Modélisation et Simulation des Plasmas

G. Fuhr
Guillaume.Fuhr@univ.amu.fr

Forward Differences

Forward finite-divided-difference formulas:

First Derivative | **Error**

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} \qquad O(h)$$

$$f'(x_i) = \frac{-f(x_{i+2}) + 4f(x_{i+1}) - 3f(x_i)}{2h} \qquad O(h^2)$$

Second Derivative

$$f''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2} \qquad O(h)$$

$$f''(x_i) = \frac{-f(x_{i+3}) + 4f(x_{i+2}) - 5f(x_{i+1}) + 2f(x_i)}{h^2} \qquad O(h^2)$$

Third Derivative

$$f'''(x_i) = \frac{f(x_{i+3}) - 3f(x_{i+2}) + 3f(x_{i+1}) - f(x_i)}{h^3} \qquad O(h)$$

$$f'''(x_i) = \frac{-3f(x_{i+4}) + 14f(x_{i+3}) - 24f(x_{i+2}) + 18f(x_{i+1}) - 5f(x_i)}{2h^3} \qquad O(h^2)$$

Backward Differences

Backward finite-divided-difference formulas:

First Derivative — **Error**

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h} \qquad O(h)$$

$$f'(x_i) = \frac{3f(x_i) - 4f(x_{i-1}) + 3f(x_{i-2})}{2h} \qquad O(h^2)$$

Second Derivative

$$f''(x_i) = \frac{f(x_i) - 2f(x_{i-1}) + f(x_{i-2})}{h^2} \qquad O(h)$$

$$f''(x_i) = \frac{2f(x_i) - 5f(x_{i-1}) + 4f(x_{i-2}) - f(x_{i-3})}{h^2} \qquad O(h^2)$$

Third Derivative

$$f'''(x_i) = \frac{f(x_i) - 3f(x_{i-1}) + 3f(x_{i-2}) - f(x_{i-3})}{h^3} \qquad O(h)$$

$$f'''(x_i) = \frac{5f(x_i) - 18f(x_{i-1}) + 24f(x_{i-2}) - 14f(x_{i-3}) + 3f(x_{i-4})}{2h^3} \qquad O(h^2)$$

Centered Differences

Centered finite-divided-difference formulas:

Error

First Derivative

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} \qquad O(h)$$

$$f'(x_i) = \frac{-f(x_{i+2}) + 8f(x_{i+1}) - 8f(x_{i-1}) + f(x_{i-2})}{12h} \qquad O(h^2)$$

Second Derivative

$$f''(x_i) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2} \qquad O(h)$$

$$f''(x_i) = \frac{-f(x_{i+2}) + 16f(x_{i+1}) - 30f(x_i) + 16f(x_{i-1}) - f(x_{i-2})}{12h^2} \qquad O(h^2)$$

Third Derivative

$$f'''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + 2f(x_{i-1}) - f(x_{i-2})}{2h^3} \qquad O(h)$$

$$f'''(x_i) = \frac{-f(x_{i+3}) + 8f(x_{i+2}) - 13f(x_{i+1}) + 13f(x_{i-1}) - 8f(x_{i-2}) + f(x_{i-3})}{8h^3} \qquad O(h^2)$$

Example: Taylor Table for the Adams-Moulton 3-steps (4 time-nodes) Method

Denoting $h \equiv \Delta t$, $\phi \equiv u$, $\dfrac{du}{dt} = u' = f(t,u)$ and $u'_n = f(t_n,u^n)$, one obtains for $K=2$:

$$u^{n+1} - u^n = \sum_{k=-K}^{1} \beta_k f(t_{n+k},u^{n+k})\,\Delta t = h\left[\beta_1 f(t_{n+1},u^{n+1}) + \beta_0 f(t_n,u^n) + \beta_{-1} f(t_{n-1},u^{n-1}) + \beta_{-2} f(t_{n-2},u^{n-2})\right]$$

Taylor Table:

