Recall: The stability of a method is determined by how it can handle the test equation $y' = \lambda y$.

Think of $\lambda$ as $\frac{df}{dy}$.

$\lambda$ large

$\lambda$ small

$y(t)$

$y(t)$

$y'' + 1001y' + 1000y = 0$

Solution: $y = c_1 e^{-t} + c_2 e^{-1000t}$

Now consider the following:

At first, we need to take small steps, because $y$ changes rapidly.

After $e^{-1000t}$ term has decayed, we want to take steps appropriate to $e^{-t}$, but we can't.

The $\lambda = -1000$ is lurking in the background.
If you open 5 books on ODEs, you will get 5 different definitions of "stiff". Here is mine:

**Def** An ODE is **stiff** if it involves at least two different time scales. The fast component disappears rapidly, and then we want to switch to larger steps appropriate for the slower components. We can't, because λ is still large, and makes things unstable.

**Sources of stiff problems**

- chemical reactions
- population models (grass - forest, insects - mammals)
- electric circuits

...and the original source of the word "stiff":

![Diagram of a spring and dashpot system with the equation y'' + dy' + ky = 0 if the damping constant d is large, system is stiff.]

To solve stiff problems, we need a method with a region of stability that extends all the way to 0. (A-stable or A(ω)-stable) All these methods are implicit.

Furthermore, fixed point iteration does not work any more. We need a nonlinear equation solver at every step.

So: - stiff methods require much more work per step
- but we can take bigger steps
Sources of stiff methods

1. Implicit RK methods
2. Some implicit multistep methods (not AM in general)

Most popular: Gear's Backward Differentiation Formulas (BDF)

Recall: Adamo-Houlton: \[ y_{n+1} = y_n + h \left[ a_0 f_{n+1} + \ldots + b_k f_{n-k+1} \right] \]
only one \( y \)-value

BDF: \[ y_{n+1} = \alpha_1 y_n + \ldots + \alpha_k y_{n-k+1} + \beta_0 h f_{n+1} \]
only one \( f \)-value

Special case: \( k=1 \) \[ y_{n+1} = y_n + h f_{n+1} \] (backward Euler)

\( k=2 \) Let's derive this, for practice

\[ y_{n+1} = \alpha_1 y_n + \alpha_2 y_{n-1} + \beta_0 h f_{n+1} \]

\[ \alpha_1 \quad y_n = y_n \]
\[ \alpha_2 \quad y_{n-1} = y_n - h y'_n + \frac{h^2}{2} y''_n - \frac{h^3}{6} y'''_n + \frac{h^4}{24} y^{(iv)}_n + \ldots \]
\[ \beta_0 \quad h f_{n+1} = h y_{n+1} + h^2 y''_n + \frac{h^3}{2} y'''_n + \frac{h^4}{6} y^{(iv)}_n + \ldots \]

\[ \alpha_1 y_n + \alpha_2 y_{n-1} + \beta_0 h f_{n+1} = (\alpha_1 + \alpha_2) y_n + (\beta_0 - \alpha_2) h y'_n + \left( \frac{\alpha_2}{2} + \beta_0 \right) h^2 y''_n + \left( \frac{\alpha_2}{2} - \alpha_2 \right) h^3 y'''_n + \ldots \]

True \[ y_{n+1} = y_n + h y'_n + \frac{h^2}{2} y''_n + \frac{h^3}{6} y'''_n + \ldots \]

\[ \alpha_1 + \alpha_2 = 1 \quad \beta_0 = \frac{2}{3} \]
\[ -\alpha_2 + \beta_0 = 1 \quad \alpha_2 = -\frac{1}{3} \]
\[ \frac{\alpha_2}{2} + \beta_0 = \frac{1}{2} \quad \alpha_1 = 1 - \alpha_2 = \frac{4}{3} \]

\[ y_{n+1} = \frac{4}{3} y_n - \frac{1}{3} y_{n-1} + \frac{2}{3} h f_{n+1} \]
Leading error term: \[ \left[ \frac{1}{6} - \left( \frac{p_0}{2} - \frac{\alpha \varepsilon}{6} \right) \right] h^3 \gamma u'' = -\frac{2}{9} h^3 \gamma u'' \]

\[ \Rightarrow \text{order 2} \]

