Other Krylov subspace methods

idea: find "best" updates inside Krylov subspaces for A (or sometimes A^T). Some methods work even with unsymmetric problems: Here are some examples:

GMRES (Generalized minimal residual)
→ works for general systems, residual guaranteed to decrease but iterations become more costly (in computations / storage)

Restarted GMRES: run k steps of GMRES, throw away "memory" and start k steps of GMRES again from where we left over.

CGN: Conjugate Gradient for normal eq: specially designed to solve $A^TAx = A^Tb$.

BCG: Bi Conjugate gradient (for general problems)
(no optimality and unstable)

Bi-CGSTAB: Stabilized version

QMR: quasi-minimal residual (another stabilization of BCG)

etc...

- Krylov subspace methods are also used for finding eigenvalues (we shall discuss this later)
- Matlab has a good collection of methods for solving linear systems:

help sparfun

Chap 5 Initial value problems for Ordinary Diff Eq (ODE)

- Objective: develop methods to solve numerically problems of the kind

$$(1)\quad \begin{cases} \dfrac{dy}{dt} = f(t,y), & \text{for } t\in[a,b] \\ y(a) = \alpha \end{cases}$$

often an exact solution to (1) is too complicated or even cannot be found.

Extension to systems:

$$\begin{cases} \dfrac{dy_1}{dt} = f_1(t, y_1, y_2, \dots, y_n) \\ \dfrac{dy_2}{dt} = f_2(t, y_1, y_2, \dots, y_n) \\ \vdots \\ \dfrac{dy_n}{dt} = f_n(t, y_1, y_2, \dots, y_n) \end{cases}$$

for $t \in [a,b]$, s.t. $y_i(a) = \alpha_i$, $i = 1 \dots n$

$$\left(\frac{d\underline{y}}{dt} = \underline{f}(t, \underline{y}) \right)$$

And to n-th order IVP:

$$y^{(n)} = f(t, y, y', \dots, y^{(n-1)}) \quad \text{s.t.} \quad \begin{cases} y(a) = \alpha_1 \\ y'(a) = \alpha_2 \\ \vdots \\ y^{(n-1)}(a) = \alpha_n \end{cases}$$

Fundamental questions to answer about the IVP (1)

- existence: does (1) admit a sol?
- uniqueness: is the sol to (1) unique?
- "stability": Do small changes in the statement of the problem introduce small changes in the solution?

⚠ this is different from notion of stability for numerical methods

A problem satisfying all 3 properties is said to be **well posed** (in the sense of Hadamard)

The IVP (1) is well posed under mild assumptions on f. We need some terminology (maybe you've seen this many times before!)

Def (Lipschitz condition)

A function $f(t,y)$ satisfies a Lipschitz condition in the variable y on some $D \subset \mathbb{R}^2$ iff $\exists L>0$

$$|f(t,y_1) - f(t,y_2)| \leq L|y_1 - y_2| \quad \forall \; (t,y_1)\in D, \; (t,y_2)\in D.$$

L = Lipschitz constant for f.

(means $f(t,\bullet)$ is Lipschitz continuous for all t).

Example:

$$D = \{(t,y) \mid 0<t<1, -1<y<1\}$$

$$f(t,y) = t|y|.$$

$$|f(t,y_1) - f(t,y_2)| = |t|y_1| - t|y_2|| = |t|\,||y_1| - |y_2|| \leq |y_1 - y_2|$$

$$L = 1$$

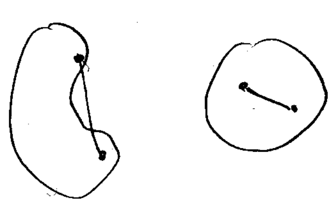
Def (Convex set)

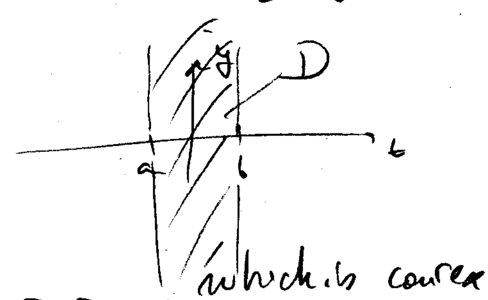
A set D is a convex set iff:

$$x,y \in D \Rightarrow \lambda x + (1-\lambda) y \in D \quad \forall \lambda \in [0,1]$$

in our case:

$$(t_1,y_1)\in D, (t_2,y_2)\in D \to (\lambda t_1 + (1-\lambda)t_2, \lambda y_1 + (1-\lambda)y_2) \quad \forall \lambda\in[0,1]$$





which is convex

For IVP we usually have:

$$D = \{(t,y) \mid a<t<b, y\in\mathbb{R}\}$$

Here is a sufficient (but not necessary) condition for Lipschitz cdt°:

