Lecture 7

MTH572/MTH472
Numerical Methods for PDEs
Alfa Heryudono

UMass Dartmouth

Main references (quotes):
Trefethen: NumPDE, ATAP, Spectral Methods in MATLAB
Fornberg: PS Guide
Leveque: NumPDE
Driscoll: Learning MATLAB

Linear Multistep Formulas: Derivation

$$\begin{aligned} u(t_{n+s}) - u(t_{n+s-1}) &= \int_{t_{n+s-1}}^{t_{n+s}} u_t(t)dt \\ &= \int_{t_{n+s-1}}^{t_{n+s}} f(t)dt \approx \int_{t_{n+s-1}}^{t_{n+s}} q(t)dt \end{aligned}$$

$q(t)$ is an interpolating polynomial that interpolates f on the interval $[t_{n+s-1}, t_{n+s}]$. Ex: you can use Lagrange interpolating polynomial for $q(t)$.

Forward Euler

Forward Euler is a one step method (Hence $s = 1$). The interpolating polynomial $q(t)$ is of degree $s - 1$. Since $s = 1$, $q(t)$ is a polynomial of degree 0, hence a constant function. Since it is an explicit method, we choose the constant function $q(t) = f^n$.

$$\begin{aligned} u(t_{n+s}) - u(t_{n+s-1}) &= \int_{t_{n+s-1}}^{t_{n+s}} q(t)dt \\ u(t_{n+1}) - u(t_n) &= \int_{t_n}^{t_{n+1}} q(t)dt \\ v^{n+1} - v^n &= \int_{nk}^{(n+1)k} f^n dt \\ v^{n+1} - v^n &= kf^n \\ v^{n+1} &= v^n + kf^n \end{aligned}$$

Backward Euler

Backward Euler is a one step method (Hence $s = 1$). The interpolating polynomial $q(t)$ is of degree $s - 1$. Since $s = 1$, $q(t)$ is a polynomial of degree 0, hence a constant function. Since it is an implicit method, we choose the constant function $q(t) = f^{n+1}$ instead of f^n.

$$u(t_{n+s}) - u(t_{n+s-1}) = \int_{t_{n+s-1}}^{t_{n+s}} q(t)dt$$

$$u(t_{n+1}) - u(t_n) = \int_{t_n}^{t_{n+1}} q(t)dt$$

$$v^{n+1} - v^n = \int_{nk}^{(n+1)k} f^{n+1}dt$$

$$v^{n+1} - v^n = kf^{n+1}$$

$$v^{n+1} = v^n + kf^{n+1}$$

Notes: In general

Explicit Method	The integrant $q(t)$ is a polynomial of degree $s-1$ that interpolates $f^n, f^{n+1}, \ldots, f^{n+s-1}$ at times $t_n, t_{n+1}, \ldots, t_{n+s-1}$
Implicit Method	The integrant $q(t)$ is a polynomial of degree s that interpolates $f^n, f^{n+1}, \ldots, f^{n+s}$ at times $t_n, t_{n+1}, \ldots, t_{n+s}$

Trapezoid

Trapezoid is a one step method (Hence $s = 1$). Since it is an implicit method, the interpolating polynomial $q(t)$ is of degree 1. Hence, $q(t)$ is a straight line. By using the Lagrange interpolating polynomial,

$$q(t) = \frac{t - t_{n+1}}{t_n - t_{n+1}} f^n + \frac{t - t_n}{t_{n+1} - t_n} f^{n+1}$$

Due to the use of uniform time-step k, we can set $t_n = 0$, $t_{n+1} = k$, $t_{n+2} = 2k$, and so on.