**Check stability**

Chow. eq: \[ r^2 = \frac{4}{3} r - \frac{1}{3} + \frac{2}{9} h^2 \]

Reduced Chow. eq: \[ r^2 = \frac{4}{3} r - \frac{1}{3} \]

Solutions: \( r_1 = 1, \ r_2 = \frac{1}{3} \)

Stable for small \( h \)

\( k=3 \)

\[ y_{n+1} = \frac{18}{11} y_n - \frac{9}{11} y_{n-1} + \frac{2}{11} y_{n-2} + \frac{6}{11} h f_{n+1} \]

Order 3

Q: How do we know this approach produces \( A(\omega) \)-stable methods?

A: We don't. It just happens to work.

In fact, for \( k \geq 7 \), it does not work any more.

See following page for pictures of regions of stability.

**Implementation:**

Predictor - Corrector

- use \( AB \) as predictor
- solve nonlinear equation based on BDF as corrector
Region of Stability for Backward Differentiation Formulas of order 1 through 6

$k = 1$

$k = 2$

$k = 3$

$k = 4$

$k = 5$

$k = 6$
Notation: \[ \alpha_k y_{n+k} + \alpha_{k-1} y_{n+k-1} + \cdots + \alpha_0 y_0 = h \left[ \beta_k f_{n+k} + \cdots + \beta_0 f_n \right] \]

\[ \alpha(r) = \sum \alpha_s r^s, \quad \beta(r) = \sum \beta_s r^s \]

Note: char. eq. is \[ \alpha(r) - h \lambda \beta(r) = 0 \]

reduced char. eq.: \[ \alpha(r) = 0 \]

Def: Truncation error at \( t_n \)

\[ (T_h y)_n = \frac{1}{h} \left[ \sum_{s=0}^{K} \alpha_s y(t_{n+s}) - h \sum_{s=0}^{K} \beta_s y'(t_{n+s}) \right] \]

Facts

1. Assume \( y \in C^{p+1} \)
   The method has order \( p \) if
   \[ (T_h y)_n = c \cdot h^p \quad y^{(p)}(3) \]

2. It suffices to check this for \( h = 1, n = 0 \)
   \[ \sum \alpha_s y(s) - \sum \beta_s y'(s) = c \cdot y^{(p+1)}(3) \]

3. The method has order \( p \)
   \[ \Rightarrow \frac{\alpha(e^z)}{z} - \beta(e^z) \text{ has a zero of order } p \text{ at } z = 0 \]
   \[ \Rightarrow \frac{\alpha(z)}{\ln(z)} - \beta(z) \quad z = 1 \]

Sketch of proof: Take \( y(t) = e^{zt} \) for arbitrary fixed \( z \) in (2)
Applications

1. Given $\alpha(r)$, we can find $\beta(r)$ based on the formulas in $\star\star$.

Since $\alpha(r)$ is the reduced char. poly, we can insure stability for small $h$.

Normally, this leads to pretty much the same calculations as just expanding everything in Taylor series, but for special cases people have developed shortcuts.

Example: For ABM, $\alpha(r) = r^k - r^{k-1}$

2. You can use $\star\star$ to prove general theorems about achievable order:
   - maximum achievable is $p = 2k$
   - maximum achievable stable method is
     $p = k+1$ (k odd)
     $k+2$ (k even)

3. Given $\beta(r)$, we can find $\alpha(r)$ based on $\star\star$

Q: How do we know this will produce a stable method?
A: We don't. We just hope.

Once again, normally this is equivalent to Taylor series matching, but in special cases you can develop shortcuts.

