots, x_j] \prod_{j=0}^{\ell-1} (x - x_j).$$

This is the Newton form of the interpolating polynomial.

Preliminaries: polynomial interpolation

To begin we review some basic concepts about (univariate) polynomial interpolation:

Let $f : \mathbb{R} \to \mathbb{R}$ be a scalar function, and let $x_0, \ldots, x_n$ be any distinct points on $\mathbb{R}$.

Theorem

There is a unique polynomial $p(x)$ of degree n such that $f(x_j) = p(x_j)$ for all $j = 0, \ldots, n$.

One way to construct this polynomial is via divided differences. Define

$$f[x_j] = f(x_j), f[x_j, \ldots, x_{j+\ell}] = \frac{f[x_{j+1}, \ldots, x_{j+\ell}] - f[x_j, \ldots, x_{j+\ell-1}]}{x_{j+\ell} - x_j},$$

which are approximations to ℓth derivatives. Then,

$$p(x) = \sum_{\ell=0}^{n} f[x_0, \ldots, x_j] \prod_{j=0}^{\ell-1} (x - x_j).$$

This is the Newton form of the interpolating polynomial.

If $x_j = x_0 + jk$ for some $k > 0$, then expressions simplify considerably and more explicit formulas can be derived.

Simple theory for linear difference equations parallels linear differential equations:

$$u^{(s)}(t) + \sum_{j=1}^{s} \alpha_j u^{(s-j)}(t) = 0, \qquad u^{(j)}(0) = u_0^j, \qquad j = 0, \ldots, s-1.$$

Solve for a function $u(t)$, $t > 0$. The order is $s > 0$.

$$u_n + \sum_{j=1}^{s} \alpha_j u_{n-j} = 0, \qquad u_{n-j} = u_{n-j,0}, \qquad j = 1, \ldots, s.$$

Solve for a sequence u_ℓ, $\ell \geqslant 0$. The order is $s > 0$.

Simple theory for linear difference equations parallels linear differential equations:

$$u^{(s)}(t) + \sum_{j=1}^{s} \alpha_j u^{(s-j)}(t) = 0, \qquad u^{(j)}(0) = u_0^j, \qquad j = 0, \ldots, s-1.$$

Solve for a function $u(t)$, $t > 0$. The order is $s > 0$.

$$\text{Ansatz } u(t) = e^{zt} \implies p(z) := \sum_{j=0}^{s} \alpha_j z^{s-j} = 0, \quad (\alpha_0 = 1)$$

Solutions take the form $u(t) \sim e^{z_j t}$, where $z_1, \ldots, z_s$ are the roots of p.

$$u_n + \sum_{j=1}^{s} \alpha_j u_{n-j} = 0, \qquad u_{n-j} = u_{n-j,0}, \qquad j = 1, \ldots, s.$$

Solve for a sequence u_ℓ, $\ell \geqslant 0$. The order is $s > 0$.

$$\text{Ansatz } u_n = z^n \implies p(z) := \sum_{j=0}^{s} \alpha_j z^{s-j} = 0, \quad (\alpha_0 = 1)$$

Solutions take the form $u_n \sim z_j^n$, where $z_1, \ldots, z_s$ are the roots of p.

Simple theory for linear difference equations parallels linear differential equations:

$$u^{(s)}(t) + \sum_{j=1}^{s} \alpha_j u^{(s-j)}(t) = 0, \qquad u^{(j)}(0) = u_0^j, \qquad j = 0, \ldots, s-1.$$

Solve for a function $u(t)$, $t > 0$. The order is $s > 0$.

$$\text{Ansatz } u(t) = e^{zt} \implies p(z) := \sum_{j=0}^{s} \alpha_j z^{s-j} = 0, \quad (\alpha_0 = 1)$$

Solutions take the form $u(t) \sim e^{z_j t}$, where $z_1, \ldots, z_s$ are the roots of p.

Solutions $u(t)$ are stable if $\Re z_j \leqslant 0$. (Asymptotically stable if $\Re z_j < 0$.)

$$u_n + \sum_{j=1}^{s} \alpha_j u_{n-j} = 0, \qquad u_{n-j} = u_{n-j,0}, \qquad j = 1, \ldots, s.$$

Solve for a sequence u_ℓ, $\ell \geqslant 0$. The order is $s > 0$.

$$\text{Ansatz } u_n = z^n \implies p(z) := \sum_{j=0}^{s} \alpha_j z^{s-j} = 0, \quad (\alpha_0 = 1)$$

Solutions take the form $u_n \sim z_j^n$, where $z_1, \ldots, z_s$ are the roots of p.

Solutions u_n are stable if $|z_j| \leqslant 1$. (Asymptotically stable if $|z_j| < 1$.)

For the IVP,

$$\boldsymbol{u}'(t) = \boldsymbol{f}(t; \boldsymbol{u}), \qquad \boldsymbol{u}(0) = \boldsymbol{u}_0.$$

$$\boldsymbol{u}_n \approx \boldsymbol{u}(t_n)$$

a general s-step multi-step scheme with timestep k has the form,

$$\frac{du}{dt} \simeq \sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}(t_{n+1-j}, \boldsymbol{u}_{n+1-j}), \simeq \int_{t_n}^{t_{n+1}} f(t, u(t))\, dt \qquad \alpha_j, \beta_j \in \mathbb{R}$$

Multi-step methods, I

For the IVP,

$$\boldsymbol{u}'(t) = \boldsymbol{f}(t;\boldsymbol{u}), \qquad \boldsymbol{u}(0) = \boldsymbol{u}_0.$$

$$\boldsymbol{u}_n \approx \boldsymbol{u}(t_n)$$

a general s-step multi-step scheme with timestep k has the form,

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}(t_{n+1-j}, \boldsymbol{u}_{n+1-j}), \qquad \alpha_j, \beta_j \in \mathbb{R}$$

Comments:

- $s = 1$ corresponds to a general single-step (and single-stage) method

$$\alpha_0 u_{n+1} + \alpha_1 u_n = k\beta_0 f(t_{n+1}, u_{n+1}) + k\beta_1 f(t_n, u_n)$$

Multi-step methods, I

For the IVP,

$$\boldsymbol{u}'(t) = \boldsymbol{f}(t; \boldsymbol{u}), \qquad \boldsymbol{u}(0) = \boldsymbol{u}_0.$$
$$\boldsymbol{u}_n \approx \boldsymbol{u}(t_n)$$

a general s-step multi-step scheme with timestep k has the form,

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}(t_{n+1-j}, \boldsymbol{u}_{n+1-j}), \qquad \alpha_j, \beta_j \in \mathbb{R}$$

