ODEs

Prateek Sharma (prateek@physics.iisc.ernet.in)
Office: D2-08

Reduction to 1st order ODEs

$$\frac{d^2y}{dx^2} + q(x)\frac{dy}{dx} = r(x)$$

$$\frac{dy}{dx} = z(x)$$

$$\frac{dz}{dx} = r(x) - q(x)z(x)$$

new variables introduced s.t. equations are well-behaved

generic set of N coupled first-order ODEs:

$$\frac{dy_i(x)}{dx} = f_i(x, y_1, \ldots, y_N), \qquad i = 1, \ldots, N$$

BVPs & IVPs

Boundary conditions are algebraic conditions on the values of the functions y_i at the boundaries. They can be satisfied at discrete specified points. Two broad categories:

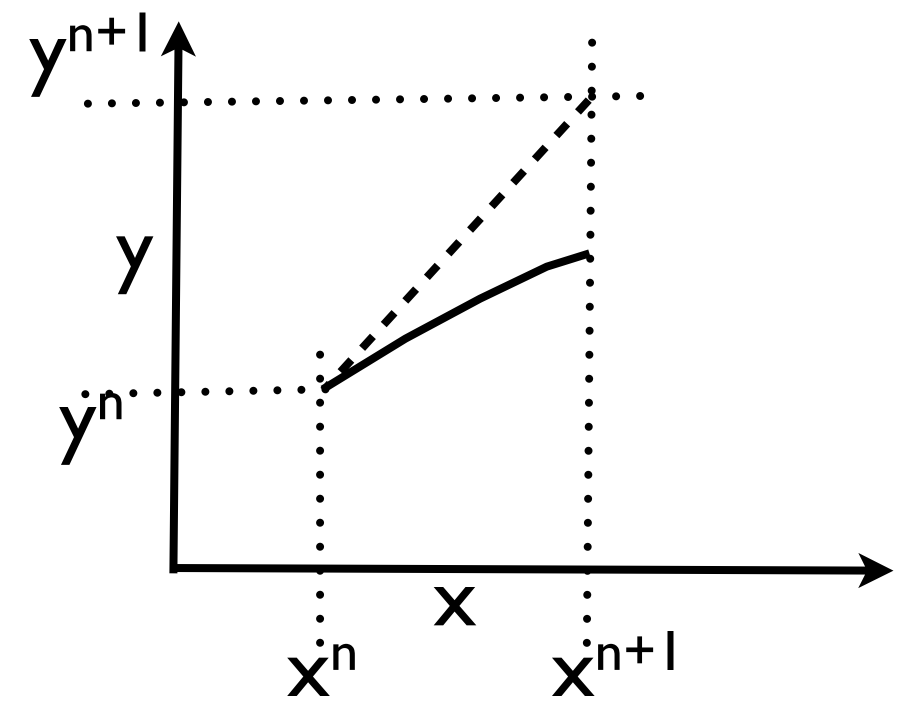
initial value problems all the y_i are given at some starting value x_s, and it is desired to find the y_i's at some final point x_f, or at some discrete list of points. e.g., planetary orbits with given initial conditions

two-point boundary value problems boundary conditions are specified at more than one x; some of the conditions will be specified at x_s and the remainder at x_f*; e.g.*, stellar structure: core density, temperature at inner boundary and radius, luminosity at outer boundary. in general more difficult.

lets consider IVPs for now

Forward Euler

$$\frac{y_i^{n+1} - y_i^n}{\Delta x} = f_i(x^n, y_1^n, .., y_N^n); i = 1, .., N$$



$$y^{n+1} = y^n + \frac{dy}{dx}\Delta x + \frac{d^2y}{dx^2}\frac{\Delta x^2}{2} + \ldots$$

$$\frac{dy}{dx} + f'\frac{\Delta x}{2} = f$$

modified eq. 1[st] order accurate

numerically unstable for large Δx! e.g., dy/dx=-ay, y(0)=1 => $y=e^{-ax}$

FE: $y^{n+1} = (1\text{-}ah)^{n+1}y^0$

|1-ah|<1 for stability!

Backward Euler

$$\frac{y_i^{n+1} - y_i^n}{\Delta x} = f_i(x^{n+1}, y_1^{n+1}, .., y_N^{n+1}); i = 1, .., N$$

also 1[st] order accurate; this is an implicit eq.
to be solved, e.g., via Newton-Raphson
unconditionally stable!

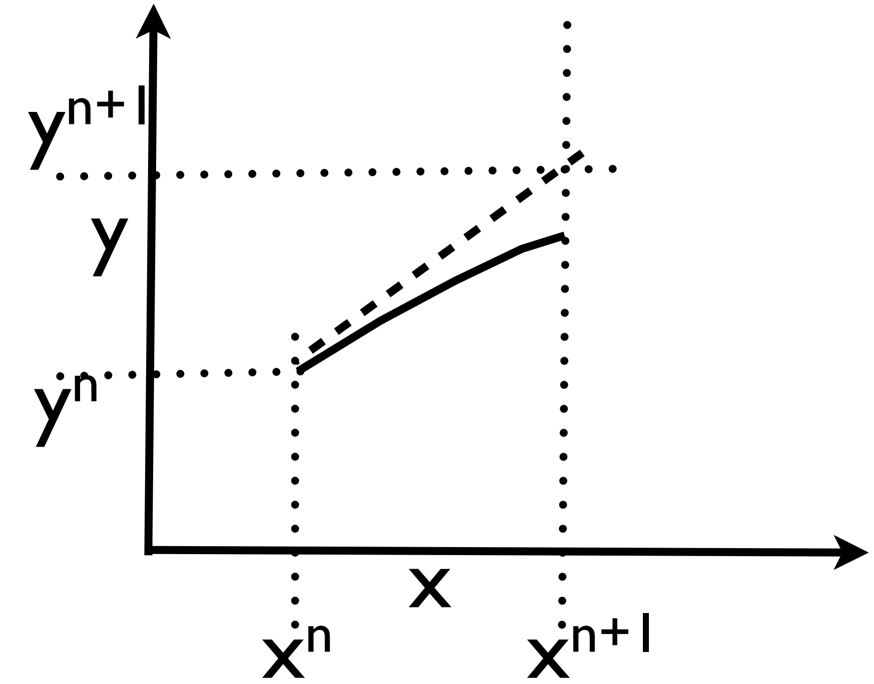


Figure 2: Numerical instability in the FE method for dt>0.2; IVP is dy/dx = -10y, y(0