I will sketch how to do this for BDF.
This will also explain where the name comes from.
Sideline to the sideline

Assume \( \{y_n\} \) is an infinite sequence.

\[(\Delta y)_n = y_{n+1} - y_n \quad \text{"forward difference"}\]
\[(\Delta^2 y)_n = y_n - y_{n-1} \quad \text{"backward difference"}\]

\[(\Delta^3 y)_n = (\Delta y)_n - (\Delta y)_{n-1} = (y_n - y_{n-1}) - (y_{n-1} - y_{n-2}) = y_n - 2y_{n-1} + y_{n-2}\]

\[(\Delta^4 y)_n = y_n - 3y_{n-1} + 3y_{n-2} - y_{n-3}\]

For completeness, \((\Delta^0 y)_n = y_n\)

**Fact** Any \( \sum_{s=0}^{k} x_s y_s \) can be written as \( \sum_{s=0}^{k} x_s (\Delta^s y)_k \)

**Example:** \[y_3 + 2y_2 - y_1 + y_0 = (-1)(\Delta^3 y)_3 + 2(\Delta^2 y)_3 + (-3)(\Delta y)_3 + 3(\Delta^0 y)_3\]

**Check:**
\[(-1) \cdot (y_3 - 3y_2 + 3y_1 - y_0) + 2 \cdot (y_3 - 2y_2 + y_1) + (-3) \cdot (y_3 - y_2) + 3 \cdot (y_3)\]
\[y_3 + 2y_2 - y_1 + y_0\]

We can do the same for functions:

\[(\Delta y)(t) = y(t+1) - y(t)\]
\[(\Delta^2 y)(t) = y(t) - y(t-1)\]

Now back to the original sideline
Backward Differentiation Formulas

I think they really should be called "backward difference formulas", but they did not ask me.

Fix $k$. In BDF, we take $\beta(t) = t^k$

\[ \sum \alpha_s y'(s) = \sum \beta_s y'(s) = c \cdot y^{(p+1)}(s) \]

Convert the $\alpha$-term to backward difference form:

\[ \sum_{s=0}^{k} \alpha_s (\Delta^s_y)(k) = y'(k) = c \cdot y^{(p+1)}(s) \]

Choose $\varepsilon \in C$ arbitrary, take $y(t) = e^{k \varepsilon t}$

\[ y(k) = e^{k \varepsilon k} = e^{k^2 \varepsilon} \]

\[ (\Delta y)(k) = e^{k^2 \varepsilon} - e^{(k-1)^2 \varepsilon} = e^{k^2 \varepsilon} \left( \frac{e^{(k-1)^2 \varepsilon}}{e^{k^2 \varepsilon}} \right) \]

\[ (\Delta^s y)(k) = e^{k^2 \varepsilon} \left( \frac{e^{(k-1)^s \varepsilon}}{e^{k^2 \varepsilon}} \right) \]

This gives

\[ e^{k^2 \varepsilon} \sum_{s=0}^{k} \alpha_s \left( \frac{e^{(k-1)^s \varepsilon}}{e^{k^2 \varepsilon}} \right)^s - \varepsilon e^{k^2 \varepsilon} = c \cdot e^{k^2 \varepsilon} \cdot \varepsilon^{p+1} \]

Make substitution

\[ J = \frac{e^{k^2 t}}{\varepsilon} \quad \iff \quad \varepsilon = \ln(1-J) \]

to end up with

\[ \sum_{s=0}^{k} \alpha_s J^s + \ln(1-J) = O(J^{p+1}) \]

\[ -3 - \frac{3}{2} J^2 - \frac{5}{3} J^3 \ldots \]

Read off: $\alpha_0 = 0, \alpha_s = \frac{1}{s}$ for $s \geq 1$
Example: \( k = 2 \)

\[
\gamma_0 (v_0^2) + x_1 (v_1^2) + x_2 (v_2^2)
\]

\[
= 0 \cdot \gamma_2 + 1 \cdot (\gamma_2 - \gamma_1) + \frac{1}{2} \cdot (\gamma_2 - 2 \gamma_1 + \gamma_0)
\]

\[
= \frac{3}{2} \gamma_2 - 2 \gamma_1 + \frac{1}{2} \gamma_0
\]

This leads to

\[
\frac{2}{3} \gamma_{n+2} - 2 \gamma_{n+1} + \frac{1}{2} \gamma_n = h f_{n+2}
\]

or

\[
\gamma_{n+2} = \frac{4}{3} \gamma_{n+1} - \frac{1}{3} \gamma_n + \frac{2}{3} h f_{n+2}
\]

(which we derived before)