Comments:

- $s = 1$ corresponds to a general single-step (and single-stage) method
- $s > 1$: we need time history, e.g., $\boldsymbol{u}_{n-2}, \boldsymbol{u}_{n-3}, \ldots$

Multi-step methods, I

For the IVP,

$$\boldsymbol{u}'(t) = \boldsymbol{f}(t; \boldsymbol{u}), \qquad \boldsymbol{u}(0) = \boldsymbol{u}_0.$$

$$\boldsymbol{u}_n \approx \boldsymbol{u}(t_n)$$

a general s-step multi-step scheme with timestep k has the form,

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}(t_{n+1-j}, \boldsymbol{u}_{n+1-j}), \qquad \alpha_j, \beta_j \in \mathbb{R}$$

Comments:

- $s = 1$ corresponds to a general single-step (and single-stage) method
- $s > 1$: we need time history, e.g., $\boldsymbol{u}_{n-2}, \boldsymbol{u}_{n-3}, \ldots$
- We assume $\alpha_0 \neq 0$. $\alpha_0 u_{n+1}$
- We can rescale the equation by a constant without changing anything: we fix this freedom by setting $\alpha_0 = 1$.

Multi-step methods, I

For the IVP,

$$\boldsymbol{u}'(t) = \boldsymbol{f}(t;\boldsymbol{u}), \qquad \boldsymbol{u}(0) = \boldsymbol{u}_0.$$
$$\boldsymbol{u}_n \approx \boldsymbol{u}(t_n)$$

a general s-step multi-step scheme with timestep k has the form,

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}(t_{n+1-j}, \boldsymbol{u}_{n+1-j}), \qquad \alpha_j, \beta_j \in \mathbb{R}$$

Comments:

- $s = 1$ corresponds to a general single-step (and single-stage) method
- $s > 1$: we need time history, e.g., $\boldsymbol{u}_{n-2}, \boldsymbol{u}_{n-3}, \ldots$
- We assume $\alpha_0 \neq 0$.
- We can rescale the equation by a constant without changing anything: we fix this freedom by setting $\alpha_0 = 1$.
- To avoid some minor pathologies, we typically assume that either $\alpha_j \neq 0$ or $\beta_j \neq 0$ for every j.

For the IVP,

$$\boldsymbol{u}'(t) = \boldsymbol{f}(t; \boldsymbol{u}), \qquad \boldsymbol{u}(0) = \boldsymbol{u}_0.$$
$$\boldsymbol{u}_n \approx \boldsymbol{u}(t_n)$$

a general s-step multi-step scheme with timestep k has the form,

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}(t_{n+1-j}, \boldsymbol{u}_{n+1-j}), \qquad \alpha_j, \beta_j \in \mathbb{R}$$

Comments:

- $s = 1$ corresponds to a general single-step (and single-stage) method
- $s > 1$: we need time history, e.g., $\boldsymbol{u}_{n-2}, \boldsymbol{u}_{n-3}, \ldots$
- We assume $\alpha_0 \neq 0$.
- We can rescale the equation by a constant without changing anything: we fix this freedom by setting $\alpha_0 = 1$.
- To avoid some minor pathologies, we typically assume that either $\alpha_j \neq 0$ or $\beta_j \neq 0$ for every j.
- $\beta_0 \neq 0$ corresponds to an implicit method. $\beta_0 = 0$ is an explicit method.

Multi-step methods, II

To simplify notation, we will assume the ODE is autonomous ($\boldsymbol{f}(t, \boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{u})$), and will abbreviate $\boldsymbol{f}(\boldsymbol{u}_j)$ as $\boldsymbol{f}_j$. Then the multi-step method takes the form,

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}_{n+1-j}$$

Multi-step methods, II

To simplify notation, we will assume the ODE is autonomous ($\boldsymbol{f}(t, \boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{u})$), and will abbreviate $\boldsymbol{f}(\boldsymbol{u}_j)$ as $\boldsymbol{f}_j$. Then the multi-step method takes the form,

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}_{n+1-j}$$

Generally speaking, the constants are chosen so that:

- The α_j approximate $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}(t_n)$
- The β_j approximate $\frac{1}{k}\int_{t_n}^{t_{n+1}} \boldsymbol{f}(\boldsymbol{u}(r))\mathrm{d}r$

To simplify notation, we will assume the ODE is autonomous ($\boldsymbol{f}(t, \boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{u})$), and will abbreviate $\boldsymbol{f}(\boldsymbol{u}_j)$ as $\boldsymbol{f}_j$. Then the multi-step method takes the form,

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}_{n+1-j}$$

Generally speaking, the constants are chosen so that:

- The α_j approximate $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u}(t_n)$
- The β_j approximate $\frac{1}{k}\int_{t_n}^{t_{n+1}} \boldsymbol{f}(\boldsymbol{u}(r))\mathrm{d}r$

There are some miscellaneous issues we'll answer later, e.g.,

- If $s \geqslant 2$, how is $\boldsymbol{u}_1$ computed from $\boldsymbol{u}_0$?
- Must we fix the time-step k?

A warmup: single-step specializations

Specializing to single-step methods ($s = 1$) yields a transparent family of methods:

$$\boldsymbol{u}_{n+1} + \alpha_1 \boldsymbol{u}_n = k \left(\beta_0 \boldsymbol{f}_{n+1} + \beta_1 \boldsymbol{f}_n \right).$$

(Recall $\alpha_0 = 1$)

A warmup: single-step specializations

Specializing to single-step methods ($s = 1$) yields a transparent family of methods:

$$\boldsymbol{u}_{n+1} + \alpha_1 \boldsymbol{u}_n = k \left(\beta_0 \boldsymbol{f}_{n+1} + \beta_1 \boldsymbol{f}_n \right).$$

(Recall $\alpha_0 = 1$)

For any reasonable notion of consistency (to approximate $\boldsymbol{u}'(t_n)$), we should take $\alpha_1 = -1$.

Specializing to single-step methods ($s = 1$) yields a transparent family of methods:

$$\boldsymbol{u}_{n+1} + \alpha_1 \boldsymbol{u}_n = k \left(\beta_0 \boldsymbol{f}_{n+1} + \beta_1 \boldsymbol{f}_n \right).$$

(Recall $\alpha_0 = 1$)

For any reasonable notion of consistency (to approximate $\boldsymbol{u}'(t_n)$), we should take $\alpha_1 = -1$.