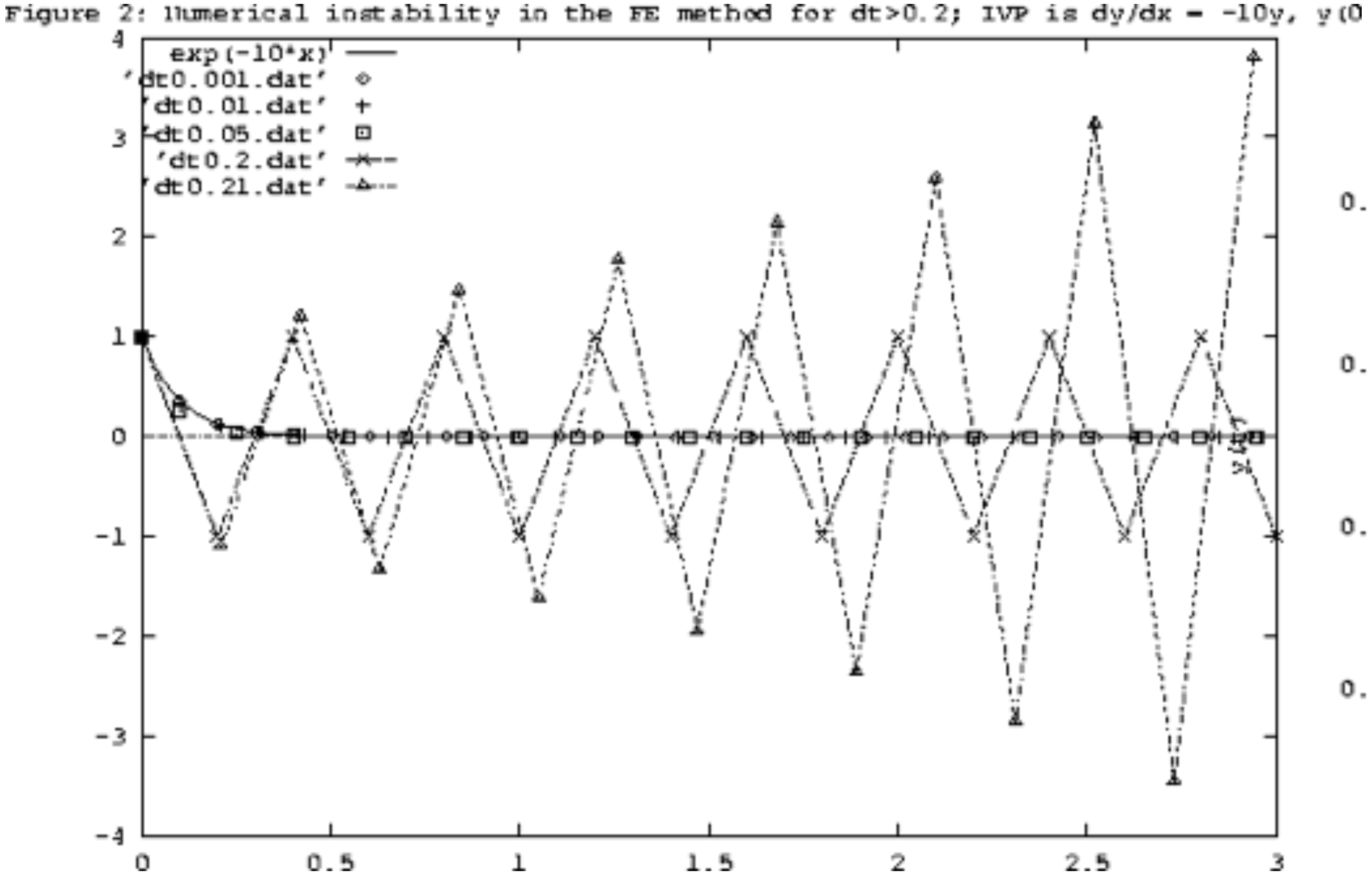
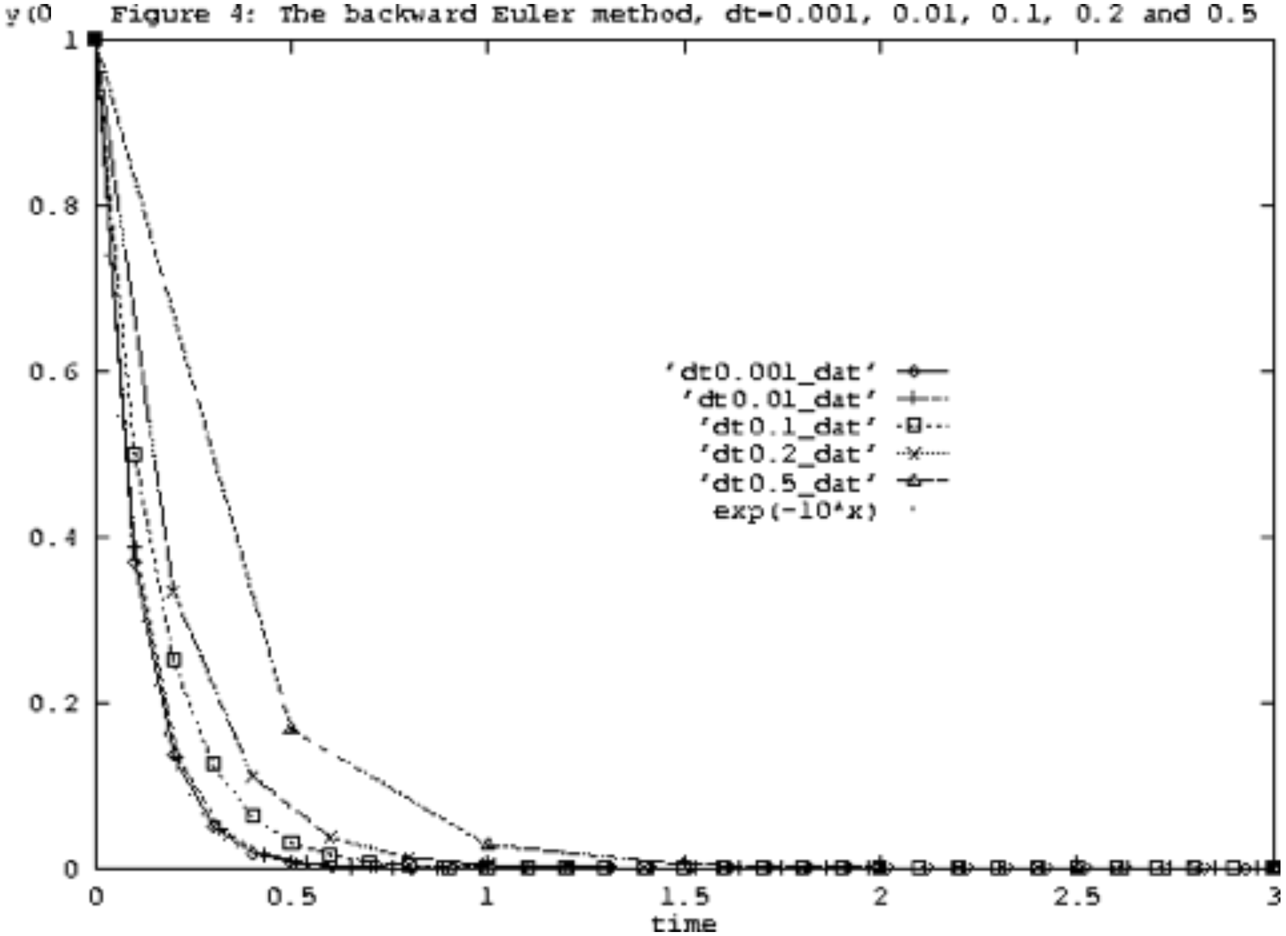