- The first row (Taylor series) + the last 5 rows (Taylor series for each term) must sum to zero
- This can be satisfied up to the 5th column (4th order term)
- Hence, the AM method with 4-time levels is 4th order accurate

	u_n	$h \cdot u'_n$	$h^2 \cdot u''_n$	$h^3 \cdot u'''_n$	$h^4 \cdot u''''_n$
u_{n+1}	1	1	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{24}$
$-u_n$	-1				
$-h\beta_1 u'_{n+1}$		$-\beta_1$	$-\beta_1$	$-\beta_1 \frac{1}{2}$	$-\beta_1 \frac{1}{6}$
$-h\beta_0 u'_n$		$-\beta_0$			
$-h\beta_{-1} u'_{n-1}$		$-\beta_{-1}$	β_{-1}	$-\beta_{-1} \frac{1}{2}$	$\beta_{-1} \frac{1}{6}$
$-h\beta_{-2} u'_{n-2}$		$-(-2)^0 \beta_{-2}$	$-(-2)^1 \beta_{-2}$	$-(-2)^2 \beta_{-2} \frac{1}{2}$	$-(-2)^3 \beta_{-2} \frac{1}{6}$

solving for the β_k's $\Rightarrow \beta_1 = 9/24,\quad \beta_0 = 19/24,\quad \beta_{-1} = -5/24$ and $\beta_{-2} = 1/24$

Example of Adams Methods for Time-Integration

Explicit Methods. (Adams-Bashforth, with ABn meaning nth order AB)

u_{n+1}	$= u_n + hu'_n$	Euler
u_{n+1}	$= u_{n-1} + 2hu'_n$	Leapfrog
u_{n+1}	$= u_n + \frac{1}{2}h\left[3u'_n - u'_{n-1}\right]$	AB2
u_{n+1}	$= u_n + \frac{h}{12}\left[23u'_n - 16u'_{n-1} + 5u'_{n-2}\right]$	AB3

Implicit Methods. (Adams-Moulton, with AMn meaning nth order AM)

u_{n+1}	$= u_n + hu'_{n+1}$	Implicit Euler
u_{n+1}	$= u_n + \frac{1}{2}h\left[u'_n + u'_{n+1}\right]$	Trapezoidal (AM2)
u_{n+1}	$= \frac{1}{3}\left[4u_n - u_{n-1} + 2hu'_{n+1}\right]$	2nd-order Backward
u_{n+1}	$= u_n + \frac{h}{12}\left[5u'_{n+1} + 8u'_n - u'_{n-1}\right]$	AM3

Practical Time-Integration Methods for CFD

- High-resolution CFD requires large discrete state vector sizes to store the spatial information
- This means that up to two times (one on each side of the current time step) have often been utilized (3 time-nodes): $u^{n+1}-u^n = h\left[\beta_1 f(t_{n+1},u^{n+1})+\beta_0 f(t_n,u^n)+\beta_{-1} f(t_{n-1},u^{n-1})\right]$
- Rewriting this equations in a way such that differences wrt. the Euler's method are easily seen, one obtains ($\theta = 0$ for explicit schemes):

$$\boxed{(1+\xi)\,u^{n+1} = \left[(1+2\xi)\,u^n - \xi\,u^{n-1}\right] + h\left[\theta\, f(t_{n+1},u^{n+1}) + (1-\theta+\varphi)\, f(t_n,u^n) - \varphi f(t_{n-1},u^{n-1})\right]}$$