With this restriction, then we have

$$\boldsymbol{u}_{n+1} - \boldsymbol{u}_n = k \left(\beta_0 \boldsymbol{f}_{n+1} + \beta_1 \boldsymbol{f}_n \right),$$

and hence the right hand side should approximate $\int_{t_n}^{t_{n+1}} \boldsymbol{f}(\boldsymbol{u}(r)) \mathrm{d}r$, requiring $\beta_0 + \beta_1 = 1$ for consistency.

Specializing to single-step methods ($s = 1$) yields a transparent family of methods:

$$\boldsymbol{u}_{n+1} + \alpha_1 \boldsymbol{u}_n = k \left(\beta_0 \boldsymbol{f}_{n+1} + \beta_1 \boldsymbol{f}_n \right).$$

(Recall $\alpha_0 = 1$)

For any reasonable notion of consistency (to approximate $\boldsymbol{u}'(t_n)$), we should take $\alpha_1 = -1$.

With this restriction, then we have

$$\boldsymbol{u}_{n+1} - \boldsymbol{u}_n = k \left(\beta_0 \boldsymbol{f}_{n+1} + \beta_1 \boldsymbol{f}_n \right),$$

and hence the right hand side should approximate $\int_{t_n}^{t_{n+1}} \boldsymbol{f}(\boldsymbol{u}(r)) \mathrm{d}r$, requiring $\beta_0 + \beta_1 = 1$ for consistency.

Then our general family of methods is

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + k \left(\beta \boldsymbol{f}_{n+1} + (1 - \beta) \boldsymbol{f}_n \right),$$

specializing to,

- $\beta = 0$: Forward Euler
- $\beta = 1$: Backward Euler
- $\beta = 1/2$: Crank-Nicolson

Can we derive 2-step methods?

$s = 2$

$$\alpha_0 U_{n+1} + \alpha_1 U_n + \alpha_2 U_{n-1} = k\beta_0 f_{n+1} + k\beta_1 f_n + k\beta_2 f_{n-1}$$

- set $\alpha_0 = 1$
- explicit scheme: $\beta_0 = 0$.

$$U_{n+1} + \alpha_1 U_n + \alpha_2 U_{n-1} = k\beta_1 f_n + k\beta_2 f_{n-1} \qquad \alpha_1, \alpha_2, \beta_1, \beta_2 = ?$$

enforce consistency by "minimizing" LTE.

$$\text{LTE: } \frac{1}{k}\left(u(t_{n+1}) + \alpha_1 u(t_n) + \alpha_2 u(t_{n-1})\right) - \beta_1 f(t_n, u(t_n)) - \beta_2 f(t_{n-1}, u(t_{n-1}))$$

Taylor expand @ $t = t_{n-1}$

abbreviations: $u = u(t_{n-1})$
$u' = u'(t_{n-1})$
$\vdots$
$u^{(j)} = u^{(j)}(t_{n-1})$

$$u(t_{n+1}) = u + 2ku' + \frac{(2k)^2}{2} u'' + \frac{(2k)^3}{6} u''' + \dots$$

using: $u' = f$, $u^{(j)} = \frac{d^{j-1}}{dt^{j-1}} f(t, u(t))$

$$f(t_n, u(t_n)) = f + kf' + k^2/2\, f'' + k^3/6\, f''' + \dots$$
$$= u' + ku'' + \frac{k^2}{2} u''' + \dots$$

$$\text{LTE: } \frac{1}{k}\left(u + 2ku' + 2k^2 u'' + \frac{4}{3}k^3 u''' + \dots\right)$$
$$+ \frac{\alpha_1}{k}\left(u + ku' + k^2/2\, u'' + k^3/6\, u''' + \dots\right)$$
$$+ \frac{\alpha_2}{k} u$$

$$-\beta_1 \left[u' + k u'' + \frac{k^2}{2} u''' + \dots \right]$$

$$-\beta_2 u'$$

$\frac{1}{k}$: $1 + \alpha_1 + \alpha_2 = 0$

1: $2 + \alpha_1 - \beta_1 - \beta_2 = 0$

k: $2 + \alpha_1/2 - \beta_1 = 0 \longrightarrow 4 + \alpha_1 - 2\beta_1 = 0$

k^2: $\frac{4}{3} + \frac{\alpha_1}{6} - \beta_1/2 = 0 \longrightarrow 8 + \alpha_1 - 3\beta_1 = 0$

$$\Downarrow$$

$$4 - \beta_1 = 0$$

$$\beta_1 = 4, \quad \alpha_1 = 4$$

$$\alpha_2 = -5, \quad \beta_2 = 2$$

$$\Rightarrow U_{n+1} + 4U_n - 5U_{n-1} = 4f_n + 2f_{n-1}$$

$$\text{LTE} = O(k^3)$$

(This didn't work...??)

The Adams Family

There are two major classes of most popular multi-step methods. The first is the family of *Adams* methods.

For these methods we start with,

$$\boldsymbol{u}(t_{n+1}) = \boldsymbol{u}(t_n) + \int_{t_n}^{t_{n+1}} \boldsymbol{f}(\boldsymbol{u}(r))\mathrm{d}r,$$

suggesting that we should take $\alpha_0 = 1$, $\alpha_1 = -1$.

There are two major classes of most popular multi-step methods. The first is the family of *Adams* methods.

For these methods we start with,

$$\boldsymbol{u}(t_{n+1}) = \boldsymbol{u}(t_n) + \int_{t_n}^{t_{n+1}} \boldsymbol{f}(\boldsymbol{u}(r))\mathrm{d}r,$$

suggesting that we should take $\alpha_0 = 1$, $\alpha_1 = -1$.

The β_j are chosen as a quadrature rule to approximate the integral:

$$\int_{t_n}^{t_{n+1}} \boldsymbol{f}(\boldsymbol{u}(r))\mathrm{d}r \approx k \sum_{j=0}^{s} \beta_j \boldsymbol{f}_{n+1-j}$$

Note that we are using points *outside* the interval of intergration (if $s > 1$).

The Adams Family

There are two major classes of most popular multi-step methods. The first is the family of *Adams* methods.

For these methods we start with,

$$\boldsymbol{u}(t_{n+1}) = \boldsymbol{u}(t_n) + \int_{t_n}^{t_{n+1}} \boldsymbol{f}(\boldsymbol{u}(r))\mathrm{d}r,$$

suggesting that we should take $\alpha_0 = 1$, $\alpha_1 = -1$.