Figure 4: The backward Euler method, dt=0.001, 0.01, 0.1, 0.2 and 0.5



Runge-Kutta Schemes

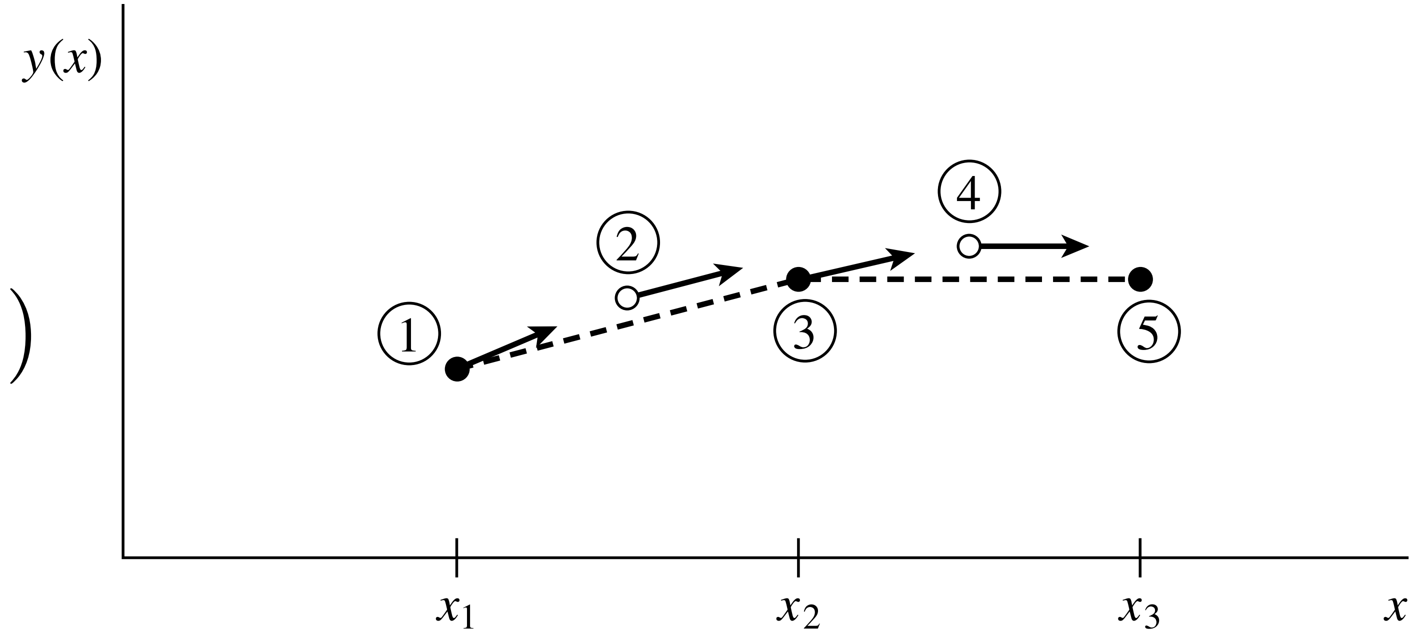
Euler is only 1st order accurate!

center the derivative at n+1/2: mid-point/RK2

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf\left(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}k_1\right)$$

$$y_{n+1} = y_n + k_2 + O(h^3)$$

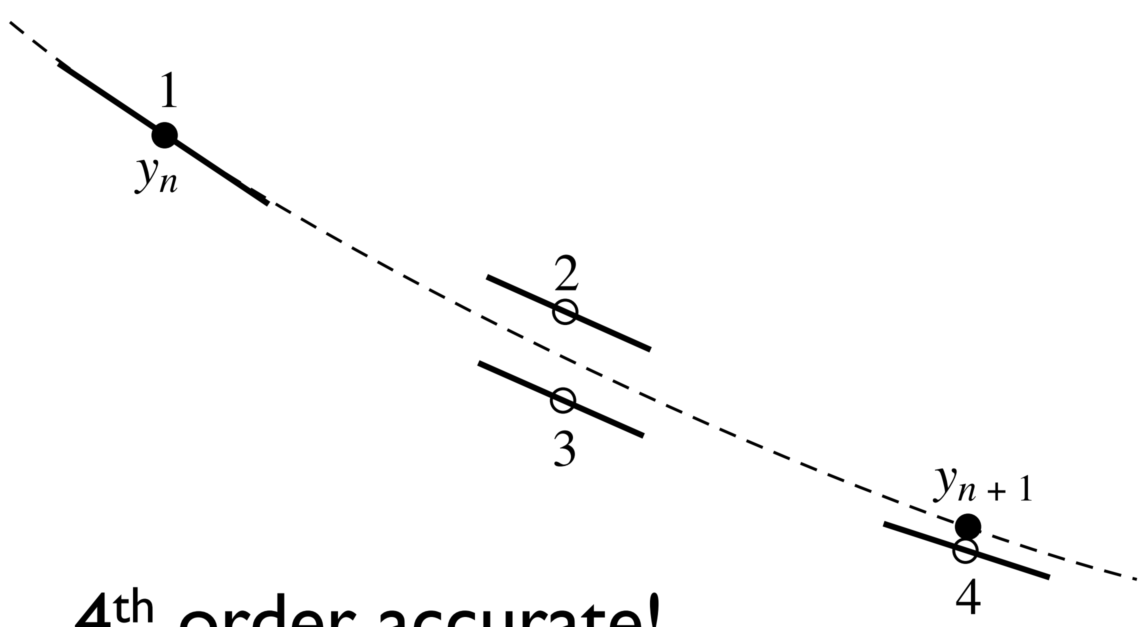


2nd order accurate!