<u>Theorem</u>: Let $f(t,y)$ be defined on some convex set $D \subset \mathbb{R}^2$.
If $\exists\, L > 0$ s.t.

$$\left| \frac{\partial f}{\partial y}(t,y) \right| \leq L \quad \forall (t,y) \in D$$

then f satisfies a Lipschitz condition on D in y with Lip. const. L.

<u>note</u>: $f(t,y) = t|y|$ satisfies the Lipschitz condition and yet $\frac{\partial f}{\partial y}$ does not exist at $y=0$.

<u>Existence and uniqueness</u> for IVP (1) are taken care by:

<u>Theorem</u> Let $D = \{(t,y) \mid a \leq t \leq b \ \&\ y \in \mathbb{R}\}$ and that

- $f(t,y)$ is continuous on D
- f satisfies Lip. cdt° on D in variable y, then the IVP (1)

<u>admits a unique solution</u>.

<u>"Stability"</u>: can be formulated as follows:

Let $y(t)$ solve $\begin{cases} \frac{dy}{dt} = f(t,y), & a \leq t \leq b \\ y(a) = \alpha \end{cases}$

$\exists\, \varepsilon_0 > 0, k > 0 \quad \forall \varepsilon$ s.t. $\varepsilon_0 > \varepsilon > 0$

$\forall\, \delta(t)$ continuous s.t. $|\delta(t)| \leq \varepsilon$

$\forall\, \delta_0 \in \mathbb{R}$ s.t. $|\delta_0| \leq \varepsilon$

$$\begin{cases} \frac{dz}{dt} = f(t,z) + \delta(t), & a \leq t \leq b, \\ z(a) = \alpha + \delta_0 \end{cases}$$ (perturbed problem)

has a unique solution $z(t)$ and:

$$|z(t) - y(t)| < k\varepsilon$$

(Essentially the mapping data $(\alpha, f(t,y))$ to solution y is continuous).

This "stability" property is crucial to trust solutions given by a numerical method (sources of error can be from the method itself or from numerical roundoff).

$\leadsto$ All numerical methods assume IVP is well-posed.

Example: $D = \{(t,y) \mid t \in [0,2],\ y \in \mathbb{R}\}$

$$\begin{cases} \dfrac{dy}{dt} = \underbrace{y - t^2 + 1}_{= f(t,y)}, & 0 \le t \le 2 \quad (IVP) \\ y(0) = \dfrac{1}{2} \end{cases}$$

$$\left|\frac{\partial f}{\partial y}\right| = |1| = 1 \Rightarrow f(t,y) \text{ satisfies Lip. cdt on } D \text{ with variable } y.$$

& f constant $\Rightarrow$ problem is stable to perturbations in init data.

We can verify this directly:

$$\begin{cases} \dfrac{dz}{dt} = z - t^2 + 1 + \delta & \text{(perturbed problem)} \\ z(0) = \dfrac{1}{2} + \delta_0 \end{cases}$$

IVP has sol: $y(t) = \frac{-1}{2} e^{t} + (t+1)^2$

$$z(t) = \left(\frac{-1}{2} + \delta_0\right)e^{t} + (t+1)^2 + (e^{t} - 1)\delta$$

$$|z - y| = |(\delta_0 + \delta)e^{t} - \delta| \le |\delta_0 + \delta| e^2 + |\delta| \le 2e^2\varepsilon + \varepsilon = (2e^2 + 1)\varepsilon$$

§5.2 Euler's method

A simple numerical method to solve IVP:

$$\begin{cases} \dfrac{dy}{dt} = f(t,y), & a \le t \le b \\ y(a) = \alpha \end{cases}$$ (assuming well-posedness)