The β_j are chosen as a quadrature rule to approximate the integral:

$$\int_{t_n}^{t_{n+1}} \boldsymbol{f}(\boldsymbol{u}(r))\mathrm{d}r \approx k \sum_{j=0}^{s} \beta_j \boldsymbol{f}_{n+1-j}$$

Note that we are using points *outside* the interval of intergration (if $s > 1$). Again, the particular type of scheme depends on whether we want an implicit or an explicit method:

- $\beta_0 = 0$ yields explicit methods (one fewer parameter to invest in LTE reduction)
- $\beta_0 \neq 0$ yields implicit methods

Adams-Bashforth Methods

The choice of explicit path yields the family of Adams-Bashforth methods.

$$\boldsymbol{u}_{n+1} - \boldsymbol{u}_n = k \sum_{j=1}^{s} \beta_j \boldsymbol{f}_{n+1-j}.$$

The β_j coefficients are used to ensure high-order LTE. E.g., two equivalent strategies:

- Expand in Taylor series, match terms by setting β_j
- Interpolate a degree-$(s-1)$ polynomial on data at $t_{n+1-s}, \ldots, t_n$, integrate the polynomial. The resulting coefficients multiplying the data are the β_j.

Adams-Bashforth Methods

The choice of explicit path yields the family of Adams-Bashforth methods.

$$\boldsymbol{u}_{n+1} - \boldsymbol{u}_n = k \sum_{j=1}^{s} \beta_j \boldsymbol{f}_{n+1-j}.$$

The β_j coefficients are used to ensure high-order LTE. E.g., two equivalent strategies:

- Expand in Taylor series, match terms by setting β_j
- Interpolate a degree-$(s-1)$ polynomial on data at $t_{n+1-s}, \ldots, t_n$, integrate the polynomial. The resulting coefficients multiplying the data are the β_j.

Coefficients for the Adams-Bashforth methods with order=steps:

	β_1	β_2	β_3	β_4	β_5	β_6
$p = s = 1$	1					
$p = s = 2$	$\frac{3}{2}$	-1				
$p = s = 3$	$\frac{23}{12}$	$-\frac{16}{12}$	$\frac{5}{12}$			
$p = s = 4$	$\frac{55}{24}$	$-\frac{59}{24}$	$\frac{37}{24}$	$-\frac{9}{24}$		
$p = s = 5$	$\frac{1901}{720}$	$-\frac{2774}{720}$	$\frac{2616}{720}$	$-\frac{1274}{720}$	$\frac{251}{720}$	
$p = s = 6$	$\frac{4277}{1440}$	$-\frac{7923}{1440}$	$\frac{9982}{1440}$	$-\frac{7298}{1440}$	$\frac{2877}{1440}$	$-\frac{475}{1440}$

Adams-Moulton Methods

The choice of implicit path yields the family of Adams-Moulton methods.

$$\boldsymbol{u}_{n+1} - \boldsymbol{u}_n = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}_{n+1-j}.$$

The β_j coefficients are used to ensure high-order LTE.
The same strategies as before are usable.

Note that technically we can take $s = 0$ here, which yields backward Euler. (Though you'd still call this a 1-step method.)

Adams-Moulton Methods

The choice of implicit path yields the family of Adams-Moulton methods.

$$\boldsymbol{u}_{n+1} - \boldsymbol{u}_n = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}_{n+1-j}.$$

The β_j coefficients are used to ensure high-order LTE.
The same strategies as before are usable.

Note that technically we can take $s = 0$ here, which yields backward Euler. (Though you'd still call this a 1-step method.) Coefficients for the Adams-Moulton methods with order=steps+1:

	β_0	β_1	β_2	β_3	β_4	β_5
$p-1=s=1$	$\frac{1}{2}$	$\frac{1}{2}$				
$p-1=s=2$	$\frac{5}{12}$	$\frac{8}{12}$	$-\frac{1}{12}$			
$p-1=s=3$	$\frac{9}{24}$	$\frac{19}{24}$	$-\frac{5}{24}$	$\frac{1}{24}$		
$p-1=s=4$	$\frac{251}{720}$	$\frac{646}{720}$	$-\frac{264}{720}$	$\frac{106}{720}$	$-\frac{19}{720}$	
$p-1=s=5$	$\frac{475}{1440}$	$\frac{1427}{1440}$	$-\frac{798}{1440}$	$\frac{482}{1440}$	$-\frac{173}{1440}$	$\frac{27}{1440}$

Backward Differentiation formulas

The Adams family of methods is not particularly robust for stiff problems.

As an alternative, consider the general form:

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}(t_{n+1-j}, \boldsymbol{u}_{n+1-j}),$$

and now instead let us focus effort on setting $\beta_j = 0$ for $j > 0$, and choosing α_j to approximate $y'(t_n)$ to high order:

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k\beta_0 \boldsymbol{f}_{n+1}.$$

This is the family of (implicit) backward differentiation formulas (BDF) methods.

Backward Differentiation formulas

The Adams family of methods is not particularly robust for stiff problems.

As an alternative, consider the general form:

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}(t_{n+1-j}, \boldsymbol{u}_{n+1-j}),$$

and now instead let us focus effort on setting $\beta_j = 0$ for $j > 0$, and choosing α_j to approximate $y'(t_n)$ to high order:

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k\beta_0 \boldsymbol{f}_{n+1}.$$

This is the family of (implicit) backward differentiation formulas (BDF) methods. Again, the BDF coefficients are explicitly computable:

	β_0	α_0	α_1	α_2	α_3	α_4	α_5	α_6
$p = s = 1$	1	1	-1					
$p = s = 2$	$\frac{2}{3}$	1	$-\frac{4}{3}$	$\frac{1}{3}$				
$p = s = 3$	$\frac{6}{11}$	1	$-\frac{18}{11}$	$\frac{9}{11}$	$-\frac{2}{11}$			
$p = s = 4$	$\frac{12}{25}$	1	$-\frac{48}{25}$	$\frac{36}{25}$	$-\frac{16}{25}$	$\frac{3}{25}$		
$p = s = 5$	$\frac{60}{137}$	1	$-\frac{300}{137}$	$\frac{300}{137}$	$-\frac{200}{137}$	$\frac{75}{137}$	$-\frac{12}{137}$	
$p = s = 6$	$\frac{60}{147}$	1	$-\frac{360}{147}$	$\frac{450}{147}$	$-\frac{400}{147}$	$\frac{225}{147}$	$-\frac{72}{147}$	$\frac{10}{147}$

Consistency and order of approximation

It's much easier to compute order conditions for multi-step methods (compared to multi-stage ones).