$$q(t) = -\frac{t - k}{k} f^n + \frac{t}{k} f^{n+1}$$

$$u(t_{n+s}) - u(t_{n+s-1}) = \int_{t_{n+s-1}}^{t_{n+s}} q(t)dt$$

$$u(t_{n+1}) - u(t_n) = \int_{t_n}^{t_{n+1}} q(t)dt$$

$$v^{n+1} - v^n = \int_0^k \left(-\frac{t - k}{k} f^n + \frac{t}{k} f^{n+1} \right) dt$$

$$v^{n+1} - v^n = \frac{k}{2}(f^n + f^{n+1})$$

$$v^{n+1} = v^n + \frac{k}{2}(f^n + f^{n+1})$$

Adams Family. (J.C. Adams 1855)

$$v^{n+s} - v^{n+s-1} = k\left(\beta_0 f^n + \beta_1 f^{n+1} + \cdots + \beta_s f^{n+s}\right) = k \sum_{j=0}^{s} \beta_j f^{n+j}$$

By using the same technique from the previous slides, you can derive the coefficients β's for the Adams-Bashforth (Explicit) and Adams-Moulton (Implicit) formula. Tables up to fourth order are given below.

Adams-Bashforth (Explicit)

Steps s	Order p	β_s	β_{s-1}	β_{s-2}	β_{s-3}	β_{s-4}	
1	1	0	1				(Fwd Euler)
2	2	0	$\frac{3}{2}$	$-\frac{1}{2}$			
3	3	0	$\frac{23}{12}$	$-\frac{16}{12}$	$\frac{5}{12}$		
4	4	0	$\frac{55}{24}$	$-\frac{59}{24}$	$\frac{37}{24}$	$-\frac{9}{24}$	

Adams-Moulton (Implicit)

Steps s	Order p	β_s	β_{s-1}	β_{s-2}	β_{s-3}	β_{s-4}	
1	1	1					(Bwd Euler)
1	2	$\frac{1}{2}$	$\frac{1}{2}$				
2	3	$\frac{5}{12}$	$\frac{8}{12}$	$-\frac{1}{12}$			
3	4	$\frac{9}{24}$	$\frac{19}{24}$	$-\frac{5}{24}$	$\frac{1}{24}$		
4	5	$\frac{251}{720}$	$\frac{646}{720}$	$-\frac{264}{720}$	$\frac{106}{720}$	$-\frac{19}{720}$	

Computing (AB)-(AM) Coefficients with Mathematica

An example how to use Mathematica to derive the AB-4 coefficients.

```
In[1]:= s = 4

Out[1]:=
4

In[2]:=
data = Table[{i k,fn[i]},{i,0,s-1}]

Out[2]:=
{{0, fn[0]}, {k, fn[1]}, {2 k, fn[2]}, {3 k, fn[3]}}

In[3]:=
ti = data[[s]][[1]]

Out[3]:=
3 k

In[4]:= Integrate[InterpolatingPolynomial[data,t],{t,ti,ti+k}]

Out[4]:=
-(3/8) k fn[0] + 37/24 k fn[1] - 59/24 k fn[2] + 55/24 k fn[3]
```

Computing (AB)-(AM) Coefficients with Mathematica

An example how to use Mathematica to derive the AM-4 (s=3) coefficients.

```
In[1]:= s = 3

Out[1]:=
3

In[2]:=
data = Table[{i k,fn[i]},{i,0,s}]

Out[2]:=
{{0, fn[0]}, {k, fn[1]}, {2 k, fn[2]}, {3 k, fn[3]}}

In[3]:=
ti = data[[s]][[1]]

Out[3]:=
2 k

In[4]:= Integrate[InterpolatingPolynomial[data,t],{t,ti,ti+k}]

Out[4]:=
1/24 k fn[0] - 5/24 k fn[1] + 19/24 k fn[2] + 3/8 k fn[3]
```

Backward Differentiation (BDF)

Curtis and Hirschfelder (1952), C.W. Gear.