Euler's method gives approx only on some mesh points:

$$t_j = a + hj \quad \text{where } h = \frac{b-a}{N} = \text{step size}$$

$$t_0 = a \quad t_1 \quad t_2 \cdots \quad t_{N-1} \quad t_N = b$$

($N+1$ equally spaced points) (can be relaxed)

Idea for Euler's method: Taylor's theorem:

$$y(t_{i+1}) = y(t_i) + y'(t_i)\underbrace{(t_{i+1}-t_i)}_{=h} + \frac{1}{2}y''(\xi_i)\underbrace{(t_{i+1}-t_i)^2}_{=h^2}$$

for some $\xi_i \in (t_i, t_{i+1})$.

$$= y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(\xi_i)$$

$$= y(t_i) + hf(t_i, y(t_i)) \underbrace{+ \tfrac{1}{2}h^2 y''(\xi_i)}_{\text{neglect for Euler's method}}$$ (y satisfies DE in IVP)

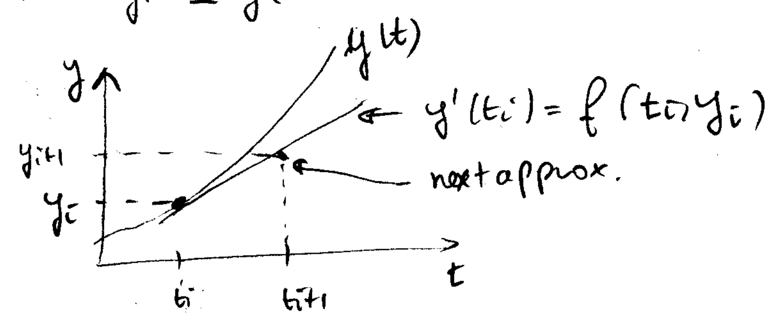
$y_0 = \alpha$

for $i = 0 .. N-1$

$$y_{i+1} = y_i + h \overset{\text{known}}{f(t_i, y_i)}$$

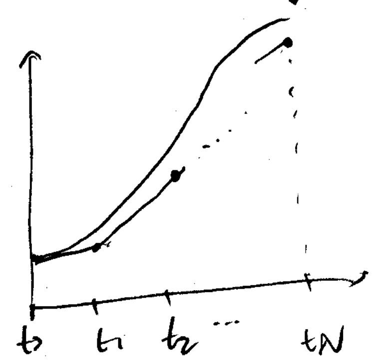
Geometrical interpretation

Assume $y_i = y(t_i)$



many times:

systematic error is introduced at every step



Several types of errors in numerical methods for DE:

- <u>Local truncation error</u>: error made in one step.

  For example in Euler's method: $O(h^2)$ since:
  $$y(t_i+h) = y(t_i) + h f(t, y(t_i)) + \underline{O(h^2)}$$
  (⚠ book defines this over $\frac{1}{h}$)

- <u>Local roundoff error</u>: precision used for computation (10^{-16} double, 10^{-8} single)

- <u>Global truncation error</u>: accumulation of all the local truncation errors. If local truncation error was $O(h^{n+1})$ then the global truncation error must be $O(h^n)$ because the number of steps $O(h)$.

- <u>Global roundoff error</u>: accumulation of local roundoff errors of previous steps.

- <u>Total error</u> = global truncation error + global roundoff error

If the global truncation error is $\underline{O(h^n)}$ then the method is of <u>order n</u>

<u>Example</u>: Euler's method is of order 1, and it is relatively simple to derive more precise bounds on the global truncation error:

<u>Thm</u> let f be continuous and satisfy Lipschitz condition with L-constant L on $D = \{(t,y) \mid t \in [a,b],\ y \in \mathbb{R}\}$ and that

$$|y''(t)| \leq M \quad \forall t \in [a,b] \text{ for some } M > 0.$$

let $y(t)$ be the sol to IVP $\begin{cases} \frac{dy}{dt} = f(t,y) \\ y(a) = \alpha \end{cases}$

and $y_0, y_1, \dots, y_N$ the approximations given by Euler's method then:

$$|y(t_i) - y_i| \leq \frac{hM}{2L}\left[e^{L(t_i-a)} - 1\right], \quad i = 0, 1, \dots, N.$$

proof (sketch)