Can be easily shown via Taylor series

RK4

$$k_1 = hf(x_n, y_n)$$
$$k_2 = hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2})$$
$$k_3 = hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2})$$
$$k_4 = hf(x_n + h, y_n + k_3)$$
$$y_{n+1} = y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + O(h^5)$$



4th order accurate!

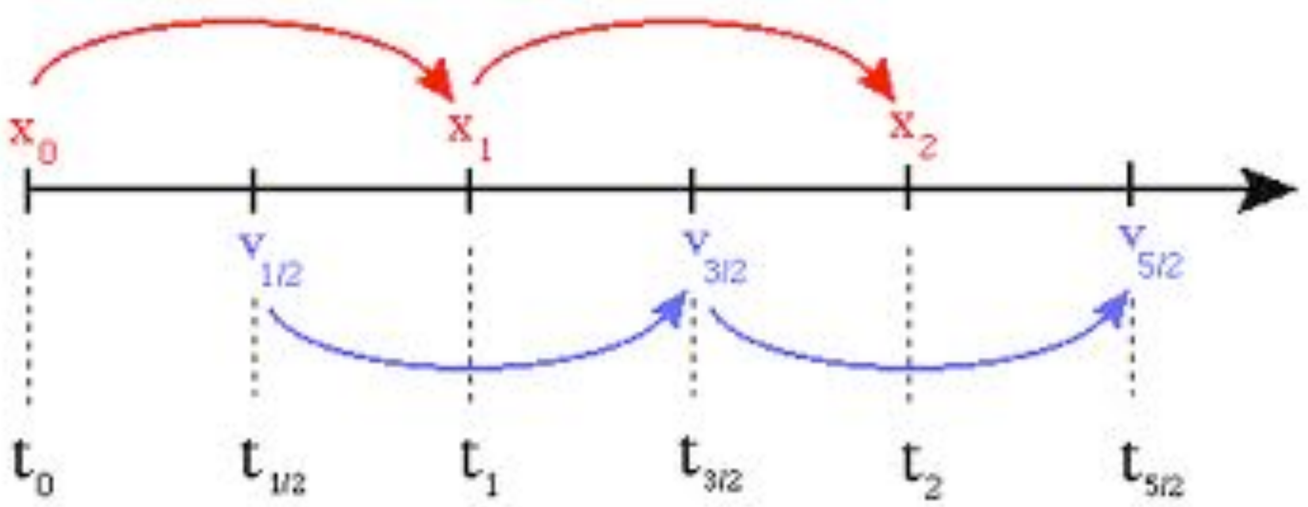
most common ODE solver; while more accurate, also requires twice the number of function evaluations

Leap-frog Method

$\ddot{x} = F(x)$ Newton's 2nd law: e.g., planetary orbits, molecular dynamics, etc.

$\dot{v} = F(x), \ \dot{x} \equiv v$ $\ddot{x} = -\nabla V(x)$ $E(x,v) = \frac{1}{2}|v|^2 + V(x)$

time-reversible, conservative, Hamiltonian system



$$x_i = x_{i-1} + v_{i-1/2}\,\Delta t,$$

$$a_i = F(x_i)$$

$$v_{i+1/2} = v_{i-1/2} + a_i\,\Delta t,$$

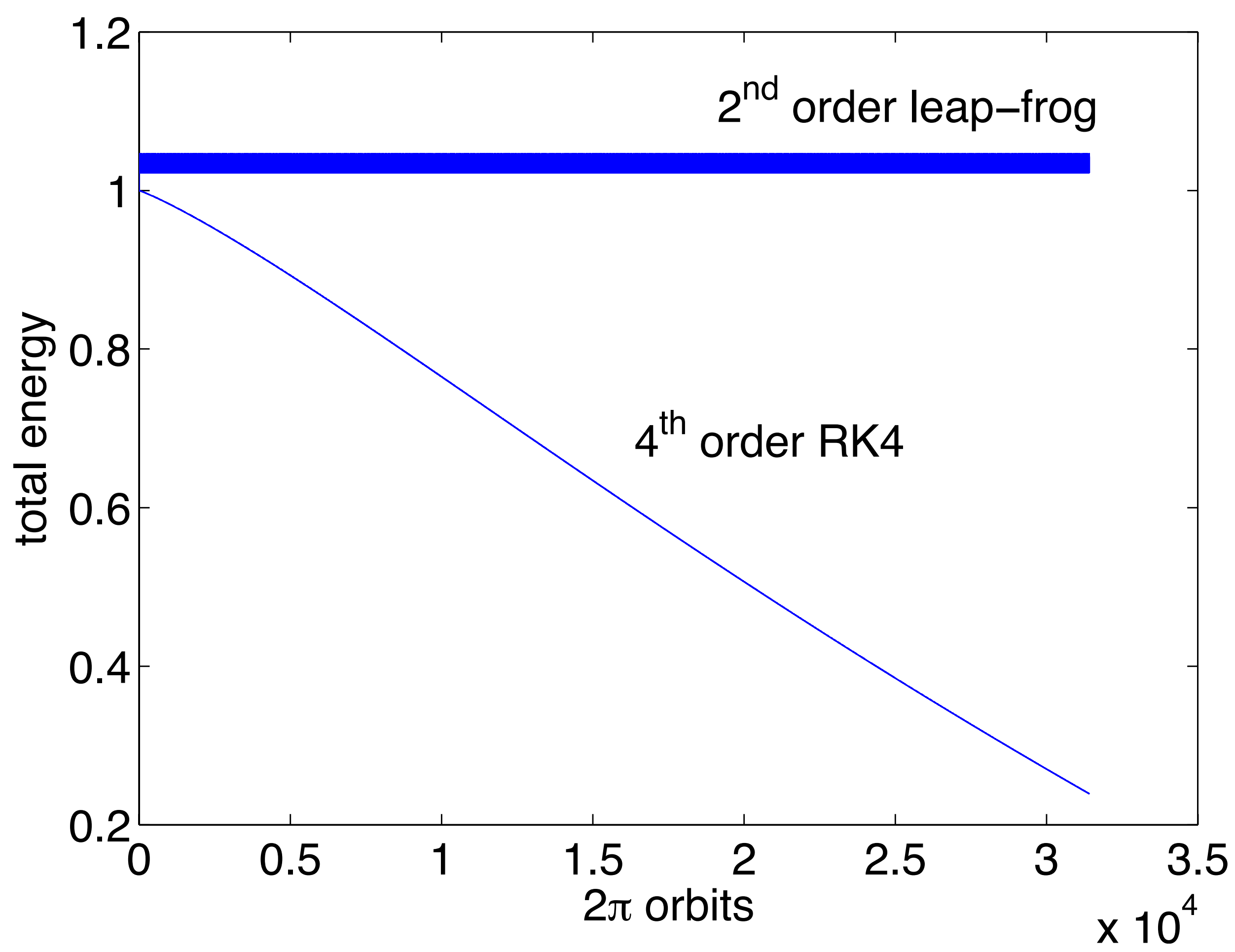
$$x_{i+1} = x_i + v_i\,\Delta t + \tfrac{1}{2}\,a_i\,\Delta t^{\,2},$$