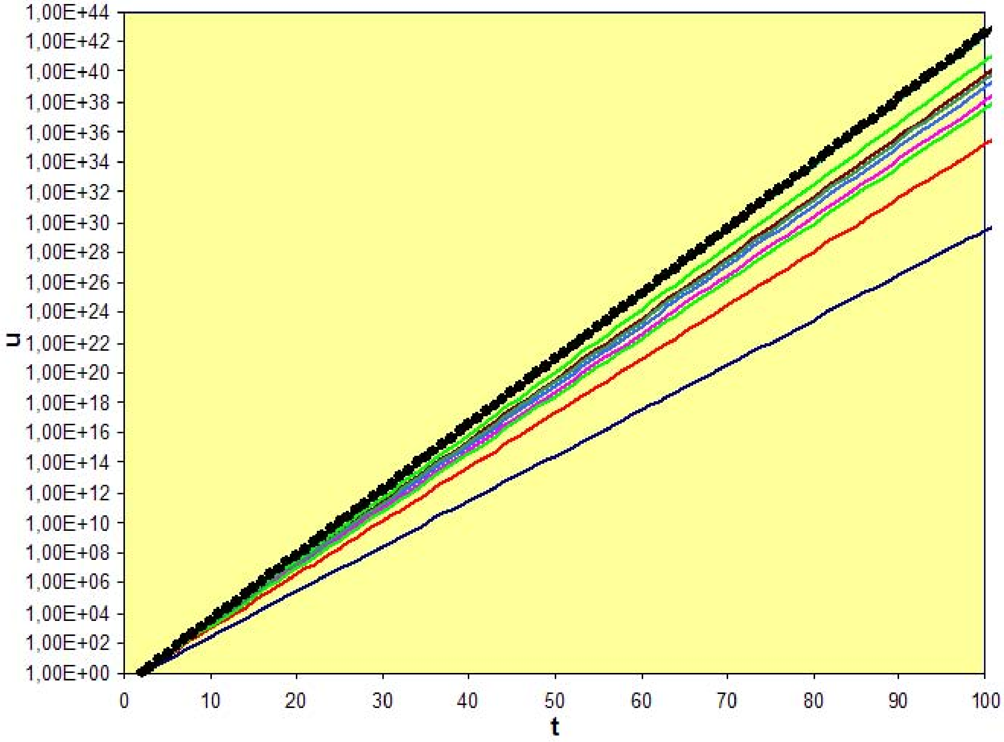
θ	ξ	φ	Method	Order
0	0	0	Euler	1
1	0	0	Implicit Euler	1
1/2	0	0	Trapezoidal or AM2	2
1	1/2	0	2nd-order Backward	2
3/4	0	−1/4	Adams type	2
1/3	−1/2	−1/3	Lees	2
1/2	−1/2	−1/2	Two-step trapezoidal	2
5/9	−1/6	−2/9	A-contractive	2
0	−1/2	0	Leapfrog	2
0	0	1/2	AB2	2
0	−5/6	−1/3	Most accurate explicit	3
1/3	−1/6	0	Third-order implicit	3
5/12	0	1/12	AM3	3
1/6	−1/2	−1/6	Milne	4

Comparaison des différents schémas

- Résolution de l'équation :

$$\frac{du}{dt} = u \ \textbf{avec} \ u(t=0) = 1$$

- *Du bas en haut et donc du moins performant au plus performant :*
  - *Euler,*
  - *Adams-Bashforth 1,*
  - *Milne 1,*
  - *Adams-Bashforth 2,*
  - *prédicteur-correcteur et trapèze,*
  - *Milne 3,*
  - *Adams-Bashforth 3,*
  - *Milne 3,*
  - *Runge-Kutta d'ordre 4*