$$y(t_{i+1}) = y(t_i) + h f(t_i, y(t_i)) + \frac{h^2}{2} y''(\xi_i)$$

$$y_{i+1} = y_i + h f(t_i, y_i)$$

$$\Rightarrow \quad |y(t_{i+1}) - y_{i+1}| \leq |y(t_i) - y_i| + h\,|f(t_i, y(t_i)) - f(t_i, y_i)| + \frac{h^2}{2}|y''(\xi_i)|$$

$$\leq (1 + hL)\,|y(t_i) - y_i| + \frac{h^2}{2} M$$

using Lemma 5.8 in book: (~ geometric series + bounding exp)

$$|y_{i+1} - w_{i+1}| \leq e^{(i+1)hL}\left(\underbrace{|y_0 - w_0|}_{=0} + \frac{h^2 M}{2hL}\right) - \frac{h^2 M}{2hL}$$

$$\leq \frac{h^2 M}{2hL}\left(e^{\overset{t_{i+1} - t_0 = t_{i+1} - a}{\overbrace{(i+1)hL}}} - 1\right)$$

<u>Lemma 5.8</u> $\{a_i\}_{i=0}^{k}$ is a seq with $a_0 \geq -\frac{t}{s}$ and

$$a_{i+1} \leq a_i(1+s) + t \quad \text{for } i = 0, \dots, k-1$$

$$\Rightarrow \quad a_{i+1} \leq e^{(i+1)s}\left(a_0 + \frac{t}{s}\right) - \frac{t}{s}$$

proof:

$$a_{i+1} \leq a_i(1+s) + t \leq ((1+s)a_{i-1} + t)(1+s) + t$$

$$\leq ((1+s)((1+s)a_{i-2} + t) + t)(1+s) + t$$

$$\vdots$$

$$\leq (1+s)^{i+1} a_0 + t \sum_{j=0}^{i} (1+s)^j$$

$$\Rightarrow a_{i+1} \leq (1+s)^{i+1} a_0 + t \left[\frac{1-(1+s)^{i+1}}{1-(1+s)}\right]$$

$$= (1+s)^{i+1}\left(a_0 + \frac{t}{s}\right) - \frac{t}{s}$$

$$\leq e^{(i+1)s}\left(a_0 + \frac{t}{s}\right) - \frac{t}{s}$$

since $(1+x)^{\alpha} = e^{\alpha \ln(1+x)} \leq e^{\alpha x}$

can be shown using Taylor's thm.

For Euler's method it's even possible to incorporate the round-off errors in the analysis:

instead of having:

$y_0 = \alpha$

for $i = 0..N-1$,

$\quad y_{i+1} = y_i + h f(t_i, y_i)$

we commit a mistake at each step:

$y_0 = \alpha + \delta_0$

for $i = 0..N-1$

$\quad y_{i+1} = y_i + h f(t_i, y_i) + \delta_{i+1}$

If $|\delta_i| < \delta$ it is possible to show:

$$|y(t_i) - y_i| \leq \frac{1}{L}\left(\frac{hM}{2} + \frac{\delta}{h}\right)\left[e^{L(t_i-a)} - 1\right] + |\delta_0|\, e^{L(t_i-a)}$$

performance degrades as $h \to 0$!!

(similar to problems with numerical differentiation)

The h giving smallest error is

$$h = \sqrt{\frac{2\delta}{M}} \quad \text{(simple calculation)}$$

§5.3 Higher order Taylor methods:

Same derivation as Euler method but carry Taylor series further:

$$y(t_{i+1}) = y(t_i) + h\,y'(t_i) + \frac{h^2}{2}y''(t_i) + \dots + \frac{h^n}{n!}y^{(n)}(t_i) + \frac{h^{n+1}}{(n+1)!}y^{(n+1)}(\xi_i)$$

for some $\xi_i \in (t_i, t_{i+1})$.

$$y'(t) = f(t, y(t))$$
$$y''(t) = \frac{d}{dt}\left[f(t, y(t))\right]$$
$$\vdots$$
$$y^{(k)}(t) = \frac{d^{k-1}}{dt^{k-1}}\left[f(t, y(t))\right]$$

Taylor method of order n

$y_0 = \alpha$,
for $i = 0 \dots N-1$,
$$y_{i+1} = y_i + h\,f(t, y(t)) + \frac{h^2}{2}\frac{d}{dt}\left[f(t, y(t))\right] + \dots + \frac{h^n}{n!}\frac{d^{n-1}}{dt^{n-1}}\left[f(t, y(t))\right]$$

Local truncation error is $O(h^{n+1})$

In general evaluating $\frac{d^k}{dt^k}\left[f(t, y(t))\right]$ can be quite involved because of the repeated application of the chain rule.