$$v_{i+1} = v_i + \tfrac{1}{2}\,(a_i + a_{i+1})\,\Delta t.$$

LF/Verlet very useful where we want to conserve energy, prevent secular errors, not just the formal one-step error RK, etc. lead to non-conservation of energy; energy drifts!

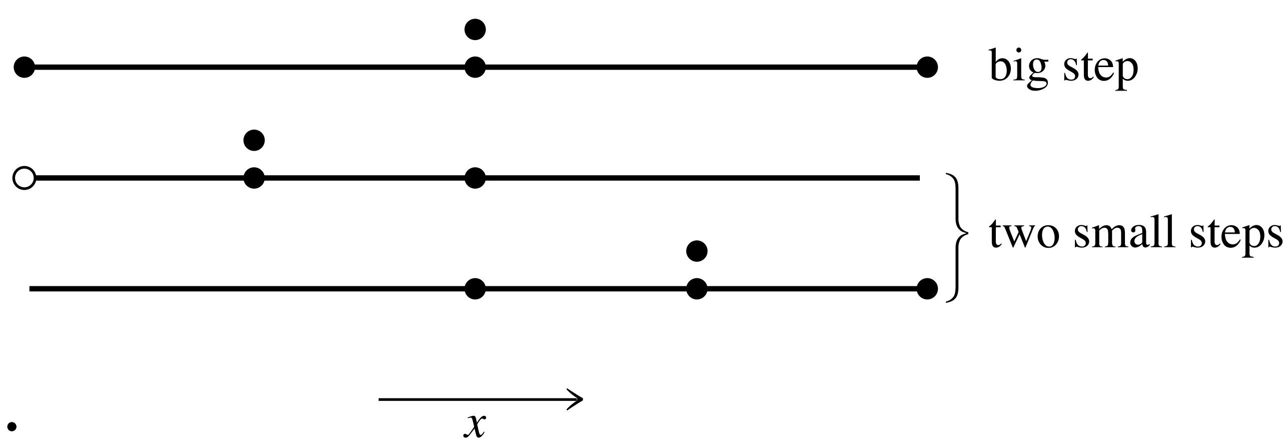
Cons: dt must be the same for time-reversibility, etc. lower order; RK, etc. fine for short-term evolution

RK, LF are explicit schemes and thus require $dt<1/\omega_0$ for stability



Adaptive step-size

stepsize (h) is chosen to achieve some pre-specified accuracy; algorithm should give an estimate of truncation error



Taylor expansion:

$$y(x+2h) = y_1 + (2h)^5\phi + O(h^6) + \dots$$

$$y(x+2h) = y_2 + 2(h^5)\phi + O(h^6) + \dots$$

$\Delta \equiv y_2 - y_1$ a good measure of truncation error

$$h_0 = h_1 \left| \frac{\Delta_0}{\Delta_1} \right|^{0.2}$$

Δ_0: desired accuracy

must use an adaptive stepsize

Embedded RK

Cash-Karp Parameters for Embedded Runga-Kutta Method								
i	a_i	b_{ij}					c_i	c_i^*
1							$\frac{37}{378}$	$\frac{2825}{27648}$
2	$\frac{1}{5}$	$\frac{1}{5}$					0	0
3	$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$				$\frac{250}{621}$	$\frac{18575}{48384}$
4	$\frac{3}{5}$	$\frac{3}{10}$	$-\frac{9}{10}$	$\frac{6}{5}$			$\frac{125}{594}$	$\frac{13525}{55296}$
5	1	$-\frac{11}{54}$	$\frac{5}{2}$	$-\frac{70}{27}$	$\frac{35}{27}$		0	$\frac{277}{14336}$
6	$\frac{7}{8}$	$\frac{1631}{55296}$	$\frac{175}{512}$	$\frac{575}{13824}$	$\frac{44275}{110592}$	$\frac{253}{4096}$	$\frac{512}{1771}$	$\frac{1}{4}$
$j =$		1	2	3	4	5		

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf(x_n + a_2 h, y_n + b_{21} k_1)$$

$$\cdots$$

$$k_6 = hf(x_n + a_6 h, y_n + b_{61} k_1 + \cdots + b_{65} k_5)$$

$$y_{n+1} = y_n + c_1 k_1 + c_2 k_2 + c_3 k_3 + c_4 k_4 + c_5 k_5 + c_6 k_6 + O(h^6)$$

$$y_{n+1}^* = y_n + c_1^* k_1 + c_2^* k_2 + c_3^* k_3 + c_4^* k_4 + c_5^* k_5 + c_6^* k_6 + O(h^5)$$

$$\Delta \equiv y_{n+1} - y_{n+1}^* = \sum_{i=1}^{6} (c_i - c_i^*) k_i$$

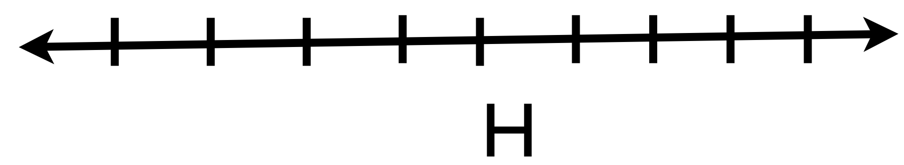
$$h_0 = h_1 \left| \frac{\Delta_0}{\Delta_1} \right|^{0.2}$$

1/2 the no. of fn. evals.!