In particular, to compute the LTE for the scheme,

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}(t_{n+1-j}, \boldsymbol{u}_{n+1-j}),$$

we need to compute the residual for the expression

$$\frac{1}{k} \sum_{j=0}^{s} \alpha_j \boldsymbol{u}(t_{n+1-j}) - \sum_{j=0}^{s} \beta_j \boldsymbol{f}(t_{n+1-j}, \boldsymbol{u}(t_{n+1-j})).$$

Consistency and order of approximation

It's much easier to compute order conditions for multi-step methods (compared to multi-stage ones).

In particular, to compute the LTE for the scheme,

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}(t_{n+1-j}, \boldsymbol{u}_{n+1-j}),$$

we need to compute the residual for the expression

$$\frac{1}{k} \sum_{j=0}^{s} \alpha_j \boldsymbol{u}(t_{n+1-j}) - \sum_{j=0}^{s} \beta_j \boldsymbol{f}(t_{n+1-j}, \boldsymbol{u}(t_{n+1-j})).$$

Noting that $\boldsymbol{u}'(t) = \boldsymbol{f}(t, \boldsymbol{u}(t))$, the above expression is equivalent to,

$$\frac{1}{k} \sum_{j=0}^{s} \alpha_j \boldsymbol{u}(t_{n+1-j}) - \sum_{j=0}^{s} \beta_j \boldsymbol{u}'(t_{n+1-j}),$$

and hence we can compute order conditions simply by computing Taylor expansions of $\boldsymbol{u}$ and $\boldsymbol{u}'$.

$$\frac{1}{k}\sum_{j=0}^{s}\alpha_j \boldsymbol{u}(t_{n+1-j}) - \sum_{j=0}^{s}\beta_j \boldsymbol{u}'(t_{n+1-j}),$$

The $\mathcal{O}(1/k)$ terms from the above come from Taylor expansions of the α_j terms, implying that we require,

$$\sum_{j=0}^{s}\alpha_j = 0.$$

$$\frac{1}{k}\sum_{j=0}^{s}\alpha_j \boldsymbol{u}(t_{n+1-j}) - \sum_{j=0}^{s}\beta_j \boldsymbol{u}'(t_{n+1-j}),$$

The $\mathcal{O}(1/k)$ terms from the above come from Taylor expansions of the α_j terms, implying that we require,

$$\sum_{j=0}^{s}\alpha_j = 0.$$

For consistency (LTE vanishing as $k \downarrow 0$), we likewise require the $\mathcal{O}(1)$ terms to vanish, i.e.,

$$\sum_{j=0}^{s}(s-j)\alpha_j - \sum_{j=0}^{s}\beta_j = 0.$$

$$\frac{1}{k}\sum_{j=0}^{s}\alpha_j \boldsymbol{u}(t_{n+1-j}) - \sum_{j=0}^{s}\beta_j \boldsymbol{u}'(t_{n+1-j}),$$

The $\mathcal{O}(1/k)$ terms from the above come from Taylor expansions of the α_j terms, implying that we require,

$$\sum_{j=0}^{s}\alpha_j = 0.$$

For consistency (LTE vanishing as $k \downarrow 0$), we likewise require the $\mathcal{O}(1)$ terms to vanish, i.e.,

$$\sum_{j=0}^{s}(s-j)\alpha_j - \sum_{j=0}^{s}\beta_j = 0.$$

These two expressions are evaluations of certain *characteristic* polynomials:

$$\left.\begin{array}{l}\rho(w) = \sum_{j=0}^{s}\alpha_j w^{s-j} \\ \sigma(w) = \sum_{j=0}^{s}\beta_j w^{s-j}\end{array}\right\} \Longrightarrow \begin{array}{l}\rho(1) = 0 \\ \rho'(1) = \sigma(1)\end{array}$$

Consistency of multi-step methods, II

$$\mathrm{LTE} = \frac{1}{k}\sum_{j=0}^{s} \alpha_j \boldsymbol{u}(t_{n+1-j}) - \sum_{j=0}^{s} \beta_j \boldsymbol{u}'(t_{n+1-j}),$$

$$\rho(w) = \sum_{j=0}^{s} \alpha_j w^{s-j}$$

$$\sigma(w) = \sum_{j=0}^{s} \beta w^{s-j}$$

We have shown the following:

Theorem

A multi-step method is consistent if and only if $\rho(1) = 0$ *and* $\rho'(1) = \sigma(1)$.

$$\text{LTE} = \frac{1}{k} \sum_{j=0}^{s} \alpha_j \boldsymbol{u}(t_{n+1-j}) - \sum_{j=0}^{s} \beta_j \boldsymbol{u}'(t_{n+1-j}),$$

$$\rho(w) = \sum_{j=0}^{s} \alpha_j w^{s-j}$$

$$\sigma(w) = \sum_{j=0}^{s} \beta w^{s-j}$$

We have shown the following:

Theorem

A multi-step method is consistent if and only if $\rho(1) = 0$ and $\rho'(1) = \sigma(1)$.

Of course, to attain more than first-order accuracy, we require more conditions.

0-Stability of multi-step methods, I

The characteristic polynomials are also integral in determining 0-stability:

Theorem

An s-step linear multi-step method is 0-stable if and only if the roots $w_1, \ldots, w_s$ of $\rho(w)$ all satisfy $|w_i| \leqslant 1$, and any roots satisfying $|w_i| = 1$ are simple. (Terminologically: "ρ satisfies the root condition")

This gives a fairly computable condition to identify 0-stability.

0-Stability of multi-step methods, I

The characteristic polynomials are also integral in determining 0-stability:

Theorem

An s-step linear multi-step method is 0-stable if and only if the roots $w_1, \ldots, w_s$ of $\rho(w)$ all satisfy $|w_i| \leqslant 1$, and any roots satisfying $|w_i| = 1$ are simple. (Terminologically: "ρ satisfies the root condition")

This gives a fairly computable condition to identify 0-stability.

Why is this related to 0-stability? Recall that the essential message of 0-stability is that

Initial data perturbations of size ϵ lead to numerical solutions with errors $C\epsilon$ for small enough ϵ.

(I.e., the k-asymptotic LTE behavior bounds the actual error in the numerical method up to a constant.)

The above is actually a more abstract version of the definition of 0-stability compared to what we saw in slides `D06`.

The characteristic polynomials are also integral in determining 0-stability:

Theorem

An s-step linear multi-step method is 0-stable if and only if the roots $w_1, \ldots, w_s$ of $\rho(w)$ all satisfy $|w_i| \leqslant 1$, and any roots satisfying $|w_i| = 1$ are simple. (Terminologically: "ρ satisfies the root condition")

This gives a fairly computable condition to identify 0-stability.