Method	*Order*	*Formula*
Forward	1	$\phi^{n+1} = \phi^n + hF(\phi^n)$
Backward	1	$\phi^{n+1} = \phi^n + hF(\phi^{n+1})$
Asselin Leapfrog	1	$\phi^{n+1} = \overline{\phi^{n-1}} + 2hF(\phi^n)$ $\overline{\phi^n} = \phi^n + \gamma(\overline{\phi^{n-1}} - 2\phi^n + \phi^{n+1})$
Leapfrog	2	$\phi^{n+1} = \phi^{n-1} + 2hF(\phi^n)$
Adams–Bashforth	2	$\phi^{n+1} = \phi^n + \frac{h}{2}\left[3F(\phi^n) - F(\phi^{n-1})\right]$
Trapezoidal	2	$\phi^{n+1} = \phi^n + \frac{h}{2}\left[F(\phi^{n+1}) + F(\phi^n)\right]$
Runge–Kutta	2	$q_1 = hF(\phi^n), \quad \phi_1 = \phi^n + q_1$ $q_2 = hF(\phi_1) - q_1, \quad \phi^{n+1} = \phi_1 + q_2/2$
Magazenkov	2	$\phi^n = \phi^{n-2} + 2hF(\phi^{n-1})$ $\phi^{n+1} = \phi^n + \frac{h}{2}\left[3F(\phi^n) - F(\phi^{n-1})\right]$
Leapfrog–Trapezoidal	2	$\phi_1 = \phi^{n-1} + 2hF(\phi^n)$ $\phi^{n+1} = \phi^n + \frac{h}{2}\left[F(\phi_1) + F(\phi^n)\right]$
Adams–Bashforth	3	$\phi^{n+1} = \phi^n + \frac{h}{12}\left[23F(\phi^n) - 16F(\phi^{n-1}) + 5F(\phi^{n-2})\right]$
Adams–Moulton	3	$\phi^{n+1} = \phi^n + \frac{h}{12}\left[5F(\phi^{n+1}) + 8F(\phi^n) - F(\phi^{n-1})\right]$
ABM Predictor–Corrector	3	$\phi_1 = \phi^n + \frac{h}{2}\left[3F(\phi^n) - F(\phi^{n-1})\right]$ $\phi^{n+1} = \phi^n + \frac{h}{12}\left[5F(\phi_1) + 8F(\phi^n) - F(\phi^{n-1})\right]$
Runge–Kutta	3	$q_1 = hF(\phi^n), \quad \phi_1 = \phi^n + q_1/3$ $q_2 = hF(\phi_1) - 5q_1/9, \quad \phi_2 = \phi_1 + 15q_2/16$ $q_3 = hF(\phi_2) - 153q_2/128, \quad \phi^{n+1} = \phi_2 + 8q_3/15$
Runge–Kutta	4	$q_1 = hF(\phi^n), \quad q_2 = hF(\phi^n + q_1/2)$ $q_3 = hF(\phi^n + q_2/2), \quad q_4 = hF(\phi^n + q_3)$ $\phi^{n+1} = \phi^n + (q_1 + 2q_2 + 2q_3 + q_4)/6$

TABLE 2.1. Summary of methods for the solution of ordinary differential equations. The second- and third-order Runge–Kutta methods are low-storage variants; $h = \Delta t$.

Method	*Storage Factor*	*Efficiency Factor*	*Amplification Factor*	*Phase Error*	*Max s*
Forward	2	0	$1 + \frac{s^2}{2}$	$1 - \frac{s^2}{3}$	0
Backward	*	∞	$1 - \frac{s^2}{2}$	$1 - \frac{s^2}{3}$	∞
Asselin Leapfrog	3	< 1	$1 - \frac{\gamma s^2}{2(1-\gamma)}$	$1 + \frac{(1+2\gamma)s^2}{6(1-\gamma)}$	< 1
Leapfrog	2	1	1	$1 + \frac{s^2}{6}$	1
Adams–Bashforth–2	3	0	$1 + \frac{s^4}{4}$	$1 + \frac{5}{12}s^2$	0
Trapezoidal	*	∞	1	$1 - \frac{s^2}{12}$	∞
Runge–Kutta–2	2	0	$1 + \frac{s^4}{8}$	$1 + \frac{s^2}{6}$	0
Magazenkov	3	0.67	$1 - \frac{s^4}{4}$	$1 + \frac{s^2}{6}$	0.67
Leapfrog–Trapezoidal	3	0.71	$1 - \frac{s^4}{4}$	$1 - \frac{s^2}{12}$	1.41
Adams–Bashforth–3	4	0.72	$1 - \frac{3}{8}s^4$	$1 + \frac{289}{720}s^4$	0.72
Adams–Moulton–3	*	0	$1 + \frac{s^4}{24}$	$1 - \frac{11}{720}s^4$	0
ABM Predictor–Corrector–3	4	0.60	$1 - \frac{19}{144}s^4$	$1 + \frac{1243}{8640}s^4$	1.20
Runge–Kutta–3	2	0.58	$1 - \frac{s^4}{24}$	$1 + \frac{s^4}{30}$	1.73
Runge–Kutta–4	4†	0.70	$1 - \frac{s^6}{144}$	$1 - \frac{s^4}{120}$	2.82

†A storage factor of 3 may be achieved following the algorithm of Blum (1962).