For example:

$$\begin{cases} y' = \cos t - \sin y + t^2 = f(t, y) \\ y(-1) = 3 \end{cases}$$

$$y'' = \frac{d}{dt}\left[f(t, y(t))\right] = -\sin t - y'\cos y + 2t$$

$$y''' = \frac{d^2}{dt^2}\left[f(t, y(t))\right] = -\cos t - y''\cos y + (y')^2\sin y + 2$$

$$y^{(4)} = \frac{d^3}{dt^3}\left[f(t, y(t))\right] = \sin t - y^{(3)}\cos y + 3y'y''\sin y + (y')^3\cos y$$

etc...

Note: • Assumes f is smooth ($n-1$ times differentiable if method of order n is desired)

• The higher the accuracy the more complicated the steps become

• differentiation can be carried out symbolically if possible!

§5.4 Runge-Kutta methods

Taylor series method for $\begin{cases} \frac{dy}{dt} = f(t,y), & t \in [a,b] \\ y(a) = A \end{cases}$

require us to compute formulas for $y'' = \frac{d}{dt}[f(t,y)]$

$y^{(3)} = \frac{d^2}{dt^2}[f(t,y)]$

etc...

which can be quite involved.

Runge-Kutta methods avoid this difficulty by carefully chosen combinations of values of $f(t,y)$.

Second order Runge Kutta methods:

Start with Taylor series:

$$y(t+h) = y(t) + h\,y'(t) + \frac{h^2}{2}y''(t) + \frac{h^3}{3!}y^{(3)}(t) + \dots$$

From the DE we get:

$$y'(t) = f(t,y) = \boxed{f}$$

$$y''(t) = \frac{d}{dt}f(t,y) = \frac{\partial f}{\partial t}(t,y) + y'\frac{\partial f}{\partial y}(t,y) = \boxed{f_t + f f_y}$$

$$y'''(t) = \frac{d^2}{dt^2}f(t,y) = f_{tt} + f f_{ty} + (f_t + f f_y)f_y + f\left(f_{ty} + f f_{yy}\right)$$

etc...

$$\Rightarrow \quad y(t+h) = y(t) + hf + \frac{h^2}{2}(f_t + f f_y) + O(h^3)$$

The idea is to eliminate the partial deriv of f by using the first few terms of the two variable Taylor series for $f(t,y)$:

$$f(t+h, y+hf) = f + \nabla f \cdot \begin{pmatrix} h \\ hf \end{pmatrix} + O(h^2)$$
$$= f + hf_t + hff_y + O(h^2)$$

Rewriting the Taylor series of y:

$$(*) \quad y(t+h) = y(t) + \frac{1}{2}hf + \frac{1}{2}h[\underbrace{f + hf_t + hff_y}_{\text{first 2 terms of 2 variable Taylor series of } f.}] + O(h^3)$$

$$= y(t) + \frac{1}{2}hf + \frac{1}{2}h\left[f(t+h, y+hf) + O(h^2)\right] + O(h^3).$$

Thus it's possible to construct an update which has the same $O(h^3)$ local truncation error as 2nd order Taylor method:

<u>Modified Euler method</u>

$y_0 = A$

for $i = 0, 1, \dots, N-1$

$F_1 = hf(t_i, y_i)$

$F_2 = hf(t_i+h, y_i+F_1)$

$y_{i+1} = y_i + \frac{1}{2}F_1 + \frac{1}{2}F_2$

The general form for Second order Runge Kutta updates is:

$$y(t+h) = y + w_1 hf + w_2 hf(t+\alpha h, y+\beta hf)$$

where w_1, w_2, α, β are parameters that we can adjust.