Why is this related to 0-stability? Recall that the essential message of 0-stability is that

Initial data perturbations of size ϵ lead to numerical solutions with errors $C\epsilon$ for small enough ϵ.

(I.e., the k-asymptotic LTE behavior bounds the actual error in the numerical method up to a constant.)

The above is actually a more abstract version of the definition of 0-stability compared to what we saw in slides `D06`.

Note: we only require control of perturbations for vanishingly small ϵ.

It turns out that while phrased as perturbations to initial data, this is conceptually similar to perturbations of $\boldsymbol{f}$, and under an ODE well-posedness result, is equivalent to considering perturbations of $\boldsymbol{f} = \boldsymbol{0}$.

I.e., it's enough to check controlled perturbations for $\boldsymbol{u}' = \boldsymbol{0}$
(This is why it's called 0-stability.)

0-Stability of multi-step methods, II

So if we only need to consider a linear multistep method for $\boldsymbol{u}' = \boldsymbol{0}$ to account for 0-stability, this means the scheme reads,

$$\sum_{j=0}^{s} \alpha_j u_{n+1-j} = 0,$$

say with starting conditions $u_0 = \cdots = u_{s-1} = 0$.

0-Stability of multi-step methods, II

So if we only need to consider a linear multistep method for $\boldsymbol{u}' = \mathbf{0}$ to account for 0-stability, this means the scheme reads,

$$\sum_{j=0}^{s} \alpha_j u_{n+1-j} = 0,$$

say with starting conditions $u_0 = \cdots = u_{s-1} = 0$.

If we replace the initial data of 0's by perturbations, then the exact solution to the difference equation, assuming unique roots $w_1, \ldots, w_s$ of ρ, is,

$$u_n \sim \epsilon_1 w_1^n + \cdots + \epsilon_s w_s^n$$

where $\epsilon_1, \ldots, \epsilon_s$ are dependent on the initial data perturbations.

0-Stability of multi-step methods, II

So if we only need to consider a linear multistep method for $\boldsymbol{u}' = \boldsymbol{0}$ to account for 0-stability, this means the scheme reads,

$$\sum_{j=0}^{s} \alpha_j u_{n+1-j} = 0,$$

say with starting conditions $u_0 = \cdots = u_{s-1} = 0$.

If we replace the initial data of 0's by perturbations, then the exact solution to the difference equation, assuming unique roots $w_1, \ldots, w_s$ of ρ, is,

$$u_n \sim \epsilon_1 w_1^n + \cdots + \epsilon_s w_s^n$$

where $\epsilon_1, \ldots, \epsilon_s$ are dependent on the initial data perturbations.

The exact solution to this problem is $u_n = 0$. As $k \downarrow 0$, then $n \uparrow \infty$. I.e., our perturbed solution is bounded relative to 0 iff $|w_j| < 1$ for all j.

$|w_j| = 1$ is allowed for simple roots, but for repeated roots with multiplicity m, then $|u_n| \sim n^{m-1}|w_j| = n^{m-1}$, which is unbounded in n for $m > 1$.

0-Stability of multi-step methods, II

So if we only need to consider a linear multistep method for $\boldsymbol{u}' = \boldsymbol{0}$ to account for 0-stability, this means the scheme reads,

$$\sum_{j=0}^{s} \alpha_j u_{n+1-j} = 0,$$

say with starting conditions $u_0 = \cdots = u_{s-1} = 0$.

If we replace the initial data of 0's by perturbations, then the exact solution to the difference equation, assuming unique roots $w_1, \ldots, w_s$ of ρ, is,

$$u_n \sim \epsilon_1 w_1^n + \cdots + \epsilon_s w_s^n$$

where $\epsilon_1, \ldots, \epsilon_s$ are dependent on the initial data perturbations.

The exact solution to this problem is $u_n = 0$. As $k \downarrow 0$, then $n \uparrow \infty$. I.e., our perturbed solution is bounded relative to 0 iff $|w_j| < 1$ for all j.

$|w_j| = 1$ is allowed for simple roots, but for repeated roots with multiplicity m, then $|u_n| \sim n^{m-1}|w_j| = n^{m-1}$, which is unbounded in n for $m > 1$.

Hence, 0-stability is equivalent to the roots of ρ lying within the unit circle (and on the boundary for simple roots).

0-stability examples

Example: any one-step ($s = 1$) method is 0-stable, since $\rho(w) = w - 1$.

$$u_{n+1} + \alpha_1 u_n = \beta_0 f_{n+1} + \beta_1 f_n$$

$\Downarrow$ $\alpha_1 = -1$ by consistency

$$\rho(w) = w - 1$$

0-stability examples

Example: any one-step ($s = 1$) method is 0-stable, since $\rho(w) = w - 1$.

Example: All Adams- methods are 0-stable, since $\rho(w) = w^s - w^{s-1}$.

$$u_{n+1} + \alpha_1 u_n = k \sum_{j=0}^{s} \beta_j f_{n+1-j}$$

$$\alpha_1 = -1 \text{ (consistency)} \Rightarrow \rho(w) = w^s - w^{s-1} = w^{s-1}(w-1)$$

0-stability examples

Example: any one-step ($s = 1$) method is 0-stable, since $\rho(w) = w - 1$.

Example: All Adams- methods are 0-stable, since $\rho(w) = w^s - w^{s-1}$.

Example: All BDF methods for $s \leqslant 6$ are 0-stable. Any BDF method with $s > 6$ is unstable.

0-stability examples

Example: any one-step ($s = 1$) method is 0-stable, since $\rho(w) = w - 1$.

Example: All Adams- methods are 0-stable, since $\rho(w) = w^s - w^{s-1}$.

Example: All BDF methods for $s \leqslant 6$ are 0-stable. Any BDF method with $s > 6$ is unstable.

There are reasonable-looking methods that violate 0-stability:

$$\boldsymbol{u}_{n+1} + 4\boldsymbol{u}_n - 5\boldsymbol{u}_{n-1} = k\left(4\boldsymbol{f}_n + 2\boldsymbol{f}_{n-1}\right),$$

and these methods are actually quite unstable.

$\rho(w) = w^2 + 4w - 5 = (w+5)(w-1)$

$w = 1, \; -5$

Convergence

Just like our analysis for simple Euler-type schemes, 0-stability and consistency *are* convergence.

Theorem (Dahlquist Equivalence Theorem)

Consider an s-step multistep method where the startup values $\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{s-1}$ are generated in a consistent way ($\boldsymbol{u}_j \to \boldsymbol{u}(0)$ as $k \downarrow 0$ for $j = 0, \ldots, s-1$.)