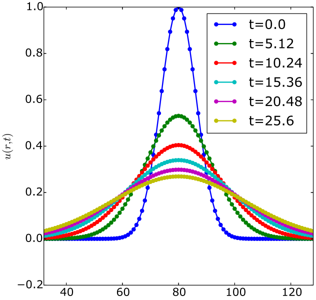
TABLE 2.2. Characteristics of the schemes listed in Table 2.1. The amplification factor and relative phase change are for well-resolved solutions to the oscillation equation, and $s = \kappa\Delta t$. "Max s" is the maximum value of $\kappa\Delta t$ for which the solution is nonamplifying. The storage and efficiency factors are defined in the text. No storage factor is given for implicit schemes.

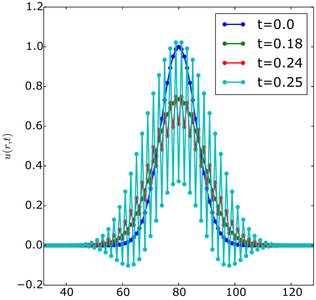
Diffusion and C.F.L.

Test Case

$$\partial_t u(r,t) = D\partial_r^2 u(r,t),$$
$$u(r,t=0) = e^{-(r-r_0)^2/\sigma^2} \quad (10)$$

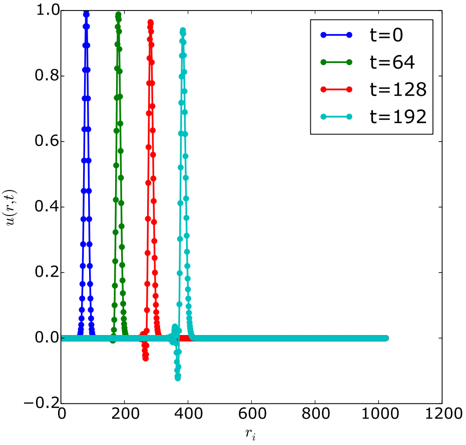
$n_{CFL} = 0.1$ $n_{CFL} = 1.0$





Advection simulation

- Numerical results using $u_0(r) = exp(-r^2/\sigma^2)$
- C.F.L : $n_{CFL} = 0.4$

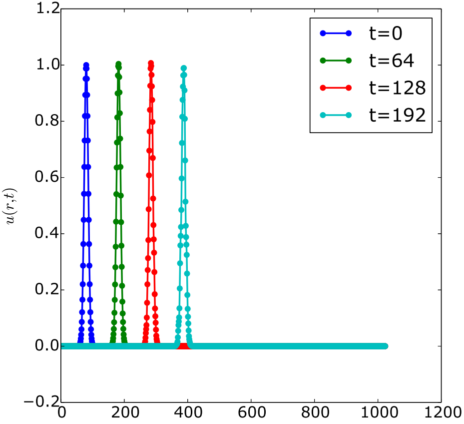


- system not conservative (diffusion)
- numerical oscillations generated
- phenomena disappears when $\Delta r \to 0$

Advection simulation

Ref. : C. S. Shu

- Numerical results using $u_0(r) = exp(-r^2/\sigma^2)$
- C.F.L : $n_{CFL} = 0.4$



- 5^{th} order WENO scheme used here
- smooth interpolation locally around discontinuities

$$\partial_t u(r,t) = -\frac{1}{\Delta r}\left(\hat{f}_{r+1/2} - \hat{f}_{r-1/2}\right)$$

$$\hat{f}_{r+1/2} = \hat{f}(u_{r-2}, \cdots, u_{r+2})$$

- Pro: parallel implementation