recall truncation error, Δ_0 can't be too small

Modified Midpoint

$$h = H/n$$



$$z_0 \equiv y(x)$$

$$z_1 = z_0 + hf(x, z_0)$$

$$z_{m+1} = z_{m-1} + 2hf(x + mh, z_m) \qquad \text{for} \quad m = 1, 2, \ldots, n-1$$

$$y(x+H) \approx y_n \equiv \frac{1}{2}[z_n + z_{n-1} + hf(x+H, z_n)]$$

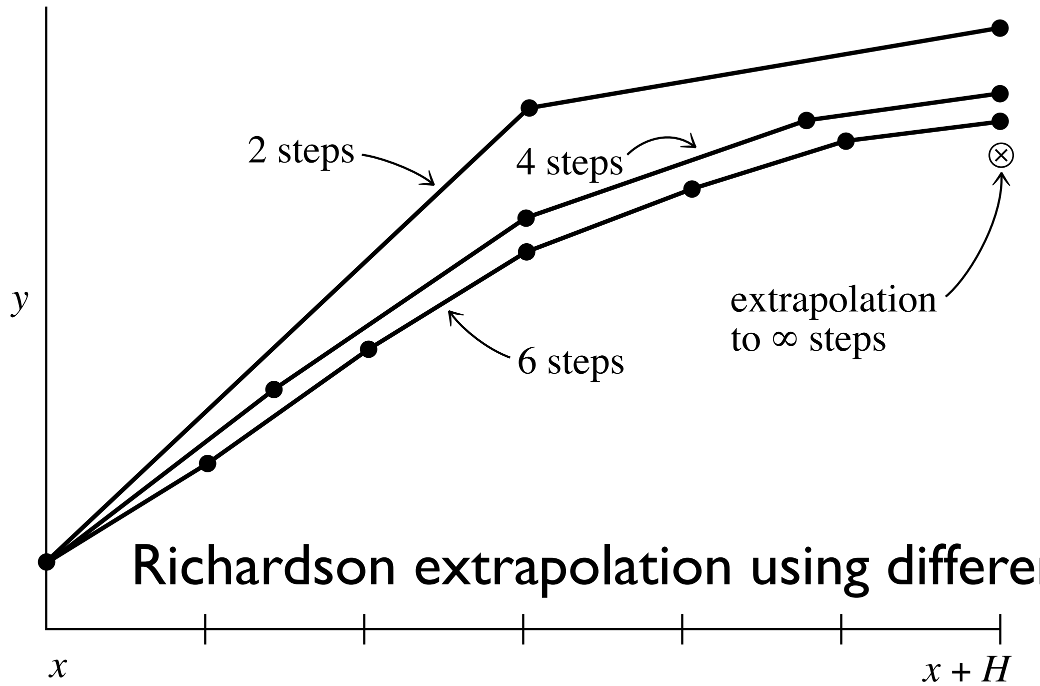
second order, with 1 derivative evaluation per h

$$y_n - y(x+H) = \sum_{i=1}^{\infty} \alpha_i h^{2i} \qquad y(x+H) \approx \frac{4y_n - y_{n/2}}{3}$$

4th order accurate,
only 1.5 fn. evaluations!

apply only for ODEs containing smooth functions

Bulirsch-Stoer



Richardson extrapolation using different h

apply only for ODEs containing smooth functions
for very high accuracy

RK4 w. adaptive stepsize for non-smooth fns.

rational vs. polynomial extrapolation of error

$$y_n - y(x+H) = \sum_{i=1}^{\infty} \alpha_i h^{2i}$$

use a method with even terms in error

$$n = 2, 4, 6, 8, 10, 12, 14, \ldots, [n_j = 2j], \ldots$$

sub-intervals; for each n obtain
approximation & error estimate
go to higher n if error large

$$H_k = H \left(\frac{\epsilon}{\epsilon_{k+1,k}} \right)^{1/(2k+1)}$$

big stepsize should be small enough, s.t.,
kth column error is smaller than specified

Stiff Equations

$$u' = 998u + 1998v$$

$$v' = -999u - 1999v$$

$$u = 2e^{-x} - e^{-1000x}$$

vanishingly small, yet determines h!

$$v = -e^{-x} + e^{-1000x}$$

if there are widely separated timescales in the problem

implicit is the way to go

$$u(0) = 1 \qquad v(0) = 0$$

$$\mathbf{y}' = \mathbf{f}(\mathbf{y})$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\mathbf{f}(\mathbf{y}_{n+1})$$

semi-implicit linearized eq.

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\left[\mathbf{f}(\mathbf{y}_n) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right|_{\mathbf{y}_n} \cdot (\mathbf{y}_{n+1} - \mathbf{y}_n)\right]$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\left[\mathbf{1} - h\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right]^{-1} \cdot \mathbf{f}(\mathbf{y}_n)$$

only first order accurate!

Higher order implicit methods

Rosenbrock Methods

$$\mathbf{y}(x_0 + h) = \mathbf{y}_0 + \sum_{i=1}^{s} c_i \mathbf{k}_i$$

autonomous differential equation

$$y' = f(y).$$

$$\begin{pmatrix} \mathbf{y} \\ x \end{pmatrix}' = \begin{pmatrix} \mathbf{f} \\ 1 \end{pmatrix}$$

linearized semi-implicit eq.; generalization of embedded RK

$$(\mathbf{1} - \gamma h \mathbf{f}') \cdot \mathbf{k}_i = h\mathbf{f}\left(\mathbf{y}_0 + \sum_{j=1}^{i-1} \alpha_{ij}\mathbf{k}_j\right) + h\mathbf{f}' \cdot \sum_{j=1}^{i-1} \gamma_{ij}\mathbf{k}_j, \qquad i = 1, \ldots, s$$

we won't go through details, see NR

$$\mathbf{g}_i = \sum_{j=1}^{i-1} \gamma_{ij}\mathbf{k}_j + \gamma\mathbf{k}_i$$

$$(\mathbf{1}/\gamma h - \mathbf{f}') \cdot \mathbf{g}_1 = \mathbf{f}(\mathbf{y}_0)$$

$$(\mathbf{1}/\gamma h - \mathbf{f}') \cdot \mathbf{g}_2 = \mathbf{f}(\mathbf{y}_0 + a_{21}\mathbf{g}_1) + c_{21}\mathbf{g}_1/h$$

$$(\mathbf{1}/\gamma h - \mathbf{f}') \cdot \mathbf{g}_3 = \mathbf{f}(\mathbf{y}_0 + a_{31}\mathbf{g}_1 + a_{32}\mathbf{g}_2) + (c_{31}\mathbf{g}_1 + c_{32}\mathbf{g}_2)/h$$

$$(\mathbf{1}/\gamma h - \mathbf{f}') \cdot \mathbf{g}_4 = \mathbf{f}(\mathbf{y}_0 + a_{41}\mathbf{g}_1 + a_{42}\mathbf{g}_2 + a_{43}\mathbf{g}_3) + (c_{41}\mathbf{g}_1 + c_{42}\mathbf{g}_2 + c_{43}\mathbf{g}_3)/h$$

solve via LU decomp.