Such a linear multistep method is convergent if and only if it is consistent and 0-stable.

I.e.,:

A linear multistep method is convergent if and only if $\rho(1) = 0$, $\rho'(1) = \sigma(1)$, and ρ satisfies the root condition.

(When convergent, a linear multistep method has order of convergence equal to the order p of the LTE.)

Just like our analysis for simple Euler-type schemes, 0-stability and consistency *are* convergence.

Theorem (Dahlquist Equivalence Theorem)

Consider an s-step multistep method where the startup values $\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{s-1}$ are generated in a consistent way ($\boldsymbol{u}_j \to \boldsymbol{u}(0)$ as $k \downarrow 0$ for $j = 0, \ldots, s-1$.)

Such a linear multistep method is convergent if and only if it is consistent and 0-stable.

I.e.,:

A linear multistep method is convergent if and only if $\rho(1) = 0$, $\rho'(1) = \sigma(1)$, and ρ satisfies the root condition.

(When convergent, a linear multistep method has order of convergence equal to the order p of the LTE.)

Note that 1 is *always* a root of ρ for multistep methods of interest.

Methods for which 1 is the *only* unity-modulus root are *strongly* stable. (Otherwise, they are weakly stable.)

Absolute stability

We have a similar notion of absolute stability for multi-step methods: We require that the iteration,

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}(t_{n+1-j}, \boldsymbol{u}_{n+1-j}),$$

produces solutions $\boldsymbol{u}_n$ that do not grow exponentially in n for the test equation $u' = \lambda u$.

We have a similar notion of absolute stability for multi-step methods: We require that the iteration,

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}(t_{n+1-j}, \boldsymbol{u}_{n+1-j}),$$

produces solutions $\boldsymbol{u}_n$ that do not grow exponentially in n for the test equation $u' = \lambda u$.

This results in the difference equation,

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k\lambda \sum_{j=0}^{s} \beta_j \boldsymbol{u}_{n+1-j},$$

whose characteristic equation is,

$$\rho(w) = k\lambda\sigma(w) \stackrel{z=\lambda k}{=} z\sigma(w).$$

Absolute stability

We have a similar notion of absolute stability for multi-step methods: We require that the iteration,

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}(t_{n+1-j}, \boldsymbol{u}_{n+1-j}),$$

produces solutions $\boldsymbol{u}_n$ that do not grow exponentially in n for the test equation $u' = \lambda u$.

This results in the difference equation,

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k\lambda \sum_{j=0}^{s} \beta_j \boldsymbol{u}_{n+1-j},$$

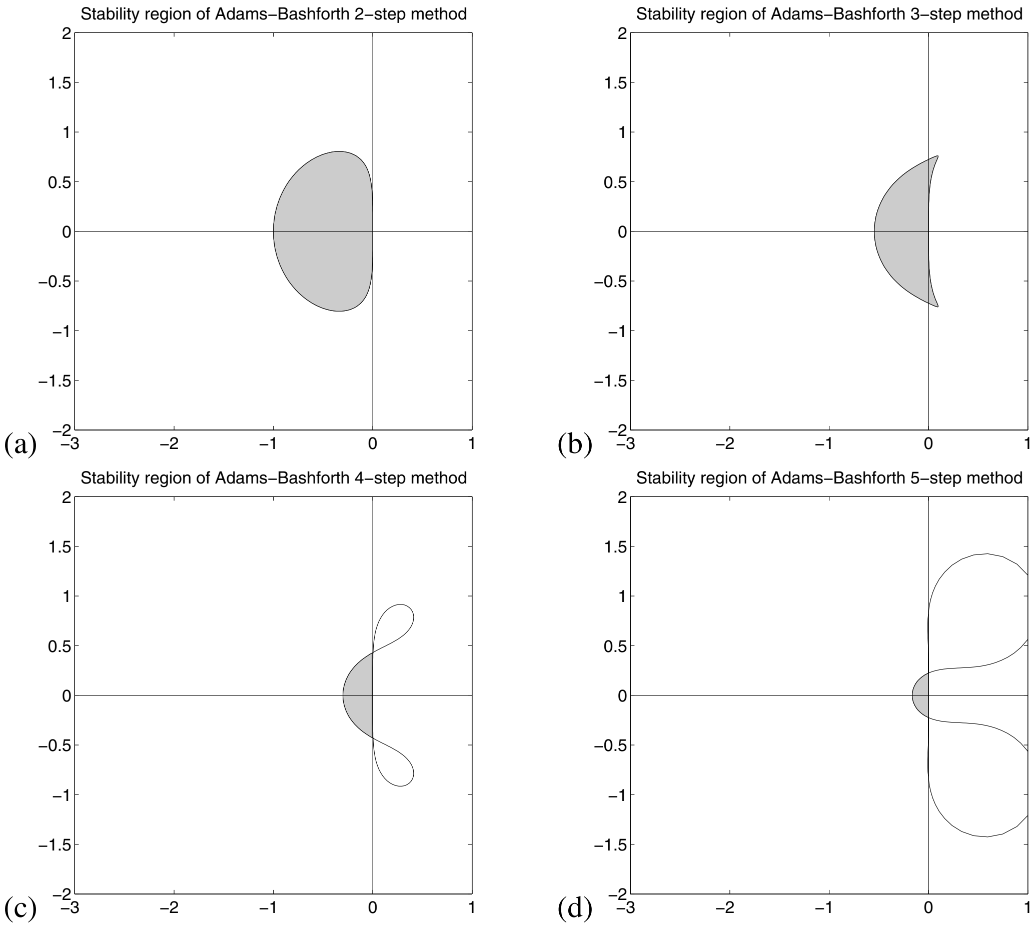
whose characteristic equation is,

$$\rho(w) = k\lambda\sigma(w) \stackrel{z=\lambda k}{=} z\sigma(w).$$

Thus, we say that the region of (absolute) stability for the scheme is the set of z values such that $\rho(w) - z\sigma(w)$ has roots $w_1, \ldots, w_s$ all satisfying $|w_j| < 1$, with $|w_j| = 1$ allowed for simple roots. (I.e., $z \in \mathrm{ROS}$ means $\rho(\cdot) - z\sigma(\cdot)$ satsifies the root condition.)