Using two variable Taylor expansion:

$$y(t+h) = y + w_1 hf + w_2 h\left[f + \alpha h f_t + \beta h f f_y\right]$$

Matching comparable terms with (*) we get:

$$\begin{cases} w_1 + w_2 = 1 \\ w_2 \alpha = \frac{1}{2} \\ w_2 \beta = \frac{1}{2} \end{cases}$$

$\rightsquigarrow$ so there is a family of RK order 2 methods

<u>Modified Euler</u>: $w_1 = w_2 = \frac{1}{2}$, $\alpha = \beta = 1$

<u>Midpoint method</u>: $w_1 = 0$, $w_2 = 1$, $\alpha = \beta = \frac{1}{2}$

(also $O(h^3)$ LTE)

$y_0 = A$

for $i = 0, 1, \ldots, N-1$

$F_1 = h f(t_i, y_i)$

$F_2 = h f(t_i + \frac{h}{2}, y_i + \frac{1}{2} F_1)$

$y_{i+1} = y_i + F_2$

<u>Heun's method</u>: $w_1 = \frac{1}{4}$, $w_2 = \frac{3}{4}$, $\alpha = \beta = \frac{2}{3}$

$y_0 = A$

for $i = 0, 1, \ldots, N-1$

$F_1 = h f(t_i, y_i)$

$F_2 = h f(t_i + \frac{2}{3} h, y_i + \frac{2}{3} F_1)$

$y_{i+1} = y_i + \frac{1}{4} F_1 + \frac{3}{4} F_2$

Runge Kutta methods of order 3 are obtained by matching terms between Taylor expansion of order 3 of $y(t)$ and $f(t+\alpha h, y + w_1 f(t+\beta h, y+ w_2 f(t,y)))$

The derivation is quite tedious, but here is one possible RK order 3 method:

$y_0 = A$

for $i = 0, \dots, N-1$

$$\begin{aligned}
F_1 &= h f(t_i, y_i) \\
F_2 &= h f(t_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}F_1) \\
F_3 &= h f(t_i + \tfrac{3}{4}h, y_i + \tfrac{3}{4}F_2) \\
y_{i+1} &= y_i + \tfrac{1}{9}(2F_1 + 3F_2 + 4F_3)
\end{aligned}$$

$$\begin{pmatrix} \alpha = w_1 = \frac{3}{4} \\ \beta = w_2 = \frac{1}{2} \end{pmatrix}$$

However it is not commonly used in practice.

The most popular RK method is that of order 4; again the derivation is tedious but the implementation straightforward.:

Runge Kutta method of order 4 (LTE $O(h^5)$)

$y_0 = A$

for $i = 0, \dots, N-1$

$$\begin{aligned}
F_1 &= h f(t_i, y_i) \\
F_2 &= h f(t_i + \tfrac{1}{2}h, x + \tfrac{1}{2}F_1) \\
F_3 &= h f(t_i + \tfrac{1}{2}h, x + \tfrac{1}{2}F_2) \\
F_4 &= h f(t_i + h, x + F_3) \\
y_{i+1} &= y_i + \tfrac{1}{6}(F_1 + 2F_2 + 2F_3 + F_4)
\end{aligned}$$

# of function eval.	1	2	3	4	5	6	7	8	9	10	11	12	....
max order of RK method	1	2	3	4	4	5	6	6	7	7	8	9	...

So # of function eval increases more rapidly than the max order of RK method. $\Rightarrow$ higher order RK methods are less attractive than the classical RK4. (it makes more sense to use smaller time steps for RK4 than using a higher order method with bigger steps).

§5.5 Adaptive Runge Kutta Fehlberg method

Idea: Estimate local truncation error and adjust the step length accordingly.

Local truncation error estimation

Assume we are given two update formulas for solving

$$\begin{cases} y' = f(t,y) \quad , t \in [a,b] \\ y(a) = \alpha \end{cases}$$

with local truncation errors differing by 1:

① $y(t_{i+1}) = y(t_i) + h\,\phi(t_i, y(t_i), h) + O(h^{n+1})$ (order n)

$y_0 = \alpha$
for $i = 0, \dots, N-1$
| $y_{i+1} = y_i + h\,\phi(t_i, y_i, h)$

② $y(t_{i+1}) = y(t_i) + h\,\tilde{\phi}(t_i, y(t_i), h) + O(h^{n+2})$ (order $n+1$)

$\tilde{y}_0 = \alpha$
for $i = 0 \dots N-1$
| $\tilde{y}_{i+1} = \tilde{y}_i + h\,\tilde{\phi}(t_i, y_i, h)$