Semi-implicit extrapolation

Semi-implicit Extrapolation implicit generalization of Bulirsch-Stoer

implicit mid-pt $$\mathbf{y}_{n+1} - \mathbf{y}_{n-1} = 2h\mathbf{f}\left(\frac{\mathbf{y}_{n+1} + \mathbf{y}_{n-1}}{2}\right)$$

$$\Delta_k \equiv \mathbf{y}_{k+1} - \mathbf{y}_k$$

$$\Delta_0 = \left[\mathbf{1} - h\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right]^{-1} \cdot h\mathbf{f}(\mathbf{y}_0)$$

$$\mathbf{y}_1 = \mathbf{y}_0 + \Delta_0$$

even order terms in error => can apply Richardson extrapolation

$k = 1, \ldots, m-1$, set

$$\Delta_k = \Delta_{k-1} + 2\left[\mathbf{1} - h\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right]^{-1} \cdot [h\mathbf{f}(\mathbf{y}_k) - \Delta_{k-1}]$$

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \Delta_k$$

$$\Delta_m = \left[\mathbf{1} - h\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right]^{-1} \cdot [h\mathbf{f}(\mathbf{y}_m) - \Delta_{m-1}]$$

$$\overline{\mathbf{y}}_m = \mathbf{y}_m + \Delta_m$$

many more methods, predictor-corrector, etc.

Two-point BVPs

BVP

$$\frac{dy_i(x)}{dx} = g_i(x, y_1, y_2, \ldots, y_N) \qquad i = 1, 2, \ldots, N$$

x_1, the solution is supposed to satisfy

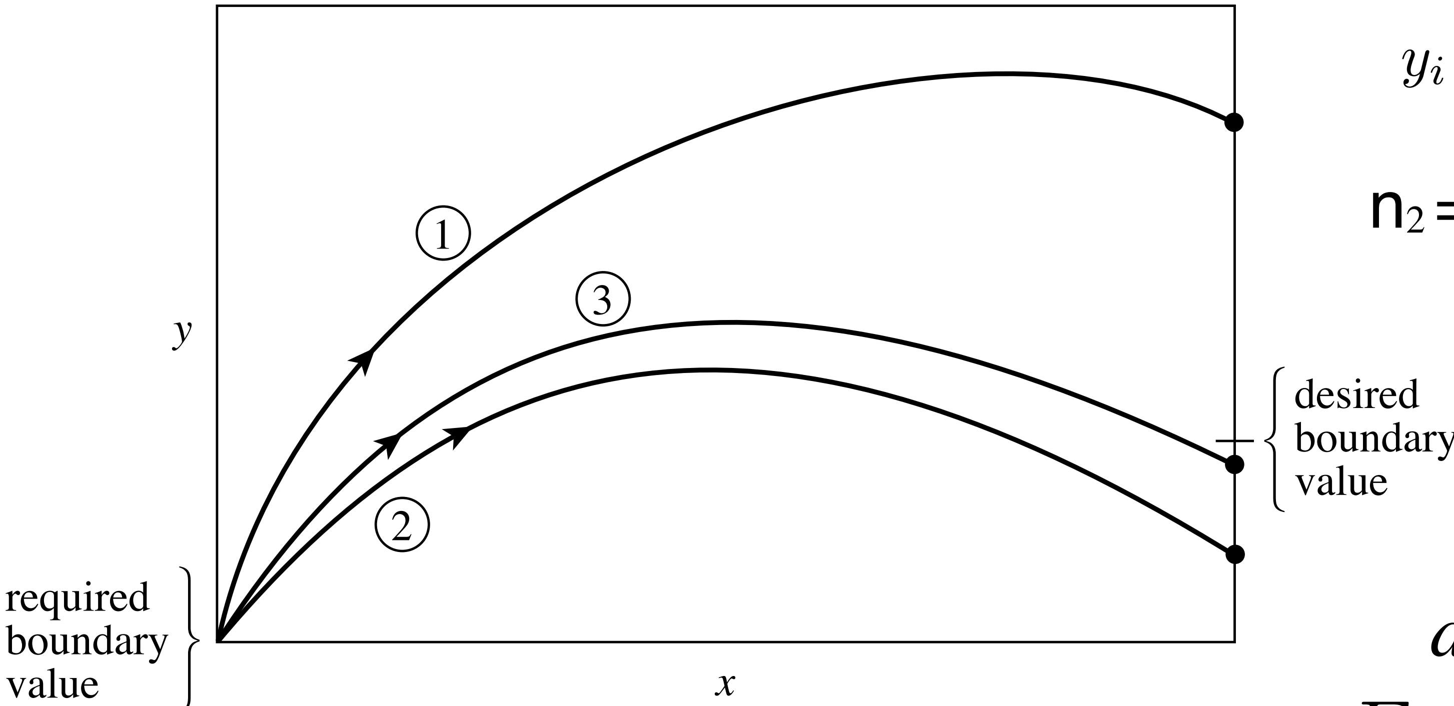
$$B_{1j}(x_1, y_1, y_2, \ldots, y_N) = 0 \qquad j = 1, \ldots, n_1$$

x_2, it is supposed to satisfy

$n_1+n_2=N$

$$B_{2k}(x_2, y_1, y_2, \ldots, y_N) = 0 \qquad k = 1, \ldots, n_2$$

Shooting Method



$$y_i(x_1) = y_i(x_1; V_1, \ldots, V_{n_2}) \qquad i = 1, \ldots, N$$

$n_2 = N - n_1$ *freely specifiable* starting values

solve for $y(x_2)$

discrepancy vector **F**

$$F_k = B_{2k}(x_2, \mathbf{y}) \qquad k = 1, \ldots, n_2$$

components of F=0 only for the desired solution

we want to solve multi-dim. roots of $F(V_1,..,Vn_2)=0$; use Newton-Raphson

$$\mathbf{J} \cdot \delta\mathbf{V} = -\mathbf{F} \qquad \mathbf{V}^{\text{new}} = \mathbf{V}^{\text{old}} + \delta\mathbf{V}$$

$$J_{ij} = \frac{\partial F_i}{\partial V_j} \approx \frac{F_i(V_1, \ldots, V_j + \Delta V_j, \ldots) - F_i(V_1, \ldots, V_j, \ldots)}{\Delta V_j}$$