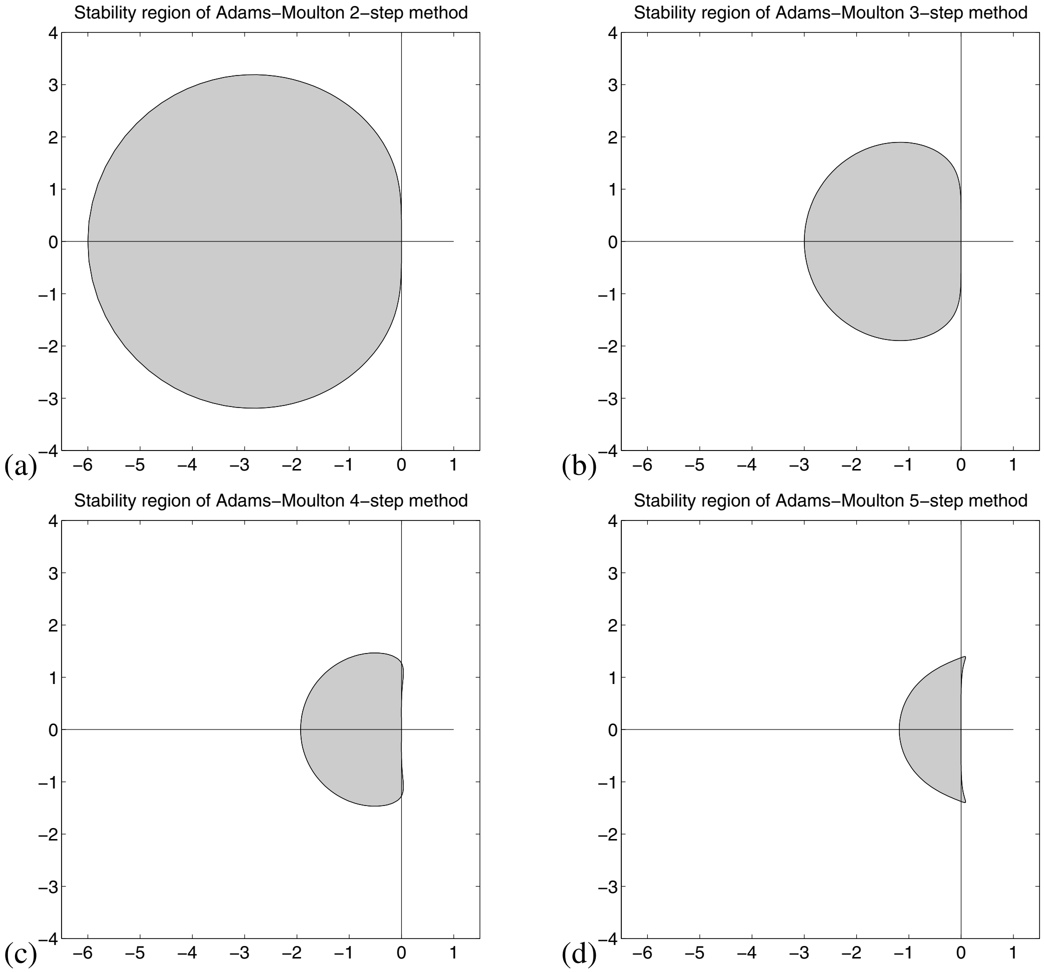
(Note that $0 \in \mathrm{ROS} \iff 0$-stability.)

Absolute stability: Adams-Bashforth



LeVeque 2007, Figure 7.2

Absolute stability: Adams-Moulton



LeVeque 2007, Figure 7.3

Dahlquist barriers

There are some somewhat dissapointing results about efficacy of multistep methods:

Theorem (First Dahlquist Barrier)

For an s-step multistep method that is 0-stable:

- *An explicit method can have order of accuracy at most $p = s$.*
- *If s is odd, the order of accuracy can be at most $p = s + 1$.*
- *If s is even, the order of accuracy can be at most $p = s + 2$.*

(Note that consistent s-step methods explicitly have $2s - 1$ degrees of freedom.)

Dahlquist barriers

There are some somewhat dissapointing results about efficacy of multistep methods:

Theorem (First Dahlquist Barrier)

For an s-step multistep method that is 0-stable:

- *An explicit method can have order of accuracy at most $p = s$.*
- *If s is odd, the order of accuracy can be at most $p = s + 1$.*
- *If s is even, the order of accuracy can be at most $p = s + 2$.*

(Note that consistent s-step methods explicitly have $2s - 1$ degrees of freedom.)

Theorem (Second Dahlquist Barrier)

No A-stable explicit multistep methods exist.

An implicit A-stable multistep method has order of accuracy at most $p = 2$.

Startup

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}(t_{n+1-j}, \boldsymbol{u}_{n+1-j}),$$

How to start from $n = 0$ if $s > 1$?

Odds and ends for multi-step methods

Startup

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}(t_{n+1-j}, \boldsymbol{u}_{n+1-j}),$$

How to start from $n = 0$ if $s > 1$?

Usually accomplished with Runge-Kutta methods of similar order.

Startup

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}(t_{n+1-j}, \boldsymbol{u}_{n+1-j}),$$

How to start from $n = 0$ if $s > 1$?

Usually accomplished with Runge-Kutta methods of similar order.

Predictor-corrector methods

Explicit and implicit methods are frequently used in *predictor-corrector* frameworks, e.g.,:

- An explicit approximation to $\boldsymbol{u}_{n+1}$ is computed with an Adams-Bashforth method.
- This approximation is used as an emulator for the unknown $\boldsymbol{u}(t_{n+1})$ on the right-hand side of an Adams-Moulton method.

Startup

$$\sum_{j=0}^{s} \alpha_j \boldsymbol{u}_{n+1-j} = k \sum_{j=0}^{s} \beta_j \boldsymbol{f}(t_{n+1-j}, \boldsymbol{u}_{n+1-j}),$$

How to start from $n = 0$ if $s > 1$?

Usually accomplished with Runge-Kutta methods of similar order.

Predictor-corrector methods

Explicit and implicit methods are frequently used in *predictor-corrector* frameworks, e.g.,:

- An explicit approximation to $\boldsymbol{u}_{n+1}$ is computed with an Adams-Bashforth method.
- This approximation is used as an emulator for the unknown $\boldsymbol{u}(t_{n+1})$ on the right-hand side of an Adams-Moulton method.

Predictor-corrector methods are an example from a more general class of methods called *general linear methods*, which encompass both multi-stage and multi-step methods.

References I

Ascher, Uri M. and Linda R. Petzold (1998). *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM. ISBN: 978-1-61197-139-2.

Butcher, J. C. (2006). "General Linear Methods". In: *Acta Numerica* 15, pp. 157–256. ISSN: 1474-0508, 0962-4929. DOI: 10.1017/S0962492906220014.

LeVeque, Randall J. (2007). *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. SIAM. ISBN: 978-0-89871-783-9.