$$
\begin{aligned}
u_t &= f \\
q_t(t_{n+s}) &= f(v^{n+s}, t_{n+s}) \\
\bar{\alpha}_0 v^n + \bar{\alpha}_1 v^{n+1} + \cdots + \bar{\alpha}_{s-1} v^{n+s-1} + \bar{\alpha}_s v^{n+s} &= f^{n+s} \\
\left(\frac{\bar{\alpha}_0}{\bar{\alpha}_s}\right) v^n + \left(\frac{\bar{\alpha}_1}{\bar{\alpha}_s}\right) v^{n+1} + \cdots + \left(\frac{\bar{\alpha}_{s-1}}{\bar{\alpha}_s}\right) v^{n+s-1} + v^{n+s} &= \left(\frac{1}{\bar{\alpha}_s}\right) f^{n+s} \\
\alpha_0 v^n + \alpha_1 v^{n+1} + \cdots + \alpha_{s-1} v^{n+s-1} + \alpha_s v^{n+s} &= k\beta_s f^{n+s}
\end{aligned}
$$

Instead of integrating the polynomial interpolant that interpolates the right hand side f as in the Adams formula, the idea is to interpolate solution values $v^n, \cdots, v^{n+s}$ at $t_n, \ldots, t_{n+s}$ and then differentiating the interpolant to obtain the coefficients. The right hand side is always f^{n+s}.

Backward Differentiation Formula

Steps s	Order p	α_s	α_{s-1}	α_{s-2}	α_{s-3}	α_{s-4}	β_s	
1	1	1	-1				1	B. Euler
2	2	1	$-\frac{4}{3}$	$\frac{1}{3}$			$\frac{2}{3}$	
3	3	1	$-\frac{18}{11}$	$\frac{9}{11}$	$-\frac{2}{11}$		$\frac{6}{11}$	
4	4	1	$-\frac{48}{25}$	$\frac{36}{25}$	$-\frac{16}{25}$	$\frac{3}{25}$	$\frac{12}{25}$	

Computing BDF Coefficients with Mathematica

An example how to use Mathematica to derive the BDF (s=4) coefficients.

```
In[1]:= s = 4
Out[1]:=
4

In[2]:= data = Table[{i k, vn[i]}, {i, 0, s}]
Out[2]:=
{{0, vn[0]}, {k, vn[1]}, {2 k, vn[2]}, {3 k, vn[3]}, {4 k, vn[4]}}

In[3]:= tns = data[[s + 1]][[1]]
Out[3]:=
4 k

In[4]:= qt = Expand[D[InterpolatingPolynomial[data, t], t] /. {t -> tns}]
Out[4]:=
vn[0]/(4 k) - (4 vn[1])/(3 k) + (3 vn[2])/k - (4 vn[3])/k + (25 vn[4])/(12 k)

In[5]:= asbar = Coefficient[qt, vn[s]]
Out[5]:=
25/(12 k)

In[6]:= Expand[qt/asbar]
Out[6]:=
(3 vn[0])/25 - (16 vn[1])/25 + (36 vn[2])/25 - (48 vn[3])/25 + vn[4]
```

Things to do in class

1. By hand, derive coefficients of AB2, AM2, and BDF2. Verify your hand calculations with Mathematica.
2. From previous lecture, find the $\rho(z), \sigma(z)$ and $R(z)$ for AB, AM, and BDF and verify their order of accuracy p using Mathematica command `Series[ ]` (compare with $\log z$ series).
3. Given the perimeter of unit circle $|z| = 1$ centered at the origin in the complex plane, find its map under the $R(z)$ for AB, AM, and BDF. Plot the circle before and after the map.
4. Let $K = 1, m = 2$ and $y^* = \frac{1}{2}$ are constants, rewrite

$$y_{tt} = -\frac{K}{m}(y - y^*), \quad y(0) = 1, \quad y_t(0) = \frac{1}{4}$$

as first order system. Solve the system with AB2, AM2, and BDF2 from $t = 0$ until $T = 1$.

Can you find the exact solution $y(t)$? Compute the error $|y(1) - v(1)|$ for $k = 10^{-3}, 5.10^{-3}, 10^{-2}, 5.10^{-2}, 10^{-1}$. For each method, plot Error vs k in loglog scale. Do the slope agree with the order of accuracy of the methods ?