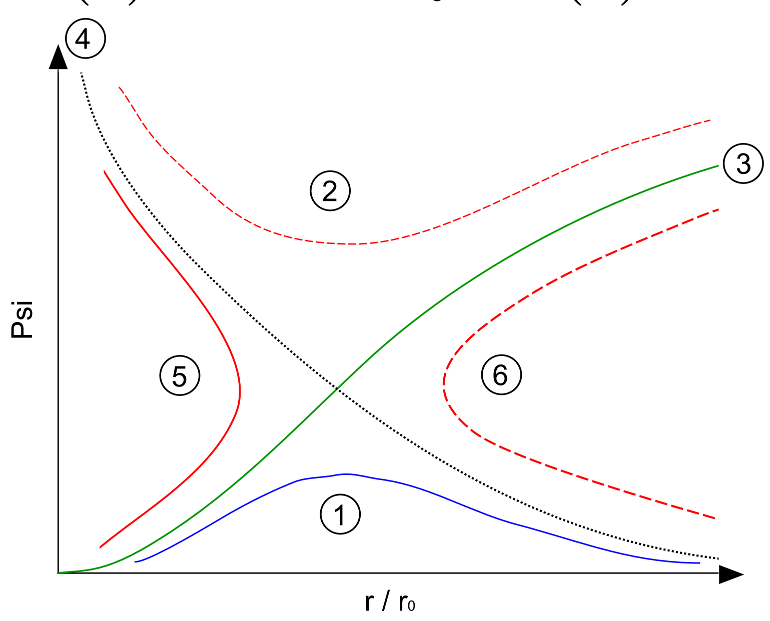
Shooting to fitting point

useful if singularities, e.g., sonic point

$$y_i(x_1) = y_i(x_1; V_{(1)1}, \ldots, V_{(1)n_2}) \qquad i = 1, \ldots, N$$

$$y_i(x_2) = y_i(x_2; V_{(2)1}, \ldots, V_{(2)n_1}) \qquad i = 1, \ldots, N$$

$$y_i(x_f; \mathbf{V}_{(1)}) = y_i(x_f; \mathbf{V}_{(2)}) \qquad i = 1, \ldots, N$$



match solution at appropriate fitting point

Relaxation Methods

$$\frac{dy}{dx} = g(x, y)$$

$$y_k - y_{k-1} \; - \; (x_k - x_{k-1})\, g\left[\tfrac{1}{2}(x_k + x_{k-1}), \tfrac{1}{2}(y_k + y_{k-1})\right] = 0$$ FDE

N(M-1) eqs. at interior pts.

$$0 = \mathbf{E}_k \equiv \mathbf{y}_k - \mathbf{y}_{k-1} - (x_k - x_{k-1})\mathbf{g}_k(x_k, x_{k-1}, \mathbf{y}_k, \mathbf{y}_{k-1}), \quad k = 2, 3, \ldots, M$$

$0 = \mathbf{E}_1 \equiv \mathbf{B}(x_1, \mathbf{y}_1)$ n_1 BCs at x_1 $\quad$ $0 = \mathbf{E}_{M+1} \equiv \mathbf{C}(x_M, \mathbf{y}_M)$ n_2 BCs at x_M, $n_1+n_2=N$

NM nonlinear eqs. for NM unknowns y^j_k ; j=1..N; k=1..M: can be solved via multi-D Newton-Raphson

Taylor series expansion at interior points; k=2..M: $\mathbf{E}_k(\mathbf{y}_k + \Delta\mathbf{y}_k, \mathbf{y}_{k-1} + \Delta\mathbf{y}_{k-1}) \approx \mathbf{E}_k(\mathbf{y}_k, \mathbf{y}_{k-1})$

$$+ \sum_{n=1}^{N} \frac{\partial \mathbf{E}_k}{\partial y_{n,k-1}} \Delta y_{n,k-1} + \sum_{n=1}^{N} \frac{\partial \mathbf{E}_k}{\partial y_{n,k}} \Delta y_{n,k}$$

$$S_{j,n} = \frac{\partial E_{j,k}}{\partial y_{n,k-1}}, \quad S_{j,n+N} = \frac{\partial E_{j,k}}{\partial y_{n,k}}, \quad n = 1, 2, \ldots, N$$

$$\sum_{n=1}^{N} S_{j,n} \Delta y_{n,k-1} + \sum_{n=N+1}^{2N} S_{j,n} \Delta y_{n-N,k} = -E_{j,k}, \quad j = 1, 2, \ldots, N$$

$$\sum_{n=1}^{N} S_{j,n} \Delta y_{n,1} = -E_{j,1}, \quad j = n_2 + 1, n_2 + 2, \ldots, N$$

inner boundary

$$\sum_{n=1}^{N} S_{j,n} \Delta y_{n,M} = -E_{j,M+1}, \quad j = 1, 2, \ldots, n_2$$

outer boundary

$$S_{j,n} = \frac{\partial E_{j,1}}{\partial y_{n,1}}, \quad n = 1, 2, \ldots, N$$

$$S_{j,n} = \frac{\partial E_{j,M+1}}{\partial y_{n,M}}, \quad n = 1, 2, \ldots, N$$

start with initial guess y0; add Δy in every iteration till desired accuracy

matrix-eq. for n=1..5 (5 dependent vars.), k=1..4 (4 grid pts.), 3/2 eqs. at inner/outer bdry

```
X X X X X                                V  B
X X X X X                                V  B
X X X X X                                V  B
X X X X X X X X X X                      V  B
X X X X X X X X X X                      V  B
X X X X X X X X X X                      V  B
X X X X X X X X X X                      V  B
X X X X X X X X X X                      V  B
          X X X X X X X X X X            V  B
          X X X X X X X X X X            V  B
          X X X X X X X X X X            V  B
          X X X X X X X X X X            V  B
          X X X X X X X X X X            V  B
                    X X X X X X X X X X  V  B
                    X X X X X X X X X X  V  B
                    X X X X X X X X X X  V  B
                    X X X X X X X X X X  V  B
                    X X X X X X X X X X  V  B
                              X X X X X  V  B
                              X X X X X  V  B
```

V:unknown increments

B: known $E_{j,k}$

X: known coupling coefficients S

block-diagonal matrix

Gaussian-Elimination inexpensive

see NR for details